Mercury LoadRunner™

MONITOR REFERENCE

Mercury LoadRunner™

Monitor Reference

Version 8.0

Mercury LoadRunner Monitor Reference, 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.

LRMREF8.0/01

Table of Contents

Welcome to Mercury LoadRunner

Welcome to LoadRunner, Mercury's tool for application performance testing. LoadRunner stresses your entire application to isolate and identify potential client, network, and server bottlenecks.

LoadRunner enables you to test your system under controlled and peak load conditions. To generate load, LoadRunner runs thousands of Virtual Users, or **Vusers**, that are distributed over a network. Using a minimum of hardware resources, these Vusers provide consistent, repeatable, and measurable load to exercise your application just as real users would. LoadRunner's in-depth reports and graphs provide the information that you need to evaluate the performance of your application.

Online Resources

LoadRunner includes the following online tools:

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

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's Customer Support Web site for updates to LoadRunner online books.

LoadRunner Function Reference gives you online access to all of LoadRunner's functions that you can use when creating Vuser scripts, including examples of how to use the functions. Check Mercury's Customer Support Web site for updates to the online *LoadRunner Function Reference*.

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

Technical Support Online uses your default Web browser to open Mercury'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's Customer Support Web site and home page, the e-mail address for sending information requests, and a list of Mercury's offices around the world.

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

LoadRunner Documentation Set

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

- install LoadRunner
- create Vuser scripts
- use the LoadRunner Controller
- configure the LoadRunner monitors
- use the LoadRunner Analysis

Using the LoadRunner Documentation Set

The LoadRunner documentation set consists of an installation guide, a Controller user's guide, a Monitor Reference, an Analysis user's guide, and a guide for creating Virtual User scripts.

Installation Guide

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

- the LoadRunner Controller—on a Windows-based machine
- Virtual User components—for both Windows and UNIX platforms
- Additional LoadRunner components

Controller User's Guide

The LoadRunner documentation set includes a Controller user's guide:

The *LoadRunner Controller User's Guide* describes how to create and run LoadRunner scenarios using the LoadRunner Controller in a Windows environment. The Vusers can run on UNIX and Windows-based platforms. The Controller user's guide presents an overview of the LoadRunner testing process.

Monitor Reference

The LoadRunner documentation set includes a 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 scenario or tuning session.

Analysis User's Guide

The LoadRunner documentation set includes an Analysis user's guide:

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

Guide for Creating Vuser Scripts

The LoadRunner documentation set includes a 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.

Note: The *Mercury Virtual User Generator User's Guide* online version is a single volume, while the printed version consists of two volumes, Volume I - *Using VuGen* and Volume II - *Protocols*.

For information on	Look here...
Installing Mercury LoadRunner	*LoadRunner Installation Guide*
The LoadRunner load testing process	*LoadRunner Controller User's Guide*
Creating Vuser scripts	*Mercury Virtual User Generator User's Guide* Volume I - *Using VuGen*, Volume II - *Protocols*
Configuring the server monitors	*LoadRunner Monitor Reference*
Creating and running load test scenarios	*LoadRunner Controller User's Guide*
Analyzing test results	*LoadRunner Analysis User's Guide*

Documentation Updates

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

To download updated documentation:

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

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."). It also indicates method or function arguments, file names or paths.
Bold	**Bold** text indicates method or function names
Italics	*Italic* text indicates 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. In a programming example, an ellipsis is used to indicate lines of a program that were intentionally omitted.
|	A vertical bar indicates that one of the options separated by the bar should be selected.

1

Introduction

You can monitor a scenario or session step execution using the LoadRunner online monitors.

The specific monitors are discussed in the next few chapters.

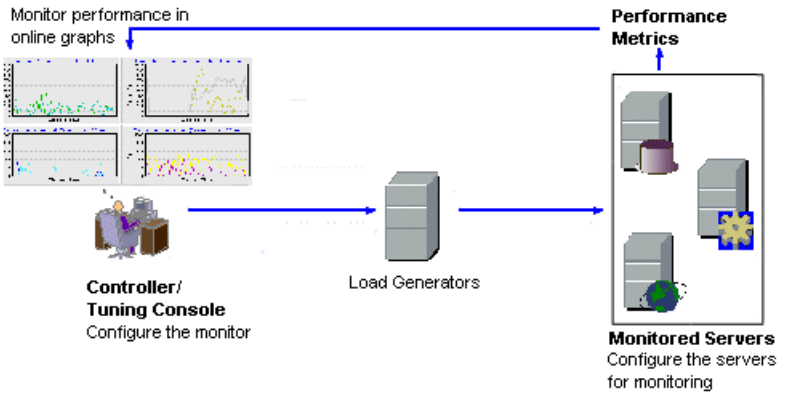
This chapter describes:

- Setting Up the Monitoring Environment
- Choosing Monitors and Measurements on the Controller
- Choosing Monitors and Measurements on the Tuning Console
- Monitor Types

Note: For information on setting monitor options, configuring graph settings and measurements, and exporting graph data, refer to the *LoadRunner Controller User's Guide* or *Mercury Tuning Module Console User's Guide*.

Setting Up the Monitoring Environment

Before monitoring a scenario or session step, you need to set up and configure the LoadRunner monitoring components. Each monitor has different configuration requirements that are explained in the specific monitoring chapters. The diagram below illustrates the LoadRunner 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 Controller/Tuning Console Machine

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

To set up a monitor, you need to perform the following:

➤ Add a monitored machine to the Controller or Tuning Console by selecting the server whose monitors you want to configure

➤ Configure the monitor by selecting the measurements that you want to monitor

For more information on setting up the monitoring environment and configuring a monitor, see the specific monitoring chapter.

Choosing Monitors and Measurements on the Controller

You specify the machines and measurements that the Controller will monitor during a scenario execution via the Controller's Run tab. During scenario execution, the collected measurement data will appear in the online graphs.

To select measurements to monitor:

1. Click the graph in the graph tree, and drag it into the right pane of the Run view.
2. Right-click the graph and select **Add Measurements**, or choose **Monitors > Add Measurements**. The monitored server machine dialog box opens.
3. In the **Monitored Server Machines** section, click **Add.** The Add Machine dialog box opens. Enter the server name or IP address of the machine you want to monitor. Select the platform on which the machine runs, and click **OK**.
4. In the **Resource Measurements** section of the monitor dialog box, click **Add**. The Monitor Configuration dialog box opens. Choose the measurements for the specific server, and click **OK** to save your configuration.

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:

1 Click the monitor server element to select it.

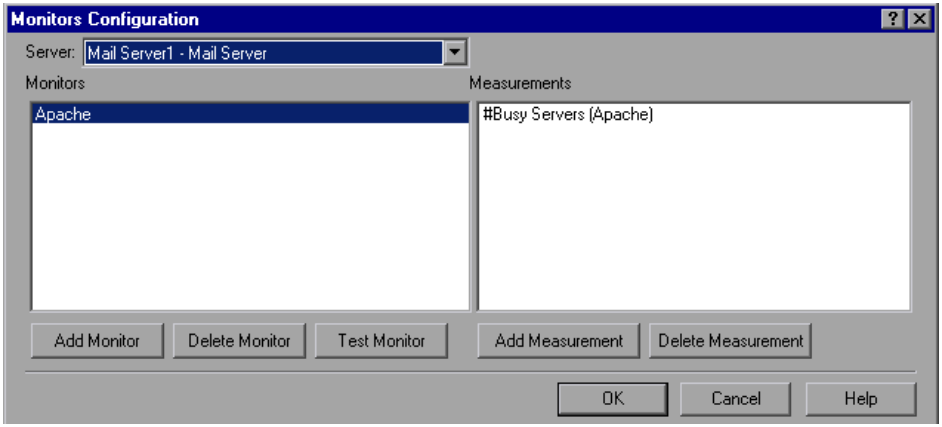
2 In the **Element Monitors** tab, click **Add**. The Select Measurements to Monitor dialog box opens.

3 Select **Show All Available Monitors**, expand the selected monitor element, and then click the monitor.

4 Click **Add**. The Monitor Configuration dialog box opens. Choose the measurements for the specific server, and click **OK** to save your configuration.

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

1 Click **Monitors**.The Monitors Configuration dialog box opens.



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.

4 Click **OK** to save your configuration.

Monitor Types

The online monitors are divided into the following categories:

- Run-Time Monitors
  Display the number and status of Vusers participating in the scenario or session step, as well as the number and types of errors that the Vusers generate. For more information, see Chapter 3, "Run-Time and Transaction Monitoring."
- Transaction Monitors
  Display the transaction rate and response time during scenario or session step execution. For more information, see Chapter 3, "Run-Time and Transaction Monitoring."
- Web Resource Monitors
  Provide information about the number of Web connections, throughput volume, HTTP responses, server retries, and downloaded pages at the Web servers during the scenario or session step. For more information, see Chapter 2, "Web Resource Monitoring."
- System Resource Monitors
  Measure the Windows, UNIX, Tuxedo, SNMP, Antara FlameThrower, and SiteScope resources used during a scenario or session step. For more information, see Chapter 4, "System Resource Monitoring."
- Network Delay Monitor
  Displays information about the network delays on your system. For more information, see Chapter 5, "Network Monitoring."
- Firewall Monitor
  Measures statistics related to the firewall servers during the scenario or session step. For more information, see Chapter 6, "Firewall Server Performance Monitoring."
- Web Server Resource Monitors
  Measure statistics related to the Apache, Microsoft IIS, iPlanet (SNMP) and iPlanet/Netscape Web servers during the scenario or session step. For more information, see Chapter 7, "Web Server Resource Monitoring."

- Web Application Server Resource Monitors
  Measure statistics related to 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 scenario or session step. For more information, see Chapter 8, "Web Application Server Resource Monitoring."

- Database Server Resource Monitors
  Measure statistics related to the SQL server, Oracle, Sybase, and DB2 databases during the scenario or session step. For more information, see Chapter 9, "Database Resource Monitoring."

- Streaming Media Monitors
  Measure statistics related to the Windows Media Server and RealPlayer audio/video servers, as well as the RealPlayer client during the scenario or session step. For more information, see Chapter 10, "Streaming Media Monitoring."

- ERP/CRM Server Resource Monitors
  Measure statistics related to the SAP R/3 system, SAP Portal, Siebel Server Manager, Siebel Web Server, and PeopleSoft (Tuxedo) servers during the scenario or session step. For more information, see Chapter 11, "ERP/CRM Server Resource Monitoring."

- Java Performance Monitors
  Measure statistics related to Java 2 Platform, Enterprise Edition (J2EE) objects, and Java-based applications, using J2EEmachines. For more information, see Chapter 12, "J2EE Performance Monitoring."

- 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 the "J2EE Diagnostics Module" chapter in the *LoadRunner Controller User's Guide*.

- Application Component Monitors
  Measure statistics related to the Microsoft COM+ and Microsoft .NET CLR servers during a scenario or session step run. For more information, see Chapter 13, "Application Component Monitoring."

- Application Deployment Solutions Monitor
  Measures statistics related to the Citrix MetaFrame XP and 1.8 servers during a scenario or session step run. For more information, see Chapter 14, "Application Deployment Solutions."
- Middleware Performance Monitors
  Measure statistics related to the Tuxedo and IBM WebSphere MQ servers during a scenario or session step run. For more information, see Chapter 15, "Middleware Performance Monitoring."
- Infrastructure Resources Monitor
  Displays information about network client data points during a session step using the Network Client graph. For more information, see Chapter 17, "Infrastructure Resources Monitoring."
- Application Traffic Management Monitor
  Measures statistics related to the F5 BIG-IP server during a session step run. For more information, see Chapter 16, "Application Traffic Management Monitoring."
- Security Monitor
  Displays information about simulated attacks on the server during a session step using the Distributed Denial of Service graph. For more information, see Chapter 18, "Security Monitoring."

All of the monitors allow you to view a summary of the collected data at the conclusion of the scenario or 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).

2

Web Resource Monitoring

You obtain information about the performance of your Web server using LoadRunner's Web Resource monitor.

This chapter describes:

- About Web Resource Monitoring
- Hits per Second Graph
- Throughput Graph
- HTTP Responses per Second Graph
- Pages Downloaded per Second Graph
- Retries per Second Graph
- Connections Graph
- Connections per Second Graph
- SSL Connections per Second Graph

About Web Resource Monitoring

The Web Resource monitor enables you to analyze the throughput on the Web server, the number of hits per second that occurred during the scenario or session step, the number of HTTP responses per second, the HTTP status codes (which indicate the status of HTTP requests, for example, "the request was successful," "the page was not found") returned from the Web server, the number of downloaded pages per second, the number of server retries per second, the number of open TCP/IP connections, the number of new TCP/IP connections per second, and the number of SSL Connections per second.

Hits per Second Graph

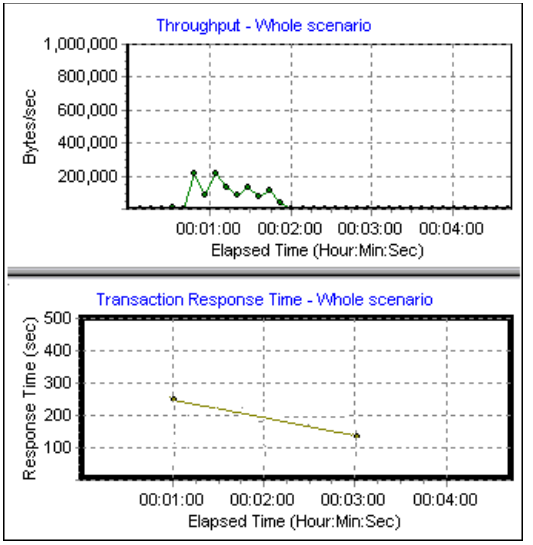
The **Hits Per Second** graph shows the number of hits (HTTP requests) to the Web server (y-axis) as a function of the elapsed time in the scenario or session step (x-axis). This graph can display the whole step, or the last 60, 180, 600, or 3600 seconds. You can compare this graph to the Transaction Response Time graph to see how the number of hits affects transaction performance.

Throughput Graph

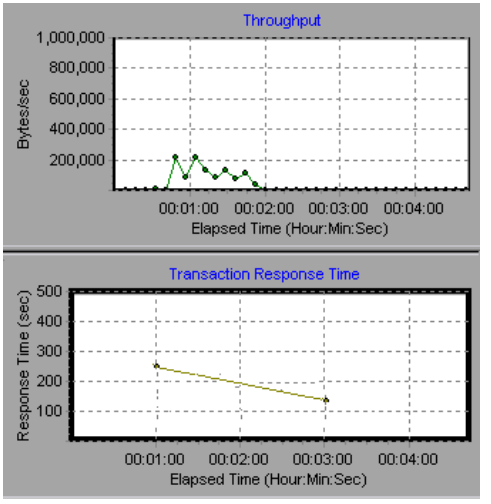
The **Throughput** graph shows the amount of throughput on the Web server (y-axis) during each second of the scenario or session step run (x-axis). Throughput is measured in bytes and represents the amount of data that the Vusers received from the server at any given second. You can compare this graph to the Transaction Response Time graph to see how the throughput affects transaction performance.

In the following example, the Transaction Response time graph is compared with the Throughput graph. It is apparent from the graph that as the throughput decreases, the transaction response time also decreases. The peak throughput occurred at approximately 1 minute into the step. The highest response time also occurred at this time.

Example in the Controller:



Example in the Tuning Console:



HTTP Responses per Second Graph

The **HTTP Responses per Second** graph shows the number of HTTP status codes—which indicate the status of HTTP requests, for example, “the request was successful,” “the page was not found”—(y-axis) returned from the Web server during each second of the scenario or session step run (x-axis), grouped by status code. You can group the results shown in this graph by script (using the "Group By" function) to locate scripts which generated error codes.

The following table displays a list of HTTP status codes:

Code	Description
200	OK
201	Created
202	Accepted
203	Non-Authoritative Information
204	No Content
205	Reset Content
206	Partial Content
300	Multiple Choices
301	Moved Permanently
302	Found
303	See Other
304	Not Modified
305	Use Proxy
307	Temporary Redirect
400	Bad Request
401	Unauthorized

Code	Description
402	Payment Required
403	Forbidden
404	Not Found
405	Method Not Allowed
406	Not Acceptable
407	Proxy Authentication Required
408	Request Timeout
409	Conflict
410	Gone
411	Length Required
412	Precondition Failed
413	Request Entity Too Large
414	Request - URI Too Large
415	Unsupported Media Type
416	Requested range not satisfiable
417	Expectation Failed
500	Internal Server Error
501	Not Implemented
502	Bad Gateway
503	Service Unavailable
504	Gateway Timeout
505	HTTP Version not supported

For more information on the above status codes and their descriptions, see http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.

Pages Downloaded per Second Graph

The **Pages Downloaded per Second** graph shows the number of Web pages (y-axis) downloaded from the server during each second of the scenario or session step run (x-axis). This graph helps you evaluate the amount of load Vusers generate, in terms of the number of pages downloaded.

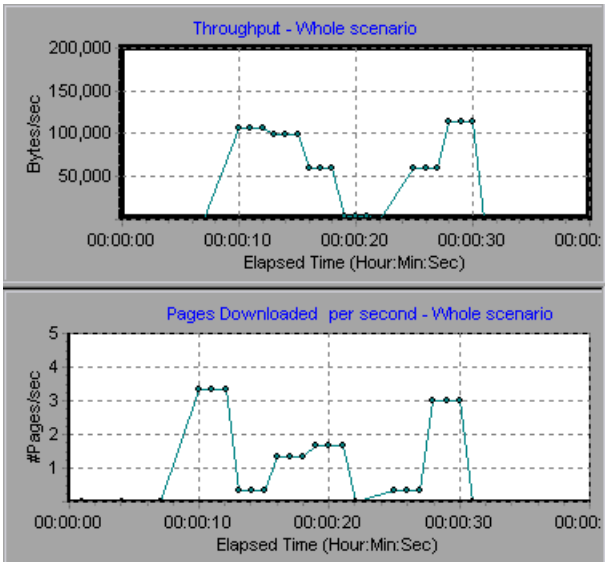
Note: To view the Pages Downloaded per Second graph, you must select **Pages per second (HTML Mode only)** from the script's run-time settings Preferences tab before running your scenario or session.

Like throughput, downloaded pages per second is a representation of the amount of data that the Vusers received from the server at any given second.

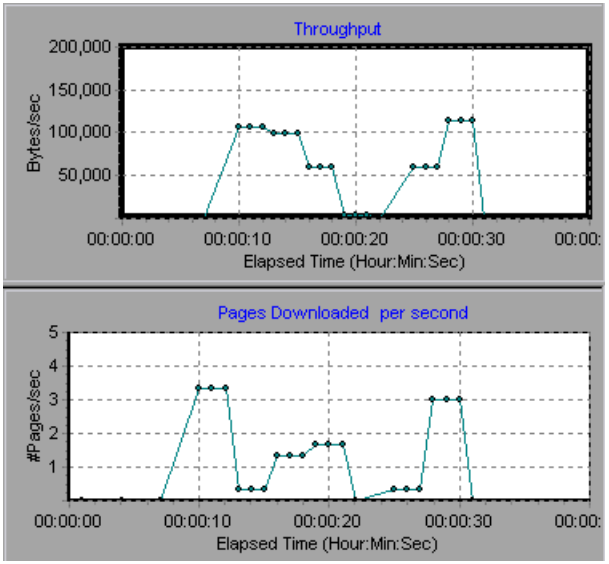
- The Throughput graph takes into account each resource and its size (for example, the size of each **.gif** file, the size of each Web page).
- The Pages Downloaded per Second graph takes into account simply the number of pages.

In the following example, the Throughput graph is compared with the Pages Downloaded per Second graph. It is apparent from the graph that throughput is not proportional to the number of pages downloaded per second. For example, between 15 and 16 seconds into the scenario or session run, the throughput decreased while the number of pages downloaded per second increased.

Example in the Controller:



Example in the Tuning Console:



Retries per Second Graph

The **Retries Per Second** graph shows the number of attempted Web server connections (y-axis) as a function of the elapsed time in the scenario or session step (x-axis). A server connection is retried when the initial connection was unauthorized, when proxy authentication is required, when the initial connection was closed by the server, when the initial connection to the server could not be made, or when the server was initially unable to resolve the load generator's IP address.

Connections Graph

The **Connections** graph shows the number of open TCP/IP connections (y-axis) at each point in time of the scenario or session step (x-axis). Note that one HTML page may cause the browser to open several connections, when links on the page go to different Web addresses. Two connections are opened for each Web server.

This graph is useful in indicating when additional connections are needed. For example, if the number of connections reaches a plateau, and the transaction response time increases sharply, adding connections would probably cause a dramatic improvement in performance (reduction in the transaction response time).

Connections per Second Graph

The **Connections Per Second** graph shows the number of new TCP/IP connections (y-axis) opened and the number of connections that are shut down each second of the scenario or session step (x-axis).

This number should be a small fraction of the number of hits per second, because new TCP/IP connections are very expensive in terms of server, router and network resource consumption. Ideally, many HTTP requests should use the same connection, instead of opening a new connection for each request.

SSL Connections per Second Graph

The **SSL Connections per Second** graph shows the number of new and reused SSL Connections (y-axis) opened in each second of the scenario or session step (x-axis). An SSL connection is opened by the browser after a TCP/IP connection has been opened to a secure server.

Because creating a new SSL connection entails heavy resource consumption, you should try to open as few new SSL connections as possible; once you've established an SSL connection, you should reuse it. There should be no more than one new SSL connection per Vuser.

If you set your run-time settings to simulate a new Vuser at each iteration (via the Browser Emulation tab in the Run-Time Settings menu), you should have no more than one new SSL connection per Vuser per iteration. Ideally, you should have very few new TCP/IP and SSL connections each second.

3

Run-Time and Transaction Monitoring

While running a scenario or session step, you use LoadRunner's Run-Time and Transaction monitors to view graphs of run-time status and transaction performance.

This chapter describes:

- About Run-Time and Transaction Graphs
- Run-Time Graphs
- User-Defined Data Points Graph
- Transaction Monitor Graphs
- Enabling the Transaction Monitor
- Adding Transactions to a Script
- Enabling Web Page Breakdown

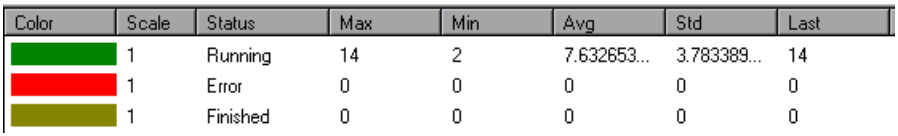
About Run-Time and Transaction Graphs

The **Run-Time** monitor provides information about the status of the Vusers participating in the scenario or session step, and the number and types of errors that the Vusers generate. In addition, the Run-Time monitor provides the User-Defined Data Points graph, which displays the real time values for user-defined points in a Vuser script.

The **Transaction** monitor displays the transaction rate and response time during scenario or session step execution. For more information about transactions, see "Adding Transactions to a Script" on page 24.

Run-Time Graphs

The monitor's **Running Vusers** graph provides information about the status of the Vusers running in the current scenario or session step on all load generator machines. The graph shows the number of running Vusers, while the information in the legend indicates the number of Vusers in each state.



Color	Scale	Status	Max	Min	Avg	Std	Last
	1	Running	14	2	7.632653...	3.783389...	14
	1	Error	0	0	0	0	0
	1	Finished	0	0	0	0	0

The Status field of each Vuser displays the current status of the Vuser. The following table describes each Vuser status.

Status	Description
Running	The total number of Vusers currently running on all load generators.
Ready	The number of Vusers that completed the initialization section of the script and are ready to run.
Finished	The number of Vusers that have finished running. This includes both Vusers that passed and failed.
Error	The number of Vusers whose execution generated an error. Check the Status field in the Vuser view or the Output window for a complete explanation of the error.

Error Statistics

The monitor's **Error Statistics** graph provides details about the number of errors that accrue during each second of the scenario or session step run. The errors are grouped by error source—for example, the location in the script or the load generator name.

Vusers with Error Statistics

The **Vusers with Error Statistics** graph provides details about the number of Vusers that generate errors during scenario or session step execution. The errors are grouped by error source.

User-Defined Data Points Graph

The **User-Defined Data Points** graph displays the real-time values of user-defined data points. You define a data point in your Vuser script by inserting an **lr_user_data_point** function at the appropriate place (**user_data_point** for GUI Vusers and **lr.user_data_point** for Java Vusers).

```
Action1()
{
	lr_think_time(1);
	lr_user_data_point ("data_point_1",1);
	lr_user_data_point ("data_point_2",2);
	return 0;
}
```

For Vuser protocols that support the graphical script representations such as Web and Oracle NCA, you insert a data point as a user-defined step. Data point information is gathered each time the script executes the function or step. For more information about data points, see the *LoadRunner Online Function Reference*.

By default, LoadRunner displays all of the data points in a single graph. The legend provides information about each data point. If desired, you can hide specific data points using the legend below the graphs.

You can also view data points offline, after the completion of the scenario or session step. For more information, see the *LoadRunner Analysis User's Guide*.

Transaction Monitor Graphs

The **Transaction** monitor provides the following graphs:

- Transaction Response Time
- Transactions per Second (Passed)
- Transactions per Second (Failed, Stopped)
- Total Transactions per Second (Passed)

The **Transaction Response Time** graph shows the average response time of transactions in seconds (y-axis) as a function of the elapsed time in the scenario or session step (x-axis).

The **Transactions per Second (Passed)** graph shows the number of successful transactions performed per second (y-axis) as a function of the elapsed time in the scenario or session step (x-axis).

The **Transactions per Second (Failed, Stopped)** graph shows the number of failed and stopped transactions per second (y-axis) as a function of the elapsed time in the scenario or session step (x-axis).

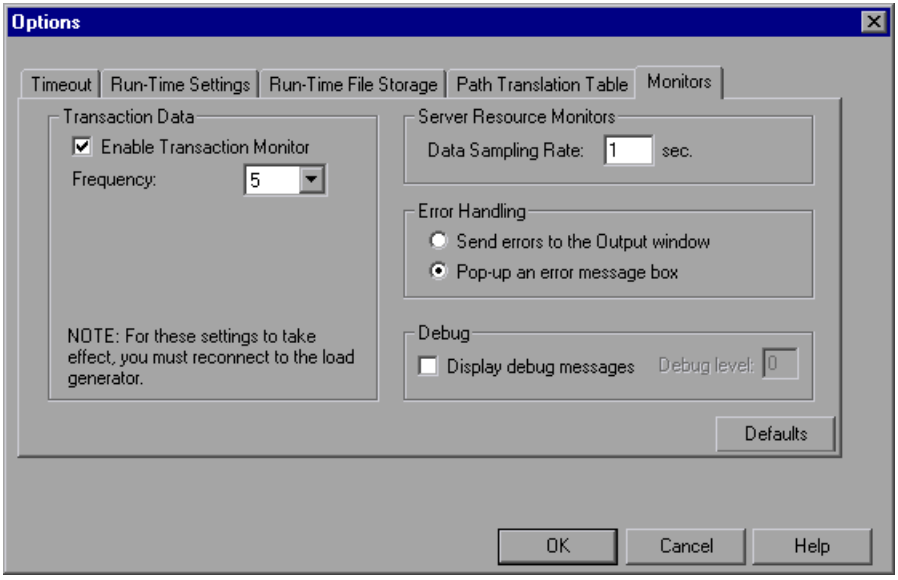
The **Total Transactions per Second (Passed)** graph shows the total number of completed, successful transactions per second (y-axis) as a function of the elapsed time in the scenario or session step (x-axis).

Enabling the Transaction Monitor

The Transaction monitor is enabled by default—it automatically begins monitoring Vuser transactions at the start of a scenario or session step. To conserve resources, you can disable the Transaction monitor.

To disable the Transaction monitor:

1 Choose **Tools** > **Options** and select the **Monitors** tab.



2 Disable transaction monitoring by clearing the **Enable Transaction Monitor** check box. To enable transaction monitoring, select the **Enable Transaction Monitor** check box.

Adding Transactions to a Script

If there are no transactions defined in your Vuser script, no data will be displayed in the online graphs. To add transactions to an existing script, edit it using the appropriate tool. The following table shows the script generation tools for each script type:

Script type	Editing tool
GUI Windows	WinRunner
non-GUI Windows	VuGen (Virtual User Generator)
SAP	QuickTest for SAP

To add a transaction to a script in the Controller:

1. Click the **Design** tab to view the list of Vuser groups and scripts.
2. To edit a script for a Vuser group, select the group and click the **View Script** button to the right of the Scenario Groups window. The script generation tool opens.

   To edit a script for an individual Vuser, click **Vusers**. Right-click the Vuser whose script you want to edit, and select **View Script** to open the script generation tool.
3. Insert Start and End Transaction functions or markers throughout your script.

To add a transaction to a script in the Tuning Console:

1. Right-click the script you want to edit in the session tree, and select **View Script**.

   The script generation tool opens.
2. Insert Start and End Transaction functions or markers throughout your script.

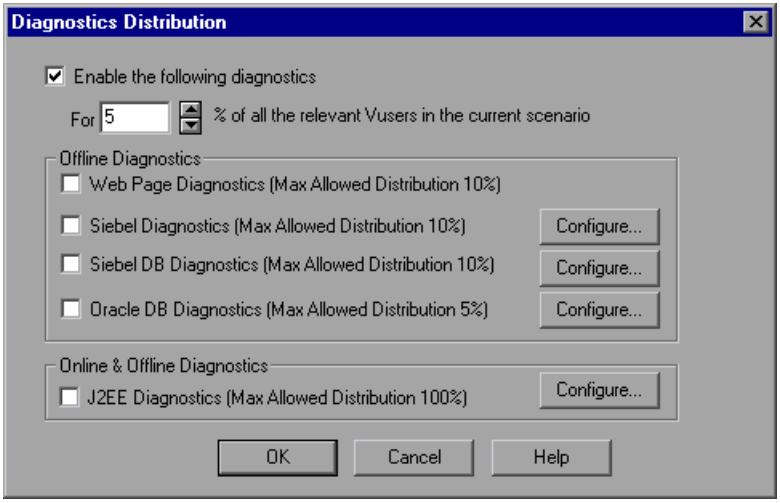
For more information, see the appropriate user's guide as described in the *Welcome* chapter.

Enabling Web Page Breakdown

In order for the Analysis to generate Web Page Breakdown graphs, which provide you with performance information for each transaction and sub-transaction defined in your script, you must enable the Web page breakdown feature in the Controller or Tuning Console before running your scenario or session step.

Enabling Web Page Breakdown in the Controller

1 From the Controller's Run tab, choose **Diagnostics > Distribution**. The Breakdown Distribution dialog box opens.



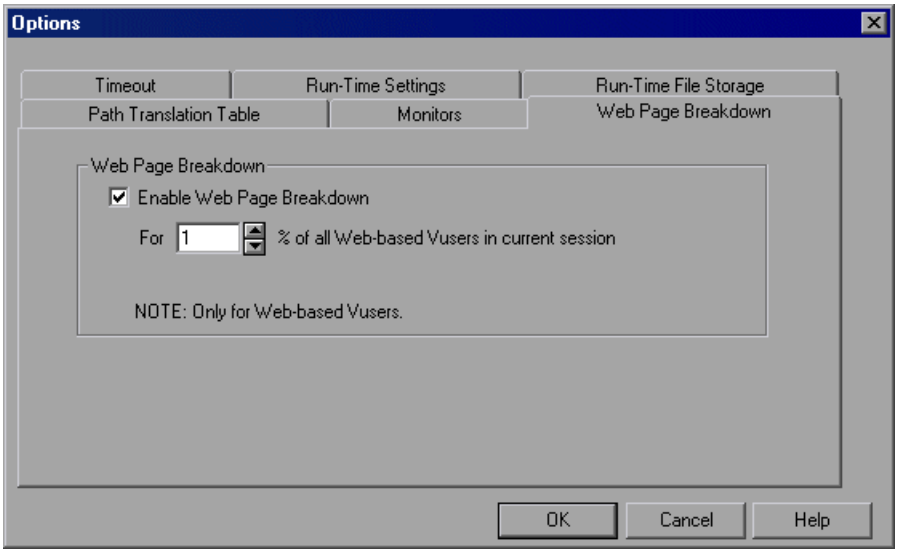
2 Select **Enable Web Page Breakdown** and specify the percentage of Vusers for which you want transaction breakdown to be performed.

3 Select **Web Page Breakdown** and click **OK**.

For more information, refer to the chapter "Using the LoadRunner Diagnostics Modules" in the *LoadRunner Controller User's Guide* or *Mercury Tuning Module Console User's Guide*.

Enabling Web Page Breakdown in the Tuning Console

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



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 scenario:** 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*.

4

System Resource Monitoring

You use LoadRunner's System Resource monitors to monitor a machine's system resource usage during a scenario or session step run and isolate server performance bottlenecks.

This chapter describes:

- About System Resource Monitoring
- Windows Resources Monitoring
- UNIX Resources Monitoring
- Server Resources Monitoring
- SNMP Resources Monitoring
- Antara FlameThrower Monitoring
- SiteScope Monitoring

About System Resource Monitoring

A primary factor in a transaction's response time is its system resource usage. Using the LoadRunner resource monitors, you can monitor the Windows, UNIX, Server Resources, SNMP, Antara FlameThrower, and SiteScope resources on a machine during a scenario or session step run, and determine why a bottleneck occurred on a particular machine.

The Windows measurements correspond to the built-in counters available from the Windows Performance Monitor.

The UNIX measurements include those available by the **rstatd** daemon: average load, collision rate, context switch rate, CPU utilization, incoming packets error rate, incoming packets rate, interrupt rate, outgoing packets error rate, outgoing packets rate, page-in rate, page-out rate, paging rate, swap-in rate, swap-out rate, system mode CPU utilization, and user mode CPU utilization.

Note: You must configure an **rstatd** daemon on all UNIX machines being monitored. For information on how to configure an **rstatd** daemon, refer to the UNIX *man* pages, or see "Setting up the Monitoring Environment", on page 35.

The Server Resources monitor can measure CPU, disk space, memory, and application resources used on remote Windows and UNIX servers.

The SNMP monitor is available for monitoring machines using the Simple Network Management Protocol (SNMP). SNMP monitoring is platform independent.

The Antara FlameThrower monitor can measure the following performance counters: Layer, TCP, HTTP, SSL/HTTPS, Sticky SLB, FTP, SMPT, POP3, DNS, and Attacks.

The SiteScope monitor can measure server, network, and processor performance counters. For detailed information on the performance counters that SiteScope can monitor, see the relevant SiteScope documentation.

The resource monitors are automatically enabled when you execute a scenario or session step. However, you must specify the machine you want to monitor and which resources to monitor for each machine. You can also add or remove machines and resources during the scenario or session step run.

Windows Resources Monitoring

The Windows Resources monitor shows the Windows resources measured during the scenario or session step. Windows NT and Windows 2000 measurements correspond to the built-in counters available from the Windows Performance Monitor.

Note: To monitor a Windows NT or 2000 machine through a firewall, use TCP, port 139.

To obtain data for this graph, you need to select the desired measurements for the online monitor (from the Controller or Tuning Console) before running the scenario or session step.

Adding a Monitored Machine to the Controller

1. Click the Windows Resources graph in the graph tree, and drag it into the right pane of the Run view.
2. Right-click the graph and select **Add Measurements**, or choose **Monitors** > **Add Measurements**. The Windows Resources dialog box opens.
3. In the **Monitored Server Machines** section, click **Add**. The Add Machines dialog opens.
4. Enter the server name or IP address of the machine you want to monitor. Select the platform on which the machine runs, and click **OK**.
5. In the **Resource Measurements** section of the Windows Resources dialog box, select the default measurements that you want to monitor.

   For a description of the available measurements, see "Windows Resources Performance Counters" on page 32.

Note: To change the default counters for the Windows machine monitor, see "Changing a Monitor's Default Counters" on page 443.

---

6 To select additional measurements, click **Add**.

Continue with Configuring the Windows Resources Monitor below.

Adding a Monitored Machine to the Tuning Console

1 Click **Monitors** to open the Monitors Configuration dialog box.

2 Select the server whose monitors you want to configure from the Server list box.

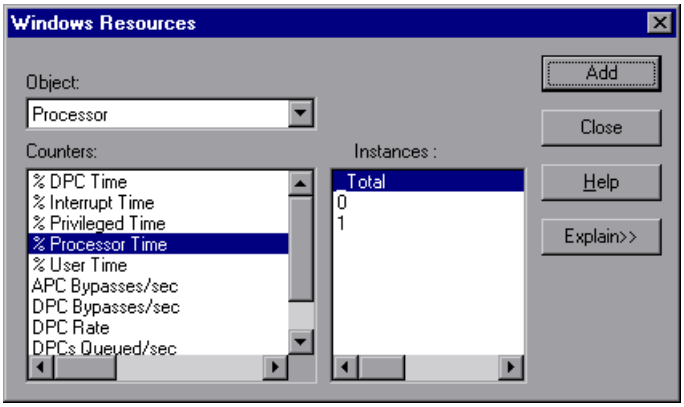
3 Click **Add Monitor** and select **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the System Resources category, select **Windows Resources**, and then click **Add**.

Continue with Configuring the Windows Resources Monitor below.

Configuring the Windows Resources Monitor

1 The Windows Resources dialog box opens displaying the available measurements and server properties.



For each measurement, select an object, counter, and instance, and then click **Add**, as described in “Understanding the Windows Resources Dialog Box” on page 32.

For a description of the available measurements, see "Windows Resources Performance Counters" on page 32.

Note: To change the default counters for the Windows machine monitor, see "Changing a Monitor's Default Counters" on page 443.

If you are monitoring a Win2000 machine, some of the NT machine default counters may not be available (such as % Total CPU usage and Interrupts/sec).

---

2. Add all the desired resources to the list, and click **Close**.
3. In the Controller, click **OK** in the Windows Resources dialog box to activate the monitor.
4. In the Tuning Console, click **OK** in the Select Measurements to Monitor dialog box to activate the monitor.

Note: If you want to monitor a remote Windows machine that does not use Windows domain security, you must authenticate the Controller or Tuning Console machine on the remote Windows machine. To authenticate the Controller or Tuning Console machine, create an account, or change the password of the account used to log on to the Controller or Tuning Console so that it matches the password and user name used to log on to the remote monitored Windows machine. When the remote Windows machine requests another machine's resources, it sends the logged-in user name and password of the machine requesting the resources.

Understanding the Windows Resources Dialog Box

The Windows Resources dialog box lets you select the Windows NT and Windows 2000 resources to monitor. The Windows resources correspond to the built-in counters available from the Windows Performance Monitor.

Object: Select the object to monitor on the specified Windows machine.

Counters: Select the resource counter to monitor. Select multiple counters using the CTRL key. For an explanation of each counter, click **Explain**.

Instances: If multiple instances of the selected counter are running, select one or more instances to monitor for the selected counter.

Explain: Displays a description of the selected counter.

Windows Resources Performance Counters

The following default measurements are available for Windows machines:

Object	Measurement	Description
System	**% Total Processor Time**	The average percentage of time that all the processors on the system are busy executing non-idle threads. On a multi-processor system, if all processors are always busy, this is 100%, if all processors are 50% busy this is 50% and if 1/4th of the processors are 100% busy this is 25%. It can be viewed as the fraction of the time spent doing useful work. Each processor is assigned an Idle thread in the Idle process which consumes those unproductive processor cycles not used by any other threads.
System	**File Data Operations/sec**	The rate at which the computer issues read and write operations to file system devices. This does not include File Control Operations.

Object	Measurement	Description
Processor	**% Processor Time (Windows 2000)**	The percentage of time that the processor is executing a non-idle thread. This counter was designed as a primary indicator of processor activity. It is calculated by measuring the time that the processor spends executing the thread of the idle process in each sample interval, and subtracting that value from 100%. (Each processor has an idle thread which consumes cycles when no other threads are ready to run). It can be viewed as the percentage of the sample interval spent doing useful work. This counter displays the average percentage of busy time observed during the sample interval. It is calculated by monitoring the time the service was inactive, and then subtracting that value from 100%.
System	**Processor Queue Length**	The instantaneous length of the processor queue in units of threads. This counter is always 0 unless you are also monitoring a thread counter. All processors use a single queue in which threads wait for processor cycles. This length does not include the threads that are currently executing. A sustained processor queue length greater than two generally indicates processor congestion. This is an instantaneous count, not an average over the time interval.
Memory	**Page Faults/sec**	This is a count of the page faults in the processor. A page fault occurs when a process refers to a virtual memory page that is not in its Working Set in the main memory. A page fault will not cause the page to be fetched from disk if that page is on the standby list (and hence already in main memory), or if it is in use by another process with which the page is shared.

Object	Measurement	Description
PhysicalDisk	**% Disk Time**	The percentage of elapsed time that the selected disk drive is busy servicing read or write requests.
Memory	**Pool Nonpaged Bytes**	The number of bytes in the nonpaged pool, a system memory area where space is acquired by operating system components as they accomplish their appointed tasks. Nonpaged pool pages cannot be paged out to the paging file. They remain in main memory as long as they are allocated.
Memory	**Pages/sec**	The number of pages read from the disk or written to the disk to resolve memory references to pages that were not in memory at the time of the reference. This is the sum of Pages Input/sec and Pages Output/sec. This counter includes paging traffic on behalf of the system cache to access file data for applications. This value also includes the pages to/from non-cached mapped memory files. This is the primary counter to observe if you are concerned about excessive memory pressure (that is, thrashing), and the excessive paging that may result.
System	**Total Interrupts/sec**	The rate at which the computer is receiving and servicing hardware interrupts. The devices that can generate interrupts are the system timer, the mouse, data communication lines, network interface cards, and other peripheral devices. This counter provides an indication of how busy these devices are on a computer-wide basis. See also Processor:Interrupts/sec.

Object	Measurement	Description
Objects	**Threads**	The number of threads in the computer at the time of data collection. Notice that this is an instantaneous count, not an average over the time interval. A thread is the basic executable entity that can execute instructions in a processor.
Process	**Private Bytes**	The current number of bytes that the process has allocated that cannot be shared with other processes.

UNIX Resources Monitoring

The UNIX Resources monitor shows the UNIX resources measured during the scenario or session step. This graph helps you determine the impact of Vuser load on the various system resources.

The UNIX kernel statistics measurements include those available by the rstatd daemon: average load, collision rate, context switch rate, CPU utilization, incoming packets error rate, incoming packets rate, interrupt rate, outgoing packets error rate, outgoing packets rate, page-in rate, page-out rate, paging rate, swap-in rate, swap-out rate, system mode CPU utilization, and user mode CPU utilization.

To obtain data for this graph, you need to select the desired measurements for the online monitor (from the Controller or Tuning Console) before running the scenario or session step.

Setting up the Monitoring Environment

To monitor UNIX resources, you must configure the rstatd daemon. Note that the rstatd daemon might already be configured, because when a machine receives an rstatd request, the inetd on that machine activates the rstatd automatically.

To verify whether the rstatd daemon is already configured:

The **rup** command reports various machine statistics, including rstatd configuration. Run the following command to view the machine statistics:

```
>rup host
```

You can also use **lr_host_monitor** and see if it returns any relevant statistics.

If the command returns meaningful statistics, the rstatd daemon is already configured and activated. If not, or if you receive an error message, the rstatd daemon is not configured.

To configure the rstatd daemon:

1. Run the command: `su root`
2. Go to **/etc/inetd.conf** and look for the rstatd row (it begins with the word rstatd). If it is commented out (with a #), remove the comment directive, and save the file.
3. From the command line, run:

   ```
   kill -1 inet_pid
   ```

   where **inet_pid** is the pid of the inetd process. This instructs the inetd to rescan the **/etc/inetd.conf** file and register all daemons which are uncommented, including the rstatd daemon.
4. Run *rup* again.

   If the command still does not indicate that the rstatd daemon is configured, contact your system administrator.

Note: To monitor a UNIX machine through a firewall, you must run a UNIX utility called **rpcinfo** and identify the rstatd's port number. By running rpcinfo -p <hostname>, you will receive a list of all RPC servers registered in the host's portmapper, along with the port number. This list will not change until rstatd is stopped and rerun.

Some firewalls allow you to open an RPC program number instead of a port. In such cases, open program 100001. If are prompted to include a version number, specify versions 3 and 4.

Adding a Monitored Machine to the Controller

1. Click the UNIX Resources graph in the graph tree, and drag it into the right pane of the Run view.
2. Right-click the graph and select **Add Measurements**, or choose **Monitors** > **Add Measurements**. The UNIX Resources dialog box opens.
3. In the **Monitored Server Machines** section, click **Add**. The Add Machines dialog opens.
4. Enter the server name or IP address of the machine you want to monitor, and click **OK**.
5. In the **Resource Measurements** section of the UNIX Resources, select the default measurements you want to monitor.

   For a description of the available measurements, see "UNIX Resources Performance Counters" on page 40.

Note: To change the default counters for the UNIX monitor, see "Changing a Monitor's Default Counters" on page 443.

---

6. To select additional measurements, click **Add**.

   Continue with Configuring the UNIX Resources Monitor below.

Adding a Monitored Machine to the Tuning Console

1 Click **Monitors** to open the Monitors Configuration dialog box.

2 Select the server whose monitors you want to configure from the Server list box.

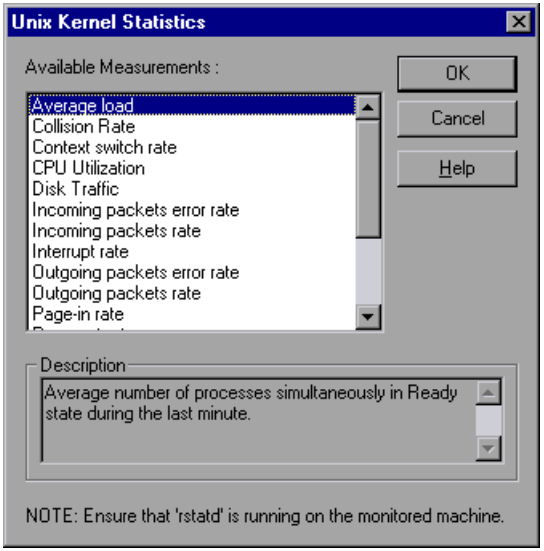
3 Click **Add Monitor** and select **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the System Resources category, select **UNIX Resources**, and then click **Add**.

Continue with Configuring the UNIX Resources Monitor below.

Configuring the UNIX Resources Monitor

1 The UNIX Kernel Statistics dialog box opens, displaying the available measurements and server properties.



Select measurements and server properties as described in “Understanding the UNIX Resources Dialog Box” on page 39, and then click **OK**.

For a description of the available measurements, see “UNIX Resources Performance Counters” on page 40.

2 In the Controller, click **OK** in the UNIX Kernel Statistics and in the UNIX Resources dialog box to activate the monitor.

3 In the Tuning Console, click **OK** in the UNIX Kernel Statistics dialog box and in the Select Measurements to Monitor dialog box to activate the monitor.

Note: Ensure that the rstatd daemon is correctly configured and running on the monitored UNIX machine. For more information, see "Setting up the Monitoring Environment", on page 35.

To change the default counters for the UNIX monitor, see "Changing a Monitor's Default Counters" on page 443.

Understanding the UNIX Resources Dialog Box

The UNIX Resources dialog box lets you select the UNIX kernel statistics to monitor.

The UNIX kernel statistics measurements include those available by the rstatd daemon: average load, collision rate, context switch rate, CPU utilization, incoming packets error rate, incoming packets rate, interrupt rate, outgoing packets error rate, outgoing packets rate, page-in rate, page-out rate, paging rate, swap-in rate, swap-out rate, system mode CPU utilization, and user mode CPU utilization.

Available Measurements: Select the measurements you want to add. To select more than one measurement, use the CTRL key.

Description: Displays a description of the selected measurement.

Note: Ensure that rstatd is running on the monitored UNIX machine. For more information, see "Setting up the Monitoring Environment".

UNIX Resources Performance Counters

The following default measurements are available for the UNIX machine:

Measurement	Description
Average load	Average number of processes simultaneously in Ready state during the last minute
Collision rate	Collisions per second detected on the Ethernet
Context switches rate	Number of switches between processes or threads, per second
CPU utilization	Percent of time that the CPU is utilized
Disk rate	Rate of disk transfers
Incoming packets error rate	Errors per second while receiving Ethernet packets
Incoming packets rate	Incoming Ethernet packets per second
Interrupt rate	Number of device interrupts per second
Outgoing packets errors rate	Errors per second while sending Ethernet packets
Outgoing packets rate	Outgoing Ethernet packets per second
Page-in rate	Number of pages read to physical memory, per second
Page-out rate	Number of pages written to pagefile(s) and removed from physical memory, per second
Paging rate	Number of pages read to physical memory or written to pagefile(s), per second
Swap-in rate	Number of processes being swapped
Swap-out rate	Number of processes being swapped
System mode CPU utilization	Percent of time that the CPU is utilized in system mode
User mode CPU utilization	Percent of time that the CPU is utilized in user mode

Server Resources Monitoring

The Server Resources monitor shows the resources of monitors (CPU, disk space, memory, or applications) used on remote Windows and UNIX servers measured during the scenario or session step. This helps you determine the impact of Vuser load on the various system resources.

The Server Resources monitor includes the following monitors:

CPU Monitor: Monitors CPU usage.

Disk Space Monitor: Monitors disk space.

Memory Monitor: Monitors Pages per Second and Percentage of Virtual Memory Used.

Service Monitor: Verifies that specific processes are listed as running and checks CPU usage.

To obtain data for this graph, you need to select the desired measurements for the online monitor (from the Controller or Tuning Console) before running the scenario or session step.

Setting up the Monitoring Environment

➤ You must ensure that SiteScope (the application that is used to monitor the Windows and UNIX servers) has been installed on a server. You can install SiteScope on the Controller or Tuning Console machine, but we recommend installing it on a dedicated server. For more information on connecting to the SiteScope machine, refer to the SiteScope documentation.

➤ Verify that SiteScope is collecting the required data from the servers it is monitoring. From the SiteScope Panel, select the monitor group polling the Server Resource machines, and check that the monitor displays a list of server measurements in the Status column.

Adding a Monitored Machine to the Controller

1. Click the Server Resource graph in the graph tree, and drag it into the right pane of the Run view.
2. Right-click the graph and select **Add Measurements**, or choose **Monitors** > **Add Measurements**. The Server Resources dialog box opens.
3. In the **Monitored Server Machines** section, click **Add.** The Add Machine dialog box opens.
4. In the **Monitored Machine Information** section, enter the server name or IP address of the machine you want to monitor, and select the platform on which the machine runs.

Note: If you are using the HTTP method of monitoring, enter the full URL of the CGI script (for example: http://demo.thiscompany.com/cgi-bin/run.sh).

In the **SiteScope Server Information** section, enter the SiteScope machine name, port (default: 8888), and version.

5. Click **OK**. The Server Resources dialog box is redisplayed.
6. In the **Resource Measurements** section of the Server Resources dialog box, click **Add** to configure the remote machine and select the measurements that you want to monitor.

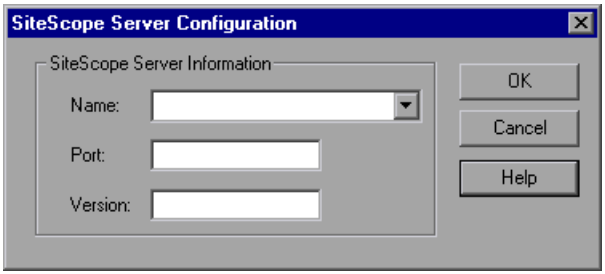
   Continue with Configuring the Server Resources Monitor below.

Adding a Monitored Machine to the Tuning Console

1. Click **Monitors** to open the Monitors Configuration dialog box.
2. From the Server list, choose the UNIX server that you want to monitor.
3. Click **Add Monitor** and select **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.
4. In the left section of the dialog box, select **Server Resources**.

5 Click **Add**.

If the SiteScope server has not been configured yet, the SiteScope Server Configuration dialog box opens.



6 Enter the SiteScope server information and click **OK**.

Continue with Configuring the Server Resources Monitor below.

Understanding the SiteScope Server Configuration Dialog Box

You use the SiteScope Server Configuration dialog box to enter information about the SiteScope server that is monitoring resources on the server in which you are interested.

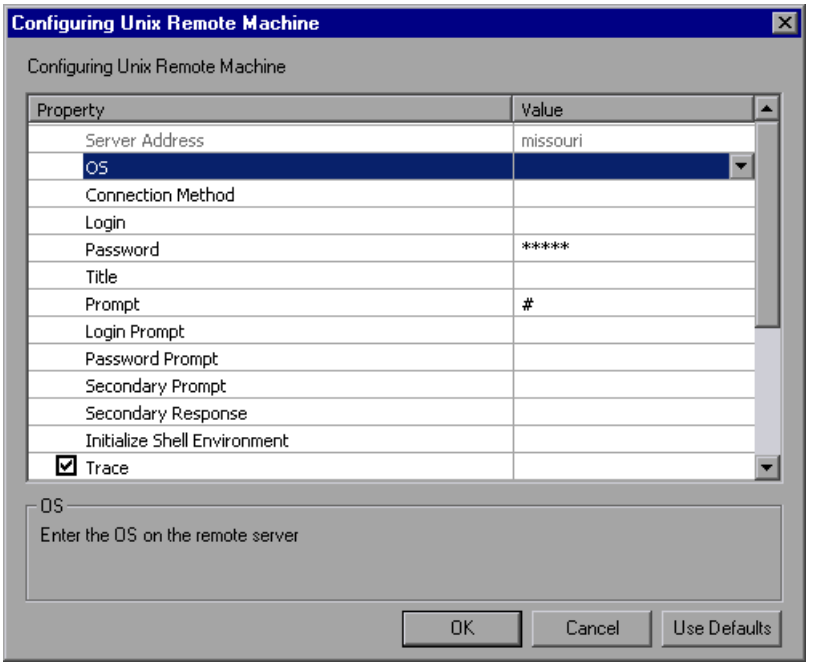
Name: Name of the SiteScope machine.

Port: The port used by the SiteScope server on the SiteScope machine.

Version: The SiteScope version.

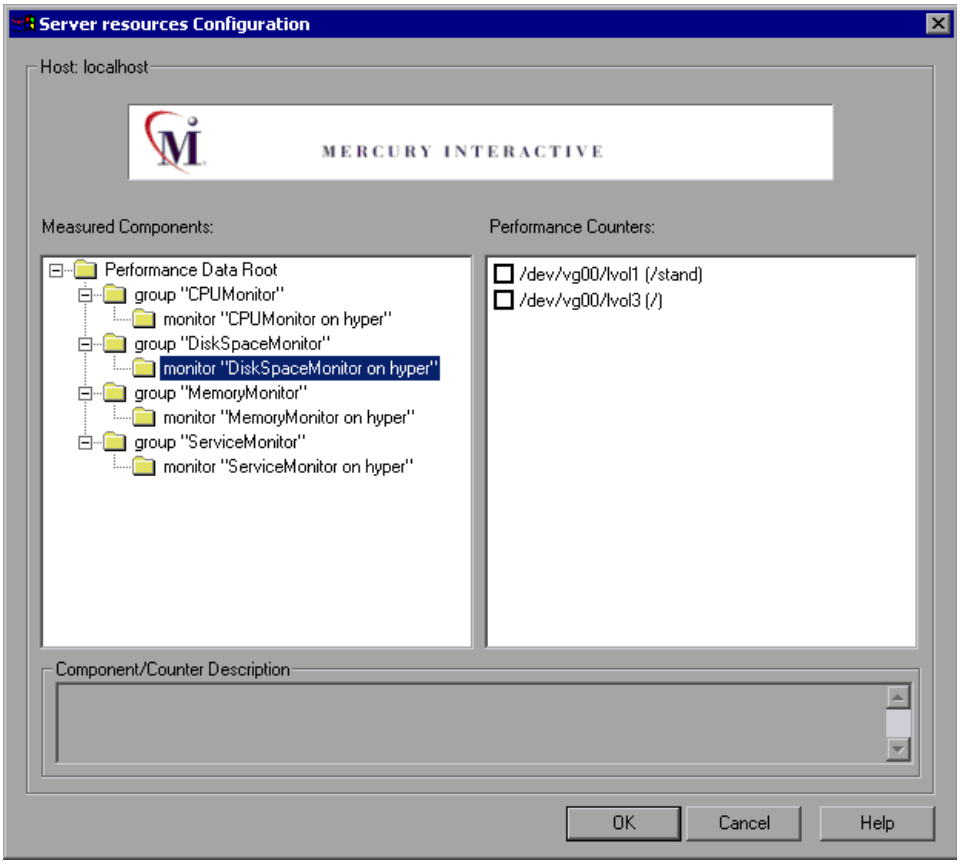
Configuring the Server Resources Monitor

1 If you are monitoring a machine on a UNIX platform, the Configuring UNIX Remote Machine dialog box opens.



Enter the remote UNIX machine configuration information, as described in "Understanding the Configuring UNIX Remote Machine Dialog Box" on page 46 and click **OK**.

2 The Server Resources Configuration dialog box opens, displaying the available measurements.



3 Browse the Measured Components tree, select the required performance counters in the right pane, and click **OK**.

For a list of the available performance counters, see "Server Resources Performance Counters" on page 51.

4 In the Controller, click **OK** in the Server Resources dialog box to activate the monitor.

5 In the Tuning Console, click **OK** in the Select Measurements to Monitor dialog box, and in the Monitors Configuration dialog box, to activate the monitor.

Note: The minimum recommended online graph refresh rate (in the Graph Configuration dialog box) is 30 seconds. If you choose a lower refresh rate, the Controller or Tuning Console may not get all the data in time.

Understanding the Configuring UNIX Remote Machine Dialog Box

The Configuring UNIX Remote Machine dialog box enables you to configure the remote UNIX machine.

Server Address: Displays the IP address or host name of the server that you entered in the Add Machine dialog box.

OS: Select the operating system running on the remote server. The following versions of UNIX are supported:

AIX	OPENSERVER
FreeBSD	SCO
HP/UX	SGI Irix
HP/UX 64-bit	Sun Solaris
Linux	Tru64 5.x
MacOSX	Tru64 Pre 4.x (Digital)

Connection Method: Select the method for connecting to the server. The supported methods are:

- **Telnet**: Log in to the remote server using Telnet.
- **SSH**: Log in to the remote server using the SSH protocol. This may require additional software and setup depending on the version of UNIX you are working with.

- **Rlogin**: Log in to the remote server using the Rlogin protocol.
- **HTTP**: Connect to an HTTP server on the remote server and run the command via a CGI. For this method, the Login and Password are optional and are used for authorizing the log on to the remote machine if required.

Login: The login for the remote server.

Password: Enter the password for the remote server.

Title: Enter a name by which the remote machine should be known. This name will appear in the drop-down list in monitors that can connect to this server.

Prompt: Enter the prompt to be displayed when the system is ready to handle a command - the default is #.

Login Prompt: Enter the prompt to be displayed when the system is waiting for the login to be entered - the default is "ogin:"

Password Prompt: Enter the prompt to be displayed when the system is waiting for the password to be entered - the default is "assword:"

Secondary Prompt: Enter the prompt to be displayed if the telnet connection to the remote server causes the remote server to prompt for more information about the connection. Separate multiple prompt strings by commas (,). For example, for Telnet connections to some remote servers, the remote server may ask what terminal type should be emulated for the connection. In this case you might need to enter Terminal type? as the secondary prompt. The response to the secondary prompt is entered in the Secondary Response field below.

Secondary Response: Enter the responses to secondary prompts required to establish connections with this remote server. Separate multiple responses with commas (,).

Initialize Shell Environment: Enter any shell commands to be executed at the beginning of the session. Separate multiple commands with a semicolon (;). This option allows you to specify shell commands to be executed on the remote machine directly after a Telnet or SSH session has been initiated. These commands can be used to customize the shell for each SiteScope remote.

Examples:

➤ The remote shell may not have the correct path set for SiteScope scripts to run. The following command will add the directory **/usr/local/bin** into the PATH of the current shell on the remote machine:

```
export PATH=$PATH:/usr/local/sbin
```

➤ The remote shell may not be initializing the pseudo terminal correctly. Enter the following command to increase the terminal width to 1024 characters:

```
stty cols 1024;${SHELL}
```

Note: Commands after a shell invocation will not be executed.

➤ There have been cases where the remote Telnet Server does not echo back the command line properly. This may cause strange behavior for monitors that rely on this behavior.

Enter the following command to force the remote terminal to echo:

```
stty echo
```

➤ Certain UNIX shells have been known to behave erratically with SiteScope. This includes bash, ksh, and csh. Enter the following command to change the shell to sh for the SiteScope connection:

```
/bin/sh
```

Trace: Select this option to trace messages to and from the remote server in the **RunMonitor.log** file.

SSH Connection Method: Select the method to use to connect to the remote server.

➤ **Internal Java Libraries**: Connect using the Java SSH client integrated with SiteScope

➤ **Plink**: Connect using an external SSH client. On Windows NT, SiteScope ships with Plink.

Disable Connection Caching: Select this to disable SSH connection caching.

Connection Limit: Enter the maximum number of connections for this remote machine.

SSH Authentication Method: Select the method to use to authenticate to the remote server (for SSH connections only).

- **Password**: Authenticate using a password.
- **Keyfile**: Authenticate using public/private key authentication. When this option is selected, SiteScope uses the private key in the file **SiteScope/groups/identity** to authenticate. The corresponding public key must be listed in the **authorized_keys** file on the remote host.

Connection Limit: Enter the maximum number of connections for this remote machine.

SSH Version 2 Only: Select this option to force SSH to only use SSH protocol version 2. This option is only supported when using the internal Java libraries connection method.

Custom Commandline: Enter the command for execution of the external SSH client. For substitutions with options listed above, use $host$, $user$, and $password$ respectively. This setting is supported only for connections using an external process.

SSH Port Number: Enter the port on which the SSH service is running.

After defining the server for SiteScope, you can test the settings by clicking on the test link. SiteScope attempts to display the working directory of the remote machine (the "pwd" command on UNIX, or "cd" on Windows NT), as a test to ensure that the remote machine can be accessed and can run commands properly.

Understanding the Server Resources Configuration Dialog Box

The Server Resources Configuration dialog box enables you to select the server resources to monitor.

Host: Displays the name of the monitored machine.

Measured Components: Displays the available components. Browse the tree and select the component you want to monitor. A description of the highlighted component appears in the Component/Counter Description box.

Performance Counters: Check the required performance counters. A description of the selected counter appears in the Component/Counter Description box.

Component/Counter Description: Displays a description of the selected component or counter.

Server Resources Performance Counters

Monitor	Measurements	Description
CPU Monitor	**Utilization**	Measures CPU utilization.
Disk Space Monitor	**Disk space**	Measures the percentage of disk space used.
Memory Monitor	**MB free**	Measures the amount of disk space free, in MB.
	Pages/sec	Measures the number of virtual memory pages that are moved between main memory and disk storage.
	Percent used	Measures the percentage of memory and paging file space used.
Services Monitor		Monitors processes locally or on remote systems. Can be used to verify that specific processes are running.

SNMP Resources Monitoring

The SNMP Resources monitor shows statistics for machines using the Simple Network Management Protocol (SNMP). The SNMP Resources monitor is available for monitoring any machine that runs an SNMP agent, using the Simple Network Management Protocol (SNMP).

To obtain data for this graph, you need to enable the SNMP Resources monitor (from the Controller or Tuning Console) and select the default measurements you want to display, before running the scenario or session step.

Note: You can specify a port number in the **snmp.cfg** file. If you do not specify a port, LoadRunner connects to default SNMP port 161. You can also specify a machine name in the following format:
<server name>:<port number>

To monitor SNMP resources through a firewall, use ports 161 or 162.

Adding a Monitored Machine to the Controller

1. Click the **SNMP Resources** graph in the graph tree, and drag it into the right pane of the Run view.
2. Right-click the graph and select **Add Measurements**, or choose **Monitors** > **Add Measurements**. The SNMP dialog box opens.
3. In the **Monitored Server Machines** section, click **Add.** The Add Machine dialog box opens.
4. Enter the server name or IP address of the machine you want to monitor. Select the platform on which the machine runs, and click **OK**.
5. In the **Resource Measurements** section of the SNMP dialog box, click **Add**.

   Continue with Configuring the SNMP Resources Monitor below.

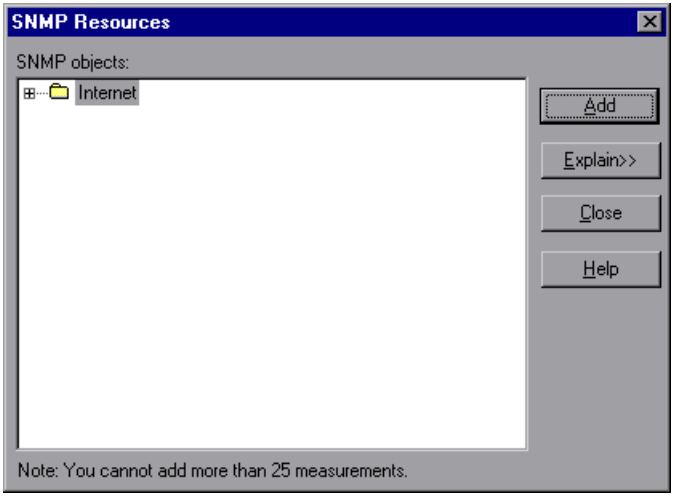
Adding a Monitored Machine to the Tuning Console

1. Click **Monitors** to open the Monitors Configuration dialog box.
2. Select the server whose monitors you want to configure from the Server list box.
3. Click **Add Monitor** and select **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.
4. In the left section of the dialog box, expand the System Resources category, select **SNMP**, and then click **Add**.

   Continue with Configuring the SNMP Resources Monitor below.

Configuring the SNMP Resources Monitor

1 The SNMP Resources dialog box opens.



2 Browse the SNMP Object tree and select SNMP objects as described in “Understanding the SNMP Resources Dialog Box” on page 54,.

3 Add all the desired resources to the list, and click **Close**.

Note: The SNMP monitor can only monitor up to 25 measurements.

---

4 In the Controller, click **OK** in the SNMP dialog box to activate the monitor.

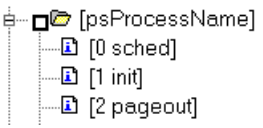
5 In the Tuning Console, click **OK** in the Select Measurements to Monitor dialog box to activate the monitor.

Note: You can modify the list of resources that you want to monitor at any point during the scenario or session step. Note that a scenario or session step does not have to be active in order for you to monitor the resources on a remote machine.

Improving the Level of Measurement Information

You can improve the level of measurement information for the SNMP monitor by enabling measurements with string values to be listed (in addition to measurements with numeric values), and by enabling the name modifier (which displays the string value as an identifying part of the measurement name).

In the following example of a measurement using the name modifier, the string value of ProcessName (sched) is displayed in addition to its instance ID (0):



To enable this feature, add the following line to the **<LoadRunner root folder>\dat\monitors\snmp.cfg** file:

```
SNMP_show_string_nodes=1
```

Usage Notes: You can select more than one name modifier, but the first in the hierarchy will be used. Each time the SNMP Add Measurements dialog box opens, the information is reread from the **snmp.cfg** file. You cannot add the same measurement twice (once with a name modifier and once without it). If you do so, an error message is issued.

Understanding the SNMP Resources Dialog Box

The SNMP Resources dialog box enables you to select the Windows or UNIX resources to monitor using the Simple Network Management Protocol (SNMP).

SNMP Objects: Select each required object and click **Add**. Click **Explain** for the ID number and a description of the selected object.

Explain: Displays a description of the selected object.

Note: The SNMP monitor can only monitor up to 25 measurements.

Antara FlameThrower Monitoring

The Antara FlameThrower monitor displays statistics about the resource usage on the Antara FlameThrower server during the scenario or session step run.

To obtain data for this graph, you need to enable the Antara FlameThrower online monitor (from the Controller or Tuning Console) and select the default measurements you want to display, before running the scenario or session step.

Adding a Monitored Machine to the Controller

1. Click the Antara FlameThrower graph in the graph tree, and drag it into the right pane of the Run view.
2. Right-click the graph and select **Add Measurements**, or choose **Monitors > Add Measurements**. The Antara FlameThrower dialog box opens.
3. In the **Monitored Server Machines** section, click **Add.** The Add Machine dialog box opens.
4. Enter the server name or IP address of the machine you want to monitor according to the following format: <server name>:<port number>.

   For example: merc1:12135

   Select the platform on which the machine runs, and click **OK**.
5. In the **Resource Measurements** section of the Antara FlameThrower dialog box, click **Add**.

   Continue with Configuring the Antara FlameThrower Monitor below.

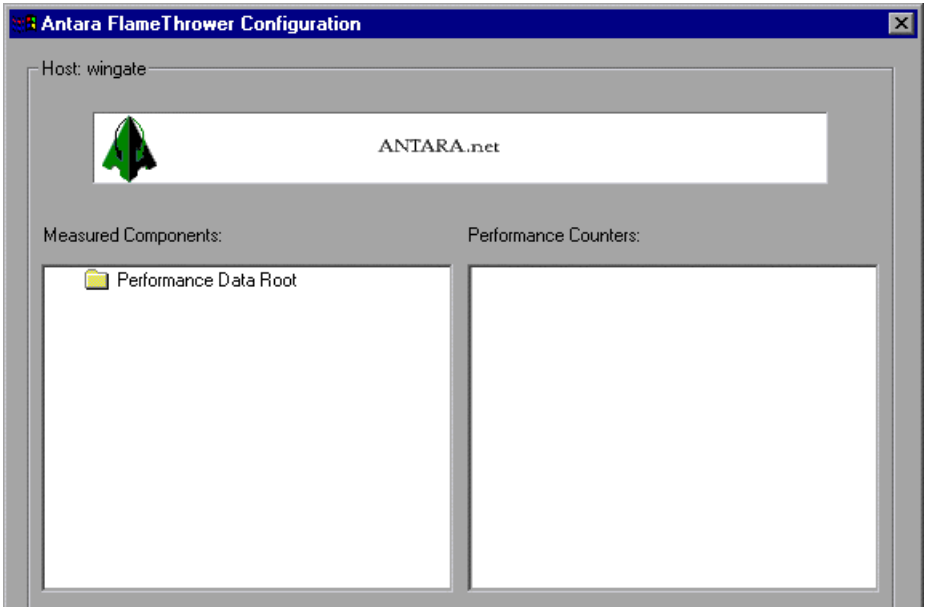
Adding a Monitored Machine to the Tuning Console

1. Click **Monitors** to open the Monitors Configuration dialog box.
2. Select the server whose monitors you want to configure from the Server list box.
3. Click **Add Monitor** and select **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.
4. In the left section of the dialog box, expand the System Resources category, select **Antara FlameThrower**, and then click **Add**.

   Continue with Configuring the Antara FlameThrower Monitor below.

Configuring the Antara FlameThrower Monitor

1. The Antara FlameThrower Monitor Configuration dialog box opens.



Browse the Measured Components tree, and select performance counters, as described in “Understanding the Antara FlameThrower Dialog Box” on page 57, and then click **OK**.

For a description of the available measurements, see “Antara FlameThrower Performance Counters” on page 58.

2 In the Controller, click **OK** in the Antara FlameThrower Monitor Configuration dialog box, and in the Antara FlameThrower dialog box, to activate the monitor.

3 In the Tuning Console, click **OK** in the Antara FlameThrower Monitor Configuration dialog box, and in the Select Measurements to Monitor dialog box, to activate the monitor.

Understanding the Antara FlameThrower Dialog Box

The Antara FlameThrower dialog box lets you select the items to monitor on the Antara FlameThrower application server.

Host: Displays the name of the monitored machine.

Measured Components: Displays the available components. Browse the tree and select the component you want to monitor. A description of the highlighted component appears in the Component/Counter Description box.

Performance Counters: Check the required performance counters. A description of the selected counter appears in the Component/Counter Description box.

Component/Counter Description: Displays a description of the selected component or counter.

Antara FlameThrower Performance Counters

The following tables describe the counters that can be monitored:

Layer Performance Counters

Measurement	Description
TxBytes	The total number of Layer 2 data bytes transmitted.
TxByteRate(/sec)	The number of Layer 2 data bytes transmitted per second.
TxFrames	The total number of packets transmitted.
TxFrameRate(/sec)	The number of packets transmitted per second.
RxBytes	The total number of Layer 2 data bytes received.
RxByteRate(/sec)	The number of Layer 2 data bytes received per second.
RxFrames	The total number of packets received.
RxFrameRate(/sec)	The number of packets received per second.

TCP Performance Counters

Measurement	Description
ActiveTCPConns	Total number of currently active TCP connections.
SuccTCPConns	Total number of SYN ACK packets received.
SuccTCPConn Rate(/sec)	Number of SYN ACK packets received per second.
TCPConnLatency (milisec)	Interval between transmitting a SYN packet and receiving a SYN ACK reply packet in msec.
MinTCPConn Latency(milisec)	Minimum TCPConnectionLatency in msec.
MaxTCPConn Latency(milisec)	Maximum TCPConnectionLatency in msec.
TCPSndConnClose	Total number of FIN or FIN ACK packets transmitted (Client).

Measurement	Description
TCPRcvConnClose	Total number of FIN or FIN ACK packets received (Client).
TCPSndResets	Total number of RST packets transmitted.
TCPRcvResets	Total number of RST packets received.
SYNSent	Total number of SYN packets transmitted.
SYNSentRate(/sec)	Number of SYN packets transmitted per second.
SYNAckSent	Total number of SYN ACK packets transmitted.
SYNAckRate(/sec)	Number of SYN ACK packets transmitted per second.

HTTP Performance Counters

Measurement	Description
HTTPRequests	Total number of HTTP Request command packets transmitted.
HTTPRequestRate (/sec)	Number of HTTP Request packets transmitted per second.
AvgHTTPData Latency(milisecs)	The average HTTP Data Latency over the past second in msec.
HTTPData Latency(milisecs)	Interval between transmitting a Request packet and receiving a response in msec.
DataThroughput (bytes/sec)	The number of data bytes received from the HTTP server per second.
MinHTTPData Latency(milisecs)	Minimum HTTPDataLatency in msec.
MaxHTTPData Latency(milisecs)	Maximum HTTPDataLatency in msec.
MinData Throughput (bytes/sec)	Minimum HTTPDataThroughput in seconds.

Measurement	Description
MaxData Throughput (bytes/sec)	Maximum HTTPDataThroughput in seconds.
SuccHTTPRequests	Total number of successful HTTP Request Replies (200 OK) received.
SuccHTTPRequest Rate(/sec)	Number of successful HTTP Request Replies (200 OK) received per second.
UnSuccHTTP Requests	Number of unsuccessful HTTP Requests.

SSL/HTTPS Performance Counters

Measurement	Description
SSLConnections	Number of ClientHello messages sent by the Client.
SSLConnection Rate(/sec)	Number of ClientHello messages sent per second.
SuccSSL Connections	Number of successful SSL Connections. A successful connection is one in which the Client receives the Server's finished handshake message without any errors.
SuccSSLConnection Rate(/sec)	Number of successful SSL connections established per second.
SSLAlertErrors	Number of SSL alert messages received by the client (e.g. bad_record_mac, decryption_failed, handshake_failure, etc.)
SuccSSLResumed Sessions	Number of SSL Sessions that were successfully resumed.
FailedSSLResumed Sessions	Number of SSL Sessions that were unable to be resumed.

Sticky SLB Performance Counters

Measurement	Description
Cookie AuthenticationFail	The number of Cookie's that were not authenticated by the Server.
SuccCookie Authentication	The number of Cookie's authenticated by the server.
SSLClientHellos	The number of Client Hello packets sent to the server.
SSLServerHellos	The number of Server Hello packets sent to back to the client.
SSLSessionsFailed	The number of Session ID's that were not authenticated by the server.
SSLSessions Resumed	The number of Session ID's authenticated by the server.
succSSLClientHellos	The number of Client Hello replies received by the client or packets received by the server.
succSSLServerHellos	The number of Server Hello's received by the client.

FTP Performance Counters

Measurement	Description
TPUsers	Total number of Ftp User command packets transmitted.
FTPUserRate(/sec)	Number of Ftp User command packets transmitted per second.
FTPUserLatency (milisecs)	Interval between transmitting a Ftp User command packet and receiving a response in msec.
MinFTPUserLatency(milisecs)	Minimum FTPUsersLatency in msec.
MaxFTPUserLatency(milisecs)	Maximum FTPUsersLatency in msec.
SuccFTPUsers	Total number of successful Ftp User command replies received.

Measurement	Description
SuccFTPUserRate (/sec)	Number of successful FTP User command replies received per second.
FTPPasses	Total number of FTP PASS packets transmitted.
FTPPassRate(/sec)	Number of FTP PASS packets transmitted per second.
FTPPassLatency (milisecs)	Interval between transmitting a Ftp PASS packet and receiving a response in msec.
MinFTPPassLatency (milisecs)	Minimum FTPPassLatency in msec.
MaxFTPPassLatency (milisecs)	Maximum FTPPassLatency in msec.
SuccFTPPasses	Total number of successful FTP PASS replies received.
SuccFTPPassRate (/sec)	Number of successful FTP PASS replies received per second.
FTPControl Connections	Total number of SYN packets transmitted by the FTP client.
FTPControl ConnectionRate (/sec)	Number of SYN packets transmitted by the FTP client per second.
SuccFTPControl Connections	Total number of SYN ACK packets received by the FTP client.
SuccFTPControl ConnectionRate (/sec)	Number of SYN ACK packets received by the FTP Client per second.
FTPData Connections	Number of SYN ACK packets received by the FTP client per second.
FTPDataConnection Rate(/sec)	Number of SYN ACK packets transmitted by the FTP Client or received by the FTP Server per second.
SuccFTPData Connections	Total number of SYN ACK packets transmitted by the FTP Client or received by the FTP Server.

Measurement	Description
SuccFTPData ConnectionRate (/sec)	Number of SYN ACK packets received by the FTP server per second.
FtpAuthFailed	Total number of error replies received by the FTP client.
FTPGets	Total number of client Get requests.
FTPPuts	Total number of client Put requests.
SuccFTPGets	Total number of successful Get requests (data has been successfully transferred from server to client).
SuccFTPPuts	Total number of successful Put requests (data has been successfully transferred from client to server).

SMTP Performance Counters

Measurement	Description
SMTPHelos	Total number of HELO packets transmitted.
SMTPHeloRate(/sec)	Number of HELO packets transmitted per second.
SMTPHeloLatency (milisecs)	Interval between transmitting a HELO packet and receiving a response in msec.
MinSMTPHelo Latency(milisecs)	Minimum SMTPHeloLatency in msec.
MaxSMTPHelo Latency(milisecs)	Maximum SMTPHeloLatency in msec.
SuccSMTPHelos	Total number of successful HELO replies received.
SuccSMTPHelo Rate(/sec)	Number of successful HELO replies received per second.
SMTPMailFroms	Total number of Mail From packets transmitted.
SMTPMailFromRate (/sec)	Number of Mail From packets transmitted per second.
SMTPMailFrom Latency(milisecs)	Interval between transmitting a Mail From packet and receiving a response in msec.

Measurement	Description
MinSMTPMailFrom Latency(milisecs)	Minimum SMTPMailFromLatency in msec.
MaxSMTPMailFrom Latency(milisecs)	Maximum SMTPMailFromLatency in msec.
SuccSMTPMail Froms	Total number of successful Mail From replies received.
SuccSMTPMailFrom Rate(/sec)	Number of successful Mail From replies received per second.
SMTPRcptTos	Total number of RcptTo packets transmitted.
SMTPRcptToRate (/sec)	Number of RcptTo packets transmitted per second.
SMTPRcptTo Latency(milisecs)	Interval between transmitting a RcptTo packet and receiving a response in msec.
MinSMTPRcptTo Latency(milisecs)	Minimum SMTPRcptToLatency in msec.
MaxSMTPRcptTo Latency(milisecs)	Maximum SMTPRcptToLatency in msec.
SuccSMTPRcptTos	Total number of successful RcptTo replies received.
SuccSMTPRcptTo Rate(/sec)	Number of successful RcptTo replies received per second.
SMTPDatas	Total number of Data packets transmitted.
SMTPDataRate(/sec)	Number of Data packets transmitted per second.
SMTPDataLatency (milisecs)	Interval between transmitting a Data packet and receiving a response in msec.
MinSMTPData Latency(milisecs)	Minimum SMTPDataLatency in msec.
MaxSMTPData Latency(milisecs)	Maximum SMTPDataLatency in msec.

Measurement	Description
SuccSMTPDatas	Total number of successful Data replies received.
SuccSMTPDataRate (/sec)	Number of successful Data replies received per second.

POP3 Performance Counters

Measurement	Description
POP3Users	Total number of Pop3 User command packets transmitted.
POP3UserRate(/sec)	Number of Pop3 User command packets transmitted per second.
POP3UserLatency (milisecs)	Interval between transmitting a Pop3 User command packet and receiving a response in msec.
MinPOP3User Latency(milisecs)	Minimum POP3UserLatency in msec.
MaxPOP3User Latency(milisecs)	Maximum POP3UserLatency in msec.
SuccPOP3Users	Total number of successful Pop3 User replies received.
SuccPOP3UserRate (/sec)	Number of successful Pop3 User replies received per second.
POP3Passes	Total number of Pop3 Pass command packets transmitted.
POP3PassRate(/sec)	Number of Pop3 Pass command packets transmitted per second.
POP3PassLatency (milisecs)	Interval between transmitting a Pop3 Pass packet and receiving a response in msec.
MinPOP3Pass Latency(milisecs)	Minimum POP3PassLatency in msec.
MaxPOP3Pass Latency(milisecs)	Maximum POP3PassLatency in msec.
SuccPOP3Passes	Total number of successful Pop3 Pass replies received.

Measurement	Description
SuccPOP3PassRate (/sec)	Number of successful Pop3 Pass replies received per second.
POP3Stats	Total number of Pop3 Stat command packets sent.
POP3StatRate(/sec)	Number of Pop3 Stat command packets transmitted per second.
POP3StatLatency (milisecs)	Interval between transmitting a Pop3 Stat packet and receiving a response in msec.
MinPOP3Stat Latency(milisecs)	Minimum POP3StartLatency in msec.
MaxPOP3Stat Latency(milisecs)	Maximum POP3StartLatency in msec.
SuccPOP3Stats	Total number of successful Pop3 Stat replies received.
SuccPOP3StatRate (/sec)	Number of successful Pop3 Stat replies received per second.
POP3Lists	Total number of Pop3 List command packets transmitted.
POP3ListRate(/sec)	Number of Pop3 List command packets transmitted per second.
POP3ListLatency (milisecs)	Interval between transmitting a Pop3 List packet and receiving a response in msec.
MinPOP3List Latency(milisecs)	Minimum POP3ListLatency in msec.
MaxPOP3List Latency(milisecs)	Maximum POP3ListLatency in msec.
SuccPOP3Lists	Total number of successful Pop3Lists received.
SuccPOP3ListRate (/sec)	Number of successful Pop3Lists received per second.
POP3Retrs	Total number of Pop3 Retr packets transmitted.
POP3RetrRate(/sec)	Number of Pop3 Retr packets transmitted per second.
POP3RetrLatency (milisecs)	Interval between transmitting a Pop3 Retr packet and receiving a response in msec.

Measurement	Description
MinPOP3Retr Latency(milisecs)	Minimum POP3RetrLatency in msec.
MaxPOP3Retr Latency(milisecs)	Maximum POP3RetrLatency in msec.
SuccPOP3Retrs	Total number of successful Pop3Retrs received.
SuccPOP3RetrRate (/sec)	Number of successful Pop3Retrs received per second.

DNS Performance Counters

Measurement	Description
SuccPrimaryDNS Request	Total number of Successful DNS requests made to the Primary DNS server.
SuccSecondaryDNS Request	Total number of Successful DNS requests made to the Secondary DNS server.
SuccDNSData RequestRate(/sec)	Number of Successful DNS Request packets transmitted per second.
PrimaryDNSFailure	Total number of DNS requests failures received from the Primary DNS server.
PrimaryDNSRequest	Total number of DNS requests made to the Primary DNS server.
SecondaryDNS Failure	Total number of DNS requests failures received from the Secondary DNS server.
SecondaryDNS Request	Total number of DNS requests made to the Secondary DNS server.
MinDNSData Latency	Minimum DNS Data Latency in msec.
MaxDNSData Latency	Maximum DNS Data Latency in msec.
CurDNSData Latency	Interval between sending a DNS request packet and receiving a response in msec.

Measurement	Description
DNSDataRequest Rate(/sec)	Number of DNS Request packets transmitted per second.
NoOf ReTransmission	Total number of DNS Request packets retransmitted.
NoOfAnswers	Total number of Answers to the DNS Request packets.

Attacks Performance Counters

Measurement	Description
Attacks	Total number of attack packets transmitted (All Attacks)
AttackRate(/sec)	Number of attack packets transmitted per second (ARP, Land, Ping, SYN, and Smurf)
Havoc Flood	Number of Havoc packets generated (Stacheldraht only)
Icmp Flood	Number of ICMP attack packets generated (TFN, TFN2K, & Stacheldraht)
Mix Flood	Number of Mix packets generated (TFN2K only)
Mstream Flood	Number of Mstream packets generated (Stacheldraht only)
Null Flood	Number of Null packets generated (Stacheldraht only)
Smurf Flood	Number of Smurf packets generated (TFN, TFN2K, & Stacheldraht)
Syn Flood	Number of SYN packets generated (TFN, TFN2K, & Stacheldraht)
Targa Flood	Number of Targa packets generated (TFN2K only)
Udp Flood	Number of UDP packets generated (All DDoS Attacks only)

SiteScope Monitoring

The SiteScope Resources monitor graph shows the SiteScope resources measured during the scenario or session step. The SiteScope monitor can measure server, network, and processor performance counters. For detailed information on the performance counters that SiteScope can monitor, refer to the relevant SiteScope documentation.

To obtain data for this graph, you need to select the desired measurements for the online monitor (from the Controller or Tuning Console) before running the scenario or session step.

Setting up the SiteScope Monitor

You select measurements to poll from SiteScope using the SiteScope Monitor Configuration dialog box.

Before setting up the SiteScope monitor:

➤ Make sure that SiteScope has been installed on a server. Although you can install SiteScope on the Controller or Tuning Console machine, we recommend installing it on a dedicated server. If SiteScope is installed on a machine other than the Controller, verify that the SiteScope machine is accessible from the Controller machine.

➤ On the machine where SiteScope is installed, configure SiteScope to monitor the required servers. When you assign a name to a monitor, include the server name in the monitor name. This avoids any confusion about the host to which the machine belongs.

Note: The monitor title cannot include "\", (for example, "CPU Utilization on \\Superman".

SiteScope's default sampling rate is 10 minutes, and its minimum rate 15 seconds.

➤ Verify that SiteScope is collecting the required data from the machines it is monitoring. From the SiteScope Panel, select the monitor group polling the server machines, and check that the monitor displays a list of server measurements in the Status column.

For Tuning Console only:

➤ In the System Topology window, add a SiteScope Server element for the machine on which SiteScope is running.

You can configure the SiteScope monitor in one of the following ways:

➤ In the System Topology window

➤ In the Session tab

Adding a Monitored Machine to the Controller

1. Click the SiteScope graph in the graph tree, and drag it into the right pane of the Run view.
2. Right-click the graph and select **Add Measurements**, or choose **Monitors** > **Add Measurements**. The SiteScope dialog box opens.
3. In the **Monitored Server Machines** section, click **Add**.
4. Enter the server name or IP address of the machine you want to monitor, and select the platform on which the machine runs, and then click **OK**.
5. In the **Resource Measurements** section of the SiteScope dialog box, click **Add**.

Continue with Configuring the SiteScope Monitor below.

Adding a Monitored Machine to the Tuning Console

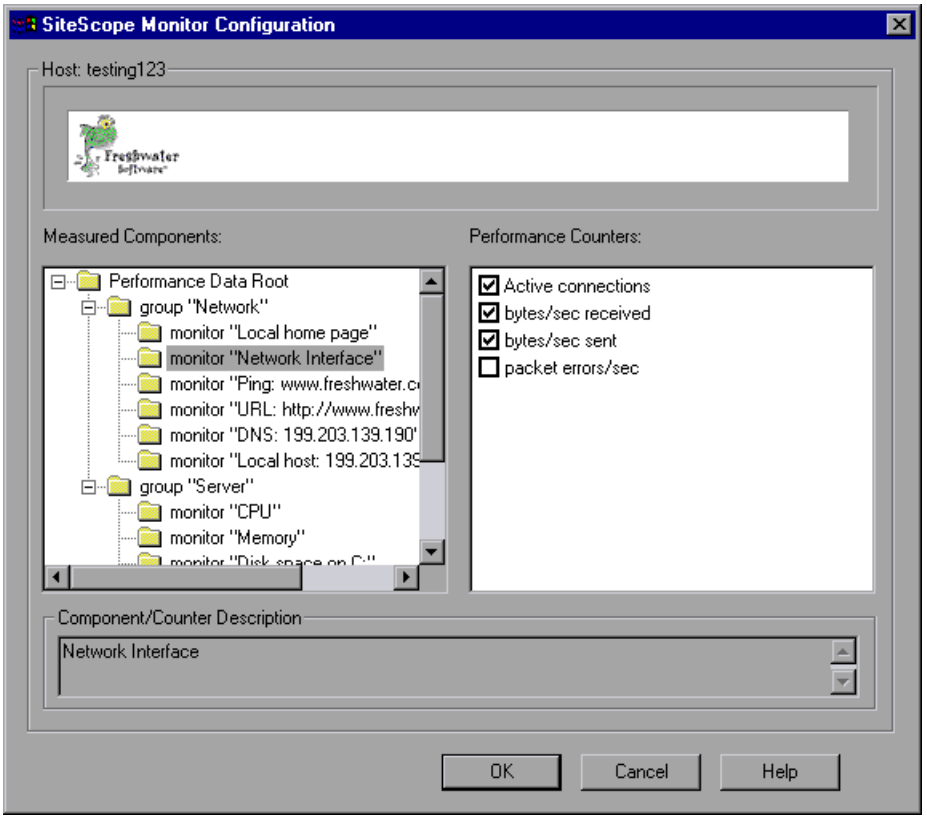
1. Click **Monitors** to open the Monitors Configuration dialog box.
2. From the Server list, choose the SiteScope server that you want to monitor.
3. Click **Add Monitor** and select **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the SiteScope Server category, select **SiteScope**, and then click **Add**.

Continue with Configuring the SiteScope Monitor below.

Configuring the SiteScope Monitor

1 The SiteScope Monitor Configuration dialog box opens displaying the available measurements.



2 Browse the Measured Components tree, and select performance counters, as described in "Understanding the SiteScope Dialog Box" on page 72, and then click **OK**.

3 In the Controller, click **OK** in the SiteScope Monitor Configuration dialog box and in the SiteScope dialog box to activate the monitor.

4 In the Tuning Console, click **OK** in the Select Measurements to Monitor dialog box, and in the Monitors Configuration dialog box, to activate the monitor.

Note: SiteScope can only be monitored by one Controller or Tuning Console at a time.

Understanding the SiteScope Dialog Box

Lets you select the items to monitor on the SiteScope application server.

Host: Displays the name of the monitored machine.

Measured Components: Displays the available components. Browse the tree and select the component you want to monitor. A description of the highlighted component appears in the Component/Counter Description box.

Performance Counters: Check the required performance counters. A description of the selected counter appears in the Component/Counter Description box.

Component/Counter Description: Displays a description of the selected component or counter.

5

Network Monitoring

You use Network monitoring to determine whether your network is causing a delay in the scenario or session step. You can also determine the problematic network segment.

Note: To run the Network monitor, you must have administrator privileges on the Windows source machine (unless you are using the ICMP protocol).

This chapter describes:

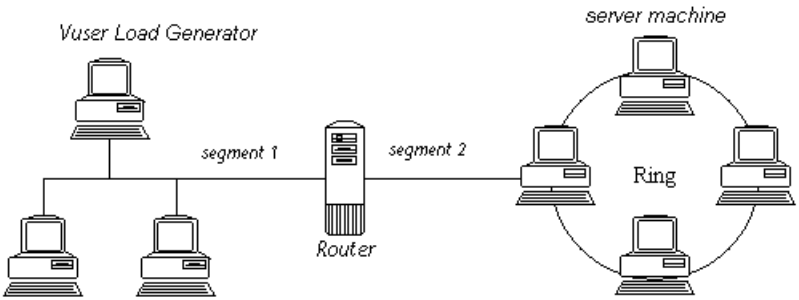
- About Network Monitoring
- Network Monitoring from a UNIX Source Machine
- Network Delay Monitoring
- Viewing the Network Delay Time Graph

About Network Monitoring

Network configuration is a primary factor in the performance of applications. A poorly designed network can slow client activity to unacceptable levels.

In a true Web or client/server system, there are many network segments. A single network segment with poor performance can affect the entire system.

The following diagram shows a typical network. To go from the server machine to the Vuser machine, data must travel over several segments.



To measure network performance, the Network monitor sends packets of data across the network. When a packet returns, the monitor calculates the time it takes for the packet to go to the requested node and return. This time is the delay which appears in the Network Delay Time graph.

Using the online Network Delay Time graph, you can locate the network-related problem so that it can be fixed.

Note: The delays from the source machine to each of the nodes are measured concurrently, yet independently. It is therefore possible that the delay from the source machine to one of the nodes could be greater than the delay for the complete path between the source and destination machines.

Network Monitoring from a UNIX Source Machine

You can run the Network monitor on UNIX machines, using UDP or ICMP. Before running the Network monitor from a UNIX source machine:

- configure the source machine by assigning root permissions to the **merc_webtrace** process.
- make the necessary adjustments to either connect to the source machine through RSH, or through the agent.

Configuring the Source Machine

To configure the source machine, where LoadRunner is installed locally:

To assign root permissions to the **merc_webtrace** process, add an s-bit to **merc_webtrace**'s permissions, as follows:

1. Log in to the source machine as root.
2. Type: cd <LoadRunner_installation>/bin to change to the **bin** directory.
3. Type: chown root merc_webtrace to make the root user the owner of the **merc_webtrace** file.
4. Type: chmod +s merc_webtrace to add the s-bit to the file permissions.
5. To verify, type ls -l merc_webtrace. The permissions should look like: -rwsrwsr-x.

To configure the source machine, where LoadRunner is installed on the network:

In a LoadRunner network installation, the **merc_webtrace** process is on the network, not on the source machine disk. The following procedure copies the **merc_webtrace** file to the local disk, configures **mdrv.dat** to recognize the process, and assigns root permissions to **merc_webtrace**:

1 Copy **merc_webtrace** from <**LoadRunner_installation**>/**bin** to anywhere on the local disk of the source machine. For example, to copy the file to the /**local**/<**LoadRunner**> directory, type: cp /net/tools/LoadRunner_installation/bin/merc_webtrace /local/<LoadRunner>

Note: All of the source machines that use the same network installation must copy **merc_webtrace** to the identical directory path on their local disk (for example, /local/<LoadRunner>), since all of them use the same **mdrv.dat**.

---

2 Add the following line to the <**LoadRunner_installation**>/**dat**/**mdrv.dat** file, in the [monitors_server] section:

```
ExtCmdLine=-merc_webtrace_path /local/xxx
```

3 Log in to the source machine as root.

4 Type: cd LoadRunner_installation/bin to change to the **bin** directory.

5 Type: chown root merc_webtrace to make the root user the owner of the **merc_webtrace** file.

6 Type: chmod +s merc_webtrace to add the s-bit to the file permissions.

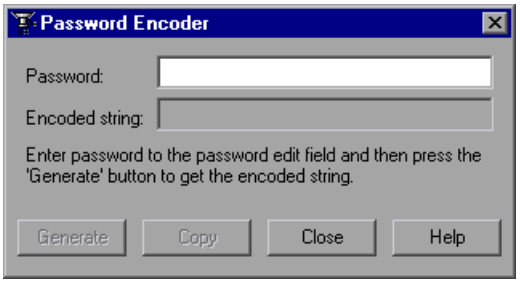
7 To verify, type ls -l merc_webtrace. The permissions should look like: -rwsrwsr-x.

Connecting to the Source Machine Through RSH

If the Controller or Tuning Console is connected to the source machine through RSH (default connection mode), then you don't need to activate the agent daemon. Before running the Network monitor the first time, you enter an encrypted user name and password in the Network monitor configuration file.

To create an encrypted user name and password:

1 On the Windows taskbar, click **Start**, point to **Programs** > **LoadRunner** > **Tools**, and click **Password Encoder**. The Password Encoder window opens.



2 In the **Password** box, type your RSH user name and password, separated by a vertical bar symbol. For example, myname|mypw.

3 Click **Generate**. An encoded string is displayed in the Encoded string field.

4 Click **Copy** to copy the encoded string to the clipboard.

5 Add the following line to the **<LoadRunner_installation>/dat/monitors/ndm.cfg** file, in the [hosts] section:

Host = <encrypted string copied from clipboard>

6 Close and open the current scenario or session step. LoadRunner will read the updated configuration file and recognize the source machine for monitoring.

Connecting to the Source Machine Through the Agent

If the Controller or Tuning Console is not connected to the source machine through RSH, then make sure that the agent daemon is active on the source machine before running the Network monitor. For more information about working without RSH, refer to the section titled "UNIX Shell" in Appendix D, "Troubleshooting the Controller" in the *LoadRunner Controller User's Guide* or *Mercury Tuning Module Console User's Guide*.

To activate the agent daemon:

If you are not working in RSH, invoke the agent daemon on the source machine.

1. Type m_daemon_setup -install from the **<LoadRunner_installation>/bin** directory.
2. Make sure that the agent daemon is running whenever you activate the Network monitor.
3. To stop the Network Delay Monitor agent daemon, type m_daemon_setup -remove.

Network Delay Monitoring

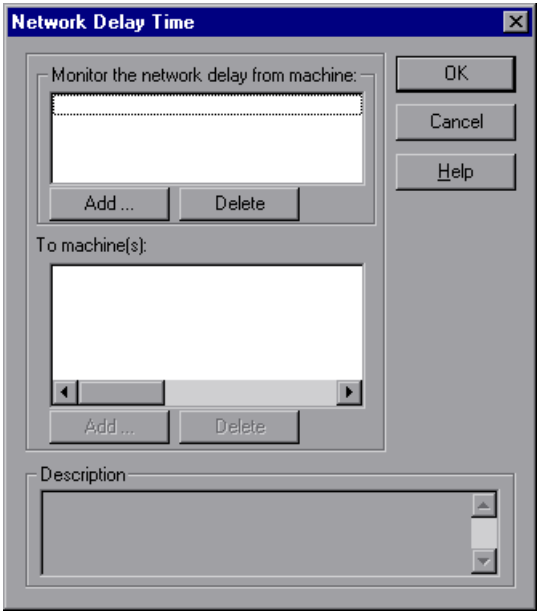
The Network Delay Time monitor shows the delays for the complete path between the source and destination machines (for example the database server and Vuser host). The graph maps the delay as a function of the elapsed scenario or session step time. Each defined path is represented by a separate line with a different color in the graph.

You configure the Network monitor from the Run view of the Controller or the topology window of the Tuning Console before you begin running a scenario or session step.

Note: To enable network monitoring, you must install the LoadRunner agent on the source machine. You do not have to install the LoadRunner agent on the destination machine.

Adding a Monitored Machine to the Controller

1. In the graph tree view, select the **Network Delay Time** graph and drag it into the right pane.
2. Right-click the graph and select **Add Measurements**, or choose **Monitors** > **Add Measurements**. The Network Delay Time dialog box opens.



3 In the **Monitor the network delay from machine** section, click **Add**, and enter the server name or IP address of the source machine, from which you want the network path monitoring to begin. Select the platform on which the machine runs, and click **OK**.

4 In the **To machine(s)** section of the Network Delay Time dialog box, click **Add** to enter the name of the machine at the final destination of the path you want to monitor.

Continue with Configuring the Network Delay Time Monitor below.

Adding a Monitored Machine to the Tuning Console

1 Click **Monitors** to display the Monitors Configuration dialog box.

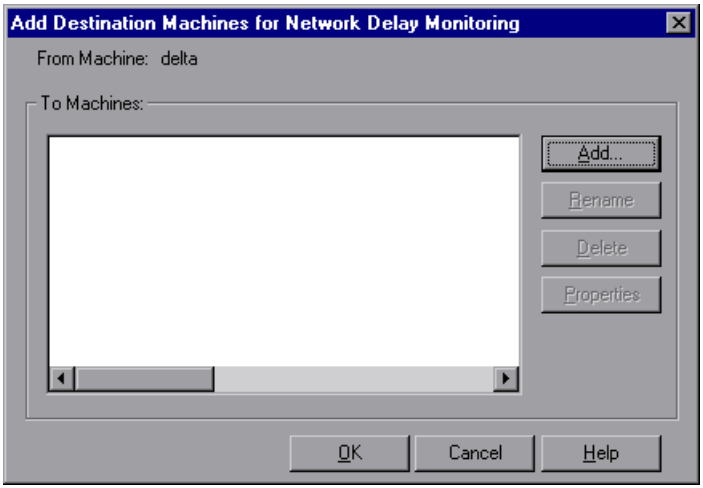
2 Click **Add Monitor** and select **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

3 In the list of monitors in the left section, expand the Load Generator category, click **Network Delay**, and then click **Add**.

Continue with Configuring the Network Delay Time Monitor below.

Configuring the Network Delay Time Monitor

1 The Add Destination Machines for Network Delay Monitoring dialog box opens.



To add a machine, click **Add**. The New Machine Name dialog box opens.

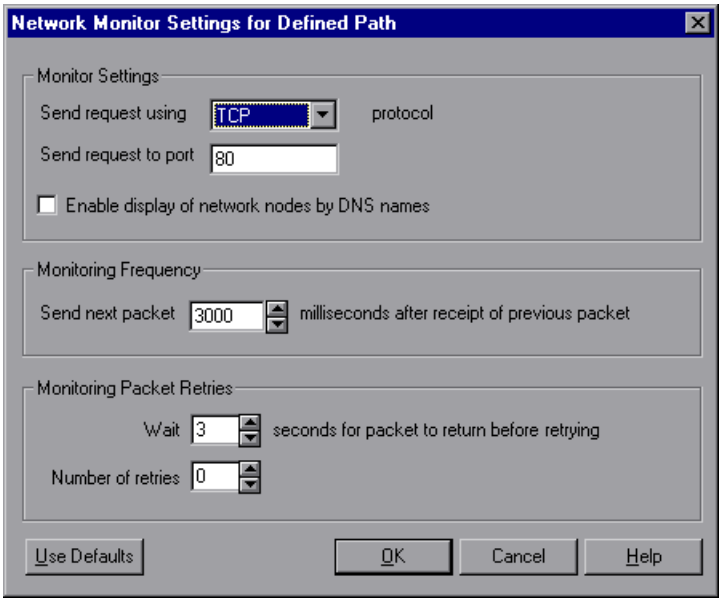
2 Enter the name of the destination machine and click **OK**. The name of the machine appears in the Add Destination Machines for Network Delay Monitoring dialog box. Repeat this procedure for each path you want to monitor, and then click **OK**.

Note: If the destination machine is localhost, enter the local machine's name and not localhost.

To rename a machine, click **Rename**, and enter a new name for the machine.

To delete a machine, select it and click **Delete**.

3 Click **Properties** to configure additional network monitor settings. The Network Monitor Settings for Defined Path dialog box opens.



4 In the Monitor Settings box, select the protocol and enter the port number being used by the network path. The Network monitor supports three protocols: TCP, UDP, and ICMP. It is recommended that you use the default protocol. In Windows, the default is TCP, and in UNIX, the default is UDP.

5 Select **Enable display of network nodes by DNS names** if you want to view the DNS name of each node along the network path, in addition to its IP address. Note that selecting this option will decrease the speed of the Network monitor.

6 In the Monitoring Frequency box, select the number of milliseconds the monitor should wait between receiving a packet and sending out the next packet. The default value is 3000 milliseconds. If you have a long, steady scenario or session step, you can increase the interval by several seconds.

7 In the Monitoring Packet Retries box, select the maximum number of seconds that the monitor should wait for a packet to return before it retries to send the packet. The default value is 3 seconds. If your network is very large and loaded (an internet connection with a low capacity), you should increase the value by several seconds. If you have a small network (such as a LAN), you can decrease the value.

In addition, select the number of times the Network monitor should try resending a packet to a node if the packet is not initially returned. The default value is 0.

Understanding the Network Delay Time Dialog Box (Controller only)

The Network Delay Time dialog box enables you to select the network path you want to monitor.

Note: To run the Network monitor, you must have administrator privileges on the source machine (unless you are using the ICMP protocol).

Monitor the network delay from machine: Displays the name of the machine from which network monitoring begins. Click **Add** to enter the name of a machine.

- **Add:** Opens the Add Machine dialog box. Enter the server name or IP address of the source machine, from which you want the network path monitoring to begin. Select the platform on which the machine runs, and click **OK**. Repeat this for each path you want to monitor.
- **Delete**: Removes the source machine.

To machine(s):

- **Add**: Opens the Adding Destination Machines for Network Delay Monitoring dialog box.
- **Delete**: Removes the destination machine.

Description: Displays the name and platform of the highlighted machine.

Adding Destination Machines for Network Delay Monitoring dialog box

Enables you to add destination machines for network delay monitoring, and configure additional network monitor settings.

From Machine: Displays the name of the source machine.

To Machines: Displays the names or URLs of the destination machines.

Add: Enter the name or URL of the machine at the final destination of the path you want to monitor in the New Machine Name dialog box. Repeat this for each path you want to monitor.

Rename: Renames the destination machine.

Delete: Deletes the destination machine.

Properties: Opens the Configuring Network Monitor Settings for Defined Path dialog box.

Configuring Network Monitor Settings for Defined Path dialog box

Lets you set the protocol, port, monitoring frequency, and monitoring packet retries.

Monitor Settings

➤ **Send request using X protocol:** Select the network protocol you want the monitor to use. Note that the Network monitor supports three protocols: TCP, UDP, and ICMP. It is recommended that you use the default protocol, TCP, unless it is necessary for you to use UDP or ICMP.

Note: When you use TCP or UDP protocols, administrator privileges are required on the source machine.

➤ **Send request to port:** Enter the port number to be used by the network path.

➤ **Enable display of network nodes by DNS names:** Enables you to view the DNS name of each node along the network path, in addition to its IP address. Note that selecting this option will decrease the speed of the Network monitor.

Monitoring Frequency

➤ **Send next packet X milliseconds after receipt of previous packet:** Select the number of milliseconds the monitor should wait between receiving a packet and sending out the next packet. The default value is 3000 milliseconds. Note that if you have a long, steady scenario or session, you can increase the interval by several seconds.

Monitoring Packet Retries

➤ **Wait X seconds for packet to return before retrying:** Select the maximum number of seconds that the monitor should wait for a packet to return before it retries to send the packet. The default value is 3 seconds.

➤ If your network is very large and loaded (an internet connection with a low capacity), you should increase the value by several seconds. If you have a small network (such as a LAN), you can decrease the value.

➤ **Number of retries:** Select the number of times the monitor should try resending a packet to a node if the packet is not initially returned. The default value is 0.

Use Defaults: Sets all settings to their default values.

Network Monitoring over a Firewall

If you are monitoring a network in which there are firewalls between the source and the destination machines, you must configure the firewalls to allow the network data packets to reach their destinations.

➤ If you are using the TCP protocol, the firewall that protects the destination machine should not block outgoing ICMP_TIMEEXCEEDED packets (packets that are sent outside the firewall from the machine). In addition, the firewall protecting the source machine should allow ICMP_TIMEEXCEEDED packets to enter, as well as TCP packets to exit.

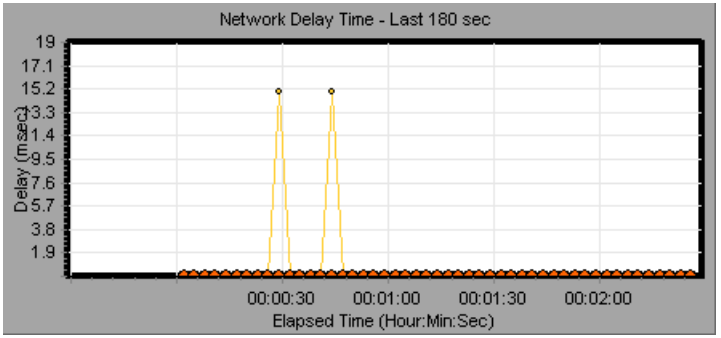
- If you are using the ICMP protocol, the destination machine's firewall should not block incoming ICMP_ECHO_REQUEST packets, or outgoing ICMP_ECHO_REPLY and ICMP_ECHO_TIMEEXCEEDED packets. In addition, the firewall protecting the source machine should allow ICMP_ECHO_REPLY and ICMP_ECHO_TIMEEXCEEDED packets to enter, and ICMP_ECHO_REQUEST packets to exit.
- If you are using the UDP protocol, ensure that the UDP protocol can access the destination machine from the source machine. The destination machine's firewall should not block outgoing ICMP_DEST_UNREACHABLE and ICMP_ECHO_TIMEEXCEEDED packets. In addition, the firewall protecting the source machine should allow ICMP_DEST_UNREACHABLE and ICMP_ECHO_TIMEEXCEEDED packets to enter.

Note: To run the Network Delay Monitor when there are firewalls between the Controller or Tuning Console machine and the source machine, you must configure the LoadRunner agent, MI Listener, and Network monitor for monitoring over a firewall. For more information see "Configuring LoadRunner Agents Inside the Firewall", "Installing and Configuring the MI Listener Outside the Firewall", and "Configuring the Network Delay Monitor over a Firewall" in the *LoadRunner Controller User's Guide* or *Mercury Tuning Module Console User's Guide*.

Viewing the Network Delay Time Graph

The **Network Delay Time** graph shows the delay for the complete path between the source and destination machines (y-axis) as a function of the elapsed scenario or session step time (x-axis).

Each path defined in the Add Destination Machines for Network Delay Monitoring dialog box is represented by a separate line with a different color in the graph.



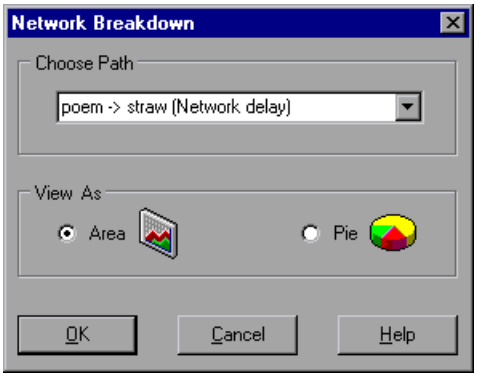
To view the DNS names of the measurements displayed in the legend, right-click the graph and select **View as DNS Name**.

To view the delay time from the source machine to each of the nodes along the network path, right-click the graph and select **Configure**. In the Graph Configuration dialog box, click **SubPaths**.

In addition, you can view the delay time for each segment of the path.

To view the delay time for the network segments:

1 Right-click the Network Delay Time graph, and select **View Segments**. The Network Breakdown dialog box opens.



2 Select the path that you want to break down.

3 Choose whether you want to view the network segments of the graph you chose as an area graph or a pie graph.

4 Click **OK** to close the Network Breakdown dialog box. The delay time for the network segments of the path you chose is displayed in the graph view area.

Note: The segment delays are measured approximately, and do not add up to the network path delay which is measured exactly. The delay for each segment of the path is estimated by calculating the delay from the source machine to one node and subtracting the delay from the source machine to another node. For example, the delay for segment B to C is calculated by measuring the delay from the source machine to point C, and subtracting the delay from the source machine to point B.

To return to the complete path delay time view, select **Hide Segments** from the right-click menu.

6

Firewall Server Performance Monitoring

To isolate server performance bottlenecks, you can monitor the firewall server during a scenario or session step run.

This chapter describes:

- About the Firewall Server Monitor
- CheckPoint FireWall-1 Server Monitoring

About the Firewall Server Monitor

The Firewall server online monitor measures the performance of a Firewall server during scenario or session step execution. To obtain performance data, you must activate the Firewall server monitor (before executing the scenario or session step) and indicate which statistics and measurements you want to monitor.

CheckPoint FireWall-1 Server Monitoring

The Check Point FireWall-1 monitor displays statistics about the resource usage on Check Point's FireWall during the scenario run.

To monitor the Check Point FireWall-1 server, you must select the counters you want the Check Point FireWall-1 server monitor to measure (from the Controller or Tuning Console). You select these counters using the Check Point FireWall-1 SNMP Resources dialog box.

Note: You can specify a port number in the **snmp.cfg** file. If you do not specify a port number, Mercury Tuning Console connects to port 260, the default port for the CheckPoint FireWall-1 SNMP agent.

Adding a Monitored Machine to the Controller

1. Click the Check Point FireWall-1 graph in the graph tree, and drag it into the right pane of the Run view.
2. Right-click the graph and select **Add Measurements**, or choose **Monitors > Add Measurements**. The Check Point FireWall-1 dialog box opens.
3. In the **Monitored Server Machines** section, click **Add.** The Add Machine dialog box opens.
4. Enter the server name or IP address of the machine you want to monitor. Select the platform on which the machine runs, and click **OK**.

Note: You can specify a machine name and port number in the Add Machine dialog box using the following format:
<machine name>:<port number>

---

5. In the **Resource Measurements** section of the Check Point FireWall-1 dialog box, click **Add**.

Continue with Configuring the Check Point FireWall-1 Server Monitor below.

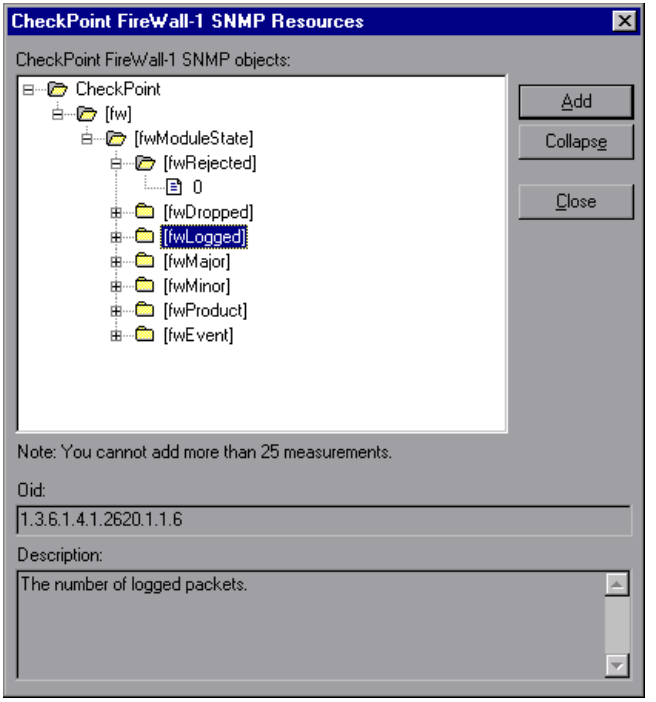
Adding a Monitored Machine to the Tuning Console

1. Click **Monitors** to open the Monitors Configuration dialog box.
2. Select the server whose monitors you want to configure from the Server list box.
3. Click **Add Monitor** and select **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.
4. In the left section of the dialog box, expand the Firewall category, select **CheckPoint FireWall-1**, and then click **Add**.

   Continue with Configuring the Check Point FireWall-1 Server Monitor below.

Configuring the Check Point FireWall-1 Server Monitor

1 The Check Point FireWall-1 SNMP Resources dialog box opens.



2 Browse the FireWall-1 Objects list and select the measurements you want to monitor, as described in “Understanding the CheckPoint FireWall-1 Dialog Box” on page 93.

For a description of the available measurements, see “Check Point FireWall-1 Performance Counters” on page 94.

3 Add all the desired resources to the list, and click **Close**.

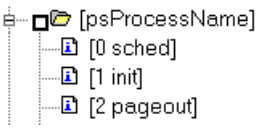
4 In the Controller, click **OK** in the Check Point FireWall-1 dialog box to activate the monitor.

5 In the Tuning Console, click **OK** in the Select Measurements to Monitor dialog box to activate the monitor.

Improving the Level of Measurement Information

You can improve the level of measurement information for the Check Point FireWall-1 monitor by enabling measurements with string values to be listed (in addition to measurements with numeric values), and by enabling the name modifier (which displays the string value as an identifying part of the measurement name).

In the following example of a measurement using the name modifier, the string value of ProcessName (sched) is displayed in addition to its instance ID (0):



To enable this feature, add the following line to the **<LoadRunner root folder>\dat\monitors\snmp.cfg** file:

```
SNMP_show_string_nodes=1
```

Usage Notes: You can select more than one name modifier, but the first in the hierarchy will be used. Each time the Check Point FireWall-1 Add Measurements dialog box opens, the information is reread from the **snmp.cfg** file. You cannot add the same measurement twice (once with a name modifier and once without it). If you do so, an error message is issued.

Understanding the CheckPoint FireWall-1 Dialog Box

The CheckPoint FireWall-1 dialog box lets you select the CheckPoint FireWall-1 server resources to monitor using the Simple Network Management Protocol (SNMP).

CheckPoint FireWall-1 SNMP Objects: Select each required object and click **Add**. Click **Explain** for the ID number and a description of the selected object.

Explain: Displays a description of the selected object.

Note: The CheckPoint FireWall-1 monitor can only monitor up to 25 measurements.

Check Point FireWall-1 Performance Counters

The following default counters can be monitored:

Measurement	Description
fwRejected	The number of rejected packets.
fwDropped	The number of dropped packets.
fwLogged	The number of logged packets.

7

Web Server Resource Monitoring

Using LoadRunner's Web Server Resource monitors, you monitor the Apache, Microsoft IIS, iPlanet (SNMP), and iPlanet/Netscape servers during a scenario or session step run and isolate server performance bottlenecks.

This chapter describes:

- About Web Server Resource Monitors
- Apache Monitoring
- Microsoft IIS Monitoring
- iPlanet/Netscape Monitoring
- iPlanet (SNMP) Monitoring
- Monitoring Using a Proxy Server

About Web Server Resource Monitors

Web Server Resource monitors provide you with information about the resource usage of the Apache, Microsoft IIS, iPlanet (SNMP), and iPlanet/Netscape Web servers during scenario or session step execution. To obtain this data, you need to activate the online monitor for the server and specify which resources you want to measure before executing the scenario or session step.

The procedures for selecting monitor measurements and configuring the monitors vary according to server type. The following sections contain specific configuration instructions for each server type.

Note: Certain measurements or counters are especially useful for determining server performance and isolating the cause of a bottleneck during an initial stress test on a Web server. For more information about these counters, see "Useful Counters for Stress Testing" on page 444.

Apache Monitoring

This Web Server Resource monitor displays statistics about the resource usage on the Apache server during the scenario or session step run.

To obtain data for this graph, you need to enable the Apache online monitor (from the Controller or Tuning Console) and select the default measurements you want to display, before running the scenario or session step.

Setting up the Monitoring Environment

To monitor an Apache server you need to know the server statistics information URL. A simple way to verify the statistics information URL is to try to view it through the browser.

The URL should be in the following format:

http://<server name/IP address>:<port number>/server-status?auto

For example:

http://stimpy:80/server-status?auto

Adding a Monitored Machine to the Controller

1. Click the Apache graph in the graph tree, and drag it into the right pane of the Run view.
2. Right-click the graph and select **Add Measurements**, or choose **Monitors > Add Measurements**. The Apache dialog box opens.

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

4 Enter the server name or IP address of the machine you want to monitor. Select the platform on which the machine runs, and click **OK**.

5 In the **Resource Measurements** section of the Apache dialog box, click **Add**.

Continue with Configuring the Apache Monitor below.

Adding a Monitored Machine to the Tuning Console

1 Click **Monitors** to open the Monitors Configuration dialog box.

2 Select the server whose monitors you want to configure from the Server list box.

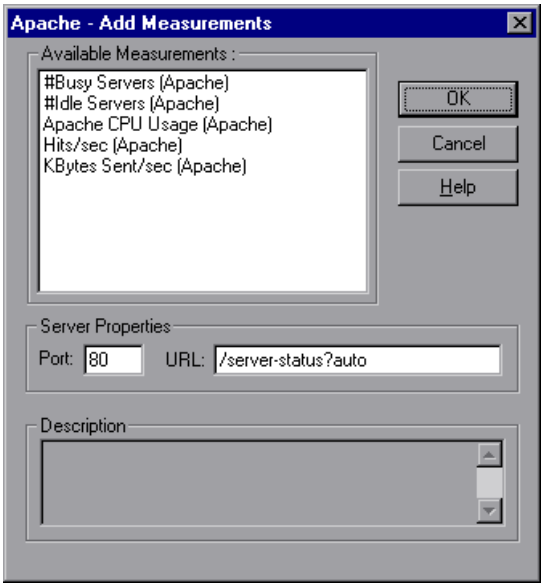
3 Click **Add Monitor** and select **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the Web Server category, select **Apache**, and then click **Add**.

Continue with Configuring the Apache Monitor below.

Configuring the Apache Monitor

1 The Apache Add Measurements dialog box opens, displaying the available measurements and server properties.



Select the required measurements, as described in "Understanding the Apache Dialog Box" on page 99.

For a description of the available measurements, see "Apache Performance Counters" on page 100.

2 In the Server Properties section, enter the Port number and URL (without the server name), and click **OK**. The default URL is /server-status?auto.

3 In the Controller, click **OK** in the Apache dialog box to activate the monitor.

Note: The default port number and URL can vary from one server to another. Please consult your Web server administrator.

---

4 In the Tuning Console, click **OK** in the Select Measurements to Monitor dialog box, and in the Monitors Configuration dialog box, to activate the monitor.

Understanding the Apache Dialog Box

The Apache dialog box allows you select the items to monitor on the Apache server.

Available Measurements: Select the required measurement(s). Select multiple measurements using the CTRL key. A description of the selected measurement appears in the Description box.

Server Properties

- **Port:** Type the port number of the server.
- **URL:** To monitor an Apache server you need to know the server statistics information URL. A simple way to verify the statistics information URL is to try to view it through the browser.

  The URL should be in the following format:

  http://<server_name/IP address>:<port_number>/server-status?auto

  For example:

  http://stimpy:80/server-status?auto

  Enter the server statistics information URL, without the server name. The default URL is /server-status?auto.

Note: The default port number and URL can vary from one server to another. Please consult your Web server administrator.

Description: Displays a description of the selected measurement.

Apache Performance Counters

The following table describes the measurements and server properties that can be monitored:

Measurement	Description
# Busy Servers	The number of servers in the Busy state
# Idle Servers	The number of servers in the Idle state
Apache CPU Usage	The percentage of time the CPU is utilized by the Apache server
Hits/sec	The HTTP request rate
KBytes Sent/sec	The rate at which data bytes are sent from the Web server

To change the default server properties:

1 Open the **apache.cfg** file in the **<LoadRunner root folder>\dat\monitors** directory.

2 Edit the following parameters after the Delimiter=: statement:

InfoURL server statistics information URL

ServerPort server port number

SamplingRate rate (milliseconds) at which the LoadRunner monitor will poll the server for the statistics information. If this value is greater than 1000, LoadRunner will use it as its sampling rate. Otherwise, it will use the sampling rate defined in the Monitors tab of the Options dialog box.

Note: To monitor an Apache server through a firewall, use the Web server port (by default, port 80).

Microsoft IIS Monitoring

This Web Server Resource monitor displays statistics about the resource usage on the Microsoft Internet Information Server (IIS) during the scenario or session step run.

To obtain data for this graph, you need to enable the MS IIS online monitor (from the Controller or Tuning Console) and select the default measurements you want to display, before running the scenario or session step. You select measurements for the Microsoft IIS Server monitor using the MS IIS dialog box.

Note: To monitor an IIS server through a firewall, use TCP, port 139.

Adding a Monitored Machine to the Controller

1. Click the MS IIS graph in the graph tree, and drag it into the right pane of the Run view.
2. Right-click the graph and select **Add Measurements**, or choose **Monitors** > **Add Measurements**. The MS IIS dialog box opens.
3. In the **Monitored Server Machines** section, click **Add.** The Add Machine dialog box opens.
4. Enter the server name or IP address of the machine you want to monitor. Select the platform on which the machine runs, and click **OK**.
5. In the **Resource Measurements** section of the MS IIS dialog box, select the measurements you want to monitor.

   For a description of the available measurements, see "Microsoft IIS Performance Counters" on page 103.

Note: To change the default counters for the Microsoft IIS Server monitor, see "Changing a Monitor's Default Counters" on page 443.

---

6 To select additional measurements, click **Add**.

Continue with Configuring the Microsoft IIS Monitor below.

Adding a Monitored Machine to the Tuning Console

1 Click **Monitors** to open the Monitors Configuration dialog box.

2 Select the server whose monitors you want to configure from the Server list box.

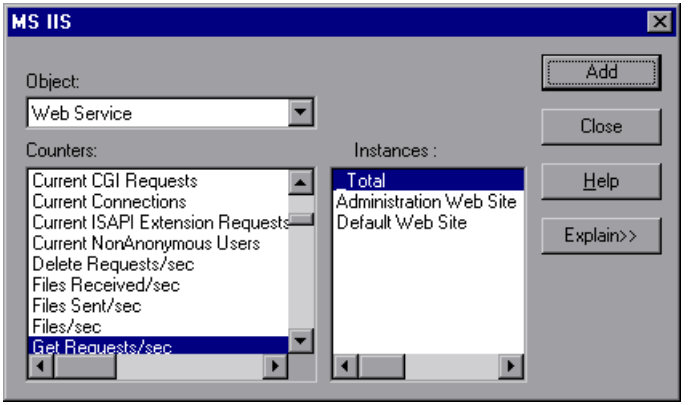
3 Click **Add Monitor** and select **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the Web Server category, select **MS IIS**, and then click **Add**.

Continue with Configuring the Microsoft IIS Monitor below.

Configuring the Microsoft IIS Monitor

1 The MS IIS dialog box opens displaying the Web Service object, its counters, and instances opens.



For each measurement, select an object, counter, and instance, and then click **Add**, as described in “Understanding the Microsoft IIS Dialog Box” on page 103.

For a description of the available measurements, see "Microsoft IIS Performance Counters" on page 103.

Note: To change the default counters for the Microsoft IIS Server monitor, see "Changing a Monitor's Default Counters" on page 443.

---

2 Add all the desired resources to the list, and click **Close**.

3 In the Controller, click **OK** in the MS IIS dialog box to activate the monitor.

4 In the Tuning Console. click **OK** in the MS IIS dialog box and in the Select Measurements to Monitor dialog box to activate the monitor.

Understanding the Microsoft IIS Dialog Box

The Microsoft IIS dialog box lets you select the items to monitor on the MS IIS server.

Object: Select the object being monitored on the specified machine.

Counters: Select a resource counter to monitor. Select multiple counters using the CTRL key. For an explanation of each counter, click **Explain**.

Instances: If multiple instances of the selected counter are running, select one or more instances to monitor for the selected counter.

Explain: Displays a description of the selected counter.

Microsoft IIS Performance Counters

The following table describes the default measurements that can be monitored:

Object	Measurement	Description
Web Service	**Bytes Sent/sec**	The rate at which the data bytes are sent by the Web service
Web Service	**Bytes Received/sec**	The rate at which the data bytes are received by the Web service

Object	Measurement	Description
Web Service	**Get Requests/sec**	The rate at which HTTP requests using the GET method are made. Get requests are generally used for basic file retrievals or image maps, though they can be used with forms.
Web Service	**Post Requests/sec**	The rate at which HTTP requests using the POST method are made. Post requests are generally used for forms or gateway requests.
Web Service	**Maximum Connections**	The maximum number of simultaneous connections established with the Web service
Web Service	**Current Connections**	The current number of connections established with the Web service
Web Service	**Current NonAnonymous Users**	The number of users that currently have a non-anonymous connection using the Web service
Web Service	**Not Found Errors/sec**	The rate of errors due to requests that could not be satisfied by the server because the requested document could not be found. These are generally reported to the client as an HTTP 404 error code.
Process	**Private Bytes**	The current number of bytes that the process has allocated that cannot be shared with other processes.

iPlanet/Netscape Monitoring

This Web Server Resource monitor displays statistics about the resource usage on the iPlanet/Netscape Web server during the scenario or session step run.

To obtain data for this graph, you need to enable the iPlanet/Netscape online monitor (from the Controller or Tuning Console) and select the default measurements you want to display, before running the scenario or session step.

Setting up the Monitoring Environment

To monitor an iPlanet/Netscape server, you need to know the administration server URL. A simple way to verify the administration server URL, is to try to view it through the browser.

The URL should be in the following format:

http://<admin_srv_name/IP address>:<port number>/https-<admin_srv_name/IP address>/bin/sitemon?doit

For example:

http://lazarus:12000/https-lazarus.mercury.co.il/bin/sitemon?doit

---

Note: In some server configurations, the URL must contain the administration server name and not the IP address.

In addition, the administration server name may differ from the iPlanet/Netscape server name.

Adding a Monitored Machine to the Controller

1. Click the iPlanet/Netscape graph in the graph tree, and drag it into the right pane of the Run view.
2. Right-click the graph and select **Add Measurements**, or choose **Monitors** > **Add Measurements**. The iPlanet/Netscape dialog box opens.
3. In the **Monitored Server Machines** section, click **Add.** The Add Machine dialog box opens.
4. Enter the server name or IP address of the machine you want to monitor. Select the platform on which the machine runs, and click **OK**.
5. In the **Resource Measurements** section of the iPlanet/Netscape dialog box, click **Add**.

Continue with Configuring the iPlanet/Netscape Monitor below.

Adding a Monitored Machine to the Tuning Console

1 Click **Monitors** to open the Monitors Configuration dialog box.

2 Select the server whose monitors you want to configure from the Server list box.

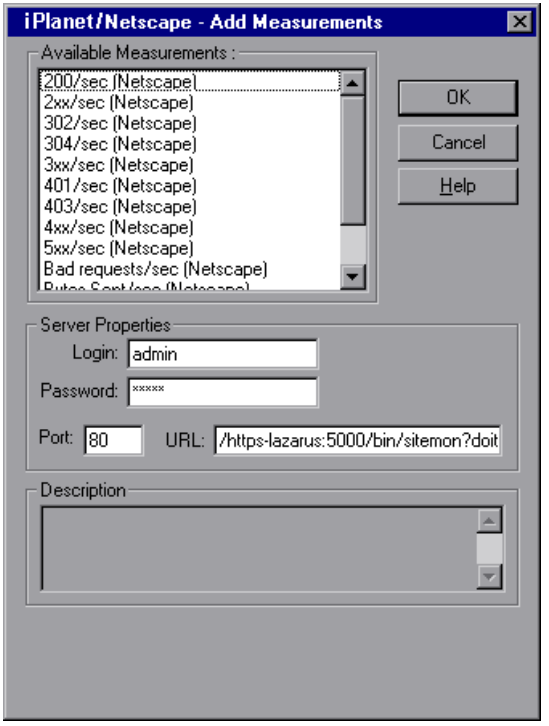
3 Click **Add Monitor** and select **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the Web Server category, select **iPlanet/Netscape**, and then click **Add**.

Continue with Configuring the iPlanet/Netscape Monitor below.

Configuring the iPlanet/Netscape Monitor

1 The iPlanet/Netscape - Add Measurements dialog box opens, displaying the available measurements and server properties.



Select the required measurements.

For a description of the available measurements, see "iPlanet/Netscape Performance Counters" on page 109.

2 Fill in the Server Properties:

➤ Enter the user login name and password. The user must have administrator permissions on the server.

➤ Enter the port number and URL (without the server name), and click **OK**. The default URL is /https-<admin_server>/bin/sitemon?doit.

3 In the Controller, click **OK** in the iPlanet/Netscape - Add Measurements dialog box and in the iPlanet/Netscape dialog box to activate the monitor.

4 In the Tuning Console, click **OK** in the iPlanet/Netscape - Add Measurements dialog box and in the Select Measurements to Monitor dialog box to activate the monitor.

Understanding the iPlanet/Netscape Dialog Box

The iPlanet/Netscape dialog box lets you select the items to monitor on the iPlanet/Netscape server.

Available Measurements: Select the required measurements. Select multiple measurements using the CTRL key. A description of the selected measurement appears in the Description box.

Server Properties

➤ **Login:** Enter the user login name. The user must have administrator permissions on the server.

➤ **Password:** Enter the user password.

➤ **Port:** Enter the server's port number.

➤ **URL:** To monitor an iPlanet/Netscape server, you need to know the server statistics information URL. A simple way to verify the statistics information URL is to try to view it through the browser.

The URL should be in the following format:

http://<admin_srv_name/IP address>:<port number>/https-<admin_srv_name/IP address>/bin/sitemon?doit

For example:

http://lazarus:80/https-lazarus/bin/sitemon?doit

Enter the server's URL, without the server name. The default URL is /https-<server>/bin/sitemon?doit.

Note: The default port number and URL can vary from one server to another. Please consult the Web server administrator. In some server configurations, the URL must contain the server name and not the IP address.

Description: Displays a description of the selected measurement.

iPlanet/Netscape Performance Counters

The following table describes the measurements and server properties that can be monitored:

Measurement	Description
200/sec	The rate of successful transactions being processed by the server
2xx/sec	The rate at which the server handles status codes in the 200 to 299 range
302/sec	The rate of relocated URLs being processed by the server
304/sec	The rate of requests for which the server tells the user to use a local copy of a URL instead of retrieving a newer version from the server
3xx/sec	The rate at which the server handles status codes in the 300 to 399 range
401/sec	The rate of unauthorized requests handled by the server
403/sec	The rate of forbidden URL status codes handled by the server
4xx/sec	The rate at which the server handles status codes in the 400 to 499 range
5xx/sec	The rate at which the server handles status codes 500 and higher
Bad requests/sec	The rate at which the server handles bad requests
Bytes sent/sec	The rate at which bytes of data are sent from the Web server
Hits/sec	The HTTP request rate
xxx/sec	The rate of all status codes (2xx-5xx) handled by the server, excluding timeouts and other errors that did return an HTTP status code

To change the default server properties:

1. Open the **Netscape.cfg** file in the **<LoadRunner root folder>\dat\monitors** directory.
2. Edit the following parameters in the [Netscape] section:

Counters	number of counters that the LoadRunner iPlanet/Netscape monitor will show you. This value should match the number of counters defined in the file.
InfoURL	server statistics information URL
ServerPort	server port number
ServerLogin	login name to the server
ServerPassword	login password for the login name
SamplingRate	rate (milliseconds) at which the LoadRunner monitor will poll the server for the statistics information. If this value is greater than 1000, LoadRunner will use it as its sampling rate. Otherwise, it will use the sampling rate defined in the Monitors tab of the Options dialog box.

Note: To monitor an iPlanet/Netscape server through a firewall, use the iPlanet/Netscape Administration server port. Configure this port during the server installation process.

---

iPlanet (SNMP) Monitoring

This Web Server Resource monitor displays statistics about the resource usage on the iPlanet (SNMP) Web server during the scenario or session step run. The iPlanet (SNMP) monitor uses the Simple Network Management Protocol (SNMP) to retrieve iPlanet (SNMP) server statistics.

To obtain data for this graph, you need to enable the iPlanet (SNMP) online monitor (from the Controller or Tuning Console) and select the default measurements you want to display, before running the scenario or session step. You define the measurements for the iPlanet (SNMP) monitor using the iPlanet (SNMP) dialog box.

Note: To monitor a iPlanet (SNMP) server, use port 161 or 162, depending on the configuration of the agent.

Adding a Monitored Machine to the Controller

1. Click the iPlanet (SNMP) graph in the graph tree, and drag it into the right pane of the Run view.
2. Right-click the graph and select **Add Measurements**, or choose **Monitors** > **Add Measurements**. The iPlanet (SNMP) dialog box opens.
3. In the **Monitored Server Machines** section, click **Add.** The Add Machine dialog box opens.

4 Enter the server name or IP address of the machine you want to monitor. Select the platform on which the machine runs, and click **OK**.

Note: If the iPlanet SNMP agent is running on a different port than the default SNMP port, you need to define the port number. Enter the following information in the Add Machine dialog box:
<server name:port number>
For example: digi:8888

In addition, you can define the default port for your iPlanet server in the configuration file, **snmp.cfg**, located in **<LoadRunner root folder>\dat\monitors**. For example, if the port used by the SNMP agent on your iPlanet server is 8888, you should edit the **snmp.cfg** file as follows:
; iPlanet (WebServer)
[cm_snmp_mon_iws60]
port=8888

---

5 In the **Resource Measurements** section of the iPlanet (SNMP) dialog box, click **Add**.

Continue with Configuring the iPlanet (SNMP) Monitor below.

Adding a Monitored Machine to the Tuning Console

1 Click **Monitors** to open the Monitors Configuration dialog box.

2 Select the server whose monitors you want to configure from the Server list box.

3 Click **Add Monitor** and select **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the Web Server category, select **iPlanet (SNMP)**, and then click **Add**.

Note: If the iPlanet SNMP agent is running on a different port than the default SNMP port, you need to define the port number.

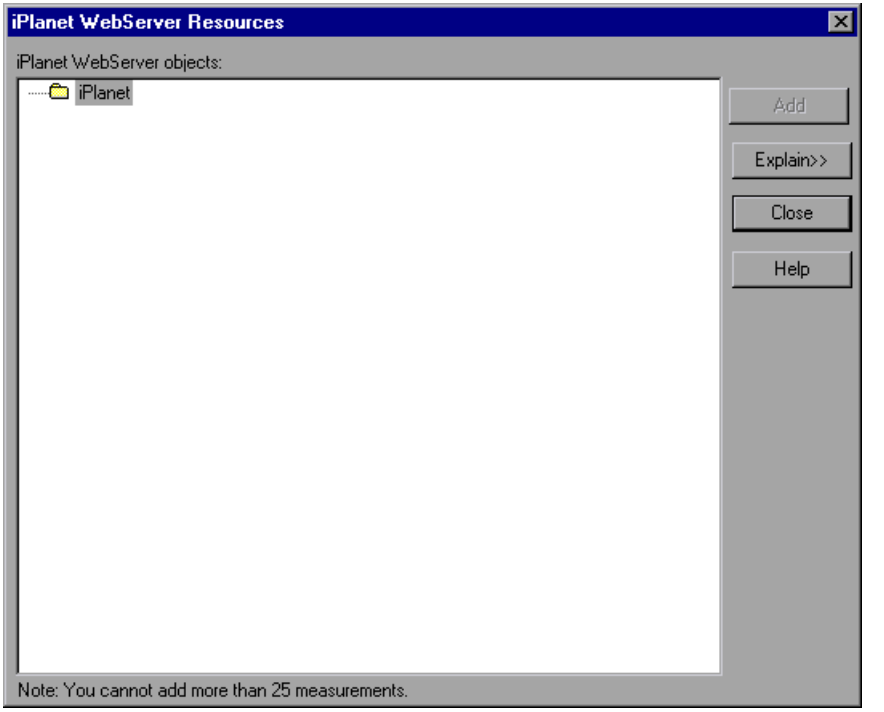
You can define the default port for your iPlanet server in the configuration file, **snmp.cfg**, located in **<LoadRunner root folder>\dat\monitors**. For example, if the port used by the SNMP agent on your iPlanet server is 8888, you should edit the **snmp.cfg** file as follows:

```
; iPlanet (WebServer)
[cm_snmp_mon_iws60]
port=8888
```

Continue with Configuring the iPlanet (SNMP) Monitor below.

Configuring the iPlanet (SNMP) Monitor

1 The iPlanet WebServer Resources dialog box opens.



Browse the iPlanet WebServer Resources Object tree, and select performance counters.

For a description of the available measurements, see "iPlanet (SNMP) Performance Counters" on page 116.

Note: The iPlanet (SNMP) monitor can only monitor up to 25 measurements.

---

2 Add all the desired resources to the list, and click **Close**.

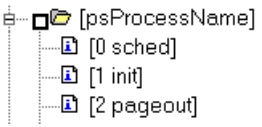
3 In the Controller, click **OK** in the iPlanet (SNMP) dialog box to activate the monitor.

4 In the Tuning Console, click **OK** in the Select Measurements to Monitor dialog box to activate the monitor.

Improving the Level of Measurement Information

You can improve the level of measurement information for the iPlanet (SNMP) monitor by enabling measurements with string values to be listed (in addition to measurements with numeric values), and by enabling the name modifier (which displays the string value as an identifying part of the measurement name).

In the following example of a measurement using the name modifier, the string value of ProcessName (sched) is displayed in addition to its instance ID (0):



To enable this feature, add the following line to the **<LoadRunner root folder>\dat\monitors\snmp.cfg** file:

```
SNMP_show_string_nodes=1
```

Usage Notes: You can select more than one name modifier, but the first in the hierarchy will be used. Each time the iPlanet SNMP Add Measurements dialog box opens, the information is reread from the **snmp.cfg** file. You cannot add the same measurement twice (once with a name modifier and once without it). If you do so, an error message is issued.

Understanding the iPlanet (SNMP) Dialog Box

The iPlanet (SNMP) dialog box lets you select the iPlanet (SNMP) server resources to monitor using the Simple Network Management Protocol (SNMP).

iPlanet (SNMP) Objects: Select each required object and click **Add**. Click **Explain** for the ID number and a description of the selected object.

Explain: Displays a description of the selected object.

Note: The iPlanet (SNMP) monitor can only monitor up to 25 measurements.

---

iPlanet (SNMP) Performance Counters

The following table describes the measurements and server properties that can be monitored:

Measurement	Description
iwsInstanceTable	iPlanet Web Server instances
iwsInstanceEntry	iPlanet Web Server instances
iwsInstanceIndex	Server instance index
iwsInstanceId	Server instance identifier
iwsInstanceVersion	Server instance software version
iwsInstanceDescription	Description of server instance
iwsInstanceOrganization	Organization responsible for server instance
iwsInstanceContact	Contact information for person(s) responsible for server instance
iwsInstanceLocation	Location of server instance
iwsInstanceStatus	Server instance status
iwsInstanceUptime	Server instance uptime

Measurement	Description
iwsInstanceDeathCount	Number of times server instance processes have died
iwsInstanceRequests	Number of requests processed
iwsInstanceInOctets	Number of octets received
iwsInstanceOutOctets	Number of octets transmitted
iwsInstanceCount2xx	Number of 200-level (Successful) responses issued
iwsInstanceCount3xx	Number of 300-level (Redirection) responses issued
iwsInstanceCount4xx	Number of 400-level (Client Error) responses issued
iwsInstanceCount5xx	Number of 500-level (Server Error) responses issued
iwsInstanceCountOther	Number of other (neither 2xx, 3xx, 4xx, nor 5xx) responses issued
iwsInstanceCount200	Number of 200 (OK) responses issued
iwsInstanceCount302	Number of 302 (Moved Temporarily) responses issued
iwsInstanceCount304	Number of 304 (Not Modified) responses issued
iwsInstanceCount400	Number of 400 (Bad Request) responses issued
iwsInstanceCount401	Number of 401 (Unauthorized) responses issued
iwsInstanceCount403	Number of 403 (Forbidden) responses issued
iwsInstanceCount404	Number of 404 (Not Found) responses issued
iwsInstanceCount503	Number of 503 (Unavailable) responses issued
iwsInstanceLoad 1MinuteAverage	System load average for 1 minute
iwsInstanceLoad 5MinuteAverage	System load average for 5 minutes

Measurement	Description
iwsInstanceLoad 15MinuteAverage	System load average for 15 minutes
iwsInstanceNetwork InOctets	Number of octets transmitted on the network per second
iwsInstanceNetwork OutOctets	Number of octets received on the network per second
iwsVsTable	iPlanet Web Server virtual servers
iwsVsEntry	iPlanet Web Server virtual server
iwsVsIndex	Virtual server index
iwsVsId	Virtual server identifier
iwsVsRequests	Number of requests processed
iwsVsInOctets	Number of octets received
iwsVsOutOctets	Number of octets transmitted
iwsVsCount2xx	Number of 200-level (Successful) responses issued
iwsVsCount3xx	Number of 300-level (Redirection) responses issued
iwsVsCount4xx	Number of 400-level (Client Error) responses issued
iwsVsCount5xx	Number of 500-level (Server Error) responses issued
iwsVsCountOther	Number of other (neither 2xx, 3xx, 4xx, nor 5xx) responses issued
iwsVsCount200	Number of 200 (OK) responses issued
iwsVsCount302	Number of 302 (Moved Temporarily) responses issued
iwsVsCount304	Number of 304 (Not Modified) responses issued
iwsVsCount400	Number of 400 (Bad Request) responses issued

Measurement	Description
iwsVsCount401	Number of 401 (Unauthorized) responses issued
iwsVsCount403	Number of 403 (Forbidden) responses issued
iwsVsCount404	Number of 404 (Not Found) responses issued
iwsVsCount503	Number of 503 (Unavailable) responses issued
iwsProcessTable	iPlanet Web Server processes
iwsProcessEntry	iPlanet Web Server process
iwsProcessIndex	Process index
iwsProcessId	Operating system process identifier
iwsProcessThreadCount	Number of request processing threads
iwsProcessThreadIdle	Number of request processing threads currently idle
iwsProcessConnection QueueCount	Number of connections currently in connection queue
iwsProcessConnection QueuePeak	Largest number of connections that have been queued simultaneously
iwsProcessConnection QueueMax	Maximum number of connections allowed in connection queue
iwsProcessConnection QueueTotal	Number of connections that have been accepted
iwsProcessConnection QueueOverflows	Number of connections rejected due to connection queue overflow
iwsProcessKeepalive Count	Number of connections currently in keepalive queue
iwsProcessKeepaliveMax	Maximum number of connections allowed in keepalive queue
iwsProcessSizeVirtual	Process size in kbytes
iwsProcessSizeResident	Process resident size in kbytes

Measurement	Description
iwsProcessFraction SystemMemoryUsage	Fraction of process memory in system memory
iwsListenTable	iPlanet Web Server listen sockets
iwsListenEntry	iPlanet Web Server listen socket
iwsListenIndex	Listen socket index
iwsListenId	Listen socket identifier
iwsListenAddress	Address socket is listening on
iwsListenPort	Port socket is listening on
iwsListenSecurity	Encryption support
iwsThreadPoolTable	iPlanet Web Server thread pools
iwsThreadPoolEntry	iPlanet Web Server thread pool
iwsThreadPoolIndex	Thread pool index
iwsThreadPoolId	Thread pool identifier
iwsThreadPoolCount	Number of requests queued
iwsThreadPoolPeak	Largest number of requests that have been queued simultaneously
iwsThreadPoolMax	Maximum number of requests allowed in queue
iwsCpuTable	iPlanet Web Server CPUs
iwsCpuEntry	iPlanet Web Server CPU
iwsCpuIndex	CPU index
iwsCpuId	CPU identifier
iwsCpuIdleTime	CPU Idle Time
iwsCpuUserTime	CPU User Time
iwsCpuKernelTime	CPU Kernel Time

Monitoring Using a Proxy Server

LoadRunner allows you to monitor using the Apache and Netscape monitors when there is a proxy server between the Controller or Tuning Console and the monitored server. To enable this, you must define settings in your configuration file: in **<LoadRunner root folder>\dat\monitors\apache.cfg** for the Apache monitor, or in **<LoadRunner root folder>\dat\monitors\Netscape.cfg** for the Netscape monitor.

Before defining settings, you need to determine whether you want LoadRunner to obtain proxy settings from your Internet Explorer connection configuration, or from the proxy settings in the configuration file.

To have LoadRunner read proxy settings from your Internet Explorer connection:

1. In the Proxy Settings section of the configuration file, assign **useProxy** a value of 1.
2. If the proxy requires a username, password, or domain, enter these parameters on the lines **proxyUsername**, **proxyPassword**, and **proxyDomain**.

To have LoadRunner read proxy settings from the configuration file:

1. In the Proxy Settings section of the configuration file, enter the proxy information on the **httpProxy** line. Use the format:
   [<protocol>=][<scheme>://]<proxy>[:<port>][[<protocol>=][<scheme>://]<proxy>[:<port>]]

   For example:
   httpProxy=http=http://my_http_proxy:8080 https=https://my_https_proxy:9000
2. If the proxy requires a username, password, or domain, enter these parameters on the lines **proxyUsername**, **proxyPassword**, and **proxyDomain**.

To have LoadRunner connect directly to the server (any proxy settings are ignored):

In the Proxy Settings section of the configuration file, assign **useProxy** a value of 0.

8

Web Application Server Resource Monitoring

You use LoadRunner's Web Application Server Resource monitors to monitor Web application servers during a scenario or session step run and isolate application server performance bottlenecks.

This chapter describes:

- About Web Application Server Resource Monitors
- Ariba Monitoring
- ATG Dynamo Monitoring
- BroadVision Monitoring
- ColdFusion Monitoring
- Fujitsu INTERSTAGE Monitoring
- iPlanet (NAS) Monitoring
- Microsoft Active Server pages Monitoring
- Oracle9iAS HTTP Monitoring
- SilverStream Monitoring
- WebLogic (SNMP) Monitoring
- WebLogic (JMX) Monitoring
- WebSphere Monitoring
- WebSphere (EPM) Monitoring

About Web Application Server Resource Monitors

Web Application Server Resource monitors provide you with information about the resource usage of the Ariba, ATG Dynamo, BroadVision, ColdFusion, Fujitsu INTERSTAGE, iPlanet (NAS), Microsoft ASP, Oracle9iAS HTTP, SilverStream, WebLogic (SNMP), WebLogic (JMX), and WebSphere application servers during scenario or session step execution. To obtain performance data, you need to activate the online monitor for the server and specify which resources you want to measure before executing the scenario or session step.

The procedures for selecting monitor measurements and configuring the monitors vary according to server type. The following sections contain specific configuration instructions for each server type.

Ariba Monitoring

The Ariba monitor displays statistics about the resource usage on the Ariba server during the scenario or session step run.

To obtain data for this graph, you need to enable the Ariba online monitor (from the Controller or Tuning Console) and select the default measurements you want to display, before running the scenario or session step. You select measurements to monitor the Ariba server using the Ariba Monitor Configuration dialog box.

Note: The port you use to monitor an Ariba server through a firewall depends on the configuration of your server.

Adding a Monitored Machine to the Controller

1. Click the Ariba graph in the graph tree, and drag it into the right pane of the Run view.
2. Right-click the graph and select **Add Measurements**, or choose **Monitors > Add Measurements**. The Ariba dialog box opens.

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

4 Enter the server name or IP address of the machine you want to monitor according to the following format: <server name>:<port number>.

For example: merc1:12130

Select the platform on which the machine runs, and click **OK**.

5 In the **Resource Measurements** section of the Ariba dialog box, click **Add**.

Continue with Configuring the Ariba Monitor below.

Adding a Monitored Machine to the Tuning Console

1 Click **Monitors** to open the Monitors Configuration dialog box.

2 Select the server whose monitors you want to configure from the Server list box.

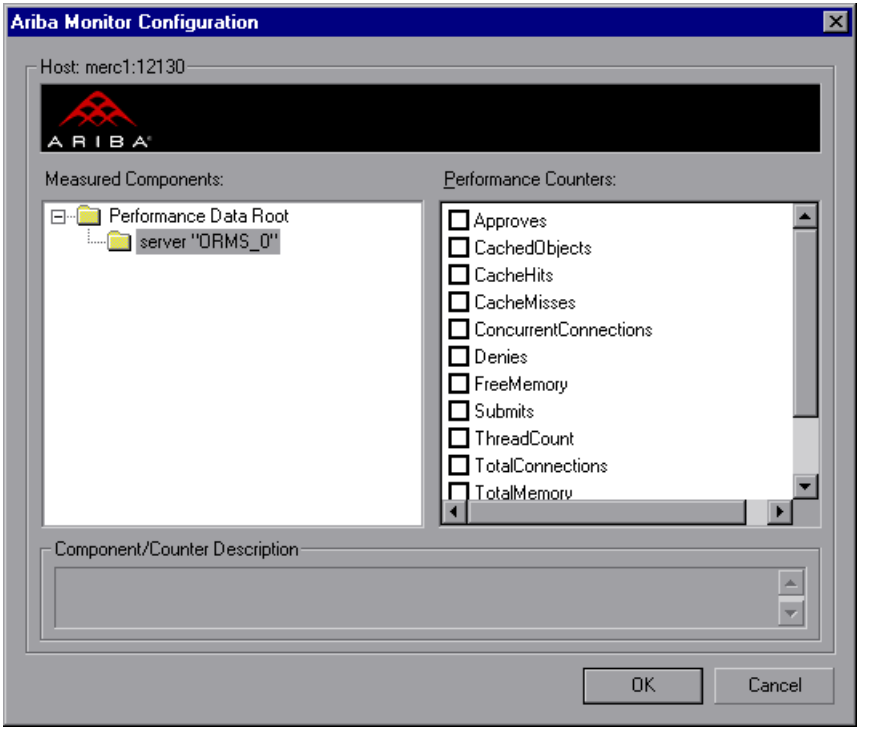
3 Click **Add Monitor** and select **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the Application Server category, select **Ariba**, and then click **Add**.

Continue with Configuring the Ariba Monitor below.

Configuring the Ariba Monitor

1 The Ariba Monitor Configuration dialog box opens.



Browse the Measured Components tree, and select performance counters as described in "Understanding the Ariba Dialog Box" on page 127.

For a description of the available measurements, see "Ariba Performance Counters" on page 128.

2 In the Controller, click **OK** in the Ariba Monitor Configuration dialog box, and in the Ariba dialog box, to activate the Ariba monitor.

3 In the Tuning Console, click **OK** in the Ariba Monitor Configuration dialog box, and in the Select Measurements to Monitor dialog box, to activate the Ariba monitor.

XML Accessibility Verification

Only browsers that are XML-compatible will allow you to view the performance XML file.

To verify whether the XML file is accessible:

Display the XML file through the browser. The URL should be in the following format: http://<server name:port number>/metrics?query=getStats

For example: http://merc1:12130/metrics?query=getStats

Note: In some cases, although the browser is XML-compatible, it may still return the error: The XML page cannot be displayed. In these cases, the XML file can be accessed by the Ariba performance monitor, although it cannot be viewed by the browser.

Understanding the Ariba Dialog Box

The Ariba dialog box lets you select the items to monitor on the Ariba application server.

Host: Displays the name of the monitored machine.

Measured Components: Displays the available components. Browse the tree and select the component you want to monitor. A description of the highlighted component appears in the Component/Counter Description box.

Performance Counters: Check the required performance counters. A description of the selected counter appears in the Component/Counter Description box.

Component/Counter Description: Displays a description of the selected component or counter.

Note: Only browsers that are XML-compatible will allow you to view the performance XML file.

To verify whether the XML file is accessible, display the XML file through the browser. The URL should be in the following format: http://<server name:server port>/metrics?query=getStats

For example: http://merc1:12130/metrics?query=getStats

Ariba Performance Counters

The following tables describe the counters that can be monitored:

Core Server Performance Counters

Measurement	Description
Requisitions Finished	The instantaneous reading of the length of the worker queue at the moment this metric is obtained. The longer the worker queue, the more user requests are delayed for processing.
Worker Queue Length	The instantaneous reading of the length of the worker queue at the moment this metric is obtained. The longer the worker queue, the more user requests are delayed for processing.
Concurrent Connections	The instantaneous reading of the number of concurrent user connections at the moment this metric is obtained
Total Connections	The cumulative number of concurrent user connections since Ariba Buyer was started.
Total Memory	The instantaneous reading of the memory (in KB) being used by Ariba Buyer at the moment this metric is obtained
Free Memory	The instantaneous reading of the reserved memory (in bytes) that is not currently in use at the moment this metric is obtained

Measurement	Description
Up Time	The amount of time (in hours and minutes) that Ariba Buyer has been running since the previous time it was started
Number of Threads	The instantaneous reading of the number of server threads in existence at the moment this metric is obtained
Number of Cached Objects	The instantaneous reading of the number of Ariba Buyer objects being held in memory at the moment this metric is obtained
Average Session Length	The average length of the user sessions (in seconds) of all users who logged out since previous sampling time. This value indicates on average how long a user stays connected to server.
Average Idle Time	The average idle time (in seconds) for all the users who are active since previous sampling time. The idle time is the period of time between two consecutive user requests from the same user.
Approves	The cumulative count of the number of approves that happened during the sampling period. An Approve consists of a user approving one Approvable.
Submits	The cumulative count of the number of Approvables submitted since previous sampling time
Denies	The cumulative count of the number of submitted Approvables denied since previous sampling time
Object Cache Accesses	The cumulative count of accesses (both reads and writes) to the object cache since previous sampling time
Object Cache Hits	The cumulative count of accesses to the object cache that are successful (cache hits) since previous sampling time

System Related Performance Counters

Measurement	Description
Database Response Time	The average response time (in seconds) to the database requests since the previous sampling time
Buyer to DB server Traffic	The cumulative number of bytes that Ariba Buyer sent to DB server since the previous sampling time.
DB to Buyer server Traffic	The cumulative number of bytes that DB server sent to Ariba Buyer since the previous sampling time
Database Query Packets	The average number of packets that Ariba Buyer sent to DB server since the previous sampling time
Database Response Packets	The average number of packets that DB server sent to Ariba Buyer since the previous sampling time

ATG Dynamo Monitoring

The ATG Dynamo monitor displays statistics about the resource usage on the ATG Dynamo server during the scenario or session step run. The ATG Dynamo monitor uses SNMP to retrieve ATG Dynamo server statistics.

To obtain data for this graph, you need to enable the ATG Dynamo online monitor (from the Controller or Tuning Console) and select the default measurements you want to display, before running the scenario or session step. You define the measurements for the ATG Dynamo monitor using the ATG Dynamo Resources dialog box.

Adding a Monitored Machine to the Controller

1. Click the ATG Dynamo graph in the graph tree, and drag it into the right pane of the Run view.
2. Right-click the graph and select **Add Measurements**, or choose **Monitors** > **Add Measurements**. The ATG Dynamo dialog box opens.
3. In the **Monitored Server Machines** section, click **Add.** The Add Machine dialog box opens.
4. Enter the server name or IP address of the machine you want to monitor. Select the platform on which the machine runs, and click **OK**.

Note: If the ATG SNMP agent is running on a different port than the default ATG SNMP port 8870, you need to define the port number. Enter the following information in the Add Machine dialog box:
<server name:port number>
For example: digi:8888

In addition, you can define the default port for your ATG server in the configuration file, **snmp.cfg,** located in **<LoadRunner root folder>\dat\monitors**. For example, if the port used by the SNMP agent on your ATG system is 8888, you should edit the **snmp.cfg** file as follows:

```
; ATG Dynamo
[cm_snmp_mon_atg]
port=8888
```

---

5. In the **Resource Measurements** section of the ATG Dynamo dialog box, click **Add**.

   Continue with Configuring the ATG Dynamo Monitor below.

Adding a Monitored Machine to the Tuning Console

1. Click **Monitors** to open the Monitors Configuration dialog box.
2. Select the server whose monitors you want to configure from the Server list box.
3. Click **Add Monitor** and select **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.
4. In the left section of the dialog box, expand the Application Server category, select **ATG Dynamo**, and then click **Add**.

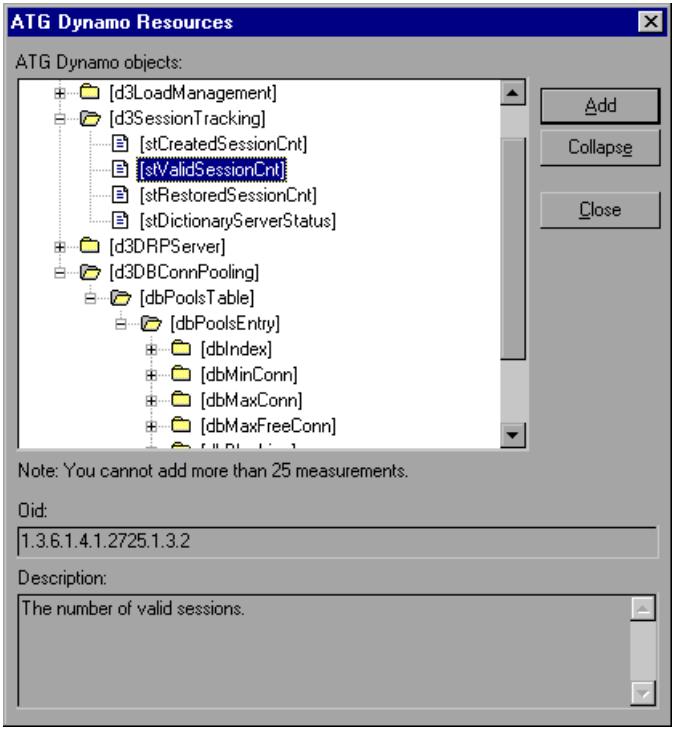
Note: If the ATG SNMP agent is running on a different port than the default ATG SNMP port 8870, you need to define the port number. You can define the default port for your ATG server in the configuration file, **snmp.cfg,** located in **<LoadRunner root folder>\dat\monitors**. For example, if the port used by the SNMP agent on your ATG system is 8888, you should edit the **snmp.cfg** file as follows:

```
; ATG Dynamo
[cm_snmp_mon_atg]
port=8888
```

Continue with Configuring the ATG Dynamo Monitor below.

Configuring the ATG Dynamo Monitor

1 The ATG Dynamo Resources dialog box opens.



Browse the ATG Dynamo Object tree, and select the measurements you want to monitor as described in “Understanding the ATG Dynamo Dialog Box” on page 135.

For a description of the available measurements, see “ATG Dynamo Performance Counters” on page 135.

2 Add all the desired resources to the list, and click **Close**.

Note: The ATG Dynamo monitor can only monitor up to 25 measurements.

---

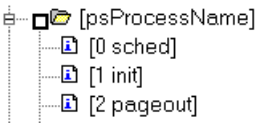
3 In the Controller, click **OK** in the ATG Dynamo dialog box to activate the monitor.

4 In the Tuning Console, click **OK** in the Select Measurements to Monitor dialog box to activate the monitor.

Improving the Level of Measurement Information

You can improve the level of measurement information for the ATG Dynamo monitor by enabling measurements with string values to be listed (in addition to measurements with numeric values), and by enabling the name modifier (which displays the string value as an identifying part of the measurement name).

In the following example of a measurement using the name modifier, the string value of ProcessName (sched) is displayed in addition to its instance ID (0):



To enable this feature, add the following line to the **<LoadRunner root folder>\dat\monitors\snmp.cfg** file:

```
SNMP_show_string_nodes=1
```

Usage Notes: You can select more than one name modifier, but the first in the hierarchy will be used. Each time the ATG Dynamo Add Measurements dialog box opens, the information is reread from the **snmp.cfg** file. You cannot add the same measurement twice (once with a name modifier and once without it). If you do so, an error message is issued.

Understanding the ATG Dynamo Dialog Box

The ATG Dynamo dialog box lets you select the ATG Dynamo server resources to monitor using the Simple Network Management Protocol (SNMP).

ATG Dynamo Objects: Select each required object and click **Add**. Click **Explain** for the ID number and a description of the selected object.

Explain: Displays a description of the selected object.

Note: The ATG Dynamo monitor can only monitor up to 25 measurements.

ATG Dynamo Performance Counters

The following tables describe the measurements that can be monitored:

d3System

Measurement	Description
sysTotalMem	The total amount of memory currently available for allocating objects, measured in bytes
sysFreeMem	An approximation of the total amount of memory currently available for future allocated objects, measured in bytes
sysNumInfoMsgs	The number of system global info messages written
sysNumWarningMsgs	The number of system global warning messages written
sysNumErrorMsgs	The number of system global error messages written

d3LoadManagement

Measurement	Description
lmIsManager	True if the Dynamo is running a load manager
lmManagerIndex	Returns the Dynamo's offset into the list of load managing entities
lmIsPrimaryManager	True if the load manager is an acting primary manager
lmServicingCMs	True if the load manager has serviced any connection module requests in the amount of time set as the connection module polling interval
lmCMLDRPPort	The port of the connection module agent
lmIndex	A unique value for each managed entity
lmSNMPPort	The port for the entry's SNMP agent
lmProbability	The probability that the entry will be given a new session
lmNewSessions	Indicates whether or not the entry is accepting new sessions, or if the load manager is allowing new sessions to be sent to the entry. This value is inclusive of any override indicated by lmNewSessionOverride.
lmNewSessionOverride	The override set for whether or not a server is accepting new sessions

d3SessionTracking

Measurement	Description
stCreatedSessionCnt	The number of created sessions
stValidSessionCnt	The number of valid sessions
stRestoredSessionCnt	The number of sessions migrated to the server
StDictionaryServerStatus	d3Session Tracking

d3DRPServer

Measurement	Description
drpPort	The port of the DRP server
drpTotalReqsServed	Total number of DRP requests serviced
drpTotalReqTime	Total service time in msecs for all DRP requests
drpAvgReqTime	Average service time in msecs for each DRP request
drpNewessions	True if the Dynamo is accepting new sessions

d3DBConnPooling

Measurement	Description
dbPoolsEntry	A pooling service entry containing information about the pool configuration and current status
dbIndex	A unique value for each pooling service
dbPoolID	The name of the DB connection pool service
dbMinConn	The minimum number of connections pooled
dbMaxConn	The maximum number of connections pooled
dbMaxFreeConn	The maximum number of free pooled connections at a time
dbBlocking	Indicates whether or not the pool is to block out check outs
dbConnOut	Returns the number of connections checked out

Measurement	Description
dbFreeResources	Returns the number of free connections in the pool. This number refers to connections actually created that are not currently checked out. It does not include how many more connections are allowed to be created as set by the maximum number of connections allowed in the pool.
dbTotalResources	Returns the number of total connections in the pool. This number refers to connections actually created and is not an indication of how many more connections may be created and used in the pool.

BroadVision Monitoring

The BroadVision monitor displays performance statistics for all the servers/services available on the BroadVision application during the scenario or session step run.

To monitor a BroadVision server, you must grant the client permission to invoke or launch services on the server. You should also ensure that your Java environment is configured properly.

To obtain data for this graph, you need to enable the BroadVision online monitor (from the Controller or Tuning Console) and select the default measurements you want to display, before running the scenario or session step.

Note: The port you use to monitor a BroadVision server through a firewall depends on the configuration of your server.

Setting up the Monitoring Environment

Before monitoring a BroadVision server, you must set up the server monitor environment.

To grant permission for a BroadVision server:

➤ Use the Iona Technologies (Orbix) command for setting user and access permission on a load generator machine:

```
chmodit [-h <host>] [-v] { <server> | -a <dir> }

{i{+,-}{user,group} | l{+,-}{user,group} }
```

➤ If you experience problems connecting to the BroadVision monitor, you may need to redefine the permissions to "all."

To invoke permission for all, enter the following command at the BroadVision server command prompt:

```
# chmodit <server> i+all
```

To launch permission for all, enter the following command at the BroadVision server command prompt:

```
# chmodit <server> l+all
```

➤ Alternatively, set ORBIX_ACL. Setting ORBIX_ACL=i+all l+all in the BroadVision/Orbix configuration file gives permission to all.

In addition, to monitor a BroadVision server, you need to have JDK 1.2 or higher installed on the Controller or Tuning Console machine.

You can install JDK 1.2 by following the download and installation instructions at the following Web site: http://java.sun.com/products/jdk/1.2/

Before activating the monitor, make sure that your Java environment is configured properly.

To configure your Java environment:

1. Open the Windows Registry.
2. The registry should contain the correct path to the Java executable (java.exe) under the JDK 1.2 installation directory. Verify the following registry key:

   HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\App Paths\java.exe

3. The registry should contain the correct path to the Java run-time environment (JRE) under the JRE 1.2 installation directory. Verify the following registry key:

   HKEY_LOCAL_MACHINE\SOFTWARE\JavaSoft\Java Runtime Environment\1.2\JavaHome

Adding a Monitored Machine to the Controller

1. Click the BroadVision graph in the graph tree, and drag it into the right pane of the Run view.
2. Right-click the graph and choose **Add Measurements**, or choose **Monitors** > **Add Measurements**. The BroadVision dialog box opens.
3. In the **Monitored Server Machines** section, click **Add.** The Add Machine dialog box opens.
4. Enter the BroadVision server name or IP address with the port number according to the format: <server name>:<port number>. For example: dnsqa:1221.

   Select the machine platform, and click **OK**.
5. In the **Resource Measurements** section of the BroadVision dialog box, click **Add**.

   Continue with Configuring the BroadVision Monitor below.

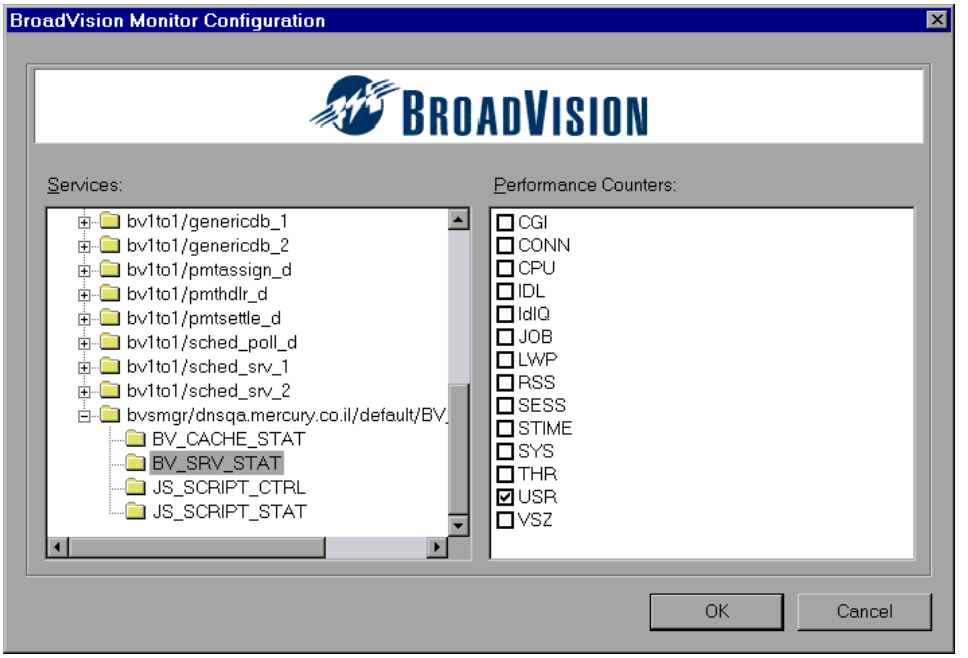
Adding a Monitored Machine to the Tuning Console

1. Click **Monitors** to open the Monitors Configuration dialog box.
2. Select the server whose monitors you want to configure from the Server list box.
3. Click **Add Monitor** and select **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.
4. In the left section of the dialog box, expand the Application Server category, select **BroadVision (5.5 - 6.x)** or **BroadVision (4.5 -5.4)**, and then click **Add**.

   Continue with Configuring the BroadVision Monitor below.

Configuring the BroadVision Monitor

1 The BroadVision Monitor Configuration dialog box opens, displaying the available measurements.



Browse the Services tree and select the required performance counters in the BroadVision Monitor Configuration window's right pane.

For a description of the available measurements, see "BroadVision Performance Counters" on page 143.

2 In the Controller, click **OK** in the BroadVision Monitor Configuration dialog box, and in the BroadVision dialog box, to activate the monitor.

3 In the Tuning Console, click **OK** in the BroadVision Monitor Configuration dialog box, and in the Select Measurements to Monitor dialog box, to activate the monitor.

Understanding the BroadVision Dialog Box

The BroadVision dialog box lets you select the items to monitor on the BroadVision application server.

Services: Displays the services available on the BroadVision application server. Browse the tree and select the service you want to monitor.

Performance Counters: Check the performance counters you want to monitor.

BroadVision Performance Counters

The following table describes the servers/services that can be monitored:

Server	Multiple Instances	Description
adm_srv	No	One-To-One user administration server. There must be one.
alert_srv	No	Alert server handles direct IDL function calls to the Alert system.
bvconf_srv	No	One-To-One configuration management server. There must be one.
cmsdb	Yes	Visitor management database server.
cntdb	Yes	Content database server.
deliv_smtp_d	Yes	Notification delivery server for e-mail type messages. Each instance of this server must have its own ID, numbered sequentially starting with "1".
deliv_comp_d	No	Notification delivery completion processor.
extdbacc	Yes	External database accessor. You need at least one for each external data source.
genericdb	No	Generic database accessor handles content query requests from applications, when specifically called from the application. This is also used by the One-To-One Command Center.

Server	Multiple Instances	Description
hostmgr	Yes	Defines a host manager process for each machine that participates in One-To-One, but doesn't run any One-To-One servers. For example, you need a hostmgr on a machine that runs only servers. You don't need a separate hostmgr on a machine that already has one of the servers in this list.
g1_ofbe_srv	No	Order fulfillment back-end server.
g1_ofdb	Yes	Order fulfillment database server.
g1_om_srv	No	Order management server.
pmtassign_d	No	The payment archiving daemon routes payment records to the archives by periodically checking the invoices table, looking for records with completed payment transactions, and then moving those records into an archive table.
pmthdlr_d	Yes	For each payment processing method, you need one or more authorization daemons to periodically acquire the authorization when a request is made.
pmtsettle_d	Yes	Payment settlement daemon periodically checks the database for orders of the associated payment processing method that need to be settled, and then authorizes the transactions.
sched_poll_d	No	Notification schedule poller scans the database tables to determine when a notification must be run.
sched_srv	Yes	Notification schedule server runs the scripts that generate the visitor notification messages.

Performance Counters

Performance counters for each server/service are divided into logical groups according to the service type.

The following section describes all the available counters under each group. Note that the same group can have a different number of counters, depending on the service.

Counter groups:

BV_DB_STAT
BV_SRV_CTRL
BV_SRV_STAT
NS_STAT
BV_CACHE_STAT
JS_SCRIPT_CTRL
JS_SCRIPT_STAT

BV_DB_STAT

The database accessor processes have additional statistics available from the BV_DB_STAT memory block. These statistics provide information about database accesses, including the count of selects, updates, inserts, deletes, and stored procedure executions.

DELETE	Count of deletes executions
INSERT	Count of inserts executions
SELECT	Count of selects executions
SPROC	Count of stored procedure executions
UPDATE	Count of updates executions

BV_SRV_CTRL

SHUTDOWN

BV_SRV_STAT

The display for Interaction Manager processes includes information about the current count of sessions, connections, idle sessions, threads in use, and count of CGI requests processed.

HOST	Host machine running the process
ID	Instance of the process (of which multiple can be configured in the **bv1to1.conf** file), or engine ID of the Interaction Manager
CGI	Current count of CGI requests processed
CONN	Current count of connections
CPU	CPU percentage consumed by this process. If a process is using most of the CPU time, consider moving it to another host, or creating an additional process, possibly running on another machine. Both of these specifications are done in the **bv1to1.conf** file. The CPU % reported is against a single processor. If a server is taking up a whole CPU on a 4 processor machine, this statistic will report 100%, while the Windows Task Manager will report 25%. The value reported by this statistic is consistent with "% Processor Time" on the Windows Performance Monitor
GROUP	Process group (which is defined in the **bv1to1.conf** file), or Interaction Manager application name.
STIME	Start time of server. The start times should be relatively close. Later times might be an indication that a server crashed and was automatically restarted.
IDL	Total count of IDL requests received, not including those to the monitor.

IdlQ	
JOB	
LWP	Number of light-weight processes (threads).
RSS	Resident memory size of server process (in kilobytes).
STIME	System start time.
SESS	Current count of sessions.
SYS	Accumulated system mode CPU time (seconds).
THR	Current count of threads.
USR	Accumulated user mode CPU time (seconds).
VSZ	Virtual memory size of server process (in kilobytes). If a process is growing in size, it probably has a memory leak. If it is an Interaction Manager process, the culprit is most likely a component or dynamic object (though Interaction Manager servers do grow and shrink from garbage collection during normal use).

NS_STAT

The NS process displays the namespace for the current One-To-One environment, and optionally can update objects in a name space.

Bind
List
New
Rebnd
Rsolv
Unbnd

BV_CACHE_STAT

Monitors the request cache status.

The available counters for each request are:

CNT- Request_Name-HIT	Count of requests found in the cache.
CNT- Request_Name-MAX	Maximum size of the cache in bytes.
CNT- Request_Name-SWAP	Count of items that got swapped out of the cache.
CNT- Request_Name-MISS	Count of requests that were not in the cache.
CNT- Request_Name-SIZE	Count of items currently in the cache.

Cache Metrics

Cache metrics are available for the following items:

AD	
ALERTSCHED	Notification schedules are defined in the BV_ALERTSCHED and BV_MSGSCHED tables. They are defined by the One-To-One Command Center user or by an application.
CATEGORY_CONTENT	
DISCUSSION	The One-To-One discussion groups provide moderated system of messages and threads of messages aligned to a particular topic. Use the Discussion group interfaces for creating, retrieving and deleting individual messages in a discussion group. To create, delete, or retrieve discussion groups, use the generic content management API. The BV_DiscussionDB object provides access to the threads and messages in the discussion group database.
EXT_FIN_PRODUCT	

EDITORIAL	Using the Editorials content module, you can point cast and community cast personalized editorial content, and sell published text on your One-To-One site. You can solicit editorial content, such as investment reports and weekly columns, from outside authors and publishers, and create your own articles, reviews, reports, and other informative media. In addition to text, you can use images, sounds, music, and video presentations as editorial content.
INCENTIVE	Contains sales incentives
MSGSCHED	Contains the specifications of visitor-message jobs. Notification schedules are defined in the BV_ALERTSCHED and BV_MSGSCHED tables. They are defined by the One-To-One Command Center user or by an application.
MSGSCRIPT	Contains the descriptions of the JavaScripts that generate visitor messages and alert messages. Contains the descriptions of the JavaScripts that generate targeted messages and alert messages. Use the Command Center to add message script information to this table by selecting the Visitor Messages module in the Notifications group. For more information, see the Command Center User's Guide.
PRODUCT	BV_PRODUCT contains information about the products that a visitor can purchase.
QUERY	BV_QUERY contains queries.
SCRIPT	BV_SCRIPT contains page scripts.

SECURITIES	
TEMPLATE	The Templates content module enables you to store in the content database any BroadVision page templates used on your One-To-One site. Combining BroadVision page templates with BroadVision dynamic objects in the One-To-One Design Center application is one way for site developers to create One-To-One Web sites. If your developers use these page templates, you can use the Command Center to enter and manage them in your content database. If your site doesn't use BroadVision page template, you will not use this content module.

JS_SCRIPT_CTRL

CACHE
DUMP
FLUSH
METER
TRACE

JS_SCRIPT_STAT

ALLOC
ERROR
FAIL
JSPPERR
RELEASE
STOP
SUCC
SYNTAX

ColdFusion Monitoring

The ColdFusion monitor displays statistics about the resource usage on the ColdFusion server during the scenario or session step run.

To obtain data for this graph, you need to enable the ColdFusion online monitor (from the Controller or Tuning Console) and select the default measurements you want to display, before running the scenario or session step. You select measurements to monitor the ColdFusion server using the ColdFusion Configuration dialog box.

Note: The ColdFusion monitor works via HTTP and supports UNIX platforms. If you want to monitor the ColdFusion server on Windows platforms, you can also use the Windows Resource monitor.

Setting up the Monitoring Environment

Copy the **<LoadRunner installation>\dat\monitors\perfmon.cfm** file into the **<ColdFusion Home>\cfide\administrator** directory. By default, the ColdFusion monitor checks for the **<ColdFusion Home>\cfide\administrator\perfmon.cfm** file.

Note: The port you use to monitor a ColdFusion server through a firewall depends on the configuration of your server.

Adding a Monitored Machine to the Controller

1. Click the ColdFusion graph in the graph tree, and drag it into the right pane of the Run view.
2. Right-click the graph and select **Add Measurements**, or choose **Monitors > Add Measurements**. The ColdFusion dialog box opens.
3. In the **Monitored Server Machines** section, click **Add.** The Add Machine dialog box opens.

4 Enter the server name or IP address of the machine you want to monitor. Select the platform on which the machine runs, and click **OK**.

5 In the **Resource Measurements** section of the ColdFusion dialog box, click **Add**.

Continue with Configuring the ColdFusion Monitor below.

Adding a Monitored Machine to the Tuning Console

1 Click **Monitors** to open the Monitors Configuration dialog box.

2 Select the server whose monitors you want to configure from the Server list box.

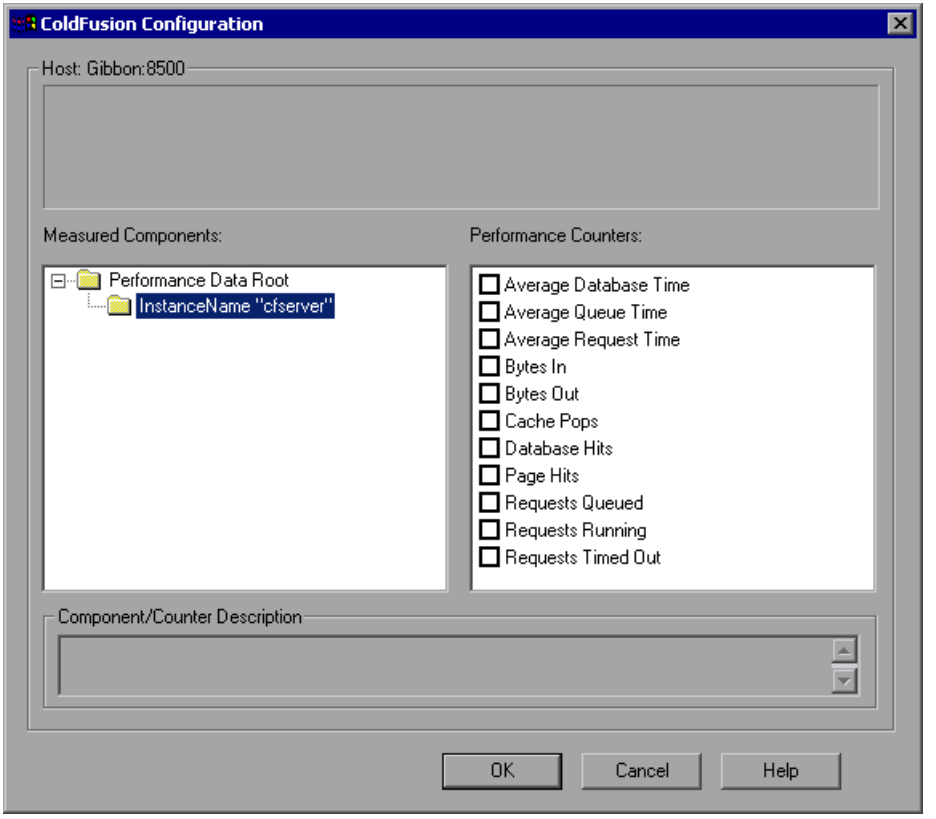
3 Click **Add Monitor** and select **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the Application Server category, select **ColdFusion**, and then click **Add**.

Continue with Configuring the ColdFusion Monitor below.

Configuring the ColdFusion Monitor

1 The ColdFusion Monitor Configuration dialog box opens, displaying the available measurements.



Browse the Measured Components tree, and check the required performance counters, as described in “Understanding the ColdFusion Dialog Box” on page 154.

For a description of the available measurements, see “ColdFusion Performance Counters” on page 155.

2 In the Controller, click **OK** in the ColdFusion Monitor Configuration dialog box, and in the ColdFusion dialog box, to activate the monitor.

3 In the Tuning Console, click **OK** in the ColdFusion Configuration dialog box, and in the Select Measurements to Monitor dialog box, to activate the monitor.

Understanding the ColdFusion Dialog Box

The ColdFusion dialog box lets you select the items to monitor on the ColdFusion application server.

Host: Displays the name of the monitored machine.

Measured Components: Displays the available components. Browse the tree and select the component you want to monitor. A description of the highlighted component appears in the Component/Counter Description box.

Performance Counters: Check the required performance counters. A description of the selected counter appears in the Component/Counter Description box.

Component/Counter Description: Displays a description of the selected component or counter.

ColdFusion Performance Counters

The following table describes the default counters that can be measured:

Measurement	Description
Avg. Database Time (msec)	The running average of the amount of time, in milliseconds, that it takes ColdFusion to process database requests.
Avg. Queue Time (msec)	The running average of the amount of time, in milliseconds, that requests spent waiting in the ColdFusion input queue before ColdFusion began to process the request.
Avg Req Time (msec)	The running average of the total amount of time, in milliseconds, that it takes ColdFusion to process a request. In addition to general page processing time, this value includes both queue time and database processing time.
Bytes In/sec	The number of bytes per second sent to the ColdFusion server.
Bytes Out/sec	The number of bytes per second returned by the ColdFusion server.
Cache Pops	Cache pops.
Database Hits/sec	This is the number of database hits generated per second by the ColdFusion server.
Page Hits/sec	This is the number of Web pages processed per second by the ColdFusion server.
Queued Requests	The number of requests currently waiting to be processed by the ColdFusion server.
Running Requests	The number of requests currently being actively processed by the ColdFusion server.
Timed Out Requests	The number of requests that timed out due to inactivity timeouts.

Fujitsu INTERSTAGE Monitoring

The Fujitsu INTERSTAGE monitor displays statistics about the resource usage on the Fujitsu server during the scenario or session step run. The Fujitsu INTERSTAGE monitor uses SNMP to retrieve Fujitsu INTERSTAGE server statistics.

To obtain data for this graph, you need to enable the Fujitsu INTERSTAGE online monitor (from the Controller or Tuning Console) and select the default measurements you want to display, before running the scenario or session step. You define the measurements for the Fujitsu INTERSTAGE monitor using the Fujitsu INTERSTAGE SNMP Resources dialog box.

Adding a Monitored Machine to the Controller

1. Click the Fujitsu INTERSTAGE graph in the graph tree, and drag it into the right pane of the Run view.
2. Right-click the graph and select **Add Measurements**, or choose **Monitors** > **Add Measurements**. The Fujitsu INTERSTAGE dialog box opens.
3. In the **Monitored Server Machines** section, click **Add.** The Add Machine dialog box opens.
4. Enter the server name or IP address of the machine you want to monitor. Select the platform on which the machine runs, and click **OK**.

Note: If the Fujitsu INTERSTAGE SNMP agent is running on a different port than the default SNMP port 161, you need to define the port number. Enter the following information in the Add Machine dialog box:
<server name:port number>
For example: digi:8888

In addition, you can define the default port for your Fujitsu INTERSTAGE server in the configuration file, **snmp.cfg**, located in **<LoadRunner root folder>\dat\monitors**. For example, if the port used by the SNMP agent on your Fujitsu INTERSTAGE system is 8888, you should edit the **snmp.cfg** file as follows:

```
; Fujitsu INTERSTAGE
[cm_snmp_mon_isp]
port=8888
```

---

5 In the **Resource Measurements** section of the Fujitsu INTERSTAGE dialog box, click **Add**.

Continue with Configuring the Fujitsu INTERSTAGE Monitor below.

Adding a Monitored Machine to the Tuning Console

1 Click **Monitors** to open the Monitors Configuration dialog box.

2 Select the server whose monitors you want to configure from the Server list box.

3 Click **Add Monitor** and select **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the Application Server category, select **Fujitsu INTERSTAGE**, and then click **Add**.

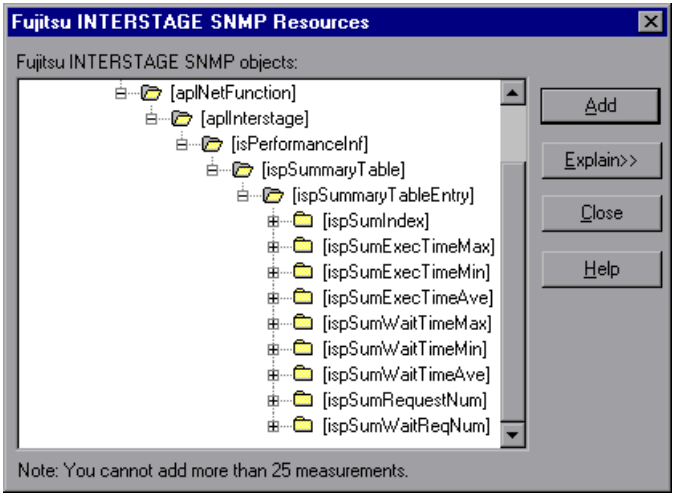
Note: If the Fujitsu INTERSTAGE SNMP agent is running on a different port than the default SNMP port 161, you need to define the port number. You can define the default port for your Fujitsu INTERSTAGE server in the configuration file, **snmp.cfg**, located in **<LoadRunner root folder>\dat\monitors**. For example, if the port used by the SNMP agent on your Fujitsu INTERSTAGE system is 8888, you should edit the **snmp.cfg** file as follows:

```
; Fujitsu INTERSTAGE
[cm_snmp_mon_isp]
port=8888
```

Continue with Configuring the Fujitsu INTERSTAGE Monitor below.

Configuring the Fujitsu INTERSTAGE Monitor

1 The Fujitsu INTERSTAGE SNMP Resources dialog box opens.



2 Browse the Fujitsu INTERSTAGE SNMP Object tree, and select the measurements you want to monitor, as described in "Understanding the Fujitsu INTERSTAGE Dialog Box" on page 159.

For a description of the available measurements, see "Fujitsu INTERSTAGE Performance Counters" on page 160.

3 Add all the desired resources to the list, and click **Close**.

Note: The Fujitsu INTERSTAGE monitor can only monitor up to 25 measurements.

---

4 In the Controller, click **OK** in the Fujitsu INTERSTAGE dialog box to activate the monitor.

5 In the Tuning Console, click **OK** in the Select Measurements to Monitor dialog box to activate the monitor.

Understanding the Fujitsu INTERSTAGE Dialog Box

The Fujitsu INTERSTAGE dialog box lets you select the Fujitsu INTERSTAGE server resources to monitor using the Simple Network Management Protocol (SNMP).

Fujitsu INTERSTAGE SNMP Objects: Select each required object and click **Add**. Click **Explain** for the ID number and a description of the selected object.

Explain: Displays a description of the selected object.

Note: The Fujitsu INTERSTAGE monitor can only monitor up to 25 measurements.

Fujitsu INTERSTAGE Performance Counters

The following tables describe the measurements that can be monitored:

Measurement	Description
IspSumObjectName	The object name of the application for which performance information is measured
IspSumExecTimeMax	The maximum processing time of the application within a certain period of time
IspSumExecTimeMin	The minimum processing time of the application within a certain period of time
IspSumExecTimeAve	The average processing time of the application within a certain period of time
IspSumWaitTimeMax	The maximum time required for INTERSTAGE to start an application after a start request is issued
IspSumWaitTimeMin	The minimum time required for INTERSTAGE to start an application after a start request is issued
IspSumWaitTimeAve	The average time required for INTERSTAGE to start an application after a start request is issued
IspSumRequestNum	The number of requests to start an application
IspSumWaitReqNum	The number of requests awaiting application activation

iPlanet (NAS) Monitoring

This Web Application Server Resource displays statistics about the resource usage on the iPlanet (NAS) Web application server during the scenario or session step run. The iPlanet (NAS) monitor uses the SNMP to retrieve iPlanet (NAS) server statistics.

To obtain data for this graph, you must first configure the iPlanet SNMP Service on the application server. You can then enable the iPlanet (NAS) online monitor (from the Controller or Tuning Console) and select the default measurements you want to display, before running the scenario or session step. You define the measurements for the iPlanet (NAS) monitor using the iPlanet (NAS) dialog box.

Setting Up the Monitoring Environment

Before you can configure the monitor, you need to set up the application server for SNMP monitoring.

This section offers a short explanation on setting up SNMP monitoring of the iPlanet Application Server. It is intended to supplement the iPlanet documentation, not act as a replacement. For an explanation of the SNMP reporting architecture and theory, refer to the iPlanet documentation.

Note: The instructions below assume that SNMP statistics will be collected on the standard SNMP port 161.

SNMP Summary

- Solaris has a native SNMP agent, **snmpdx**, that is started automatically at boot time by the script **/etc/rc3.d/S76snmpdx**. This daemon communicates on the standard SNMP port 161. The port number can be changed with the -p <port> option.
- Planet Products are shipped with their own SNMP agents. The architecture is such that there is one "master agent" per host, which a network management station communicates with, and one or more "subagents" that collect data from various iPlanet products and forward statistics to the master agent. The master agent also defaults to communicating on port 161.
- To run both the Solaris SNMP agent and the iPlanet SNMP agent, a proxy must be used that makes the Sun agent look like a subagent to the iPlanet master agent.

Steps Overview

- Login to the system as root
- Change the port number for the Solaris SNMP agent
- Configure and run the iPlanet agents "magt" and "sagt"
- Start the Solaris SNMP agent
- Configure iPlanet Application Server for SNMP statistics
- Start SNMP subagents for iPlanet Directory Server and iPlanet Web Server (optional)

Changing the Port Number for the Solaris SNMP Agent

1. Login to the system as root. (Only a root user can change the port number and run the agents).
2. Stop the SNMP agent by running /etc/rc2.d/K76snmpdx stop.
3. Edit /etc/rc3.d/S76snmpdx to run the Solaris daemon on a non-standard port number. For example, 1161:
   Replace
   /usr/lib/snmp/snmpdx -y -c /etc/snmp/conf
   with
   /usr/lib/snmp/snmpdx -p 1161 -y -c /etc/snmp/conf

Configuring and Running the iPlanet Agents

The master and proxy agents and startup scripts are found in **<ias install directory>\snmp**.

1. In the script S75snmpagt, add a line to the environment variable GX_ROOTDIR so that it points to your iAS installation. For example, if the iPlanet Application Server is installed in /usr/iplanet/ias6/ias:

   ```
   GX_ROOTDIR=/usr/iplanet/ias6/ias
   exprt GX_ROOTDIR
   ```

2. Copy the script S75snmpagt to /etc/rc3.d
3. chmod 755 /etc/rc3.d/S75snmpagt
4. In /etc/rc3.d/S75snmpagt /etc/rc2.d/K07snmpagt
5. You can configure system information and traps.

   In the example below, information has been added about the system owner and location, and SNMP traps have been sent to a network manager station ("mde.uk.sun.com").

   ```
   COMMUNITY      public
   ALLOW ALL OPERATIONS
   INITIAL sysLocation "Under Joe Bloggs' Desk in Headquarters"
   INITIAL sysContact "Joe Bloggs
   Email: Joe.Bloggs@Sun.COM
   Voice: +1 650 555 1212"
   MANAGER mde.uk.sun.com
   SEND ALL TRAPS TO PORT 162
   WITH COMMUNITY public
   ```

Note: There is no need to edit the proxy agent's configuration file (CONFIG_SAGT).

---

6. Start the iPlanet agents by running the command:
   /etc/rc3.d/S75snmpagt start

To start the Solaris SNMP agent:

Restart the Solaris SNMP agent by running the command:
/etc/rc3.d/S76snmpdx start

Configuring the iPlanet Application Server for SNMP Statistics

1. Start the iPlanet Application Server admin tool ksvradmin.
2. In the General View, select the instance name that you want to manage.
3. Click the **SNMP** tab in the management frame.
4. Select **Enable SNMP Administration and Monitoring** and **Enable SNMP Debug**.
5. Type 60 in the Connection Attempt Interval field, and exit ksvradmin.
6. Restart the iPlanet Application Server with the commands:

   iascontrol stop
   iascontrol kill
   iascontrol start

7. Check in the logfile **<iASInstallDir>/logs/ias.log** that the application server successfully connected to the master agent. You should see the following line:
   kas> SNMP: Connected to master agent

Starting SNMP Subagents for the iPlanet Web Server

1. Use your Web browser to access the iPlanet Web Server.
2. Choose the Web server you wish to administer, and click the **Manage** button.
3. Select the **Monitor** tab, and click **SNMP Subagent Configuration** on the left side of the page.
4. Type in the configuration information and set the radio button **Enable SNMP Statistics Collection** to **On**.
5. Click **SNMP Subagent Control**.
6. Click the **Start** button.

Starting SNMP Subagents for the iPlanet Directory Server

1. Use the Netscape Administration Console to manage the iPlanet Directory Server.
2. Select the **Configuration** tab.
3. Click the **SNMP** tab in the Configuration frame.
4. Select the **Enable statistics collection** check box.
5. Set "Master Host" to "localhost".
6. Set "Master port" to 199.
7. In the other fields, enter the appropriate information.
8. Click the **Start Subagent** button.

Summary note

Use your SNMP management tool to query the SNMP master agent on port 161. You should see all the information provided by the Solaris SNMP agent as well as any iPlanet subagents that you have configured.

The next time that you boot Solaris, the Sun and iPlanet SNMP agents will be started automatically by the boot scripts which you have configured.

Adding a Monitored Machine to the Controller

To configure the iPlanet (NAS) Resources monitor:

1. Click the iPlanet (NAS) graph in the graph tree, and drag it into the right pane of the Run view.
2. Right-click the graph and select **Add Measurements**, or choose **Monitors** > **Add Measurements**. The iPlanet (NAS) dialog box opens.
3. In the **Monitored Server Machines** section, click **Add.** The Add Machine dialog box opens.

4 Enter the server name or IP address of the machine you want to monitor. Select the platform on which the machine runs, and click **OK**.

Note: If the iPlanet SNMP agent is running on a different port than the default SNMP port, you must define the port number. Enter the following information in the Add Machine dialog box:
<server name:port number>
For example: digi:8888

In addition, you can define the default port for your iPlanet server in the configuration file, **snmp.cfg**, located in **<LoadRunner root folder>\dat\monitors**. For example, if the port used by the SNMP agent on your iPlanet server is 8888, you should edit the **snmp.cfg** file as follows:

```
; iPlanet (NAS)
[cm_snmp_mon_nas]
port=8888
```

---

5 In the **Resource Measurements** section of the iPlanet (NAS) dialog box, click **Add**.

Continue with Configuring the iPlanet (NAS) Monitor below.

Adding a Monitored Machine to the Tuning Console

1. Click **Monitors** to open the Monitors Configuration dialog box.
2. Select the server whose monitors you want to configure from the Server list box.
3. Click **Add Monitor** and select **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.
4. In the left section of the dialog box, expand the Application Server category, select **iPlanet (NAS)**, and then click **Add**.

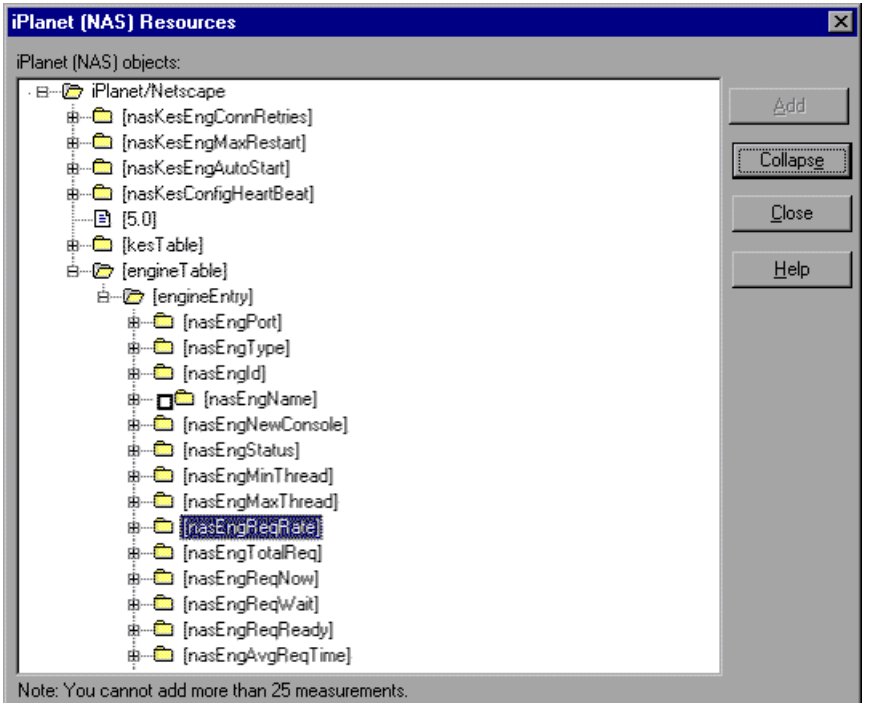
Note: If the iPlanet SNMP agent is running on a different port than the default SNMP port, you need to define the port number. You can define the default port for your iPlanet server in the configuration file, **snmp.cfg**, located in **<LoadRunner root folder>\dat\monitors**. For example, if the port used by the SNMP agent on your iPlanet server is 8888, you should edit the **snmp.cfg** file as follows:

```
; iPlanet (NAS)
[cm_snmp_mon_nas]
port=8888
```

Continue with Configuring the iPlanet (NAS) Monitor below.

Configuring the iPlanet (NAS) Monitor

1 The iPlanet (NAS) Resources dialog box opens.



2 Browse the iPlanet (NAS) Resources Object tree, and select performance counters, as described in “Understanding the iPlanet (NAS) Dialog Box” on page 170.

For a description of the available measurements, see “iPlanet (NAS) Performance Counters” on page 170.

3 Add all the desired resources to the list, and click **Close**.

Note: The iPlanet (NAS) monitor can only monitor up to 25 measurements.

---

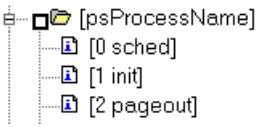
4 In the Controller, click **OK** in the iPlanet (NAS) dialog box to activate the monitor.

5 In the Tuning Console, click **OK** in the Select Measurements to Monitor dialog box to activate the monitor.

Improving the Level of Measurement Information

You can improve the level of measurement information for the iPlanet (NAS) monitor by enabling measurements with string values to be listed (in addition to measurements with numeric values), and by enabling the name modifier (which displays the string value as an identifying part of the measurement name).

In the following example of a measurement using the name modifier, the string value of ProcessName (sched) is displayed in addition to its instance ID (0):



To enable this feature, add the following line to the **<LoadRunner root folder>\dat\monitors\snmp.cfg** file:

```
SNMP_show_string_nodes=1
```

Usage Notes: You can select more than one name modifier, but the first in the hierarchy will be used. Each time the iPlanet (NAS) Add Measurements dialog box opens, the information is reread from the **snmp.cfg** file. You cannot add the same measurement twice (once with a name modifier and once without it). If you do so, an error message is issued.

Understanding the iPlanet (NAS) Dialog Box

The iPlanet (NAS) dialog box lets you select the iPlanet (NAS) server resources to monitor using the Simple Network Management Protocol (SNMP).

iPlanet (NAS) SNMP Objects: Select each required object and click **Add**. Click **Explain** for the ID number and a description of the selected object.

Explain: Displays a description of the selected object.

Note: The iPlanet (NAS) monitor can only monitor up to 25 measurements.

---

iPlanet (NAS) Performance Counters

The following tables describe the counters that can be monitored:

Netscape Performance Counters

Measurement	Description
nasKesEngConn Retries	The maximum number of times the administration server will try to connect to an engine.
nasKesEngMax Restart	The maximum number of times the administration server will restart an engine after a failure.
nasKesEngAutoStart	Start all the engines at startup of the administration server.
nasKesConfigHeart Beat	Heart Beat.

KES Performance Counters

Measurement	Description
nasKesId	The ID of the KES this engine belongs to.
nasKesMinThread	The default minimum number of threads per engine.

Measurement	Description
nasKesMaxThread	The default maximum number of threads per engine.
nasKesLoadBalancer Disable	Enable or Disable the load balancer service.
nasKesCpuLoad	The total CPU usage on this host.
nasKesDiskLoad	The total disk usage on this host.
nasKesMemLoad	The total memory usage on this host.
nasKesRequestLoad	The number of requests on this NAS.
nasKesCpuLoad Factor	The relative importance of CPU usage in computing the server load. This number is specified as a percent. The sum of all server load factors, CPULoad, DiskLoad, MemLoad and ExecReqs must equal 100%.
nasKesDiskLoad Factor	The relative importance of Disk usage in computing the server load. This number is specified as a percent. The sum of all server load factors, CPULoad, DiskLoad, MemLoad and ExecReqs must equal 100%.
nasKesMemLoad Factor	The relative importance of Memory usage in computing the server load. This number is specified as a percent. The sum of all server load factors, CPULoad, DiskLoad, MemLoad and ExecReqs must equal 100%.
nasKesAppLogics RunningFactor	The relative importance of the number of times an AppLogic is run in computing the AppLogic execution performance. This figure is specified as a percent. The sum of all agent load factors, ResultCached, AvgExecTime, LastExecTime, and ServerLoad must equal 100%.
nasKesResultsCached Factor	The relative importance of the cached results of an AppLogic in computing the AppLogic execution performance. This figure is specified as a percent. The sum of all agent load factors, ResultCached, AvgExecTime, LastExecTime, and ServerLoad must equal 100%

Measurement	Description
nasKesAvgExecTime Factor	The relative importance of the average execution time of an AppLogic in computing the AppLogic execution performance. This figure is specified as a percent. The sum of all agent load factors, ResultCached, AvgExecTime, LastExecTime, and ServerLoad must equal 100%.
nasKesLastExecTime Factor	The relative importance of the last execution time of an AppLogic in computing the AppLogic execution performance. This figure is specified as a percent. The sum of all agent load factors, ResultCached, AvgExecTime, LastExecTime, and ServerLoad must equal 100%.
nasKesHitsFactor	The relative importance of the number of AppLogics running in computing the AppLogic execution performance. This figure is specified as a percent. The sum of all agent load factors, ResultCached, AvgExecTime, LastExecTime, and ServerLoad must equal 100%.
nasKesServerLoad Factor	The relative importance of the server load (computed using the four server load factors) in computing AppLogic execution performance. The sum of all agent load factors, ResultCached, AvgExecTime, LastExecTime, and ServerLoad must equal 100%.
nasKesBroadcast Interval	The length of time in seconds, between each broadcast attempt from the load balancer daemon.
nasKesApplogic BroadcastInterval	The length of time in seconds, between each broadcast of AppLogics load information across all the server in the cluster. This should be greater than nasKesBroacastInterval.
nasKesServer BroadcastInterval	The length of time in seconds, between each broadcast of server load information across all the server in the cluster. This should be greater than nasKesBroacastInterval.

Measurement	Description
nasKesServerLoad UpdateInterval	The length of time in seconds between each update of server load informations. A server load update applies the server load data that has been sampled up until the moment when the update occurs.
nasKesCpuLoad UpdateInterval	The length of time, in seconds, between each sampling of CPU usage.
nasKesDiskLoad UpdateInterval	The length of time, in seconds, between each sampling of disk usage.
nasKesMemLoad UpdateInterval	The length of time, in seconds, between each sampling of memory thrashes.
nasKesTotalReqs UpdateInterval	The length of time, in seconds, between each sampling of the number of requests.
nasKesMaxHops	The maximum number of times a request can be load-balanced to another server.
nasKesODBCReqMin Thread	The minimum number of threads reserved to process asynchronous requests.
nasKesODBCReq MaxThread	The maximum number of threads reserved to process asynchronous requests.
nasKesODBCCache MaxConns	The maximum number of connections opened between NAS and the database.
nasKesODBCCache FreeSlots	The minimum number of cached connections established between NAS and the database.
nasKesODBCCache Timeout	The time after which an idle connection is dropped.
nasKesODBCCache Interval	The interval in seconds at which the cache cleaner will try to disconnect connections already idle for longer than the specified timeout.
nasKesODBCConn GiveupTime	Maximum time the driver will try to connect to the database.
nasKesODBCCache Debug	Turns on the connection cache debug information.

Measurement	Description
nasKesODBCResult SetInitRows	The number of rows fetched at once from the database.
nasKesODBCResult SetMaxRows	The maximum number of rows the cached result set can contain.
nasKesODBCResult SetMaxSize	The maximum size of result set the driver will cache
nasKesODBCSql Debug	Turns on SQL debug information.
nasKesODBCEnable Parser	Turns on SQL parsing.
nasKesORCLReqMin Thread	The minimum number of threads reserved to process asynchronous requests.
nasKesORCLReqMax Thread	The maximum number of threads reserved to process asynchronous requests.
nasKesORCLCache MaxConns	The maximum number of connections opened between NAS and the database.
nasKesORCLCache FreeSlots	The minimum number of cached connections established between NAS and the database.
nasKesORCLCache Timeout	The time after which an idle connection is dropped.
nasKesORCLCache Interval	The interval in seconds at which the cache cleaner will try to disconnect connections already idle for longer than the specified timeout.
nasKesORCLConn GiveupTime	The maximum time the driver will spend trying to obtain a connection to Oracle.
nasKesORCLCache Debug	Turns on the connection cache debug information.
nasKesORCLResult SetInitRows	The number of rows fetched at once from the database.
nasKesORCLResult SetMaxRows	The maximum number of rows the cached result set can contain.

Measurement	Description
nasKesORCLResult SetMaxSize	The maximum size of result set the driver will cache.
nasKesORCLSql Debug	Turns on SQL debug information.
nasKesSYBReqMin Thread	The minimum number of threads reserved to process asynchronous requests.
nasKesSYBReqMax Thread	The maximum number of threads reserved to process asynchronous request.
nasKesSYBCache MaxConns	The maximum number of connections opened between NAS and the database.
nasKesSYBCache FreeSlots	The minimum number of cached connections established between NAS and the database.
nasKesSYBCache Timeout	The time after which an idle connection is dropped.
nasKesSYBCache Interval	The interval time between cached connections.
nasKesSYBConn GiveupTime	The maximum time the driver will spend trying to obtain a connection to Sybase before giving up.
nasKesSYBCache Debug	Turns on the connection cache debug information.
nasKesSYBResult SetInitRows	The number of rows fetched at once from the database.
nasKesSYBResult SetMaxRows	The maximum number of rows the cached result set can contain.
nasKesSYBResult SetMaxSize	The maximum size of result set the driver will cache.

Engine Performance Counters

Measurement	Description
nasEngKesPort	The port of the KXS this engine serves. This is supplied as part of the object ID and cannot be modified after creation.
nasEngPort	The TCP/IP port this engine is listening on. The port can only be specified at the creation of the engine. It is not allowed to modify it.
nasEngType	Type of the engine: executive(0), Java(1000), C++(3000).
nasEngId	The ID is an incremental number starting at 0. The ID cannot be modified.
nasEngName	The name of this engine. This is an informational string that contains kcs, kxs ot kjs.
nasEngNewConsole	Starts each engine in a new console window.
nasEngStatus	The status column used to add, remove, enable or disable an engine. To create an engine, one needs to set. This follows rfc1443.
nasEngMinThread	The default minimum number of threads per engine.
nasEngMaxThread	The default maximum number of threads per engine.
nasEngReqRate	The rate at which requests arrive.
nasEngTotalReq	The total number of requests processed since engine startup.
nasEngReqNow	The number of requests being processed.
nasEngReqWait	The requests waiting to be serviced.
nasEngReqReady	The requests that are ready to be serviced.
nasEngAvgReqTime	The average request processing time.
nasEngThreadNow	Number of threads in use by the request manager.
nasEngThreadWait	The number of idle threads.
nasEngWebReq Queue	The number of web requests that are queued.

Measurement	Description
nasEngFailedReq	The number of requests that failed.
nasEngTotalConn	The total number of connections opened.
nasEngTotalConn Now	The total number of connections in use.
nasEngTotalAccept	The total number of connections listening to incoming requests.
nasEngTotalAccept Now	The total number of connections listening to incoming connections in use.
nasEngTotalSent	The total number of packets sent.
nasEngTotalSentBytes	The total number of bytes sent.
nasEngTotalRecv	The total number of packets received.
nasEngTotalRecvBytes	The total number of bytes received.
nasEngBindTotal	The number of AppLogic bound since startup.
nasEngBindTotal Cached	The number of AppLogic cached since startup.
nasEngTotalThreads	Total number of threads created in this process.
nasEngCurrent Threads	Total number of threads in use in this process.
nasEngSleeping Threads	Number of threads sleeping in this process.
nasEngDAETotal Query	Total number of queries executed since startup.
nasEngDAEQuery Now	The number of queries being processed.
nasEngDAETotal Conn	The number of logical connections created since startup.
nasEngDAEConn Now	The number of logical connections in use.

Measurement	Description
nasEngDAECache Count	The number of caches.
nasEngODBCQuery Total	Total number of queries executed since startup.
nasEngODBC PreparedQueryTotal	Total number of odbc prepared queries executed since startup.
nasEngODBCConn Total	Total number of connections opened since startup.
nasEngODBCConn Now	Number of connections currently opened.
nasEngORCLQuery Total	Total number of queries executed since startup.
nasEngORCL PreparedQueryTotal	Total number of prepared queries executed since startup.
nasEngORCLConn Total	Total number of connections established with Oracle since startup.
nasEngORCLConn Now	Number of connections opened with Oracle now.
nasEngSYBQuery Total	Total number of queries the driver processed since startup.
nasEngSYBPrepared QueryTotal	Total number of prepared queries processed since startup.
nasEngSYBConnTotal	Total number of connections opened since startup.
nasEngSYBConnNow	Number of SYB connections opened now.
nasStatusTrapEntry	The KES definition.
nasTrapKesIpAddress	The IP Address of KES host.
nasTrapKesPort	The port of the main engine of this NAS.
nasTrapEngPort	The port of the engine generating this event.
nasTrapEngState	The port of the engine generating this event.

Microsoft Active Server pages Monitoring

The Microsoft Active Server Pages (ASP) montor displays statistics about the resource usage on the ASP server during the scenario or session step run.

To obtain data for this graph, you need to enable the Microsoft ASP online monitor (from the Controller or Tuning Console) and select the default measurements you want to display, before running the scenario or session step. You select measurements to monitor the Microsoft ASP application server using the MS Active Server Pages dialog box.

Note: To monitor an ASP server through a firewall, use TCP, port 139.

Adding a Monitored Machine to the Controller

1. Click the MS Active Server Pages graph in the graph tree, and drag it into the right pane of the Run view.
2. Right-click the graph and select **Add Measurements**, or choose **Monitors > Add Measurements**. The MS Active Server Pages dialog box opens.
3. In the **Monitored Server Machines** section, click **Add.** The Add Machine dialog box opens.
4. Enter the server name or IP address of the machine you want to monitor. Select the platform on which the machine runs, and click **OK**.
5. In the **Resource Measurements** section of the MS Active Server Pages dialog box, select the default measurements you want to monitor.

   For a description of the available measurements, see "MS Active Server Pages Performance Counters" on page 182.

Note: To change the default counters for the Microsoft ASP monitor, see "Changing a Monitor's Default Counters" on page 443.

---

6 To select additional measurements, click **Add** in the **Resource Measurements** section.

Continue with Configuring the Microsoft Active Server Pages Monitor below.

Adding a Monitored Machine to the Tuning Console

1 Click **Monitors** to open the Monitors Configuration dialog box.

2 Select the server whose monitors you want to configure from the Server list box.

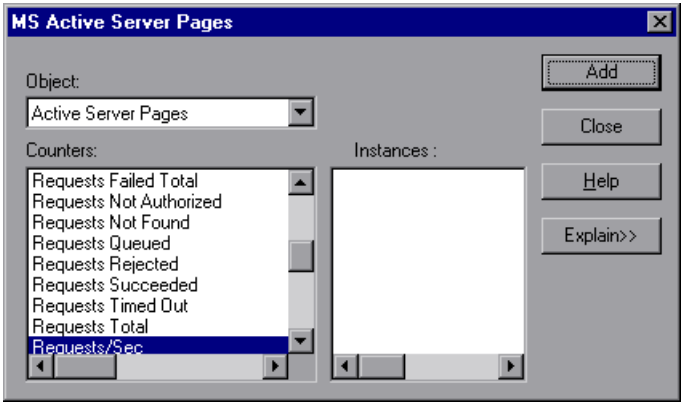
3 Click **Add Monitor** and select **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the Application Server category, select **MS Active Server Pages**, and then click **Add**.

Continue with Configuring the Microsoft Active Server Pages Monitor below.

Configuring the Microsoft Active Server Pages Monitor

1 The MS Active Server Pages dialog box opens displaying the Active Server Pages object, its counters, and instances.



For each measurement, select an object, counter, and instance, and then click **Add**, as described in "Understanding the Microsoft Active Server Pages Dialog Box" on page 181.

For a description of the available measurements, see "MS Active Server Pages Performance Counters" on page 182.

Note: To change the default counters for the Microsoft ASP monitor, see "Changing a Monitor's Default Counters" on page 443.

---

2 Add all the desired resources to the list, and click **Close**.

3 In the Controller, click **OK** in the MS Active Server Pages dialog box to activate the monitor.

4 In the Tuning Console, click **OK** in the Select Measurements to Monitor dialog box to activate the monitor.

Understanding the Microsoft Active Server Pages Dialog Box

The MS Active Server Pages dialog box lets you select the items to monitor on the MS Active Server Pages application server.

Object: Select the object being monitored on the specified machine.

Counters: Select a resource counter to monitor. Select multiple counters using the CTRL key. For an explanation of each counter, click **Explain**.

Instances: If multiple instances of the selected counter are running, select one or more instances to monitor for the selected counter.

Explain: Displays a description of the selected counter.

MS Active Server Pages Performance Counters

The following table describes the default counters that can be monitored:

Measurement	Description
Errors per Second	The number of errors per second.
Requests Wait Time	The number of milliseconds the most recent request was waiting in the queue.
Requests Executing	The number of requests currently executing.
Requests Queued	The number of requests waiting in the queue for service.
Requests Rejected	The total number of requests not executed because there were insufficient resources to process them.
Requests Not Found	The number of requests for files that were not found.
Requests/sec	The number of requests executed per second.
Memory Allocated	The total amount of memory, in bytes, currently allocated by Active Server Pages.
Errors During Script Run-Time	The number of failed requests due to run-time errors.
Sessions Current	The current number of sessions being serviced.
Transactions/sec	The number of transactions started per second.

Oracle9iAS HTTP Monitoring

The Oracle9iAS HTTP monitor displays statistics about the resource usage on the Oracle9iAS HTTP server during the scenario or session step run.

To obtain data for this graph, you need to enable the Oracle9iAS HTTP online monitor (from the Controller or Tuning Console), and select the default measurements you want to display, before running the scenario or session step. You select measurements to monitor the Oracle9iAS HTTP server using the Oracle HTTP Server Monitor Configuration dialog box. Note that you must start running the Oracle9iAS HTTP server before you begin selecting the measurements you want to monitor.

Note: The port you use to monitor an Oracle9iAS HTTP server through a firewall depends on the configuration of your server.

Adding a Monitored Machine to the Controller

1. Click the Oracle9iAS HTTP graph in the graph tree, and drag it into the right pane of the Run view.
2. Right-click the graph and select **Add Measurements**, or choose **Monitors** > **Add Measurements**. The Oracle9iAS HTTP Server dialog box opens.
3. In the **Monitored Server Machines** section, click **Add.** The Add Machine dialog box opens.
4. Enter the server name or IP address of the machine you want to monitor. Select any platform, and click **OK**.
5. In the **Resource Measurements** section of the Oracle9iAS HTTP Server dialog box, click **Add**.

   Continue with Configuring the Oracle9iAS HTTP Monitor below.

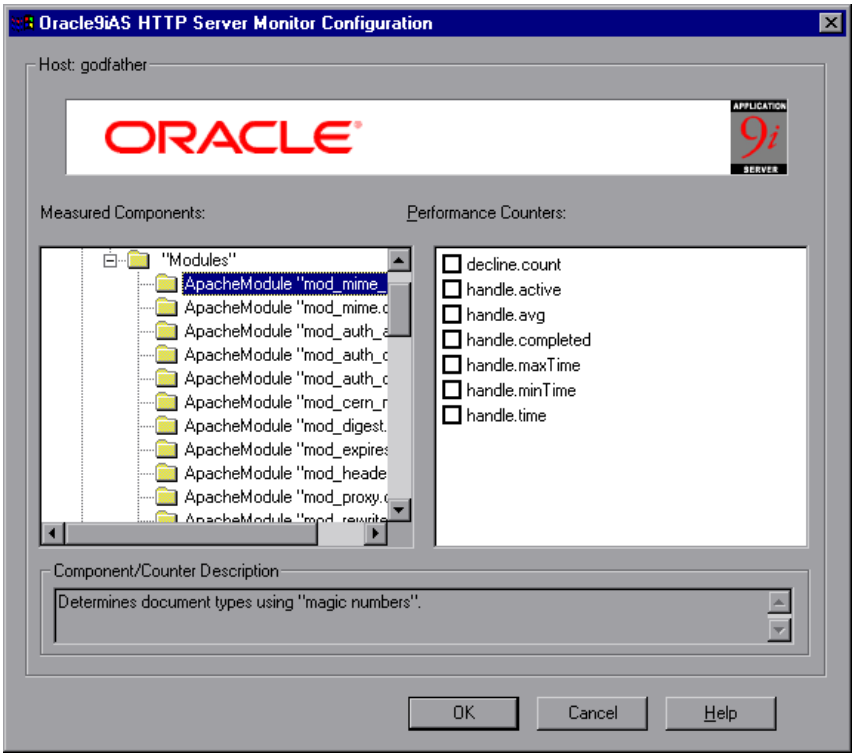
Adding a Monitored Machine to the Tuning Console

1. Click **Monitors** to open the Monitors Configuration dialog box.
2. Select the server whose monitors you want to configure from the Server list box.
3. Click **Add Monitor** and select **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.
4. In the left section of the dialog box, expand the Application Server category, select **Oracle9iAS HTTP Server**, and then click **Add**.

   Continue with Configuring the Oracle9iAS HTTP Monitor below.

Configuring the Oracle9iAS HTTP Monitor

1 The Oracle HTTP Server Monitor Configuration dialog box opens, displaying the counters that can be monitored.



2 Browse the Measured Components tree, and select performance counters, as described in "Understanding the Oracle9iAS HTTP Server Dialog Box" on page 186.

For a description of the available measurements, see "Oracle9iAS HTTP Server Performance Counters" on page 186.

3 In the Controller, click **OK** in the Oracle HTTP Server Monitor Configuration dialog box, and in the Oracle9iAS HTTP Server dialog box, to activate the monitor.

4 In the Tuning Console, click **OK** in the Oracle9iAS HTTP Server Monitor Configuration dialog box, and in the Select Measurements to Monitor dialog box, to activate the monitor.

Understanding the Oracle9iAS HTTP Server Dialog Box

The Oracle 9iAS HTTP Server dialog box lets you select the items to monitor on the Oracle 9iAS HTTP Web server. Note that you must start running the Oracle9iAS HTTP server before you begin selecting the measurements you want to monitor.

Host: Displays the name of the monitored machine.

Measured Components: Displays the available components. Browse the tree and select the component you want to monitor. A description of the highlighted component appears in the Component/Counter Description box.

Performance Counters: Check the required performance counters. A description of the selected counter appears in the Component/Counter Description box.

Component/Counter Description: Displays a description of the selected component or counter.

Oracle9iAS HTTP Server Performance Counters

The following table describes some of the modules that can be monitored:

Measurement	Description
mod_mime.c	Determines document types using file extensions
mod_mime_magic.c	Determines document types using "magic numbers"
mod_auth_anon.c	Provides anonymous user access to authenticated areas
mod_auth_dbm.c	Provides user authentication using DBM files
mod_auth_digest.c	Provides MD5 authentication
mod_cern_meta.c	Supports HTTP header metafiles

Measurement	Description
mod_digest.c	Provides MD5 authentication (deprecated by mod_auth_digest)
mod_expires.c	Applies Expires: headers to resources
mod_headers.c	Adds arbitrary HTTP headers to resources
mod_proxy.c	Provides caching proxy abilities
mod_rewrite.c	Provides powerful URI-to-filename mapping using regular expressions
mod_speling.c	Automatically corrects minor typos in URLs
mod_info.c	Provides server configuration information
mod_status.c	Displays server status
mod_usertrack.c	Provides user tracking using cookies
mod_dms.c	Provides access to DMS Apache statistics
mod_perl.c	Allows execution of perl scripts
mod_fastcgi.c	Supports CGI access to long-lived programs
mod_ssl.c	Provides SSL support
mod_plsql.c	Handles requests for Oracle stored procedures
mod_isapi.c	Provides Windows ISAPI extension support
mod_setenvif.c	Sets environment variables based on client information
mod_actions.c	Executes CGI scripts based on media type or request method
mod_imap.c	Handles imagemap files
mod_asis.c	Sends files that contain their own HTTP headers
mod_log_config.c	Provides user-configurable logging replacement for mod_log_common
mod_env.c	Passes environments to CGI scripts

Measurement	Description
mod_alias.c	Maps different parts of the host file system in the document tree, and redirects URLs
mod_userdir.c	Handles user home directories
mod_cgi.c	Invokes CGI scripts
mod_dir.c	Handles the basic directory
mod_autoindex.c	Provides automatic directory listings
mod_include.c	Provides server-parsed documents
mod_negotiation.c	Handles content negotiation
mod_auth.c	Provides user authentication using text files
mod_access.c	Provides access control based on the client hostname or IP address
mod_so.c	Supports loading modules (.so on UNIX, .dll on Win32) at run-time
mod_oprocmgr.c	Monitors JServ processes and restarts them if they fail
mod_jserv.c	Routes HTTP requests to JServ server processes. Balances load across multiple JServs by distributing new requests in round-robin order
mod_ose.c	Routes requests to the JVM embedded in Oracle's database server
http_core.c	Handles requests for static Web pages

The following table describes the counters that can be monitored:

Measurement	Description
handle.minTime	The minimum time spent in the module handler
handle.avg	The average time spent in the module handler
handle.active	The number of threads currently in the handle processing phase

Measurement	Description
handle.time	The total amount of time spent in the module handler
handle.completed	The number of times the handle processing phase was completed
request.maxTime	The maximum amount of time required to service an HTTP request
request.minTime	The minimum amount of time required to service an HTTP request
request.avg	The average amount of time required to service an HTTP request
request.active	The number of threads currently in the request processing phase
request.time	The total amount of time required to service an HTTP request
request.completed	The number of times the request processing phase was completed
connection.maxTime	The maximum amount of time spent servicing any HTTP connection
connection.minTime	The minimum amount of time spent servicing any HTTP connection
connection.avg	The average amount of time spent servicing HTTP connections
connection.active	The number of connections with currently open threads
connection.time	The total amount of time spent servicing HTTP connections
connection.completed	The number of times the connection processing phase was completed
numMods.value	The number of loaded modules
childFinish.count	The number of times the Apache parent server started a child server, for any reason

Measurement	Description
childStart.count	The number of times “children” finished “gracefully.”There are some ungraceful error/crash cases that are not counted in childFinish.count
Decline.count	The number of times each module declined HTTP requests
internalRedirect.count	The number of times that any module passed control to another module using an “internal redirect”
cpuTime.value	The total CPU time utilized by all processes on the Apache server (measured in CPU milliseconds)
heapSize.value	The total heap memory utilized by all processes on the Apache server (measured in kilobytes)
pid.value	The process identifier of the parent Apache process
upTime.value	The amount of time the server been running (measured in milliseconds)

SilverStream Monitoring

The SilverStream monitor displays statistics about the resource usage on the SilverStream server during the scenario or session step run. To obtain data for this graph, you need to enable the SilverStream online monitor (from the Controller or Tuning Console) and select the default measurements you want to display, before running the scenario or session step.

To monitor a SilverStream server you need to know the server statistics information URL. A simple way to verify the statistics URL is to access it from a browser.

The URL should be in the following format:

http://<server_name/IP_address>:<port_number>/SilverStream/Statistics

For example:

http://199.203.78.57:80/SilverStream/Statistics

Note: To monitor a SilverStream server through a firewall, use the Web server port (by default, port 80).

Adding a Monitored Machine to the Controller

1. Click the SilverStream graph in the graph tree, and drag it into the right pane of the Run view.
2. Right-click the graph and select **Add Measurements**, or choose **Monitors** > **Add Measurements**. The SilverStream dialog box opens.
3. In the **Monitored Server Machines** section, click **Add.** The Add Machine dialog box opens.
4. Enter the server name or IP address of the machine you want to monitor. Select the platform on which the machine runs, and click **OK**.
5. In the **Resource Measurements** section of the SilverStream dialog box, click **Add**.

Continue with Configuring the SilverStream Monitor below.

Adding a Monitored Machine to the Tuning Console

1 Click **Monitors** to open the Monitors Configuration dialog box.

2 Select the server whose monitors you want to configure from the Server list box.

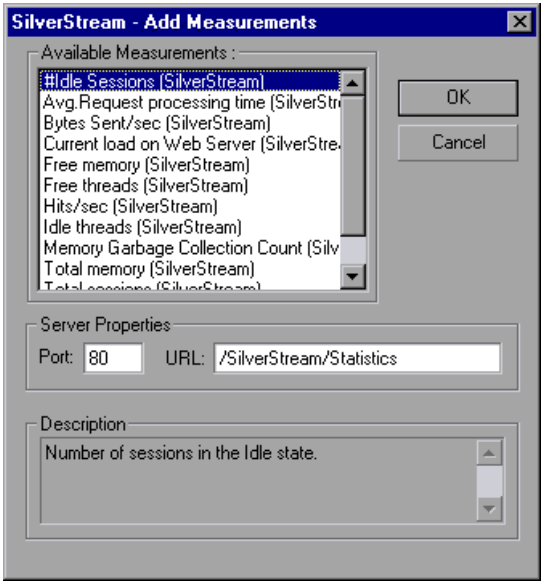
3 Click **Add Monitor** and select **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the Application Server category, select **SilverStream**, and then click **Add**.

Continue with Configuring the SilverStream Monitor below.

Configuring the SilverStream Monitor

1 The SilverStream dialog box opens displaying the available measurements and server properties.



2 Select the required measurements, as described in "Understanding the SilverStream Dialog Box" on page 194.

For a description of the available measurements, see "SilverStream Performance Counters" on page 195.

3 In the Server Properties section, enter the Port number and URL (without the server name), and click **OK**. The default URL is /SilverStream/Statistics.

4 In the Controller, click **OK** in the SilverStream dialog box to activate the monitor.

5 In the Tuning Console, click **OK** in the Select Measurements to Monitor dialog box to activate the monitor.

Note: The default port number and URL can vary from one server to another. Please consult the Web server administrator.

To change the default server properties:

1 Open the **SilverStream.cfg** file in the **<LoadRunner root folder>\dat\monitors** directory.

2 Edit the following parameters at the end of the file:

InfoURL	server statistics information URL
ServerPort	server port number
SamplingRate	rate (milliseconds) at which the LoadRunner monitor will poll the server for the statistics information. If this value is greater than 1000, LoadRunner will use it as its sampling rate. Otherwise, it will use the sampling rate defined in the Monitors tab of the Options dialog box.

Note: To monitor a SilverStream server through a firewall, use the Web server port (by default, port 80).

Understanding the SilverStream Dialog Box

The SilverStream dialog box lets you select the items to monitor on the SilverStream application server.

Available Measurements: Select the required measurement(s). Select multiple measurements using the CTRL key. A description of the selected measurement appears in the Description box.

Server Properties

- **Port:** Enter the server's port number.
- **URL:** To monitor a SilverStream server you need to know the server statistics information URL. A simple way to verify the statistics URL is to access it from a browser.

  The URL should be in the following format:

  http://<server_name/IP_address>:<port_number>/SilverStream/Statistics

  For example:

  http://199.203.78.57:80/SilverStream/Statistics

  Enter the server's URL, without the server name. The default URL is /SilverStream/Statistics.

Note: The default port number and URL can vary from one server to another. Please consult the Web server administrator.

Description: Displays a description of the selected measurement.

SilverStream Performance Counters

The following table describes the measurements and server properties that can be monitored:

Measurement	Description
#Idle Sessions	The number of sessions in the Idle state.
Avg. Request processing time	The average request processing time.
Bytes Sent/sec	The rate at which data bytes are sent from the Web server.
Current load on Web Server	The percentage of load utilized by the SilverStream server, scaled at a factor of 25.
Hits/sec	The HTTP request rate.
Total sessions	The total number of sessions.
Free memory	The total amount of memory in the Java Virtual Machine currently available for future allocated objects.
Total memory	The total amount of memory in the Java Virtual Machine.
Memory Garbage Collection Count	The total number of times the JAVA Garbage Collector has run since the server was started.
Free threads	The current number of threads not associated with a client connection and available for immediate use.
Idle threads	The number of threads associated with a client connection, but not currently handling a user request.
Total threads	The total number of client threads allocated.

WebLogic (SNMP) Monitoring

The WebLogic (SNMP) monitor displays statistics about the resource usage on the WebLogic (SNMP) server (version 6.0 and earlier) during the scenario or session step run. The WebLogic (SNMP) monitor uses SNMP to retrieve server statistics.

To use this monitor, you must make sure that a version prior to WebLogic 6.0 is installed on your server, and that the SNMP agent is installed and activated on the server. For instructions on installing the SNMP agent, see http://www.weblogic.com/docs51/admindocs/snmpagent.html.

To obtain data for this graph, you need to enable the WebLogic (SNMP) online monitor (from the Controller or Tuning Console) and select the default measurements you want to display, before running the scenario or session step.

Note: To monitor a WebLogic (SNMP) server, use port 161 or 162, depending on the configuration of the agent.

Adding a Monitored Machine to the Controller

1. Click the WebLogic (SNMP) graph in the graph tree, and drag it into the right pane of the Run view.
2. Right-click the graph and select **Add Measurements**, or choose **Monitors > Add Measurements**. The WebLogic (SNMP) dialog box opens.
3. In the **Monitored Server Machines** section, click **Add.** The Add Machine dialog box opens.
4. Enter the server name or IP address of the machine you want to monitor. Select the platform on which the machine runs, and click **OK**.

Note: If the WebLogic SNMP agent is running on a different port than the default SNMP port, you must define the port number. Enter the following information in the Add Machine dialog box:
<server name:port number>
For example: digi:8888

In addition, you can define the default port for your WebLogic server in the configuration file, **snmp.cfg**, located in **<LoadRunner root folder>\dat\monitors**. For example, if the port used by the SNMP agent on your WebLogic server is 8888, you should edit the **snmp.cfg** file as follows:
; WebLogic
[cm_snmp_mon_isp]
port=8888

---

5 In the **Resource Measurements** section of the WebLogic (SNMP) dialog box, click **Add**.

Continue with Configuring the WebLogic (SNMP) Monitor below.

Adding a Monitored Machine to the Tuning Console

1 Click **Monitors** to open the Monitors Configuration dialog box.

2 Select the server whose monitors you want to configure from the Server list box.

3 Click **Add Monitor** and select **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the Application Server category, select **WebLogic**, and then click **Add**.

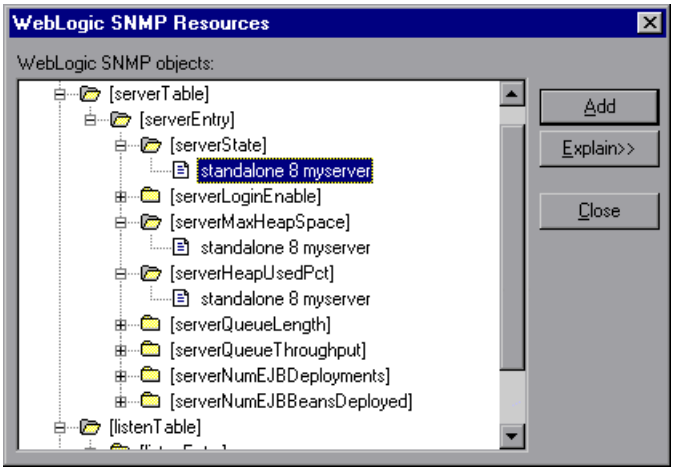
Note: If the WebLogic SNMP agent is running on a different port than the default SNMP port, you must define the port number. You can define the default port for your WebLogic server in the configuration file, **snmp.cfg**, located in **<LoadRunner root folder>\dat\monitors**. For example, if the port used by the SNMP agent on your WebLogic server is 8888, you should edit the **snmp.cfg** file as follows:

```
; WebLogic
[cm_snmp_mon_isp]
port=8888
```

Continue with Configuring the WebLogic (SNMP) Monitor below.

Configuring the WebLogic (SNMP) Monitor

1 The WebLogic SNMP Resources dialog box opens, displaying the available measurements.



2 Browse the WebLogic SNMP Objects tree, and select performance counters, as described in "Understanding the WebLogic (SNMP) Dialog Box" on page 199.

For a description of the available measurements, see "WebLogic (SNMP) Performance Counters" on page 200.

Note: The WebLogic (SNMP) monitor can only monitor up to 25 measurements.

---

3. After selecting and adding the required objects, click **Close**.
4. In the Controller, click **OK** in the WebLogic (SNMP) dialog box to activate the monitor.
5. In the Tuning Console, click **OK** in the Select Measurements to Monitor dialog box to activate the monitor.

Understanding the WebLogic (SNMP) Dialog Box

The WebLogic (SNMP) dialog box lets you select the items to monitor on the WebLogic (SNMP) application server. To use this monitor, you must make sure that a version prior to WebLogic 6.0 is installed on your server, and that the SNMP agent is installed and activated on the server. For instructions on installing the SNMP agent, see http://www.weblogic.com/docs51/admindocs/snmpagent.html.

WebLogic SNMP Objects: Select each required object and click **Add** (you can select only one object at a time). Click **Explain** for a description of the selected object.

Explain: Displays a description of the selected object.

Note: The WebLogic (SNMP) monitor can only monitor up to 25 measurements.

WebLogic (SNMP) Performance Counters

The following tables describe the measurements and server properties that can be monitored:

Server Table

The Server Table lists all WebLogic (SNMP) servers that are being monitored by the agent. A server must be contacted or be reported as a member of a cluster at least once before it will appear in this table. Servers are only reported as a member of a cluster when they are actively participating in the cluster, or shortly thereafter.

Measurement	Description
ServerState	The state of the WebLogic server, as inferred by the SNMP agent. **Up** implies that the agent can contact the server. **Down** implies that the agent cannot contact the server.
ServerLoginEnable	This value is true if client logins are enabled on the server.
ServerMaxHeapSpace	The maximum heap size for this server, in KB
ServerHeapUsedPct	The percentage of heap space currently in use on the server
ServerQueueLength	The current length of the server execute queue
ServerQueueThroughput	The current throughput of execute queue, expressed as the number of requests processed per second
ServerNumEJBDeployment	The total number of EJB deployment units known to the server
ServerNumEJBBeansDeployed	The total number of EJB beans actively deployed on the server

Listen Table

The Listen Table is the set of protocols, IP addresses, and port combinations on which servers are listening. There will be multiple entries for each server: one for each protocol, ipAddr, port combination. If clustering is used, the clustering-related MIB objects will assume a higher priority.

Measurement	Description
ListenPort	Port number.
ListenAdminOK	True if admin requests are allowed on this (protocol, ipAddr, port); otherwise false
ListenState	Listening if the (protocol, ipAddr, port) is enabled on the server; not Listening if it is not. The server may be listening but not accepting new clients if its server Login Enable state is false. In this case, existing clients will continue to function, but new ones will not.

ClassPath Table

The ClassPath Table is the table of classpath elements for Java, WebLogic (SNMP) servers, and servlets. There are multiple entries in this table for each server. There may also be multiple entries for each path on a server. If clustering is used, the clustering-related MIB objects will assume a higher priority.

Measurement	Description
CPType	The type of CP element: Java, WebLogic, servlet. A Java CPType means the cpElement is one of the elements in the normal Java classpath. A WebLogic CPType means the cpElement is one of the elements in weblogic.class.path. A servlet CPType means the cpElement is one of the elements in the dynamic servlet classpath.
CPIndex	The position of an element within its path. The index starts at 1.

WebLogic (JMX) Monitoring

The WebLogic (JMX) monitor displays statistics about the resource usage on the BEA WebLogic (JMX) server during the scenario or session step run. The BEA WebLogic (JMX) monitor uses the Java JMX interface to access run-time MBeans on the server. An MBean is a container that holds the performance data.

To use this monitor, you must set ensure permissions for a user to be able to monitor MBeans and ensure that Java 1.3 or later is installed on the Controller or Tuning Console machine. In addition, if the servlet is disabled, or if you do not want to use the servlet, you can load classes directly from the file system.

To obtain data for this graph, you need to enable the WebLogic (JMX) online monitor (from the Controller or Tuning Console) and select the default measurements you want to display, before running the scenario or session step.

Note: To use the WebLogic (JMX) monitor, you must make sure that WebLogic 6.0 or above is installed on your server.

The WebLogic (JMX) monitor utilizes a built-in server called the ClasspathServlet to load classes directly and automatically from the server. For information about loading classes from the server, see the "Loading Classes from the Server" on page 205.

Setting up the Monitoring Environment

Before using the WebLogic (JMX) monitor, you need to set up the WebLogic (JMX) monitoring environment.

- Install Java on the Controller or Tuning Console machine. You must install Java 1.3 or later to work with Weblogic 6.x or 7.x, and Java 1.4 to work with Weblogic 8.1.
- Set Permissions for user to be able to monitor MBeans

➤ Ensure that classes load directly from the file system if the servlet is disabled or not used

Installing Java 1.3.x

1. Install Java 1.3 or later on the Controller or Tuning Console machine. If Java 1.3 or later is already installed, but is not the default Java version being used, specify the full path to the updated version.
2. Specify the path in the **<LoadRunner root folder>\dat\monitors\WebLogicMon.ini** file. Edit the JVM entry in the [WebLogicMon] section. For example:

   ```
   JVM="E:\Program Files\JavaSoft\JRE\1.3.1\bin\javaw.exe
   ```

Installing Java 1.4

1. Install JDK 1.4 on the Controller or Tuning Console machine. If Java 1.4 is already installed, but is not the default Java version being used, specify the full path to the updated version in the command line.
2. Copy **weblogic.jar** from the **<WebLogic Server installation>\..\lib** to **<LoadRunner root folder>\classes**.
3. Remove **jmxri.jar** from **<LoadRunner root folder>\classes**.
4. Specify the path in the **<LoadRunner root folder>\dat\monitors\WebLogicMon.ini** file. Edit the JVM entry in the [WebLogicMon] section. For example:

   ```
   JVM="E:\Program Files\JavaSoft\JRE\1.4\bin\javaw.exe
   ```

5. Edit the JavaVersion entry in the [WebLogicMon] section.

Setting Permissions for Monitoring

You must set certain permissions for a user to be able to monitor MBeans.

To log on to the WebLogic (JMX) server:

Enter the username and password of a user with administrative privileges.

To set permissions for WebLogic version 6.x:

1. Open the WebLogic console (http://<host:port>/console).
2. In the tree on the left, select **Security** > **ACLs**.

   If you are working with the WebLogic 6.1 console, click **Create a new ACL...** in the screen on the right.
3. In the New ACL Name box, type weblogic.admin.mbean, and click **Create**.

   If you are working with the WebLogic 6.1 console, click **Add a new Permission...** in the screen on the right.
4. In the New Permission box (or Permission box, in the WebLogic 6.1 console), type access. In the WebLogic 6.0 console, click **Create**.
5. In the Users box and Groups box, enter the name of any user or group you want to use for monitoring.
6. Click **Grant Permission** in the WebLogic 6.0 console. In the WebLogic 6.1 console, click **Apply**.

To set permissions for WebLogic versions 7.x and 8.x:

1. Open the WebLogic console (http://<host:port>/console).
2. In the tree on the left, select **Security** > **Realms** > **myrealm** > **Users**, and click **Configure a new User...** in the screen on the right. The Create User: General tab opens.
3. In the Name box, type weblogic.admin.mbean, enter a password, confirm the password, and then click **Apply**.
4. In the Groups tab, enter the name of any user or group you want to use for monitoring, and then click **Apply**.

Loading Classes from the Server

The WebLogic (JMX) monitor utilizes a built-in server called the ClasspathServlet to load classes directly and automatically from the server. The advantages of this are easy installation and version independence. The disadvantages are a slight decrease in performance when loading classes for the first time (due to the size of the servlet), and the possibility of the servlet becoming disabled.

If the servlet is disabled, or if you do not want to use the servlet, you can load classes directly from the file system.

Note: For Weblogic version 8.1, you must load classes directly from the file system, as the WebLogic (JMX) monitor will not load the classes automatically.

To load classes directly from the file system:

1. Copy the **weblogic.jar** file from the application server install folder (under the lib folder) to **<LoadRunner root folder>\classes**.
2. If the classes file is not located in the default **<LoadRunner root folder>** folder, you need to specify the full path to it in the **<LoadRunner root folder>\dat\monitors\WebLogicMon.ini** file. In this file, change the line Weblogic=weblogic.jar to Weblogic=<full path to weblogic.jar>.

   You select measurements to monitor the WebLogic (JMX) application server using the BEA WebLogic Monitor Configuration dialog box.

Adding a Monitored Machine to the Controller

1. Click the WebLogic (JMX) graph in the graph tree, and drag it into the right pane of the Run view.
2. Right-click the graph and select **Add Measurements**, or choose **Monitors > Add Measurements**. The Add Machine dialog box opens.
3. In the **Monitored Server Machines** section, click **Add.** The WebLogic (JMX) dialog box opens.

4 Enter the server name or IP address of the machine you want to monitor according to the format: <server name>:<port number>.

For example: mercury:8111

Select the platform on which the machine runs, and click **OK**.

5 In the **Resource Measurements** of the WebLogic (JMX) dialog box, click **Add**.

Continue with Configuring the WebLogic (JMX) Monitor below.

Adding a Monitored Machine to the Tuning Console

1 Click **Monitors** to open the Monitors Configuration dialog box.

2 Select the server whose monitors you want to configure from the Server list box.

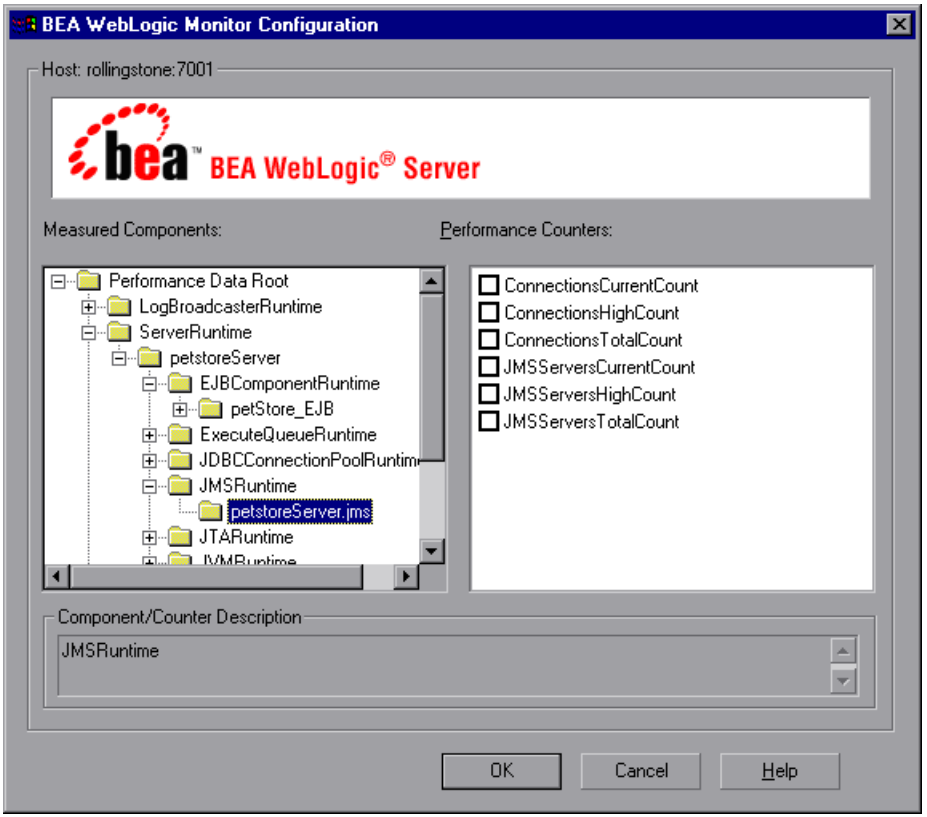
3 Click **Add Monitor** and select **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the Application Server category, select **WebLogic (JMX)**, and then click **Add**.

Continue with Configuring the WebLogic (JMX) Monitor below.

Configuring the WebLogic (JMX) Monitor

1 In the Enter Login Information dialog box, enter the username and password of a user with administrative privileges to the WebLogic server, and then click **OK**. The BEA WebLogic Monitor Configuration dialog box opens, displaying the available measurements.



For details on creating user permissions, see "Setting Permissions for Monitoring" on page 204.

2 Browse the Measured Components tree, and select performance counters, as described in "Understanding the WebLogic (JMX) Dialog Box" on page 208.

For a description of the available measurements, see "BEA WebLogic (JMX) Performance Counters" on page 208.

3 In the Controller, click **OK** in the BEA WebLogic Monitor Configuration dialog box, and in the WebLogic (JMX) dialog box, to activate the monitor.

4 In the Tuning Console, click **OK** in the BEA WebLogic Monitor Configuration dialog box, and in the Select Measurements to Monitor dialog box, to activate the monitor.

Understanding the WebLogic (JMX) Dialog Box

The WebLogic (JMX) dialog box lets you select the items to monitor on the BEA WebLogic (JMX) application server.

Host: Displays the name of the monitored machine.

Measured Components: Displays the available components. Browse the tree and select the component you want to monitor. A description of the highlighted component appears in the Component/Counter Description box.

Performance Counters: Check the required performance counters. A description of the selected counter appears in the Component/Counter Description box.

Component/Counter Description: Displays a description of the selected component or counter.

BEA WebLogic (JMX) Performance Counters

The following measurements are available for the WebLogic (JMX) server:

LogBroadcasterRuntime

Measurement	Description
MessagesLogged	The number of total log messages generated by this instance of the WebLogic server.
Registered	Returns “false” if the MBean represented by this object has been unregistered.
CachingDisabled	Private property that disables caching in proxies.

ServerRuntime

For more information on the measurements contained in each of the following measurement categories, see Mercury's Load Testing Monitors Web site (http://www.mercuryinteractive.com/products/loadrunner/load_testing_monitors/bealogic.html).

- ServletRuntime
- WebAppComponentRuntime
- EJBStatefulHomeRuntime
- JTARuntime
- JVMRuntime
- EJBEntityHomeRuntime.
- DomainRuntime
- EJBComponentRuntime
- DomainLogHandlerRuntime
- JDBCConnectionPoolRuntime
- ExecuteQueueRuntime
- ClusterRuntime
- JMSRuntime
- TimeServiceRuntime
- EJBStatelessHomeRuntime
- WLECConnectionServiceRuntime

ServerSecurityRuntime

Measurement	Description
UnlockedUsersTotalCount	Returns the number of times a user has been unlocked on the server
InvalidLoginUsersHighCount	Returns the high-water number of users with outstanding invalid login attempts for the server
LoginAttemptsWhileLockedTotalCount	Returns the cumulative number of invalid logins attempted on the server while the user was locked
Registered	Returns “false” if the MBean represented by this object has been unregistered.
LockedUsersCurrentCount	Returns the number of currently locked users on the server
CachingDisabled	Private property that disables caching in proxies.
InvalidLoginAttemptsTotalCount	Returns the cumulative number of invalid logins attempted on the server
UserLockoutTotalCount	Returns the cumulative number of user lockouts done on the server

WebSphere Monitoring

The WebSphere monitor displays statistics about the resource usage on the WebSphere 3.x, 4.x, or 5.x servers during the scenario or session step run.

To obtain data for this graph, you need to enable the WebSphere online monitor (from the Controller or Tuning Console) and select the default measurements you want to display before running the scenario or session step.

Setting up the Monitoring Environment

Before you can monitor the WebSphere server, you must first deploy the performance servlet. You may also choose to enable EPM counters on the application server.

Deploying the Performance Servlet on the Application Server

To monitor WebSphere version 5.x, you need to deploy the performance servlet on the application server using the IBM WebSphere "Installing a New Application" wizard.

To deploy the performance servlet on the application server for WebSphere 5.x:

1. From the administrative console, click **Applications** > **Install New Application** in the console navigation tree.
2. For Path, specify the full path name of the source application file (**PerfServletApp.ear**) on the server machine and click **Next**.
3. Select the **Generate Default Bindings** check box and click **Next**.
4. On the Install New Application page, click **Summary**, and select the **Cell/Node/Server** option. Click **Click here**.
5. On the **Map modules to application servers** panel, select the server onto which the application files will install from the **Clusters and Servers** list, and select **Module** to select all of the application modules.

6 Click **Next**, and in the Summary panel click **Finish**.

7 Verify that the servlet is running properly and that the performance data is being generated. A simple way to verify that the performance data is accessible is to display it in a Web browser. The URL must be in the following format:

http://<server name:port number>/<servlet_folder>/com.ibm.ivb.epm.servlet.PerformanceServlet

For example: http://websphere.mercury.co.il:81/servlet/com.ibm.ivb.epm.servlet.PerformanceServlet

---

Note: Only browsers that are XML-compatible will allow you to view the performance XML file.

Enabling the EPM Counters on the WebSphere 3.5.x Server

To work with WebSphere version 3.5.x

1 The EPM counters in 3.5.x are by default set to "none". To enable the counters, choose the application server you are monitoring in the WebSphere Administrator's Console browser.

2 Right-click the application server and select **Performance**. Select Performance Modules from the pop-up window.

3 Right-click Performance Modules to choose a performance level. Selecting various levels of counters enables the application server to manage varying levels of performance data.

4 Click the **Set** button.

5 In versions 3.5.2 and 3.5.3 the Servlet counters have been disabled. To enable the Servlet counters, you need to modify the contents of the com/ibm/servlet/appserver.properties file located in **<WAS_HOME>\lib\ibmwebas.jar**.

Extract the **jar** file and modify the appserver.properties as follows:

#listeners.application=com.ibm.servlet.engine.EPMApplicationListener
com.ibm.servlet.debug.OLTServletManager

```
listeners.application=
```

Should be:

```
listeners.application=com.ibm.servlet.engine.EPMApplicationListener
com.ibm.servlet.debug.OLTServletManager
#listeners.application=
```

6 Repackage the **jar** file.

Adding a Monitored Machine to the Controller

To configure the WebSphere or WebSphere 4.x-5.x monitor:

1 Click the WebSphere or WebSphere 4.x - 5.x graph in the graph tree, and drag it into the right pane of the Run view.

2 Right-click the graph and select **Add Measurements**, or choose **Monitors** > **Add Measurements**. The WebSphere dialog box opens.

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

4 Enter the server name or IP address of the machine you want to monitor. Select the platform on which the machine runs, and click **OK**.

5 In the **Resource Measurements** section of the WebSphere dialog box, click **Add**.

Continue with Configuring the WebSphere Monitor below.

Adding a Monitored Machine to the Tuning Console

1 Click **Monitors** to open the Monitors Configuration dialog box.

2 Select the server whose monitors you want to configure from the Server list box.

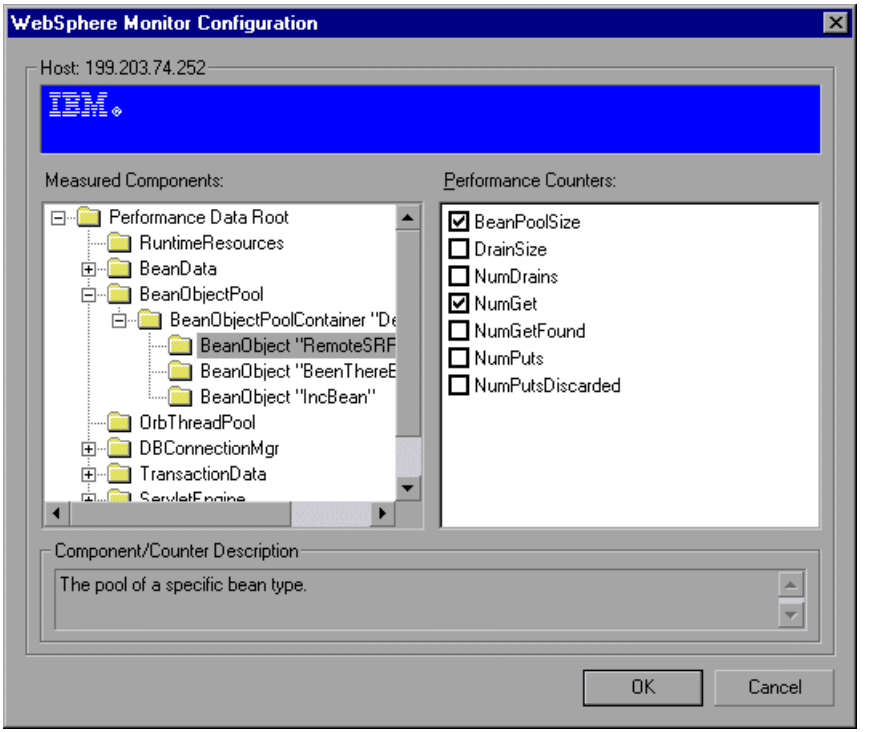
3 Click **Add Monitor** and select **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the Application Server category, select **WebSphere**, and then click **Add**.

Continue with Configuring the WebSphere Monitor below.

Configuring the WebSphere Monitor

1 The WebSphere Monitor Configuration dialog box opens, displaying the available measurements.



2 Browse the Measured Components tree, and select performance counters, as described in “Understanding the WebSphere Dialog Box” on page 215.

For a description of the available measurements, see “WebSphere Performance Counters” on page 217.

3 In the Controller, click **OK** in the WebSphere Monitor Configuration dialog box, and in the WebSphere dialog box, to activate the WebSphere monitor.

4 In the Tuning Console, click **OK** in the WebSphere Monitor Configuration dialog box, and in the Select Measurements to Monitor dialog box, to activate the WebSphere monitor.

Note: The port you use to monitor a WebSphere server through a firewall depends on the configuration of your server.

Understanding the WebSphere Dialog Box

The WebSphere dialog box lets you select the resources to monitor on the WebSphere 3.x, 4.x, or 5.x application server.

Host: Displays the name of the monitored machine.

Measured Components: Displays the available components. Browse the tree and select the component you want to monitor. A description of the highlighted component appears in the Component/Counter Description box.

Performance Counters: Check the required performance counters. A description of the selected counter appears in the Component/Counter Description box.

Component/Counter Description: Displays a description of the selected component or counter.

Specifying a Web Alias for the Servlet Directory

You can monitor as many application servers as you want by specifying a Web alias for the servlet directory.

To specify another Web alias for the servlet directory:

By default, LoadRunner uses the alias servlet as the servlet directory Web alias. For example, if the WebSphere Server machine is named mercury and the path for the servlets directory is: E:\AppServer\hosts\default_host\default_app\servlets, LoadRunner will request the XML file in the following URL: http:/mercury/servlet/com.ibm.ivb.epm.servlet.PerformanceServlet, where servlet is the default web alias for the servlet directory.

If the Web alias for the servlet directory is not servlet, you must specify the servlet directory Web alias in the Add Machine dialog box according to the following format:

http://<server name:port number>/<servlet_dir_alias>

For example: http://mercury/servlet2

Using this method, you can monitor as many application servers as you want—whether they are installed on the same machine, or on different machines.

To monitor other applications, in addition to the default application:

You can monitor as many applications as you want, regardless of whether they are installed on the same machine or different machines.

1. Copy the same files that you copied to the Servlets directory for the Default application to the Servlets directories for any other Web applications that you want to monitor.
2. Add the com.ibm.ivb.epm.servlet.PerformanceServlet to the configuration in the WebSphere Console for each Web application.
3. Add the Web application to be monitored to the WebSphere Performance Monitor using the following format:

   http://<server:port_number>/<servlet_dir_alias>/servlet

   For example: http://mercury/servlet3/servlet

WebSphere Performance Counters

The following tables describe the counters that can be monitored:

Run-Time Resources

Contains resources related to the Java Virtual Machine run-time, as well as the ORB.

Measurement	Description
MemoryFree	The amount of free memory remaining in the Java Virtual Machine
MemoryTotal	The total memory allocated for the Java Virtual Machine
MemoryUse	The total memory in use within the Java Virtual Machine

BeanData

Every home on the server provides performance data, depending upon the type of bean deployed in the home. The top level bean data holds an aggregate of all the containers.

Measurement	Description
BeanCreates	The number of beans created. Applies to an individual bean that is either 'stateful' or 'entity'
EntityBeanCreates	The number of entity beans created
BeanRemoves	The number of entity beans pertaining to a specific bean that have been removed. Applies to an individual bean that is either 'stateful' or 'entity'
EntityBeanRemoves	The number of entity beans removed
StatefulBeanCreates	The number of stateful beans created
StatefulBeanRemoves	The number of stateful bean removed

Measurement	Description
BeanPassivates	The number of bean passivates pertaining to a specific bean. Applies to an individual bean that is either 'stateful' or 'entity'.
EntityBeanPassivates	The number of entity bean passivates.
StatefulBeanPassivates	The number of stateful bean passivates.
BeanActivates	The number of bean activates pertaining to a specific bean. Applies to an individual bean that is either 'stateful' or 'entity'.
EntityBeanActivates	The number of entity bean activates.
StatefulBeanActivates	The number of stateful bean activates.
BeanLoads	The number of times the bean data was loaded. Applies to entity.
BeanStores	The number of times the bean data was stored in the database. Applies to entity.
BeanInstantiates	The number of times a bean object was created. This applies to an individual bean, regardless of its type.
StatelessBeanInstantiates	The number of times a stateless session bean object was created.
StatefulBeanInstantiates	The number of times a stateful session bean object was created.
EntityBeanInstantiates	The number of times an entity bean object was created.
BeanDestroys	The number of times an individual bean object was destroyed. This applies to any bean, regardless of its type.
StatelessBeanDestroys	The number of times a stateless session bean object was destroyed.
StatefulBeanDestroys	The number of times a stateful session bean object was destroyed.

Measurement	Description
EntityBeanDestroys	The number of times an entity bean object was destroyed.
BeansActive	The average number of instances of active beans pertaining to a specific bean. Applies to an individual bean that is either 'stateful' or 'entity'.
EntityBeansActive	The average number of active entity beans.
StatefulBeansActive	The average number of active session beans.
BeansLive	The average number of bean objects of this specific type that are instantiated but not yet destroyed. This applies to an individual bean, regardless of its type.
StatelessBeansLive	The average number of stateless session bean objects that are instantiated but not yet destroyed.
StatefulBeansLive	The average number of stateful session bean objects that are instantiated but not yet destroyed.
EntityBeansLive	The average number of entity bean objects that are instantiated but not yet destroyed.
BeanMethodRT	The average method response time for all methods defined in the remote interface to this bean. Applies to all beans.
BeanMethodActive	The average number of methods being processed concurrently. Applies to all beans.
BeanMethodCalls	The total number of method calls against this bean.

BeanObjectPool

The server holds a cache of bean objects. Each home has a cache and there is therefore one BeanObjectPoolContainer per container. The top level BeanObjectPool holds an aggregate of all the containers data.

Measurement	Description
BeanObjectPoolContainer	The pool of a specific bean type
BeanObject	The pool specific to a home
NumGet	The number of calls retrieving an object from the pool
NumGetFound	The number of calls to the pool that resulted in finding an available bean
NumPuts	The number of beans that were released to the pool
NumPutsDiscarded	The number of times releasing a bean to the pool resulted in the bean being discarded because the pool was full
NumDrains	The number of times the daemon found the pool was idle and attempted to clean it
DrainSize	The average number of beans discarded by the daemon during a clean
BeanPoolSize	The average number of beans in the pool

OrbThreadPool

These are resources related to the ORB thread pool that is on the server.

Measurement	Description
ActiveThreads	The average number of active threads in the pool
TotalThreads	The average number of threads in the pool
PercentTimeMaxed	The average percent of the time that the number of threads in the pool reached or exceeded the desired maximum number

Measurement	Description
ThreadCreates	The number of threads created
ThreadDestroys	The number of threads destroyed
ConfiguredMaxSize	The configured maximum number of pooled threads

DBConnectionMgr

These are resources related to the database connection manager. The manager consists of a series of data sources, as well as a top-level aggregate of each of the performance metrics.

Measurement	Description
DataSource	Resources related to a specific data source specified by the "name" attribute
ConnectionCreates	The number of connections created
ConnectionDestroys	The number of connections released
ConnectionPoolSize	The average size of the pool, i.e., number of connections
ConnectionAllocates	The number of times a connection was allocated
ConnectionWaiters	The average number of threads waiting for a connection
ConnectionWaitTime	The average time, in seconds, of a connection grant
ConnectionTime	The average time, in seconds, that a connection is in use
ConnectionPercentUsed	The average percentage of the pool that is in use
ConnectionPercentMaxed	The percentage of the time that all connections are in use

TransactionData

These are resources that pertain to transactions.

Measurement	Description
NumTransactions	The number of transactions processed
ActiveTransactions	The average number of active transactions
TransactionRT	The average duration of each transaction
BeanObjectCount	The average number of bean object pools involved in a transaction
RolledBack	The number of transactions rolled back
Commited	The number of transactions committed
LocalTransactions	The number of transactions that were local
TransactionMethodCount	The average number of methods invoked as part of each transaction
Timeouts	The number of transactions that timed out due to inactivity timeouts
TransactionSuspended	The average number of times that a transaction was suspended

ServletEngine

These are resources that are related to servlets and JSPs.

Measurement	Description
ServletsLoaded	The number of servlets currently loaded
ServletRequests	The number of requests serviced
CurrentRequests	The number of requests currently being serviced
ServletRT	The average response time for each request
ServletsActive	The average number of servlets actively processing requests

Measurement	Description
ServletIdle	The amount of time that the server has been idle (i.e., time since last request)
ServletErrors	The number of requests that resulted in an error or an exception
ServletBeanCalls	The number of bean method invocations that were made by the servlet
ServletBeanCreates	The number of bean references that were made by the servlet
ServletDBCalls	The number of database calls made by the servlet
ServletDBConAlloc	The number of database connections allocated by the servlet
SessionLoads	The number of times the servlet session data was read from the database
SessionStores	The number of times the servlet session data was stored in the database
SessionSize	The average size, in bytes, of a session data
LoadedSince	The time that has passed since the server was loaded (UNC time)

Sessions

These are general metrics regarding the HTTP session pool.

Measurement	Description
SessionsCreated	The number of sessions created on the server
SessionsActive	The number of currently active sessions
SessionsInvalidated	The number of invalidated sessions. May not be valid when using sessions in the database mode
SessionLifetime	Contains statistical data of sessions that have been invalidated. Does not include sessions that are still alive

WebSphere (EPM) Monitoring

The WebSphere (EPM) monitor displays statistics about the resource usage on the WebSphere 3.5.x server during the scenario or session step run.

To monitor the IBM WebSphere application server (3.5.x), you must first install the IBM WebSphere Administrator's Console on the Controller or Tuning Console machine. You may also need to copy the security keyring. Once you have installed the WebSphere Administrator's Console and enabled the EPM counters, you can configure the WebSphere (EPM) monitor before running the scenario or session step.

Setting up the Monitoring Environment

Before you can monitor the IBM WebSphere application server 3.5.x, you must install the IBM WebSphere Administrator's Console on the Controller or Tuning Console machine. You may also need to copy the security keyring and enable the EPM counters on the application server.

Installing the IBM WebSphere Administrator's Console

You must install the IBM WebSphere Administrator's Console on the Controller or Tuning Console machine.

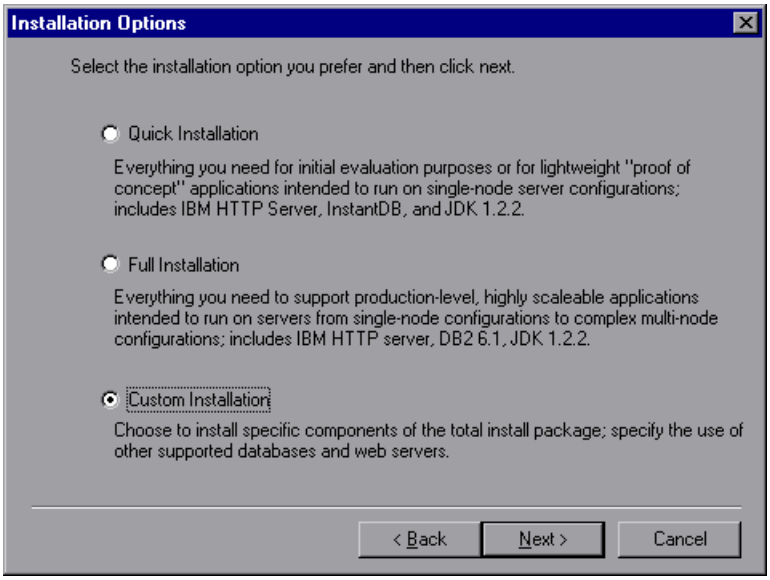
To install the IBM WebSphere Administrator's Console:

1 Start the WebSphere installation program from the WebSphere 3.5 Windows NT distribution CD-ROM. The WebSphere Application Server dialog box opens.

Disregard the instruction to shut down all Web servers that you plan to run with WebSphere. This is not relevant to the Administrator's Console installation.

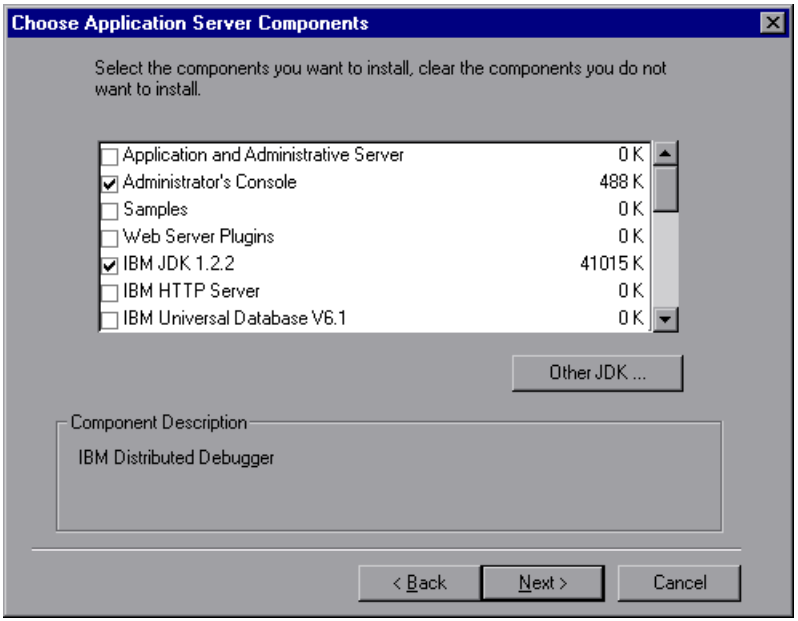
Click **Next**.

2 The Installation Options dialog box opens. Select **Custom Installation**.



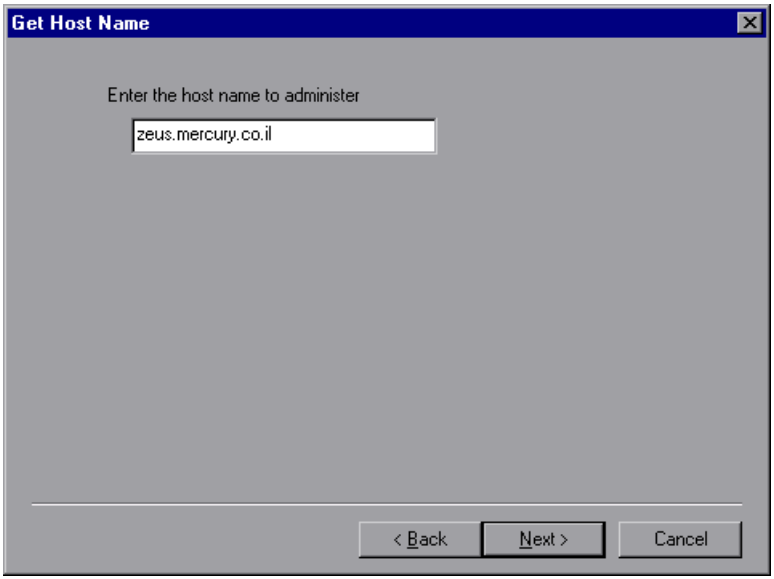
Click **Next**.

3 The Choose Application Server Components dialog box opens. Select **Administrator's Console** and **IBM JDK 1.2.2**. Clear all the other options.



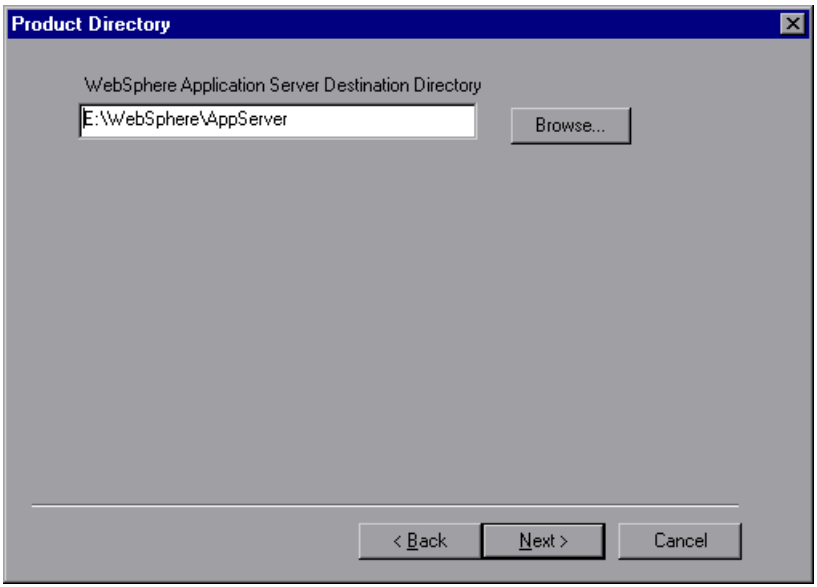
Click **Next**.

4 The Get Host Name dialog box opens. Type the name of the machine that you want to monitor.



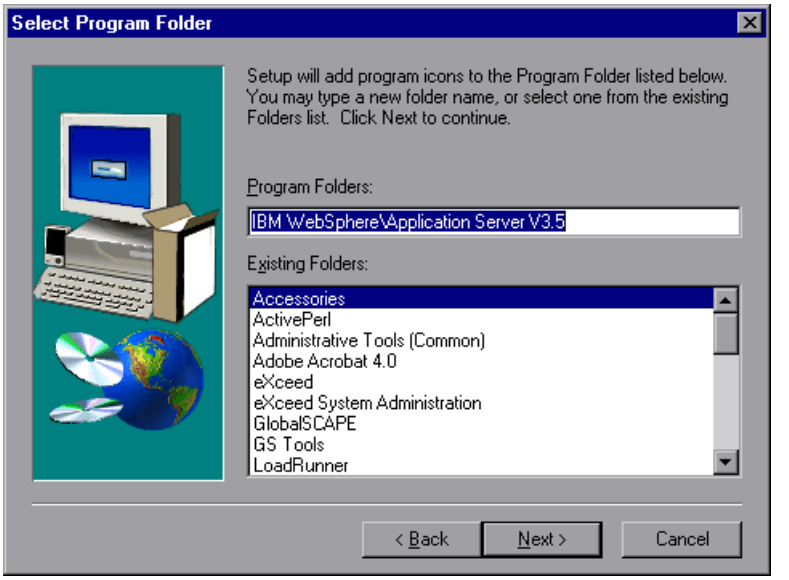
Click **Next**.

5 The Product Directory dialog box opens. Specify the folder in which to install the Administrator's Console. To select a different location, click **Browse**, choose a folder other than the default folder, and click **OK**.



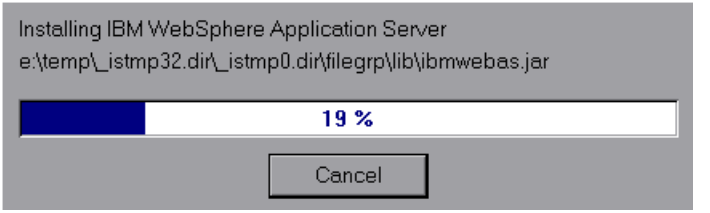
Click **Next**.

6 The Select Program Folder dialog box opens. Specify a program folder, or accept the default folder, IBM WebSphere\Application Server V3.5.



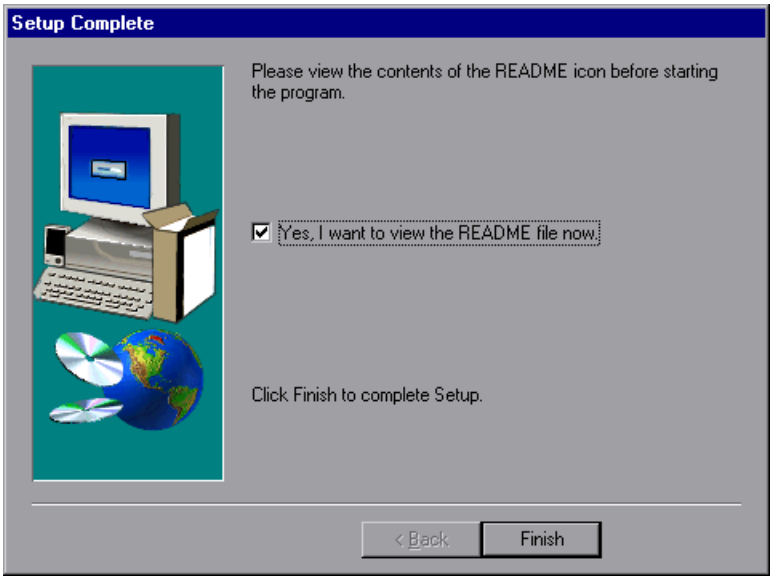
Click **Next**.

The installation process begins. To pause or quit the installation, click **Cancel**.

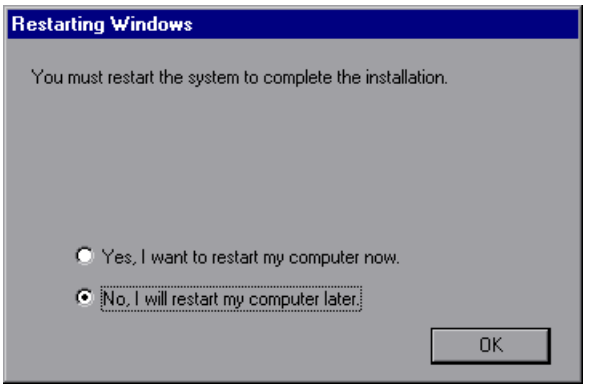


7 When the installation is complete, the Setup Complete dialog box opens. Select the check box to view the readme file before starting the program.

You can view the readme file at any time by selecting **Start** > **Programs** > **Application Server V3.5** > **IBM WebSphere** > **README**.



8 Click **Finish** to complete the installation program. The Restarting Windows dialog box opens.



9 Select either to restart your computer and complete the installation now (recommended) or to wait and complete the installation later.

10 Click **OK** to complete the installation of the Administrator's Console.

Copying the Security Keyring

If you enabled security on the WebSphere server, you must copy the security keyring from the server to the admin client. (One way to tell whether security is enabled is to see whether the Administrator's Console can connect to the admin server.) A keyring is a certification used by the server to identify the client.

You need to copy the **jar** file containing the keyring from the server lib folder to the client lib folder. You also need to add the **jar** file containing the keyring to the monitoring client command line.

Note: The keyring used in this file (**353Keyring.jar**) is the IBM dummy keyring that must be installed on servers using versions 3.52 and below. If your server is using the IBM dummy keyring and is version 3.52 or below, you do not need to change the line. If you are using the dummy keyring and are running version 3.53 or later, you do not need to do anything.

To copy the keyring:

1 Copy the keyring **jar** file from the server to the admin client lib folder (by default, **C:\Websphere\Appserver\lib**):

The **jar** file containing the keyring, **xxxKeyring.jar**, is located by default in the following location:

NT Server	C:\Websphere\Appserver\lib
UNIX Server	OPT/websphere/Appserver/lib

2 Open the **<LoadRunner root folder>\dat\monitors\WebSphere35Mon.ini** file in a text editor.

3 Locate the following line:
JVM_CLASSES4=C:\WebSphere\AppServer\lib\353Keyring.jar

Note: If you did not use the default location for the WebSphere installation, the line will be different.

---

4 Change **353Keyring.jar** to the keyring you are using.

Enabling EPM Counters on the WebSphere 3.5.x Server

To enable the EPM counters, which are by default set to "none," right-click the application you are monitoring in the WebSphere Administrator's Console browser, and select **Performance**. Expand the Performance Modules tree in the dialog box that opens. To manage different levels of performance data, right-click the performance modules and choose a performance level. Click the **Set** button.

Alternatively, ensure that the application server is started, select the **Advanced** tab in the WebSphere Administrator's Console browser, and in the EPM Specification box, type:
epm=high:epm.beanMethodData=none

Adding a Monitored Machine to the Controller

1 Click the WebSphere (EPM) graph in the graph tree, and drag it into the right pane of the Run view.

2 Right-click the graph and select **Add Measurements**, or choose **Monitors** > **Add Measurements**. The WebSphere (EPM) dialog box opens.

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

4 Enter the server name or IP address of the machine you want to monitor. Select the platform on which the machine runs, and click **OK**.

5 In the **Resource Measurements** section of the WebSphere (EPM) dialog box, click **Add**.

Continue with Configuring the WebSphere (EPM) Monitor below.

Adding a Monitored Machine to the Tuning Console

1 Click **Monitors** to open the Monitors Configuration dialog box.

2 Select the server whose monitors you want to configure from the Server list box.

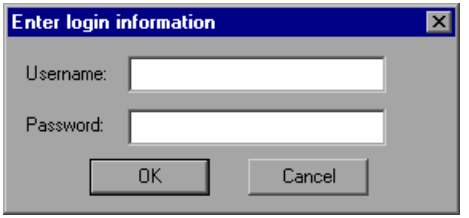
3 Click **Add Monitor** and select **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the Application Server category, select **WebSphere (EPM)**, and then click **Add**.

Continue with Configuring the WebSphere (EPM) Monitor below.

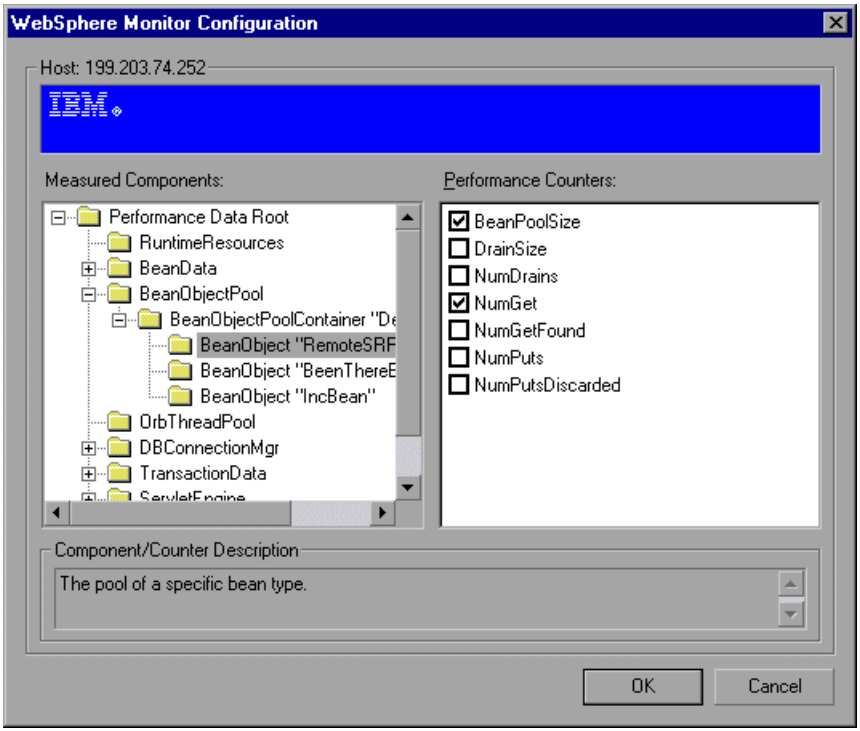
Configuring the WebSphere (EPM) Monitor

1 The WebSphere (EPM) Logon dialog box opens.



Enter your username and password and click **OK**.

2 The WebSphere Monitor Configuration dialog box opens, displaying the available measurements.



3 Browse the Measured Components tree, and select performance counters, as described in "Understanding the WebSphere (EPM) Dialog Box" on page 236, and then click **OK**.

For a description of the available measurements, see "WebSphere Performance Counters" on page 217.

4 In the Controller, click **OK** in the WebSphere Monitor Configuration dialog box, and in the WebSphere (EPM) dialog box, to activate the monitor.

5 In the Tuning Console, click **OK** in the WebSphere Monitor Configuration dialog box, and in the Select Measurements to Monitor dialog box, to activate the monitor.

Understanding the WebSphere (EPM) Dialog Box

The WebSphere (EPM) dialog box lets you select the resources to monitor on the WebSphere (EPM) application server (3.5.x).

Host: Displays the name of the monitored machine.

Measured Components: Displays the available components. Browse the tree and select the component you want to monitor. A description of the highlighted component appears in the Component/Counter Description box.

Performance Counters: Check the required performance counters. A description of the selected counter appears in the Component/Counter Description box.

Component/Counter Description: Displays a description of the selected component or counter.

9

Database Resource Monitoring

You monitor DB2, Oracle, SQL Server, or Sybase database resource usage during a scenario or session step run using LoadRunner's Database Server Resource monitors.

This chapter describes:

- About Database Resource Monitoring
- DB2 Monitoring
- Oracle Monitoring
- SQL Server Monitoring
- Sybase Monitoring

About Database Resource Monitoring

The DB2, Oracle, SQL Server, or Sybase database server resource monitors measure statistics for DB2, Oracle, SQL Server, or Sybase database servers. During a scenario or session step run, you use these monitors to isolate database server performance bottlenecks.

For each database server, you configure the measurements you want to monitor before running your scenario or session step. To run the DB2, Oracle, and Sybase monitors, you must also install the client libraries on the database server you want to monitor.

DB2 Monitoring

The DB2 monitor shows the resource usage on the DB2 database server machine.

To monitor the DB2 database server machine, you must first set up the DB2 monitor environment. You then enable the DB2 monitor (from the Controller or Tuning Console) by selecting the counters you want the monitor to measure. You select these counters using the DB2 Monitor Configuration dialog box.

Note: If there is no application working with a database, you can only monitor the database manager instance.

Setting Up the Monitoring Environment

Before monitoring a DB2 database server, you must set up the monitor environment.

To set up the DB2 monitor environment:

1. Install all the client files and libraries on the Controller or Tuning Console machine.
2. Select **Start** > **Programs** > **DB2 for Windows NT** > **Control Center**. Enter your DB2 server username and password (with administrative privileges).
3. In the console that opens, right-click **Systems**, and select **Add**.
4. Enter the following settings in the dialog box:

   System Name: <server name>

   Remote Instance: DB2

   Host Name: <server name>

   Service Name: the DB2 server port. The default value is 50000.

5 Click **Retrieve**, and then **OK**.

Note: If you receive an error message after clicking **Retrieve**, repeat steps 3 and 4, and click **OK**.

---

6 Expand the <server name> node in the console tree.

7 Right-click **Instance**, and select **Add**.

8 Enter the following settings in the dialog box:

Remote Instance: DB2

Instance Name: the database instance to be called from the Controller or Tuning Console

Host Name: <server name>

Service Name: <DB2 server port>. The default value is 50000.

9 Click **OK** and close the Control Center.

Note: You can only work with a single Database Manager instance during each monitoring session.

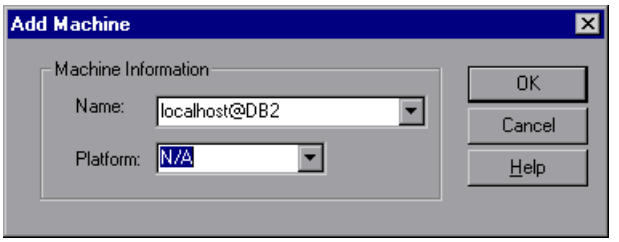
Adding a Monitored Machine to the Controller

1 Click the DB2 graph in the graph tree, and drag it into the right pane of the Run view.

2 Right-click the graph and select **Add Measurements**, or choose **Monitors** > **Add Measurements**. The DB2 dialog box opens.

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

4 Enter the DB2 server machine name followed by the @ sign and the database instance you specified in the DB2 Control Center. In the Platform box, select **N/A**.



Click **OK** to save the information you entered and close the dialog box.

5 In the **Resource Measurements** section of the DB2 dialog box, click **Add**.

Continue with Configuring the DB2 Monitor below.

Adding a Monitored Machine to the Tuning Console

1 Click **Monitors** to open the Monitors Configuration dialog box.

2 Select the server whose monitors you want to configure from the Server list box.

3 Click **Add Monitor** and select **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

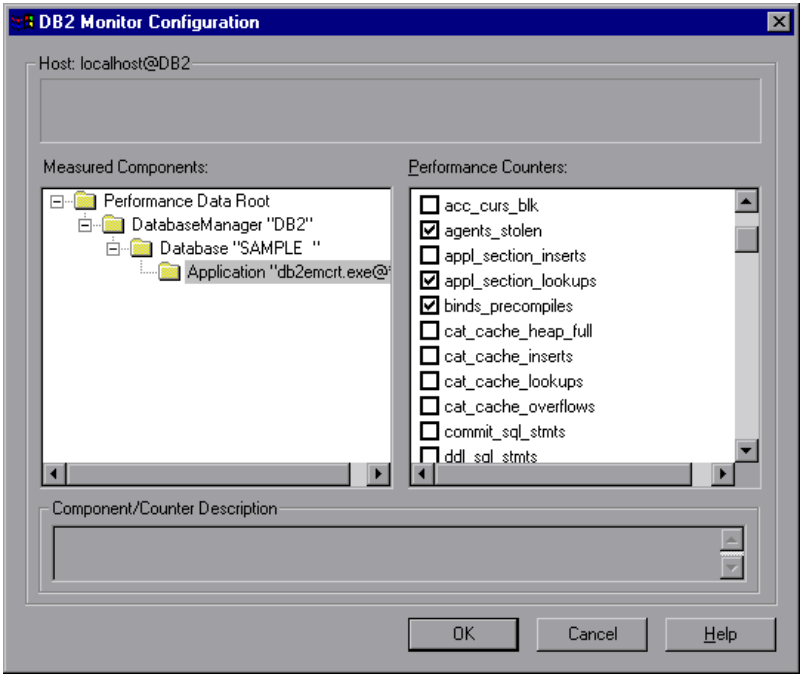
4 In the left section of the dialog box, expand the Database Server Resource category, select **DB2**, and then click **Add**.

Continue with Configuring the DB2 Monitor below.

Configuring the DB2 Monitor

1 In the dialog box that opens, enter your DB2 server username and password, and click **OK**.

The DB2 Monitor Configuration dialog box opens, displaying the available measurements.



2 Browse the Measured Components tree, and select performance counters, as described in "Understanding the DB2 Dialog Box" on page 242, and then click **OK**.

For a description of the available measurements, see "DB2 Performance Counters" on page 242.

3 Add all the desired resources to the list, and click **Close**.

4 In the Controller, click **OK** in the DB2 dialog box to activate the monitor.

5 In the Tuning Console, click **OK** in the Select Measurements to Monitor dialog box to activate the monitor.

Understanding the DB2 Dialog Box

The DB2 dialog box lets you select the measurements to monitor the resource usage on a DB2 database during a scenario or session step run.

Note: If there is no application working with a database, you can only monitor the database manager instance.

Host: Displays the name of the monitored machine.

Measured Components: Displays the available components. Browse the tree and select the component you want to monitor. A description of the highlighted component appears in the Component/Counter Description box.

Performance Counters: Check the required performance counters. A description of the selected counter appears in the Component/Counter Description box.

Component/Counter Description: Displays a description of the selected component or counter.

DB2 Performance Counters

The following tables describe the default counters that can be monitored.

DatabaseManager

Measurement	Description
rem_cons_in	The current number of connections initiated from remote clients to the instance of the database manager that is being monitored.
rem_cons_in_exec	The number of remote applications that are currently connected to a database and are currently processing a unit of work within the database manager instance being monitored.

Measurement	Description
local_cons	The number of local applications that are currently connected to a database within the database manager instance being monitored.
local_cons_in_exec	The number of local applications that are currently connected to a database within the database manager instance being monitored and are currently processing a unit of work.
con_local_dbases	The number of local databases that have applications connected.
agents_registered	The number of agents registered in the database manager instance that is being monitored (coordinator agents and subagents).
agents_waiting_on_token	The number of agents waiting for a token so they can execute a transaction in the database manager.
idle_agents	The number of agents in the agent pool that are currently unassigned to an application and are therefore "idle".
agents_from_pool	The number of agents assigned from the agent pool
agents_created_empty_pool	The number of agents created because the agent pool was empty.
agents_stolen	The number of times that agents are stolen from an application. Agents are stolen when an idle agent associated with an application is reassigned to work on a different application.
comm_private_mem	The amount of private memory that the instance of the database manager has currently committed at the time of the snapshot.
inactive_gw_agents	The number of DRDA agents in the DRDA connections pool that are primed with a connection to a DRDA database, but are inactive.

Measurement	Description
num_gw_conn_switches	The number of times that an agent from the agents pool was primed with a connection and was stolen for use with a different DRDA database.
sort_heap_allocated	The total number of allocated pages of sort heap space for all sorts at the level chosen and at the time the snapshot was taken.
post_threshold_sorts	The number of sorts that have requested heaps after the sort heap threshold has been reached.
piped_sorts_requested	The number of piped sorts that have been requested.
piped_sorts_accepted	The number of piped sorts that have been accepted.

Database

Measurement	Description
appls_cur_cons	Indicates the number of applications that are currently connected to the database.
appls_in_db2	Indicates the number of applications that are currently connected to the database, and for which the database manager is currently processing a request.
total_sec_cons	The number of connections made by a sub-agent to the database at the node.
num_assoc_agents	At the application level, this is the number of sub-agents associated with an application. At the database level, it is the number of sub-agents for all applications.
sort_heap_allocated	The total number of allocated pages of sort heap space for all sorts at the level chosen and at the time the snapshot was taken.

Measurement	Description
total_sorts	The total number of sorts that have been executed.
total_sort_time	The total elapsed time (in milliseconds) for all sorts that have been executed.
sort_overflows	The total number of sorts that ran out of sort heap and may have required disk space for temporary storage.
active_sorts	The number of sorts in the database that currently have a sort heap allocated.
total_hash_joins	The total number of hash joins executed.
total_hash_loops	The total number of times that a single partition of a hash join was larger than the available sort heap space.
hash_join_overflows	The number of times that hash join data exceeded the available sort heap space
hash_join_small_overflows	The number of times that hash join data exceeded the available sort heap space by less than 10%.
pool_data_l_reads	Indicates the number of logical read requests for data pages that have gone through the buffer pool.
pool_data_p_reads	The number of read requests that required I/O to get data pages into the buffer pool.
pool_data_writes	Indicates the number of times a buffer pool data page was physically written to disk.
pool_index_l_reads	Indicates the number of logical read requests for index pages that have gone through the buffer pool.
pool_index_p_reads	Indicates the number of physical read requests to get index pages into the buffer pool.
pool_index_writes	Indicates the number of times a buffer pool index page was physically written to disk.

Measurement	Description
pool_read_time	Provides the total amount of elapsed time spent processing read requests that caused data or index pages to be physically read from disk to buffer pool.
pool_write_time	Provides the total amount of time spent physically writing data or index pages from the buffer pool to disk.
files_closed	The total number of database files closed.
pool_async_data_reads	The number of pages read asynchronously into the buffer pool.
pool_async_data_writes	The number of times a buffer pool data page was physically written to disk by either an asynchronous page cleaner, or a pre-fetcher. A pre-fetcher may have written dirty pages to disk to make space for the pages being pre-fetched.
pool_async_index_writes	The number of times a buffer pool index page was physically written to disk by either an asynchronous page cleaner, or a pre-fetcher. A pre-fetcher may have written dirty pages to disk to make space for the pages being pre-fetched.
pool_async_index_reads	The number of index pages read asynchronously into the buffer pool by a pre-fetcher.
pool_async_read_time	The total elapsed time spent reading by database manager pre-fetchers.
pool_async_write_time	The total elapsed time spent writing data or index pages from the buffer pool to disk by database manager page cleaners.
pool_async_data_read_reqs	The number of asynchronous read requests.
pool_lsn_gap_clns	The number of times a page cleaner was invoked because the logging space used had reached a pre-defined criterion for the database.

Measurement	Description
pool_drty_pg_steal_clns	The number of times a page cleaner was invoked because a synchronous write was needed during the victim buffer replacement for the database.
pool_drty_pg_thrsh_clns	The number of times a page cleaner was invoked because a buffer pool had reached the dirty page threshold criterion for the database.
prefetch_wait_time	The time an application spent waiting for an I/O server (pre-fetcher) to finish loading pages into the buffer pool.
pool_data_to_estore	The number of buffer pool data pages copied to extended storage.
pool_index_to_estore	The number of buffer pool index pages copied to extended storage.
pool_data_from_estore	The number of buffer pool data pages copied from extended storage.
pool_index_from_estore	The number of buffer pool index pages copied from extended storage.
direct_reads	The number of read operations that do not use the buffer pool.
direct_writes	The number of write operations that do not use the buffer pool.
direct_read_reqs	The number of requests to perform a direct read of one or more sectors of data.
direct_write_reqs	The number of requests to perform a direct write of one or more sectors of data.
direct_read_time	The elapsed time (in milliseconds) required to perform the direct reads.
direct_write_time	The elapsed time (in milliseconds) required to perform the direct writes.
cat_cache_lookups	The number of times that the catalog cache was referenced to obtain table descriptor information.

Measurement	Description
cat_cache_inserts	The number of times that the system tried to insert table descriptor information into the catalog cache.
cat_cache_overflows	The number of times that an insert into the catalog cache failed due the catalog cache being full.
cat_cache_heap_full	The number of times that an insert into the catalog cache failed due to a heap-full condition in the database heap.
pkg_cache_lookups	The number of times that an application looked for a section or package in the package cache. At a database level, it indicates the overall number of references since the database was started, or monitor data was reset.
pkg_cache_inserts	The total number of times that a requested section was not available for use and had to be loaded into the package cache. This count includes any implicit prepares performed by the system.
pkg_cache_num_overflows	The number of times that the package cache overflowed the bounds of its allocated memory.
appl_section_lookups	Lookups of SQL sections by an application from its SQL work area.
appl_section_inserts	Inserts of SQL sections by an application from its SQL work area.
sec_logs_allocated	The total number of secondary log files that are currently being used for the database.
log_reads	The number of log pages read from disk by the logger.
log_writes	The number of log pages written to disk by the logger.
total_log_used	The total amount of active log space currently used (in bytes) in the database.

Measurement	Description
locks_held	The number of locks currently held.
lock_list_in_use	The total amount of lock list memory (in bytes) that is in use.
deadlocks	The total number of deadlocks that have occurred.
lock_escals	The number of times that locks have been escalated from several row locks to a table lock.
x_lock_escals	The number of times that locks have been escalated from several row locks to one exclusive table lock, or the number of times an exclusive lock on a row caused the table lock to become an exclusive lock.
lock_timeouts	The number of times that a request to lock an object timed-out instead of being granted.
lock_waits	The total number of times that applications or connections waited for locks.
lock_wait_time	The total elapsed time waited for a lock.
locks_waiting	Indicates the number of agents waiting on a lock.
rows_deleted	The number of row deletions attempted.
rows_inserted	The number of row insertions attempted.
rows_updated	The number of row updates attempted.
rows_selected	The number of rows that have been selected and returned to the application.
int_rows_deleted	The number of rows deleted from the database as a result of internal activity.
int_rows_updated	The number of rows updated from the database as a result of internal activity.
int_rows_inserted	The number of rows inserted into the database as a result of internal activity caused by triggers.

Measurement	Description
static_sql_stmts	The number of static SQL statements that were attempted.
dynamic_sql_stmts	The number of dynamic SQL statements that were attempted.
failed_sql_stmts	The number of SQL statements that were attempted, but failed.
commit_sql_stmts	The total number of SQL COMMIT statements that have been attempted.
rollback_sql_stmts	The total number of SQL ROLLBACK statements that have been attempted.
select_sql_stmts	The number of SQL SELECT statements that were executed.
uid_sql_stmts	The number of SQL UPDATE, INSERT, and DELETE statements that were executed.
ddl_sql_stmts	This element indicates the number of SQL Data Definition Language (DDL) statements that were executed.
int_auto_rebinds	The number of automatic rebinds (or recompiles) that have been attempted.
int_commits	The total number of commits initiated internally by the database manager.
int_rollbacks	The total number of rollbacks initiated internally by the database manager.
int_deadlock_rollbacks	The total number of forced rollbacks initiated by the database manager due to a deadlock. A rollback is performed on the current unit of work in an application selected by the database manager to resolve the deadlock.
binds_precompiles	The number of binds and pre-compiles attempted.

Application

Measurement	Description
agents_stolen	The number of times that agents are stolen from an application. Agents are stolen when an idle agent associated with an application is reassigned to work on a different application.
num_assoc_agents	At the application level, this is the number of sub-agents associated with an application. At the database level, it is the number of sub-agents for all applications.
total_sorts	The total number of sorts that have been executed.
total_sort_time	The total elapsed time (in milliseconds) for all sorts that have been executed.
sort_overflows	The total number of sorts that ran out of sort heap and may have required disk space for temporary storage.
total_hash_joins	The total number of hash joins executed.
total_hash_loops	The total number of times that a single partition of a hash join was larger than the available sort heap space.
hash_join_overflows	The number of times that hash join data exceeded the available sort heap space
hash_join_small_overflows	The number of times that hash join data exceeded the available sort heap space by less than 10%.
pool_data_l_reads	Indicates the number of logical read requests for data pages that have gone through the buffer pool.
pool_data_p_reads	The number of read requests that required I/O to get data pages into the buffer pool.
pool_data_writes	Indicates the number of times a buffer pool data page was physically written to disk.

Measurement	Description
pool_index_l_reads	Indicates the number of logical read requests for index pages that have gone through the buffer pool.
pool_index_p_reads	Indicates the number of physical read requests to get index pages into the buffer pool.
pool_index_writes	Indicates the number of times a buffer pool index page was physically written to disk.
pool_read_time	Provides the total amount of elapsed time spent processing read requests that caused data or index pages to be physically read from disk to buffer pool.
prefetch_wait_time	The time an application spent waiting for an I/O server (pre-fetcher) to finish loading pages into the buffer pool.
pool_data_to_estore	The number of buffer pool data pages copied to extended storage.
pool_index_to_estore	The number of buffer pool index pages copied to extended storage.
pool_data_from_estore	The number of buffer pool data pages copied from extended storage.
pool_index_from_estore	The number of buffer pool index pages copied from extended storage.
direct_reads	The number of read operations that do not use the buffer pool.
direct_writes	The number of write operations that do not use the buffer pool.
direct_read_reqs	The number of requests to perform a direct read of one or more sectors of data.
direct_write_reqs	The number of requests to perform a direct write of one or more sectors of data.
direct_read_time	The elapsed time (in milliseconds) required to perform the direct reads.

Measurement	Description
direct_write_time	The elapsed time (in milliseconds) required to perform the direct writes.
cat_cache_lookups	The number of times that the catalog cache was referenced to obtain table descriptor information.
cat_cache_inserts	The number of times that the system tried to insert table descriptor information into the catalog cache.
cat_cache_overflows	The number of times that an insert into the catalog cache failed due the catalog cache being full.
cat_cache_heap_full	The number of times that an insert into the catalog cache failed due to a heap-full condition in the database heap.
pkg_cache_lookups	The number of times that an application looked for a section or package in the package cache. At a database level, it indicates the overall number of references since the database was started, or monitor data was reset.
pkg_cache_inserts	The total number of times that a requested section was not available for use and had to be loaded into the package cache. This count includes any implicit prepares performed by the system.
appl_section_lookups	Lookups of SQL sections by an application from its SQL work area.
appl_section_inserts	Inserts of SQL sections by an application from its SQL work area.
uow_log_space_used	The amount of log space (in bytes) used in the current unit of work of the monitored application.
locks_held	The number of locks currently held.
deadlocks	The total number of deadlocks that have occurred.

Measurement	Description
lock_escals	The number of times that locks have been escalated from several row locks to a table lock.
x_lock_escals	The number of times that locks have been escalated from several row locks to one exclusive table lock, or the number of times an exclusive lock on a row caused the table lock to become an exclusive lock.
lock_timeouts	The number of times that a request to lock an object timed-out instead of being granted.
lock_waits	The total number of times that applications or connections waited for locks.
lock_wait_time	The total elapsed time waited for a lock.
locks_waiting	Indicates the number of agents waiting on a lock.
uow_lock_wait_time	The total amount of elapsed time this unit of work has spent waiting for locks.
rows_deleted	The number of row deletions attempted.
rows_inserted	The number of row insertions attempted.
rows_updated	The number of row updates attempted.
rows_selected	The number of rows that have been selected and returned to the application.
rows_written	The number of rows changed (inserted, deleted or updated) in the table.
rows_read	The number of rows read from the table.
int_rows_deleted	The number of rows deleted from the database as a result of internal activity.
int_rows_updated	The number of rows updated from the database as a result of internal activity.
int_rows_inserted	The number of rows inserted into the database as a result of internal activity caused by triggers.

Measurement	Description
open_rem_curs	The number of remote cursors currently open for this application, including those cursors counted by 'open_rem_curs_blk'.
open_rem_curs_blk	The number of remote blocking cursors currently open for this application.
rej_curs_blk	The number of times that a request for an I/O block at server was rejected and the request was converted to non-blocked I/O.
acc_curs_blk	The number of times that a request for an I/O block was accepted.
open_loc_curs	The number of local cursors currently open for this application, including those cursors counted by 'open_loc_curs_blk'.
open_loc_curs_blk	The number of local blocking cursors currently open for this application.
static_sql_stmts	The number of static SQL statements that were attempted.
dynamic_sql_stmts	The number of dynamic SQL statements that were attempted.
failed_sql_stmts	The number of SQL statements that were attempted, but failed.
commit_sql_stmts	The total number of SQL COMMIT statements that have been attempted.
rollback_sql_stmts	The total number of SQL ROLLBACK statements that have been attempted.
select_sql_stmts	The number of SQL SELECT statements that were executed.
uid_sql_stmts	The number of SQL UPDATE, INSERT, and DELETE statements that were executed.
ddl_sql_stmts	This element indicates the number of SQL Data Definition Language (DDL) statements that were executed.

Measurement	Description
int_auto_rebinds	The number of automatic rebinds (or recompiles) that have been attempted.
int_commits	The total number of commits initiated internally by the database manager.
int_rollbacks	The total number of rollbacks initiated internally by the database manager.
int_deadlock_rollbacks	The total number of forced rollbacks initiated by the database manager due to a deadlock. A rollback is performed on the current unit of work in an application selected by the database manager to resolve the deadlock.
binds_precompiles	The number of binds and pre-compiles attempted.

Oracle Monitoring

The Oracle monitor displays information from Oracle V$ tables: Session statistics, V$SESSTAT, system statistics, V$SYSSTAT, and other table counters defined by the user in the custom query.

To obtain data for this graph, you must first set up the monitoring environment. You then enable the Oracle online monitor (from the Controller or Tuning Console) and select the default measurements you want to display, before running the scenario or session step.

Note: The port you use to monitor an Oracle server through a firewall depends on the configuration of the Oracle server. Configuration information for the connection between the client and server is located in the Oracle client **tnsnames.ora** file.

Setting Up the Monitoring Environment

The Oracle server measures information from the V$SESSTAT and V$SYSSTAT Oracle V$ tables, and other table counters defined by the user in the custom query. In order to monitor the Oracle server, you must set up the monitoring environment as described below before you can configure the monitor.

To set up the Oracle monitor environment:

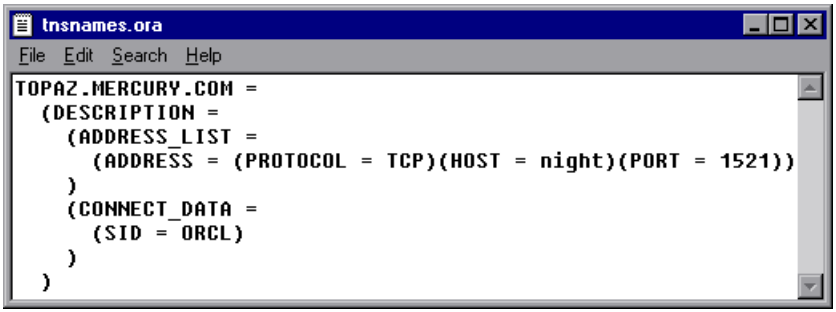
1. Ensure that the Oracle client libraries are installed on the Controller or Tuning Console machine.
2. Verify that **%OracleHome%\bin** is included in the path environment variable. If it is not, add it.
3. Configure the **tnsnames.ora** file on the Controller or Tuning Console machine so that the Oracle client can communicate with the Oracle server(s) you plan to monitor.

   You can configure connection parameters either manually, by editing the **tnsnames.ora** file in a text editor, or using the Oracle service configuration tool (for example, select **Start** > **Programs** > **Oracle for Windows NT** > **Oracle Net8 Easy Config**).

   You specify:

   - a new service name (TNS name) for the Oracle instance
   - TCP protocol
   - the host name (name of monitored server machine)
   - the port number (usually 1521)
   - the database SID (the default SID is ORCL)

For example:



Note: Only the 32-bit Oracle client should be installed on the Controller or Tuning Console machine running the Oracle monitor. If you have a 16-bit and a 32-bit Oracle client installation on the controller machine, the 16-bit installation should be uninstalled.

---

4. Obtain a username and password for the service from your database administrator, and ensure that the Controller or Tuning Console has database administrator privileges for the Oracle V$ tables (V$SESSTAT, V$SYSSTAT, V$STATNAME, V$INSTANCE, V$SESSION).
5. Verify connection with the Oracle server by performing **tns ping** from the Controller or Tuning Console machine. Note that there may be a problem connecting if the Oracle server is behind a DMZ/firewall that limits its communication to application servers accessing it.
6. Ensure that the registries are updated for the version of Oracle that you are using and that they have the following key:
HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE
7. Verify that the Oracle server you want to monitor is up and running.

Note: It is possible to monitor several Oracle database servers concurrently.

---

8. Run SQL*Plus from the Controller or Tuning Console and attempt to log in to the Oracle server(s) with the desired username/password/server combination.
9. Type SELECT * FROM V$SYSSTAT to verify that you can view the V$SYSSTAT table on the Oracle server. Use similar queries to verify that you can view the V$SESSTAT, V$SESSION, V$INSTANCE, V$STATNAME, and V$PROCESS tables on the server. Make sure that the Oracle **bin** directory is in the search path.
10. To change the length of each monitoring sample (in seconds), you need to edit the **dat\monitors\vmon.cfg** file in the LoadRunner root folder. The default rate is 10 seconds.

Note: The minimum sampling rate for the Oracle Monitor is 10 seconds. If you set the sampling rate at less than 10 seconds, the Oracle Monitor will continue to monitor at 10 second intervals.

If a problem occurs in setting up the Oracle environment, view the error message issued by the Oracle server.

Adding a Monitored Machine to the Controller

1. Click the Oracle graph in the graph tree, and drag it into the right pane of the Run view.
2. Right-click the graph and select **Add Measurements**, or choose **Monitors** > **Add Measurements**. The Oracle dialog box opens.
3. In the **Monitored Server Machines** section, click **Add.** The Add Machine dialog box opens.
4. Enter the server name or IP address of the machine you want to monitor. Select any platform, and click **OK**.
5. In the **Resource Measurements** section of the Oracle dialog box, click **Add**.

Continue with Configuring the Oracle Monitor below.

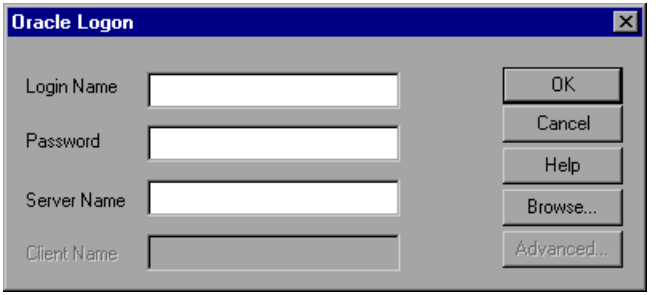
Adding a Monitored Machine to the Tuning Console

1. Click **Monitors** to open the Monitors Configuration dialog box.
2. Select the server whose monitors you want to configure from the Server list box.
3. Click **Add Monitor** and select **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.
4. In the left section of the dialog box, expand the Database Server Resource category, select **Oracle**, and then click **Add**.

   Continue with Configuring the Oracle Monitor below.

Configuring the Oracle Monitor

1. The Oracle Logon dialog box opens.



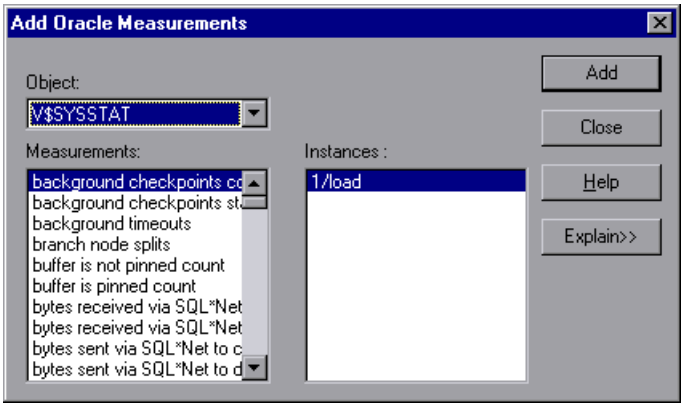
Enter the following information, and then click **OK.**

Login Name: Enter your login name.

Password: Enter your password.

Server Name: Enter the name of the server.

2 The Add Oracle Measurements dialog box opens, displaying the available measurements.



3 For each measurement, select an object, measurement, and instance, and then click **Add**, as described in "Understanding the Oracle Dialog Box" on page 262.

For a description of the available measurements, see "Oracle Performance Counters" on page 262.

4 Add all the desired resources to the list, and click **Close**.

5 In the Controller, click **OK** in the Oracle dialog box to activate the monitor.

6 In the Tuning Console, click **OK** in the Select Measurements to Monitor dialog box to activate the monitor.

Note: By default, the database returns the absolute value of a counter. However, by changing the IsRate setting in the **dat\monitors\vmon.cfg** file to 1, you can instruct the database to report a counter's rate value—the change in the counter per unit time.

Understanding the Oracle Dialog Box

The Oracle dialog box lets you select the resources to monitor from the V$SESSTAT and V$SYSSTAT Oracle V$ tables.

Object: Select the object being monitored on the specified machine.

Measurements: Select a resource measurement to monitor. Select multiple measurements using the CTRL key. For an explanation of each measurement, click **Explain**.

Instances: If multiple instances of the selected measurement are running, select one or more instances to monitor for the selected measurement.

Explain: Displays a description of the selected measurement.

Oracle Performance Counters

The following measurements are most commonly used when monitoring the Oracle server (from the V$SYSSTAT table):

Measurement	Description
CPU used by this session	This is the amount of CPU time (in 10s of milliseconds) used by a session between the time a user call started and ended. Some user calls can be completed within 10 milliseconds and, as a result, the start and end user-call time can be the same. In this case, 0 milliseconds are added to the statistic. A similar problem can exist in the operating system reporting, especially on systems that suffer from many context switches.
Bytes received via SQL*Net from client	The total number of bytes received from the client over Net8
Logons current	The total number of current logons
Opens of replaced files	The total number of files that needed to be reopened because they were no longer in the process file cache

Measurement	Description
User calls	Oracle allocates resources (Call State Objects) to keep track of relevant user call data structures every time you log in, parse, or execute. When determining activity, the ratio of user calls to RPI calls gives you an indication of how much internal work gets generated as a result of the type of requests the user is sending to Oracle.
SQL*Net roundtrips to/from client	The total number of Net8 messages sent to, and received from, the client
Bytes sent via SQL*Net to client	The total number of bytes sent to the client from the foreground process(es)
Opened cursors current	The total number of current open cursors
DB block changes	Closely related to consistent changes, this statistic counts the total number of changes that were made to all blocks in the SGA that were part of an update or delete operation. These are changes that are generating redo log entries and hence will be permanent changes to the database if the transaction is committed. This statistic is a rough indication of total database work and indicates (possibly on a per-transaction level) the rate at which buffers are being dirtied.
Total file opens	The total number of file opens being performed by the instance. Each process needs a number of files (control file, log file, database file) to work against the database.

Custom Queries

Using the custom query feature, you can define your own query to the Oracle database and view the result of this query—a single numerical value—in the Oracle online monitor graph. By defining your own query, you can monitor not only the V$SYSSTAT and V$SESSTAT table counters that are currently provided by the Oracle monitor, but other tables that contain useful performance information as well.

To create a custom query:

1. In the third line of the **vmon.cfg** file, CustomCounters=, indicate the number of custom counters you want to create.
2. Create a new section in the **vmon.cfg** file for the new counter. Each section has the following format:

   ```
   [Custom2]
   Name=Number of sessions
   Description=This counter returns the number of sessions active.
   Query=SELECT COUNT(*) FROM V$SESSION
   IsRate=1
   ```

3. In the [Custom#] line, assign the next number in the sequence of counters to the new custom counter. Note that the custom counters must be in consecutive order, beginning with the number 0.
4. In the Name line, enter the name of the new counter.
5. In the Description line, enter the description of the counter that you want the help message to contain.
6. In the Query line, enter the text of the SQL query (on one line of the **vmon.cfg** file) that returns exactly one row from the database. This row must contain one column, a numerical value.

   Note: Custom queries should not exceed 512 characters.

   ---

7. In the IsRate line, enter 0 if you want the database to report the counter as an absolute number. If you want the database to report the change in the counter per unit time, enter 1.

   Note: Custom queries cannot return negative values.

SQL Server Monitoring

The SQL Server monitor shows the standard Windows resources on the SQL server machine.

To obtain data for this graph, you enable the SQL Server online monitor (from the Controller or Tuning Console) and select the default measurements you want to display, before running the scenario or session step.

Note: To monitor an SQL server through a firewall, use TCP, port 139.

Adding a Monitored Machine to the Controller

1. Click the SQL Server graph in the graph tree, and drag it into the right pane of the Run view.
2. Right-click the graph and select **Add Measurements**, or choose **Monitors** > **Add Measurements**. The SQL Server dialog box opens.
3. In the **Monitored Server Machines** section, click **Add.** The Add Machine dialog box opens.
4. Enter the server name or IP address of the machine you want to monitor. Select the platform on which the machine runs, and click **OK**.
5. In the **Resource Measurements** section of the SQL Server dialog box, select the default measurements you want to monitor.

   For a description of the available measurements, see "SQL Server Performance Counters" on page 268.

Note: To change the default counters for the SQL Server monitor, see "Changing a Monitor's Default Counters" on page 443.

---

6. To select additional measurements, click **Add.**

   Continue with Configuring the SQL Server Monitor below.

Adding a Monitored Machine to the Tuning Console

1 Click **Monitors** to open the Monitors Configuration dialog box.

2 Select the server whose monitors you want to configure from the Server list box.

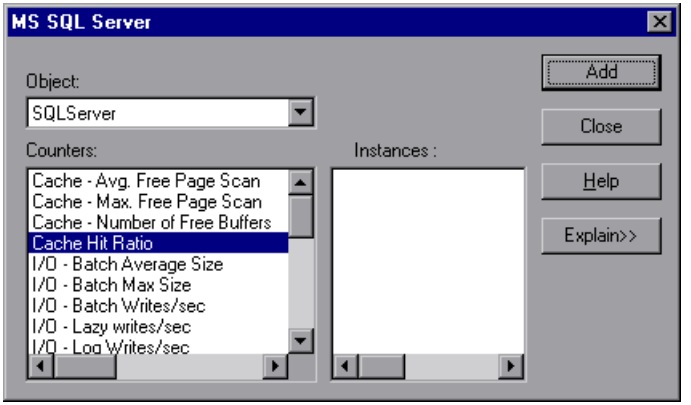
3 Click **Add Monitor** and select **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the Database Server Resource category, select **MS SQL Server**, and then click **Add**.

Continue with Configuring the SQL Server Monitor below.

Configuring the SQL Server Monitor

1 The SQL Server dialog box opens, displaying the SQL Server object, its counters, and instances opens.



For each measurement, select an object, counter, and instance, and then click **Add**, as described in "Understanding the SQL Server Dialog Box" on page 267.

For a description of the available measurements, see "SQL Server Performance Counters" on page 268.

2 Add all the desired resources to the list, and click **Close**.

3 In the Controller, click **OK** in the SQL Server dialog box to activate the monitor.

4 In the Tuning Console, click **OK** in the Select Measurements to Monitor dialog box to activate the monitor.

Note: Certain measurements or counters are especially useful for determining server performance and isolating the cause of a bottleneck during an initial stress test on the SQL Server. For more information about these counters, see "Useful Counters for Stress Testing" on page 444.

Understanding the SQL Server Dialog Box

The SQL Server dialog box lets you select additional resources to monitor on the SQL Server.

Object: Select the object being monitored on the specified machine.

Counters: Select a resource counter to monitor. Select multiple counters using the CTRL key. For an explanation of each counter, click **Explain**.

Instances: If multiple instances of the selected counter are running, select one or more instances to monitor for the selected counter.

Explain: Displays a description of the selected counter.

SQL Server Performance Counters

The following table describes the default counters that can be monitored on version 6.5 of the SQL Server:

Measurement	Description
% Total Processor Time (NT)	The average percentage of time that all the processors on the system are busy executing non-idle threads. On a multi-processor system, if all processors are always busy, this is 100%, if all processors are 50% busy this is 50% and if 1/4th of the processors are 100% busy this is 25%. It can be viewed as the fraction of the time spent doing useful work. Each processor is assigned an Idle thread in the Idle process which consumes those unproductive processor cycles not used by any other threads.
% Processor Time (Win 2000)	The percentage of time that the processor is executing a non-idle thread. This counter was designed as a primary indicator of processor activity. It is calculated by measuring the time that the processor spends executing the thread of the idle process in each sample interval, and subtracting that value from 100%. (Each processor has an idle thread which consumes cycles when no other threads are ready to run). It can be viewed as the percentage of the sample interval spent doing useful work. This counter displays the average percentage of busy time observed during the sample interval. It is calculated by monitoring the time the service was inactive, and then subtracting that value from 100%.
Cache Hit Ratio	The percentage of time that a requested data page was found in the data cache (instead of being read from disk)
I/O - Batch Writes/sec	The number of 2K pages written to disk per second, using Batch I/O. The checkpoint thread is the primary user of Batch I/O.
I/O - Lazy Writes/sec	The number of 2K pages flushed to disk per second by the Lazy Writer
I/O - Outstanding Reads	The number of physical reads pending

Measurement	Description
I/O - Outstanding Writes	The number of physical writes pending
I/O - Page Reads/sec	The number of physical page reads per second
I/O - Transactions/sec	The number of Transact-SQL command batches executed per second
User Connections	The number of open user connections

Sybase Monitoring

The Sybase monitor enables monitoring of Sybase Adaptive Server Enterprise (Sybase ASE) servers (version 11 or later) on Windows and UNIX. The monitor connects to the Sybase ASE server (via the Adaptive Server Enterprise Monitor Server) and retrieves metrics from the server using standard, Sybase-provided libraries.

To , you must first set up the Sybase monitoring environment. You then enable the Oracle online monitor (from the Controller or Tuning Console) and select the default measurements you want to display, before running the scenario or session step.

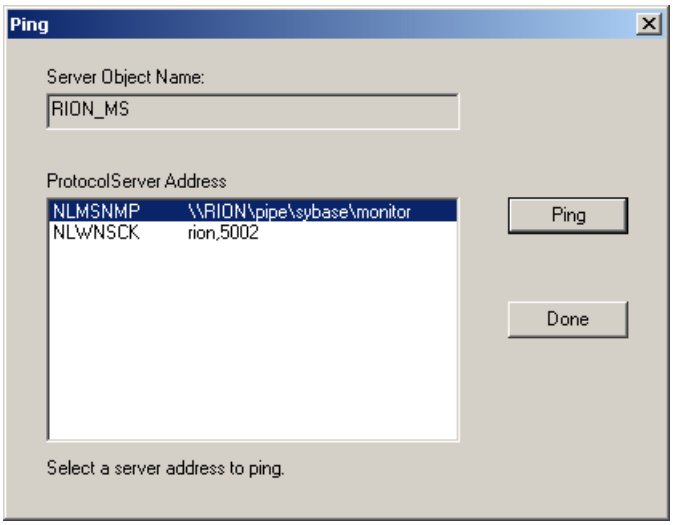
Note: When connecting to the monitored server, you connect to the Adaptive Server Enterprise Monitor Server, not the Sybase ASE server. The Adaptive Server Enterprise Monitor Server is an application that runs on the same machine as Sybase ASE server and retrieves performance information from it. The Adaptive Server Enterprise Monitor Server usually has the same name as the Sybase server, but with the suffix **_ms**.

Setting Up the Monitoring Environment

You need to set up the Sybase monitor environment before you can configure the monitor.

To set up the Sybase monitor environment:

1. Install the Sybase client files and libraries on the Controller or Tuning Console machine.
2. Verify a connection between the client and server on the Controller or Tuning Console machine. To do so, use the Sybase client's **dsedit** tool to ping the Adaptive Server Enterprise Monitor Server.



Note: The port you use to monitor a Sybase server through a firewall depends on the configuration of the Sybase server. Configuration information for the connection between the client and server is located in the Sybase client **sql.ini** file.

After you have set up the Sybase monitor environment, you can select the objects you want the monitor to measure.

Adding a Monitored Machine to the Controller

1. Click the Sybase graph in the graph tree, and drag it into the right pane of the Run view.
2. Right-click the graph and select **Add Measurements**, or choose **Monitors** > **Add Measurements**. The Sybase dialog box opens.
3. In the **Monitored Server Machines** section, click **Add.** The Add Machine dialog box opens.
4. Enter the server name or IP address of the machine you want to monitor. Select any platform, and click **OK**.
5. In the **Resource Measurements** section of the Sybase dialog box, click **Add**.

   Continue with Configuring the Sybase ASE Monitor below.

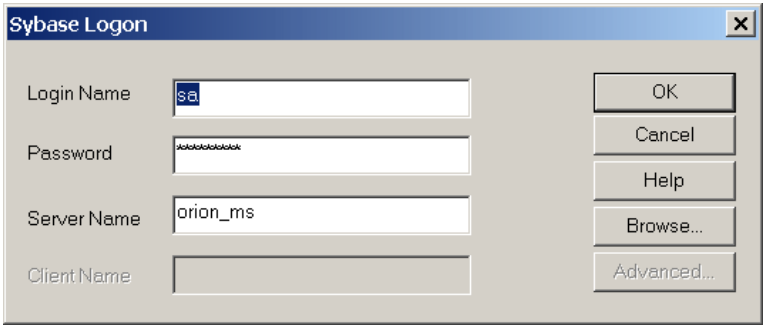
Adding a Monitored Machine to the Tuning Console

1. Click **Monitors** to open the Monitors Configuration dialog box.
2. Select the server whose monitors you want to configure from the Server list box.
3. Click **Add Monitor** and select **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.
4. In the left section of the dialog box, expand the Database Server Resource category, select **Sybase**, and then click **Add**.

   Continue with Configuring the Sybase ASE Monitor below.

Configuring the Sybase ASE Monitor

1 The Sybase Logon dialog box opens.



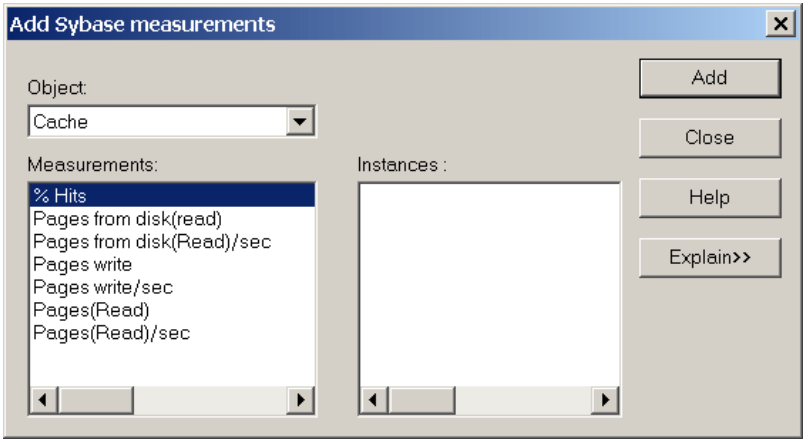
Enter the following information, and then click OK.

Login Name: Enter your login name.

Password: Enter your password.

Server Name: Enter the name of the server (usually the same name as the Sybase server but with the suffix **_ms**).

2 Click **OK**. The Add Sybase Measurements dialog box opens.



For each measurement, select an object, measurement, and instance, and then click **Add**, as described in “Understanding the Sybase Dialog Box” on page 273.

For a description of the available measurements, see "Sybase Performance Counters" on page 274.

3 Add all the desired resources to the list, and click **Close**.

4 In the Controller, click **OK** in the Sybase dialog box to activate the monitor.

5 In the Tuning Console, click **OK** in the Select Measurements to Monitor dialog box to activate the monitor.

Understanding the Sybase Dialog Box

The Sybase dialog box lets you select additional resources to monitor on the Sybase server.

Object: Select the object being monitored on the specified machine.

Measurements: Select a resource measurement to monitor. Select multiple measurements using the CTRL key. For an explanation of each measurement, click **Explain**.

Instances: If multiple instances of the selected measurement are running, select one or more instances to monitor for the selected measurement.

Explain: Displays a description of the selected measurement.

Sybase Performance Counters

The following measurements are available when monitoring a Sybase server:

Object	Measurement	Description
Network	**Average packet size (Read)**	Reports the number of network packets received
	Average packet size (Send)	Reports the number of network packets sent
	Network bytes (Read)	Reports the number of bytes received, over the sampling interval
	Network bytes (Read)/sec	Reports the number of bytes received, per second
	Network bytes (Send)	Reports the number of bytes sent, over the sampling interval
	Network bytes (Send)/sec	Reports the number of bytes sent, per second
	Network packets (Read)	Reports the number of network packets received, over the sampling interval
	Network packets (Read)/sec	Reports the number of network packets received, per second
	Network packets (Send)	Reports the number of network packets sent, over the sampling interval
	Network packets (Send)/sec	Reports the number of network packets sent, per second
Memory	**Memory**	Reports the amount of memory, in bytes, allocated for the page cache

Object	Measurement	Description
Disk	**Reads**	Reports the number of reads made from a database device
	Writes	Reports the number of writes made to a database device
	Waits	Reports the number of times that access to a device had to wait
Disk	**Grants**	Reports the number of times access to a device was granted
Engine	**Server is busy (%)**	Reports the percentage of time during which the Adaptive Server is in a "busy" state
	CPU time	Reports how much "busy" time was used by the engine
	Logical pages (Read)	Reports the number of data page reads, whether satisfied from cache or from a database device
	Pages from disk (Read)	Reports the number of data page reads that could not be satisfied from the data cache
	Pages stored	Reports the number of data pages written to a database device
Stored Procedures	**Executed (sampling period)**	Reports the number of times a stored procedure was executed, over the sampling interval
	Executed (session)	Reports the number of times a stored procedure was executed, during the session
	Average duration (sampling period)	Reports the time, in seconds, spent executing a stored procedure, over the sampling interval
	Average duration (session)	Reports the time, in seconds, spent executing a stored procedure, during the session

Object	Measurement	Description
Locks	**% Requests**	Reports the percentage of successful requests for locks
	Locks count	Reports the number of locks. This is an accumulated value.
Locks	**Granted immediately**	Reports the number of locks that were granted immediately, without having to wait for another lock to be released
	Granted after wait	Reports the number of locks that were granted after waiting for another lock to be released
	Not granted	Reports the number of locks that were requested but not granted
	Wait time (avg.)	Reports the average wait time for a lock
SqlSrvr	**Locks/sec**	Reports the number of locks. This is an accumulated value.
	% Processor time (server)	Reports the percentage of time that the Adaptive Server is in a "busy" state
	Transactions	Reports the number of committed Transact-SQL statement blocks (transactions)
	Deadlocks	Reports the number of deadlocks

Object	Measurement	Description
Cache	**% Hits**	Reports the percentage of times that a data page read could be satisfied from cache without requiring a physical page read
	Pages (Read)	Reports the number of data page reads, whether satisfied from cache or from a database device
	Pages (Read)/sec	Reports the number of data page reads, whether satisfied from cache or from a database device, per second
Cache	**Pages from disk (Read)**	Reports the number of data page reads that could not be satisfied from the data cache
	Pages from disk (Read)/sec	Reports the number of data page reads, per second, that could not be satisfied from the data cache
	Pages (Write)	Reports the number of data pages written to a database device
	Pages (Write)/sec	Reports the number of data pages written to a database device, per second

Object	Measurement	Description
Process	**% Processor time (process)**	Reports the percentage of time that a process running a given application was in the "Running" state (out of the time that all processes were in the "Running" state)
	Locks/sec	Reports the number of locks, by process. This is an accumulated value.
	% Cache hit	Reports the percentage of times that a data page read could be satisfied from cache without requiring a physical page read, by process
	Pages (Write)	Reports the number of data pages written to a database device, by process
Transaction	**Transactions**	Reports the number of committed Transact-SQL statement blocks (transactions), during the session
	Rows (Deleted)	Reports the number of rows deleted from database tables during the session

Object	Measurement	Description
Transaction	**Inserts**	Reports the number of insertions into a database table during the session
	Updates	Reports the updates to database tables during the session
	Updates in place	Reports the sum of expensive, in-place and not-in-place updates (everything except updates deferred) during the session
	Transactions/sec	Reports the number of committed Transact-SQL statement blocks (transactions) per second
	Rows (Deleted)/sec	Reports the number of rows deleted from database tables, per second
	Inserts/sec	Reports the number of insertions into a database table, per second
	Updates/sec	Reports the updates to database tables, per second
	Updates in place/sec	Reports the sum of expensive, in-place and not-in-place updates (everything except updates deferred), per second

10

Streaming Media Monitoring

To isolate server and client performance bottlenecks during a session or scenario run, you monitor the Windows Media Server and RealPlayer audio/video servers, as well as the RealPlayer and Media Player clients.

This chapter describes:

- About Streaming Media Monitoring
- Windows Media Server Monitoring
- RealPlayer Server Monitoring
- RealPlayer Client Monitoring
- Media Player Client Monitoring

Note: For instructions on recording a script containing streaming media functions, see the *Mercury Virtual User Generator User's Guide*.

About Streaming Media Monitoring

The streaming media monitors provide you with performance information for the Windows Media Server and RealPlayer audio/video servers, as well as the RealPlayer and Media Player clients. To obtain data for the Windows Media Server and RealPlayer Server, you need to activate the streaming media monitor before executing the session or scenario, and indicate which statistics and measurements you want to monitor. The RealPlayer Client and Media Player Client do not require pre-session or scenario activation or configuration.

Windows Media Server Monitoring

The Windows Media Server monitor displays statistics about the resource usage on the Windows Media Server during the scenario or session step run.

To obtain data for this graph, you need to enable the Windows Media Server online monitor (from the Controller or Tuning Console) and select the default measurements you want to display, before running the scenario or session step. You select these counters using the Windows Media Server dialog box.

Adding a Monitored Machine to the Controller

1. Click the Windows Media Server graph in the graph tree, and drag it into the right pane of the Run view.
2. Right-click the graph and select **Add Measurements**, or choose **Monitors** > **Add Measurements**. The Windows Media Server dialog box opens.
3. In the **Monitored Server Machines** section, click **Add.** The Add Machine dialog box opens.
4. Enter the server name or IP address of the machine you want to monitor. Select the platform on which the machine runs, and click **OK**.
5. In the **Resource Measurements** section of the Windows Media Server dialog box, select the default measurements you want to monitor.

   For a description of the available measurements, see "Windows Media Server Performance Counters" on page 284.
6. To select additional measurements, click **Add.**

   Continue with Configuring the Windows Media Server Monitor below.

Adding a Monitored Machine to the Tuning Console

1. Click **Monitors** to open the Monitors Configuration dialog box.
2. Select the server whose monitors you want to configure from the Server list box.

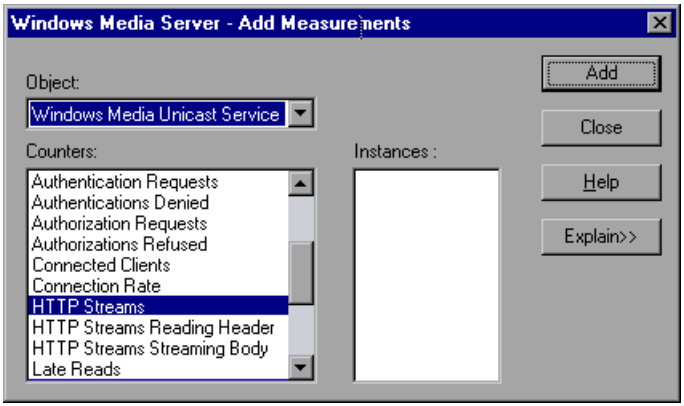
3 Click **Add Monitor** and select **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the Streaming Media Server category, select **Windows Media Server**, and then click **Add**.

Continue with Configuring the Windows Media Server Monitor below.

Configuring the Windows Media Server Monitor

1 The Windows Media Server - Add Measurements dialog box opens, displaying the Windows Media Unicast Service object, its counters, and instances.



For each measurement, select an object, a counter, and an instance, and then click **Add**, as described in “Understanding the Windows Media Server Dialog Box” on page 284.

For a description of the available measurements, see “Windows Media Server Performance Counters” on page 284.

2 Add all the desired resources to the list, and click **Close**.

3 In the Controller, click **OK** in the Windows Media Server dialog box to activate the monitor.

4 In the Tuning Console, click **OK** in the Select Measurements to Monitor dialog box to activate the monitor.

Understanding the Windows Media Server Dialog Box

The Windows Media Server dialog box lets you select the resources to monitor on the Windows Media Server.

Object: Select the object being monitored on the specified machine.

Counters: Select a resource counter to monitor. Select multiple counters using the CTRL key. For an explanation of each counter, click **Explain**.

Instances: If multiple instances of the selected counter are running, select one or more instances to monitor for the selected counter.

Explain: Displays a description of the selected counter.

Windows Media Server Performance Counters

The following table describes the default counters that can be monitored:

Measurement	Description
Active Live Unicast Streams (Windows)	The number of live unicast streams that are being streamed.
Active Streams	The number of streams that are being streamed.
Active TCP Streams	The number of TCP streams that are being streamed.
Active UDP Streams	The number of UDP streams that are being streamed.
Aggregate Read Rate	The total, aggregate rate (bytes/sec) of file reads.
Aggregate Send Rate	The total, aggregate rate (bytes/sec) of stream transmission.
Connected Clients	The number of clients connected to the server.
Connection Rate	The rate at which clients are connecting to the server.
Consoles	The number of Consoles currently connected to the server.
HTTP Streams	The number of HTTP streams being streamed.
Late Reads	The number of late read completions per second.

Measurement	Description
Pending Connections	The number of clients that are attempting to connect to the server, but are not yet connected. This number may be high if the server is running near maximum capacity and cannot process a large number of connection requests in a timely manner.
Stations	The number of station objects that currently exist on the server.
Streams	The number of stream objects that currently exist on the server.
Stream Errors	The cumulative number of errors occurring per second.

RealPlayer Server Monitoring

The Real Server monitor displays statistics about the resource usage on the RealPlayer Server during the scenario or session step run.

To obtain data for this graph, you need to enable the RealPlayer Server online monitor (from the Controller or Tuning Console) and select the default measurements you want to display, before running the scenario or session step. You select these counters using the Real Server dialog box.

Adding a Monitored Machine to the Controller

1. Click the Real Server graph in the graph tree, and drag it into the right pane of the Run view.
2. Right-click the graph and select **Add Measurements**, or choose **Monitors** > **Add Measurements**. The Real Server dialog box opens
3. In the **Monitored Server Machines** section, click **Add.** The Add Machine dialog box opens.
4. Enter the server name or IP address of the machine you want to monitor. Select the platform on which the machine runs, and click **OK**.

5 In the **Resource Measurements** section of the Real Server dialog box, click **Add**.

Continue with Configuring the RealPlayer Server Monitor below.

Adding a Monitored Machine to the Tuning Console

1 Click **Monitors** to open the Monitors Configuration dialog box.

2 Select the server whose monitors you want to configure from the Server list box.

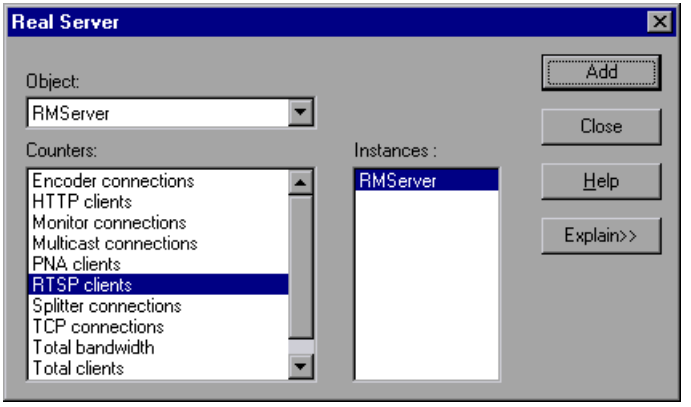
3 Click **Add Monitor** and select **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the Streaming Media Server category, select **RealPlayer Server**, and then click **Add**.

Continue with Configuring the RealPlayer Server Monitor below.

Configuring the RealPlayer Server Monitor

1 The Real Server dialog box opens, displaying the counters that can be monitored.



For each measurement, select an object, counter, and instance, and then click **Add**, as described in “Understanding the RealPlayer Server Dialog Box” on page 287.

For a description of the available measurements, see "Real Server Performance Counters" on page 288.

2. Add all the desired resources to the list, and click **Close**.
3. In the Controller, click **OK** in the Real Server dialog box to activate the monitor.
4. In the Tuning Console, click **OK** in the Select Measurements to Monitor dialog box to activate the monitor.

Understanding the RealPlayer Server Dialog Box

The Real Server dialog box lets you select the resources to monitor on the Real Server.

Object: Select the object being monitored on the specified machine.

Counters: Select a resource counter to monitor. Select multiple counters using the CTRL key. For an explanation of each counter, click **Explain**.

Instances: If multiple instances of the selected counter are running, select one or more instances to monitor for the selected counter.

Explain: Displays a description of the selected counter.

Real Server Performance Counters

The following table describes the default counters that can be monitored:

Measurement	Description
Encoder Connections	The number of active encoder connections
HTTP Clients	The number of active clients using HTTP
Monitor Connections	The number of active server monitor connections
Multicast Connections	The number of active multicast connections
PNA Clients	The number of active clients using PNA
RTSP Clients	The number of active clients using RTSP
Splitter Connections	The number of active splitter connections
TCP Connections	The number of active TCP connections
Total Bandwidth	The number of bits per second being consumed
Total Clients	The total number of active clients
UDP Clients	The number of active UDP connections

RealPlayer Client Monitoring

The Real Client monitor shows statistics on the RealPlayer client machine as a function of the elapsed scenario or session step time. The x-axis represents the time that has elapsed since the start of the scenario or session step run. The y-axis represents the resource usage.

Configuring the Real Client monitor

On the Controller:

You can view the RealPlayer Client online monitor graph by dragging it from the graph tree into the right pane of the Run view. The graph appears in the graph view area.

On the Tuning Console:

1. Right-click a graph and select **Open a New Graph**. The Open a New Graph dialog box opens.
2. Click the "+" next to **Streaming Media** to expand the graph tree, and select the **Real Client** graph.
3. Click **Open Graph**. The graph appears in the graph view area.

RealPlayer Client Performance Counters

The following table describes the RealPlayer Client measurements that are monitored:

Measurement	Description
Current Bandwidth (Kbits/sec)	The number of kilobytes in the last second
Buffering Event Time (sec)	The average time spent on buffering
Network Performance	The ratio (percentage) between the current bandwidth and the actual bandwidth of the clip
Percentage of Recovered Packets	The percentage of error packets that were recovered
Percentage of Lost Packets	The percentage of packets that were lost
Percentage of Late Packets	The percentage of late packets
Time to First Frame Appearance (sec)	The time for first frame appearance (measured from the start of the replay)
Number of Buffering Events	The average number of all buffering events
Number of Buffering Seek Events	The average number of buffering events resulting from a seek operation
Buffering Seek Time	The average time spent on buffering events resulting from a seek operation
Number of Buffering Congestion Events	The average number of buffering events resulting from network congestion
Buffering Congestion Time	The average time spent on buffering events resulting from network congestion
Number of Buffering Live Pause Events	The average number of buffering events resulting from live pause
Buffering Live Pause Time	The average time spent on buffering events resulting from live pause

Media Player Client Monitoring

The Media Player Client monitor graph shows statistics on the Windows Media Player client machine as a function of the elapsed scenario or session step time. The x-axis represents the time that has elapsed since the start of the scenario or session step run. The y-axis represents the resource usage.

Configuring the Windows Media Player Client monitor

On the Controller:

You can view the Windows Media Player Client online monitor graph by dragging it from the graph tree into the right pane of the Run view. The graph appears in the graph view area.

On the Tuning Console:

1. Right-click a graph and select **Open a New Graph**. The Open a New Graph dialog box opens.
2. Click the "+" next to **Streaming Media** to expand the graph tree, and select the **Media Player Client** graph.
3. Click **Open Graph**. The graph appears in the graph view area.

Media Player Client Performance Counters

The following table describes the Media Player Client measurements that are monitored:

Measurement	Description
Average Buffering Events	The number of times Media Player Client had to buffer incoming media data due to insufficient media content.
Average Buffering Time (sec)	The time spent by Media Player Client waiting for sufficient amount of media data in order to continue playing media clip.
Current bandwidth (Kbits/sec)	The number of kbits per second received.

Measurement	Description
Number of Packets	The number of packets sent by server for a particular media clip.
Stream Interruptions	The number of interruptions encountered by media player client while playing a media clip. This measurement includes the number of times Media Player Client had to buffer incoming media data, and any errors that occurred during playback.
Stream Quality (Packet-level)	The percentage ratio of packets received to total packets.
Stream Quality (Sampling-level)	The percentage of stream samples received on time (no delays in reception).
Total number of recovered packets	The number of lost packets that were recovered. This value is only relevant during network playback.
Total number of lost packets	The number of lost packets that were not recovered. This value is only relevant during network playback.

11

ERP/CRM Server Resource Monitoring

You use LoadRunner's ERP/CRM server resource monitors to monitor ERP/CRM servers during a scenario or session step run and isolate server performance bottlenecks.

This chapter describes:

- About ERP/CRM Server Resource Monitoring
- Choosing Between Different SAP Monitors
- SAP Server Resource Monitoring
- SAPGUI Server Resource Monitoring
- SAP Portal Server Resource Monitoring
- SAP CCMS Resource Monitoring
- Siebel Web Server Resource Monitoring
- Siebel Server Manager Resource Monitoring
- PeopleSoft (Tuxedo) Resource Monitoring
- PeopleSoft Ping Monitoring

Note: The PeopleSoft Ping monitor is only available in the Mercury Tuning Console.

About ERP/CRM Server Resource Monitoring

The ERP/CRM server resource monitors provide you with performance information for ERP/CRM servers. To display this data, you must activate the monitors before executing the scenario or session step and select the statistics and measurements you want to monitor.

Choosing Between Different SAP Monitors

You can monitor the resource usage of SAP solutions R/3, BW, CRM, APO, SAP Enterprise Portal, and SAPGUI for HTML during a scenario run using the SAP server monitors. Use the support matrix below to help you select the appropriate SAP monitor.

Support Matrix

SAP Application	Server Release	LoadRunner Monitor	Required SAPGUI
• R/3 3.1-4.6D • BW 2.x	SAP R/3 Kernel 3.1-4.6D (based on the SAP Application Server)	• SAP Monitor • SAP CCMS Monitor	SAPGUI 3.1-6.20
• R/3 4.7 Enterprise • BW 3.1 • CRM 4.0 • APO 3.x	SAP R/3 Kernel WAS 6.20 (based on the SAP Web Application Server)	• SAPGUI Monitor • SAP CCMS Monitor	SAPGUI 6.20

SAP Application	Server Release	LoadRunner Monitor	Required SAPGUI
SAP Enterprise Portal	No SAP System integration	• SAP Portal Monitor	
	SAP 3.1-4.6 kernel based SAP System integration	• SAP Monitor • SAP CCMS Monitor • SAP Portal Monitor	LoadRunner Controller machine: SAPGUI 3.1-6.20 (see note below)
	SAP 6.20 kernel based SAP System integration	• SAPGUI Monitor • SAP CCMS Monitor • SAP Portal Monitor	LoadRunner Controller machine: SAPGUI 6.20 (see note below)
SAPGUI for HTML	SAP 3.1-4.6 kernel based SAP System integration	• SAP Monitor • SAP CCMS Monitor	LoadRunner Controller machine: SAPGUI 3.1-6.20 (see note below)
	SAP 6.20 kernel based SAP System integration	• SAPGUI Monitor • SAP CCMS Monitor	LoadRunner Controller machine: SAPGUI 6.20 (see note below)

Note: A connection from the Controller to the SAP System using SAPGUI is only required for monitoring purposes. SAPGUI is not required on the load generator machines.

SAP Server Resource Monitoring

The SAP monitor displays statistics about the resource usage of a SAP R/3 system during the scenario or session step run. This monitor is appropriate for SAP R/3 server and SAPGUI (previous to version 6.20) environments.

You can use the SAP monitor to view:

- the number of configured instances for each SAP system
- data for all application instances (not just the one you logged on to)
- transactions used and the users that call them
- number of users working on the different instances
- performance history for recent periods of all instances
- response time distribution
- resource consumption for any application server
- application server workload for the current day or for a recent period.

To obtain data on the SAP R/3 system server, you need to enable the SAP online monitor (from the Controller or Tuning Console) before executing the scenario or session step, and indicate which statistics and measurements you want to monitor. You select these counters using the Add SAP Measurements dialog box.

Note: The SAP R/3 performance monitor supports SAP server versions 3.1 to 4.6, regardless of the SAP R/3 server's operating system and the platform on which it is installed.

Setting Up the Monitoring Environment

Before monitoring a SAP R/3 system server, perform the following:

- Install the SAP GUI client on the Controller or Tuning Console machine.
- Invoke the SAP GUI client and log on to the SAP server.

➤ From the SAP GUI client application, press **F6** to determine whether you can access the st03 transaction and query for **last minute load** information. If this functionality is not already enabled, enable it from the SAP R/3 client on the Controller or Tuning Console machine, using the username and password defined in the Controller or Tuning Console.

Adding a Monitored Machine to the Controller

1. Click the SAP graph in the graph tree, and drag it into the right pane of the Run view.
2. Right-click the graph and select **Add Measurements**, or choose **Monitors** > **Add Measurements**. The SAP dialog box opens.
3. In the **Monitored Server Machines** section, click **Add**. The Add Machines dialog opens.
4. Enter the server name or IP address of the machine you want to monitor, select the platform on which the machine runs, and click **OK**.

Note: You can also specify an IP address and a system number using the following format:
<IP address:system number>
For example: 199.35.106.162:00

---

5. In the **Resource Measurements** section, click **Add**.

   Continue with Configuring the SAP Monitor below.

Adding a Monitored Machine to the Tuning Console

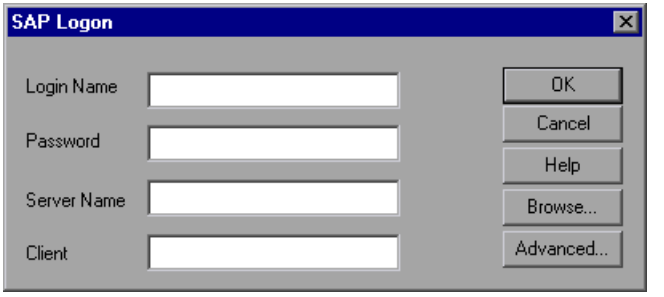
1. Click **Monitors** to open the Monitors Configuration dialog box.
2. From the Server list, choose the server running SAP.
3. Click **Add Monitor** and select **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the ERP-CRM Server category, select **SAP**, and then click **Add**.

Continue with Configuring the SAP Monitor below.

Configuring the SAP Monitor

1 The SAP Logon dialog box opens.



Enter the following information:

Login Name: Login name used to access the SAP server.

Password: Password for the login name.

Server Name: Name of the SAP server.

Client: Name of the client machine.

Note: To connect to the SAP monitor through a router, you need to enter the router string into the Server Name field. A router string has the format: <RouterString/ServerIP/S/sapdpxx>

where RouterString is /H/<IP_ADDRESS>/H/<IP_ADDRESS>/H/
ServerIP is the application server IP address
and xx is the system number.

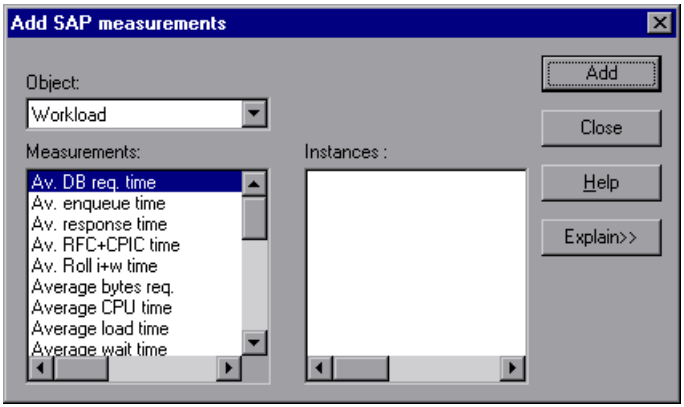
For example, if the router string = /H/199.35.107.9/H/204.79.199.244/H/, application server IP address = 172.20.11.6, and the system number = 00, you should enter the following string into the Server Name field:

/H/199.35.107.9/H/204.79.199.244/H/172.20.11.6/S/sapdp00

---

2 To change the default language, click **Advanced** and enter a 2-letter string in the **Language** field.

3 Click **OK**. The Add SAP Measurements dialog box opens.



4 For each measurement, select an object, measurement, and instance, and then click **Add**, as described in "Understanding the Add SAP Measurements Dialog Box" on page 300.

For a description of the available measurements, see "SAP Performance Counters" on page 301.

5 When you have finished adding measurements, click **Close**.

6 In the Controller, click **OK** in the SAP dialog box to activate the monitor.

7 In the Tuning Console, click **OK** in the Select Measurements to Monitor dialog box, and in the Monitors Configuration dialog box, to activate the monitor.

Understanding the Add SAP Measurements Dialog Box

The Add SAP Measurements dialog box enables you to select the SAP resources to monitor.

Object: Select the object being monitored on the specified machine.

Measurements: Select resource measurements to monitor. You can select multiple measurements using the CTRL key.

Instances: If multiple instances of the selected measurement are running, select one or more instances to monitor for the selected measurement.

Add: Click **Add** to add the selected object, measurement, and instance to the monitored measurement list.

Explain: Displays a description of the selected measurement.

SAP Performance Counters

The following table lists the most commonly monitored counters:

Measurement	Description
Average CPU time	The average CPU time used in the work process.
Average response time	The average response time, measured from the time a dialog sends a request to the dispatcher work process, through the processing of the dialog, until the dialog is completed and the data is passed to the presentation layer. The response time between the SAP GUI and the dispatcher is not included in this value.
Average wait time	The average amount of time that an unprocessed dialog step waits in the dispatcher queue for a free work process. Under normal conditions, the dispatcher work process should pass a dialog step to the application process immediately after receiving the request from the dialog step. Under these conditions, the average wait time would be a few milliseconds. A heavy load on the application server or on the entire system causes queues at the dispatcher queue.
Average load time	The time needed to load and generate objects, such as ABAP source code and screen information, from the database.
Database calls	The number of parsed requests sent to the database.
Database requests	The number of logical ABAP requests for data in the database. These requests are passed through the R/3 database interface and parsed into individual database calls. The proportion of database calls to database requests is important. If access to information in a table is buffered in the SAP buffers, database calls to the database server are not required. Therefore, the ratio of calls/requests gives an overall indication of the efficiency of table buffering. A good ratio would be 1:10.
Roll ins	The number of rolled-in user contexts.

Measurement	Description
Roll outs	The number of rolled-out user contexts.
Roll in time	The processing time for roll ins.
Roll out time	The processing time for roll outs.
Roll wait time	The queue time in the roll area. When synchronous RFCs are called, the work process executes a roll out and may have to wait for the end of the RFC in the roll area, even if the dialog step is not yet completed. In the roll area, RFC server programs can also wait for other RFCs sent to them.
Average time per logical DB call	The average response time for all commands sent to the database system (in milliseconds). The time depends on the CPU capacity of the database server, the network, the buffering, and on the input/output capabilities of the database server. Access times for buffered tables are many magnitudes faster and are not considered in the measurement.

SAPGUI Server Resource Monitoring

The SAPGUI monitor displays statistics about the resource usage of a SAP R/3 system during the scenario or session step run.

You can use the SAPGUI monitor to view:

- the number of configured instances for each SAP system
- data for all application instances (not just the one you logged on to)
- transactions used and the users that call them
- number of users working on the different instances
- performance history for recent periods of all instances
- response time distribution
- resource consumption for any application server
- application server workload for the current day or for a recent period

To obtain data on the SAP R/3 system server, you need to enable the SAPGUI online monitor (from the Controller or Tuning Console) before executing the scenario or session step, and indicate which statistics and measurements you want to monitor. You select these counters using the Add SAPGUI Monitor Measurements dialog box.

Note: The SAP R/3 performance monitor supports SAP server versions 3.1 to 4.6, regardless of the SAP R/3 server's operating system and the platform on which it is installed.

Setting Up the Monitoring Environment

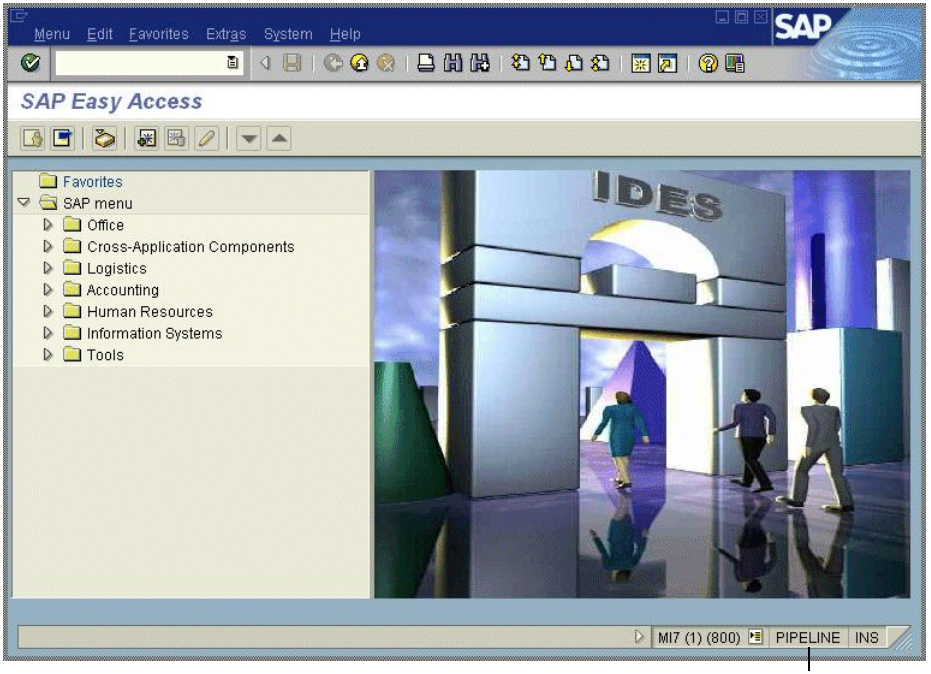
Before monitoring a SAP R/3 system server, perform the following:

- Install the SAPGUI for Windows 6.20 client on the Controller or Tuning Console machine.
- Install the latest patch for the SAPGUI for Windows 6.20 client. The lowest supported level is patch 36. (SAPGUI patches can be downloaded from https://websmp104.sap-ag.de/patches.)
- From the SAP GUI client application, press **F6** to determine whether you can access the st03 transaction and query for **last minute load** information. If this functionality is not already enabled, enable it from the SAP R/3 client on the Controller or Tuning Console machine, using the username and password defined in the Controller or Tuning Console.

For Tuning Console only:

- In the Tuning Console's System Topology window, add an ERP/CRM Server element representing the server running SAPGUI, and specify the following settings in the Element Properties tab:
  - In the Product field, choose **SAPGUI**.

➤ In the Logical Name field, enter the name that is displayed in the status bar of the SAP user interface when you are connected to a server, as displayed in the following figure.



Adding a Monitored Machine to the Controller

1. Click the SAPGUI graph in the graph tree, and drag it into the right pane of the Run view.
2. Right-click the graph and select **Add Measurements**, or choose **Monitors** > **Add Measurements**. The SAPGUI dialog box opens.
3. In the **Monitored Server Machines** section, click **Add**. The Add Machine dialog box opens.

4 Enter the logical name of the server you want to monitor, select the platform on which the machine runs, and click **OK**. To determine the logical name, see the status bar of the SAP user interface when you are connected to a server, as displayed in the figure above.

The SAPGUI dialog box is redisplayed.

5 In the **Resource Measurements** section of the SAPGUI dialog box, click **Add**.

Continue with Configuring the SAPGUI Monitor below.

Adding a Monitored Machine to the Tuning Console

1 Click **Monitors** to open the Monitors Configuration dialog box.

2 From the Server list, choose the server running SAPGUI.

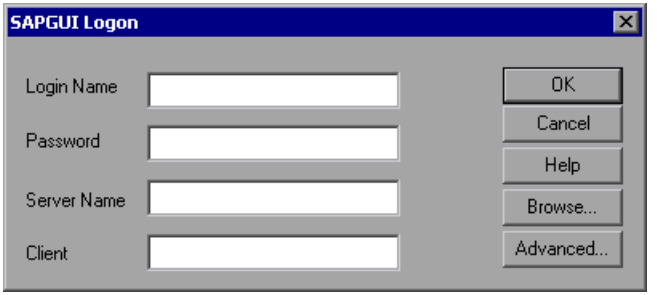
3 Click **Add Monitor** and select **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the ERP/CRM Server category, select **SAPGUI**, and then click **Add**.

Continue with Configuring the SAPGUI Monitor below.

Configuring the SAPGUI Monitor

1 The SAPGUI Logon dialog box opens.



Enter the following information in the SAPGUI Logon dialog box:

Login Name: Login name used to access the SAPGUI server.

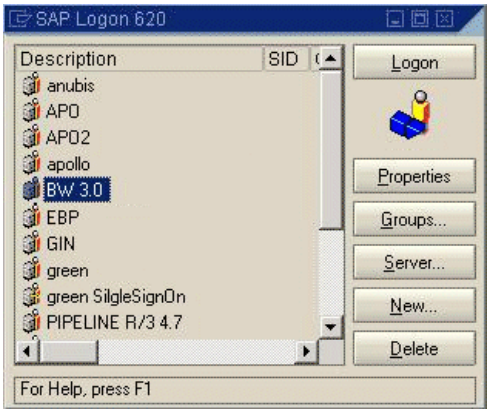
Password: Password for the login name.

Server Name: Name of the SAPGUI server.

Client: Name of the client machine.

You can enter the server name in the following ways:

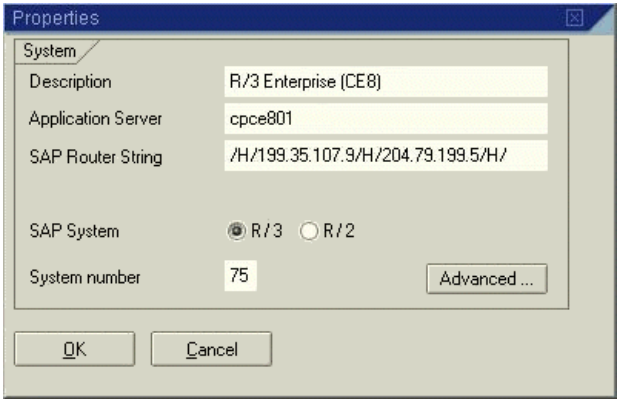
➤ The server description, as displayed in the SAP Logon application (for example BW 3.0 in the SAP Logon dialog box displayed below).



➤ A string, in the format: server_network_name[:system_number]

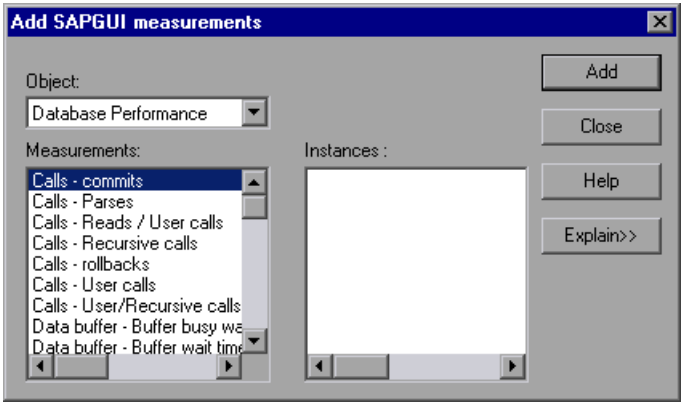
where server_network_name is the name or IP address of the application server as it is displayed in the Server Name field of the LoadRunner SAPGUI Logon dialog box (for example: pipeline.mercury.com), and system_number (preceded by ":") is the system number as it is displayed in the Properties dialog box. If the system number is omitted, "00" is used by default.

If a SAP router string is also specified in the Properties dialog box, the server_network_name should be the concatenation of the router string and the application server (for example, /H/199.35.107.9/H/204.79.199.5/H/cpce801 in the Properties dialog box displayed below).



2 To change the default language, click **Advanced** in the LoadRunner SAPGUI Logon dialog box, and enter a 2-letter string in the Language field.

3 Click **OK**. The Add SAPGUI Measurements dialog box opens.



4 For each measurement, select an object, measurement, and instance, and then click **Add**, as described in "Understanding the Add SAPGUI Measurements Dialog Box" on page 309.

For a description of the available measurements, see "SAPGUI Performance Counters" on page 310.

5 Click **Add** to place the selected measurement on the resource list. Add all the desired resources to the list, and click **Close**.

6 In the Controller, click **OK** in the SAPGUI dialog box to activate the monitor.

7 In the Tuning Console, click **OK** in the Select Measurements to Monitor dialog box, and in the Monitors Configuration dialog box, to activate the monitor.

Understanding the Add SAPGUI Measurements Dialog Box

The Add SAPGUI Measurements dialog box enables you to select the SAP resources to monitor.

Object: Select the object being monitored on the specified machine.

Measurements: Select resource measurements to monitor. You can select multiple measurements using the CTRL key.

Instances: If multiple instances of the selected measurement are running, select one or more instances to monitor for the selected measurement.

Add: Click to add the selected object, measurement, and instance to the monitored measurement list.

Explain: Displays a description of the selected measurement.

SAPGUI Performance Counters

The following table lists the most commonly monitored counters:

Measurement	Description
Average CPU time	The average CPU time used in the work process.
Average response time	The average response time, measured from the time a dialog sends a request to the dispatcher work process, through the processing of the dialog, until the dialog is completed and the data is passed to the presentation layer. The response time between the SAP GUI and the dispatcher is not included in this value.
Average wait time	The average amount of time that an unprocessed dialog step waits in the dispatcher queue for a free work process. Under normal conditions, the dispatcher work process should pass a dialog step to the application process immediately after receiving the request from the dialog step. Under these conditions, the average wait time would be a few milliseconds. A heavy load on the application server or on the entire system causes queues at the dispatcher queue.
Average load time	The time needed to load and generate objects, such as ABAP source code and screen information, from the database.
Database calls	The number of parsed requests sent to the database.
Database requests	The number of logical ABAP requests for data in the database. These requests are passed through the R/3 database interface and parsed into individual database calls. The proportion of database calls to database requests is important. If access to information in a table is buffered in the SAP buffers, database calls to the database server are not required. Therefore, the ratio of calls/requests gives an overall indication of the efficiency of table buffering. A good ratio would be 1:10.
Roll ins	The number of rolled-in user contexts.

Measurement	Description
Roll outs	The number of rolled-out user contexts.
Roll in time	The processing time for roll ins.
Roll out time	The processing time for roll outs.
Roll wait time	The queue time in the roll area. When synchronous RFCs are called, the work process executes a roll out and may have to wait for the end of the RFC in the roll area, even if the dialog step is not yet completed. In the roll area, RFC server programs can also wait for other RFCs sent to them.
Average time per logical DB call	The average response time for all commands sent to the database system (in milliseconds). The time depends on the CPU capacity of the database server, the network, the buffering, and on the input/output capabilities of the database server. Access times for buffered tables are many magnitudes faster and are not considered in the measurement.

SAP Portal Server Resource Monitoring

The SAP Portal monitor displays statistics about the resource usage of a SAP Enterprise Portal environment during the scenario or session step run.

To obtain data on the SAP R/3 system server, you need to enable the SAP online monitor (from the Controller or Tuning Console) before executing the scenario or session step, and indicate which statistics and measurements you want to monitor. You select these counters using the SAP Portal dialog box.

Setting up the Monitoring Environment

Before monitoring a SAP Portal server, perform the following:

➤ Ensure that SiteScope (the application that is used to monitor the SAP Portal server) has been installed on a server. You can install SiteScope on the Controller or Tuning Console machine, but we recommend installing it on a dedicated server.

For Tuning Console only:

➤ In the System Topology window, add an element representing the server running SAP Portal. Note that this element must not be a Load Generator.

Adding a Monitored Machine to the Controller

1. Click the SAP Portal graph in the graph tree, and drag it into the right pane of the Run view.
2. Right-click the graph and select **Add Measurements**, or choose **Monitors > Add Measurements**. The SAP Portal dialog box opens.
3. In the **Monitored Server Machines** section, click **Add.** The Add Machine dialog box opens.
4. In the **Monitored Machine Information** section, enter the server name or IP address of the machine you want to monitor, and select the platform on which the machine runs.

   In the **SiteScope Server Information** section, enter the SiteScope machine name, port (default: 8888), and version.

   Click **OK**. The SAP Portal dialog box is redisplayed.

5 In the **Resource Measurements** section, click **Add**.

Continue with Configuring the SAP Portal Monitor below.

Adding a Monitored Machine to the Tuning Console

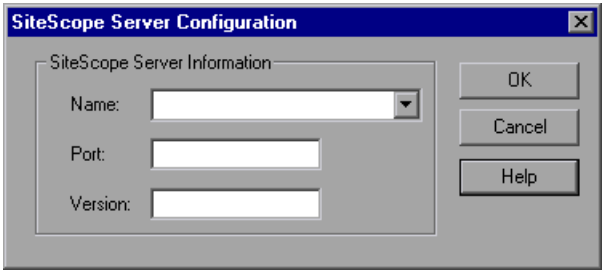
1 Click **Monitors** to open the Monitors Configuration dialog box.

2 From the Server list, choose the ERP/CRM server running SAP Portal.

3 Click **Add Monitor** and select **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the ERP-CRM Server category, select **SAP Portal**, and then click **Add**.

If the SiteScope server has not been configured yet, the SiteScope Server Configuration dialog box opens.

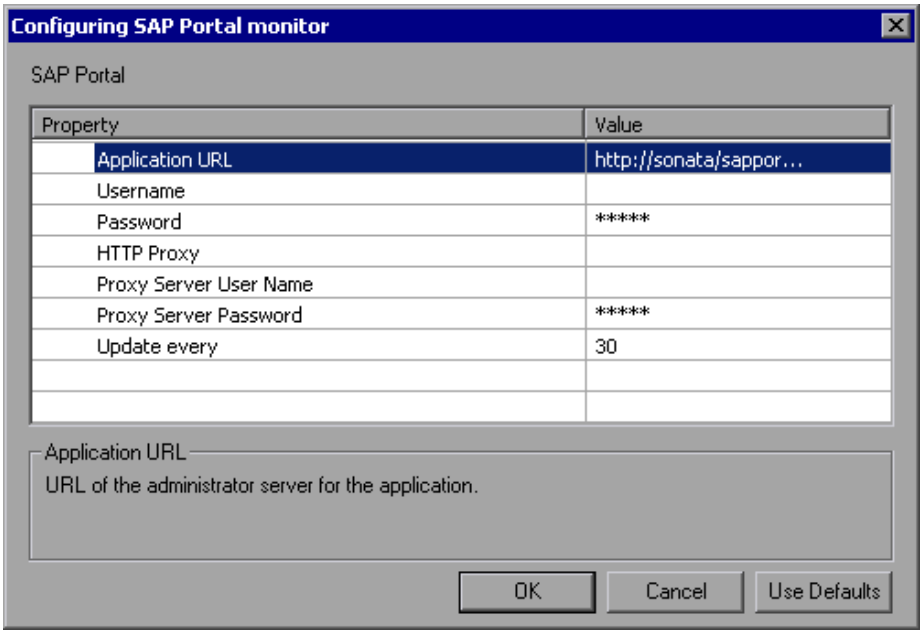


5 Enter the SiteScope server information as described in "Understanding the SiteScope Server Configuration Dialog Box" on page 43 and click **OK**.

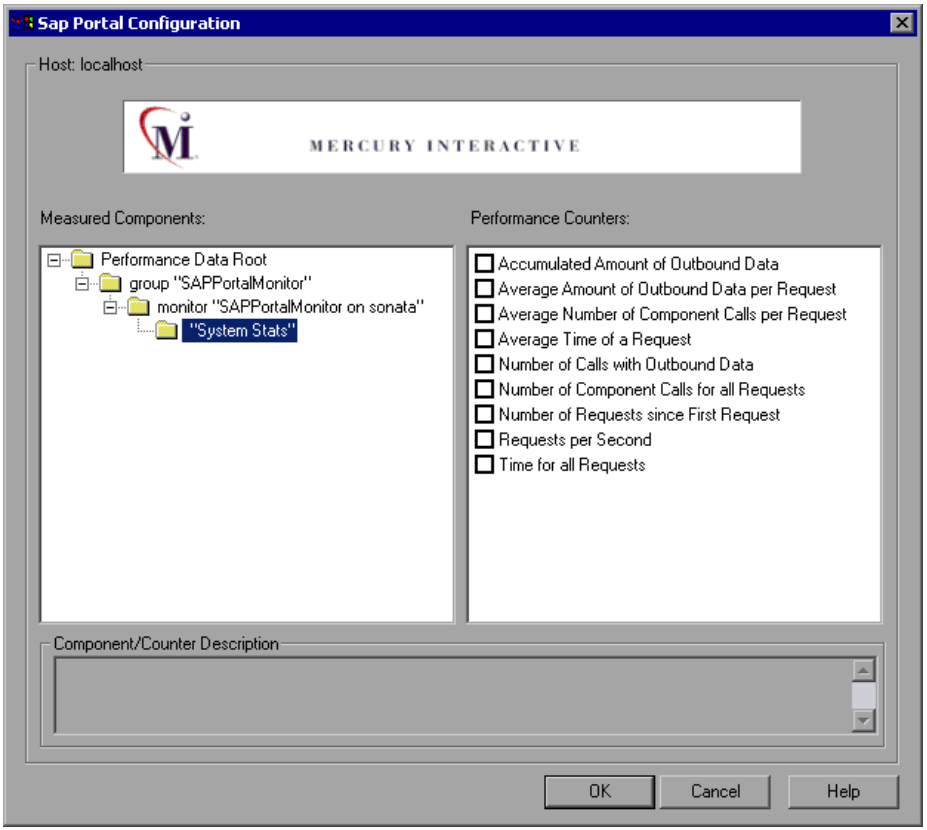
Continue with Configuring the SAP Portal Monitor below.

Configuring the SAP Portal Monitor

1 In the Configuring SAP Portal Monitor dialog box, enter the Sap Portal monitor information, as described in "Understanding the Configuring SAP Portal Monitor Dialog Box" on page 316.



2 Click **OK**. The SAP Portal Configuration dialog box opens, displaying the available measurements.



3 Browse the Measured Components tree, and select performance counters, as described in "Understanding the SAP Portal Configuration Dialog Box" on page 316, and then click **OK**.

For a description of the available measurements, see "SAP Portal Performance Counters" on page 317.

4 In the Controller, click **OK** in the SAP Portal Configuration dialog box, and in the SAP Portal dialog box, to activate the monitor.

5 In the Tuning Console, click **OK** in the Select Measurements to Monitor dialog box, and in the Monitors Configuration dialog box, to activate the monitor.

Note: The minimum recommended SiteScope monitor refresh rate is 30 seconds. If you choose a lower refresh rate, the Controller or Tuning Console may not get all the data in time.

Understanding the Configuring SAP Portal Monitor Dialog Box

The Configuring SAP Portal Monitor Dialog Box enables you to configure the SAP Portal monitor.

Application URL: Enter the URL of the administrator server for the application.

Username: Enter the user name for the server administrator page.

Password: Enter the password for the server administrator page.

HTTP Proxy: Enter a proxy server to use, including the port (optional).

Proxy Server User Name: If the proxy server requires authorization, enter the user name.

Proxy Server Password: If the proxy server requires authorization, enter the password.

Update every: Enter the amount of time lapse between the SiteScope check of the monitor.

Understanding the SAP Portal Configuration Dialog Box

The SAP Portal dialog box enables you to select the SAP Portal resources to monitor.

Host: Displays the name of the monitored machine.

Measured Components: Displays the available components. Browse the tree and select the component you want to monitor. A description of the highlighted component appears in the Component/Counter Description box.

Performance Counters: Check the required performance counters. A description of the selected counter appears in the Component/Counter Description box.

Component/Counter Description: Displays a description of the selected component or counter.

SAP Portal Performance Counters

The following table shows the default counters that can be measured:

Measurement	Description
Accumulated Amount of Outbound Data (bytes)	The accumulated amount of outbound data, measured in bytes.
Average Amount of Outbound Data per Request (bytes)	The average amount of outbound data per request, measured in bytes.
Average Number of Component Calls per Request (bytes)	The average number of component calls per request, measured in bytes.
Average Time of a Request (ms)	The average amount of time, in milliseconds, taken to process a request.
Number of Calls with Outbound Data	The total number of calls with outbound data.
Number of Component Calls for all Requests	The total number of component calls for all requests.
Number of Requests since First Request	The total number of requests since the first request was made.
Requests per Second	The number of requests made per second.
Time for all Requests (ms)	The total time, in milliseconds, taken for processing all requests.

SAP CCMS Resource Monitoring

The SAP CCMS (Computer Center Management System) monitor displays statistics about the resource usage of all servers, components, and resources in any SAP R/3 landscape during the scenario or session step run. You can also use the SAP CCMS monitor for SAP Portal and SAP GUI environments, but the amount of measurements provided by the SAP CCMS monitor is much greater.

To obtain data on the SAP R/3 landscape resources, you need to enable the SAP CCMS online monitor (from the Controller or Tuning Console) before executing the scenario or session step, and select the statistics and measurements you want to monitor. You select these counters using the SAP CCMS dialog box.

Setting up the Monitoring Environment

Before monitoring a SAP CCMS server, perform the following:

➤ Ensure that SiteScope (the application that is used to monitor the SAP R/3 server) has been installed on a server. You can install SiteScope on the Controller or Tuning Console machine, but we recommend installing it on a dedicated server.

➤ Ensure that the SAP Java Connector (SAP JCo 2.0.6 and above) component is installed on the same server where SiteScope is running (or at least is accessible on a shared or remote location). To install the SAP Java Connector, perform the following:

  ➤ Download the SAP Java Connector from the SAP Software Distribution Center at http://www.service.sap.com/connectors. Click **SAP Java Connector** and **Tools and Services**. You will need a valid Service Marketplace login to access this site.

  ➤ Follow the installation instructions that come with the SAP JCo download for your appropriate platform. On Windows, add the JCo installation location in the System Environment PATH variable. This change usually requires you to reboot Windows for the system PATH to be updated. Once completed, make the SAP JCo library file available to SiteScope by copying **sapjco.jar** (which comes with the JCo install) to **/SiteScope/java/lib/ext** before starting SiteScope.

➤ Consult your SAP documentation to determine if your R/3 landscape components may need additional software installed to run or work with CCMS.

Note: The BC-XAL 1.0 interface is supported on R/3 systems 4.5B and above only.

For Tuning Console only:

➤ In the Element Properties tab's Product field, choose **SAP CCMS**.

Adding a Monitored Machine to the Controller

1. Click the SAP CCMS graph in the graph tree, and drag it into the right pane of the Run view.
2. Right-click the graph and select **Add Measurements**, or choose **Monitors** > **Add Measurements**. The SAP CCMS dialog box opens.
3. In the **Monitored Server Machines** section, click **Add**. The Add Machine dialog box opens.
4. In the **Monitored Machine Information** section, enter the server name or IP address of the machine you want to monitor, and select the platform on which the machine runs.

   In the **SiteScope Server Information** section, enter the SiteScope machine name, port (default: 8888), and version.

   Click **OK**. The SAP CCMS dialog box is redisplayed.
5. In the **Resource Measurements** section, click **Add**.

   Continue with Configuring the SAP CCMS Monitor below.

Adding a Monitored Machine to the Tuning Console

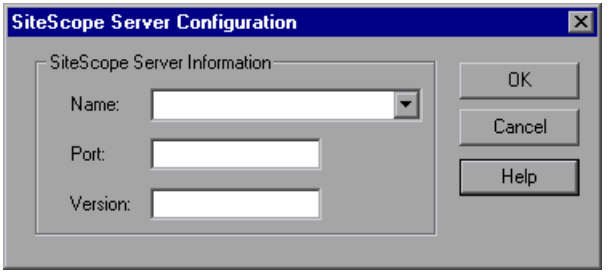
1 Click **Monitors** to open the Monitors Configuration dialog box.

2 From the Server list, choose the ERP/CRM server running SAP CCMS.

3 Click **Add Monitor** and select **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the ERP-CRM Server category, select **SAP CCMS**, and then click **Add**.

If the SiteScope server has not been configured yet, the SiteScope Server Configuration dialog box opens.

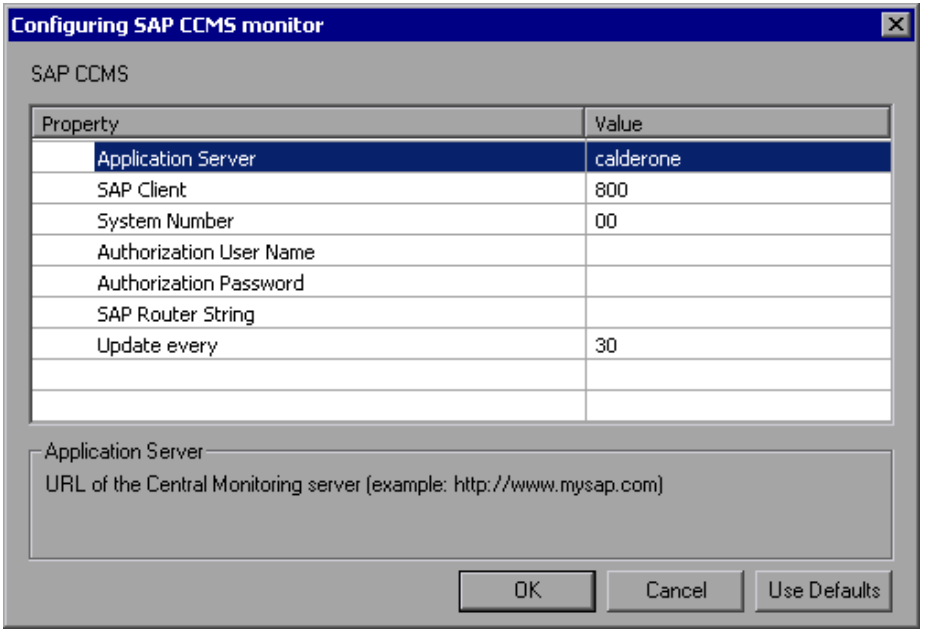


5 Enter the SiteScope server information as described in "Understanding the SiteScope Server Configuration Dialog Box" on page 43 and click **OK**.

Continue with Configuring the SAP CCMS Monitor below.

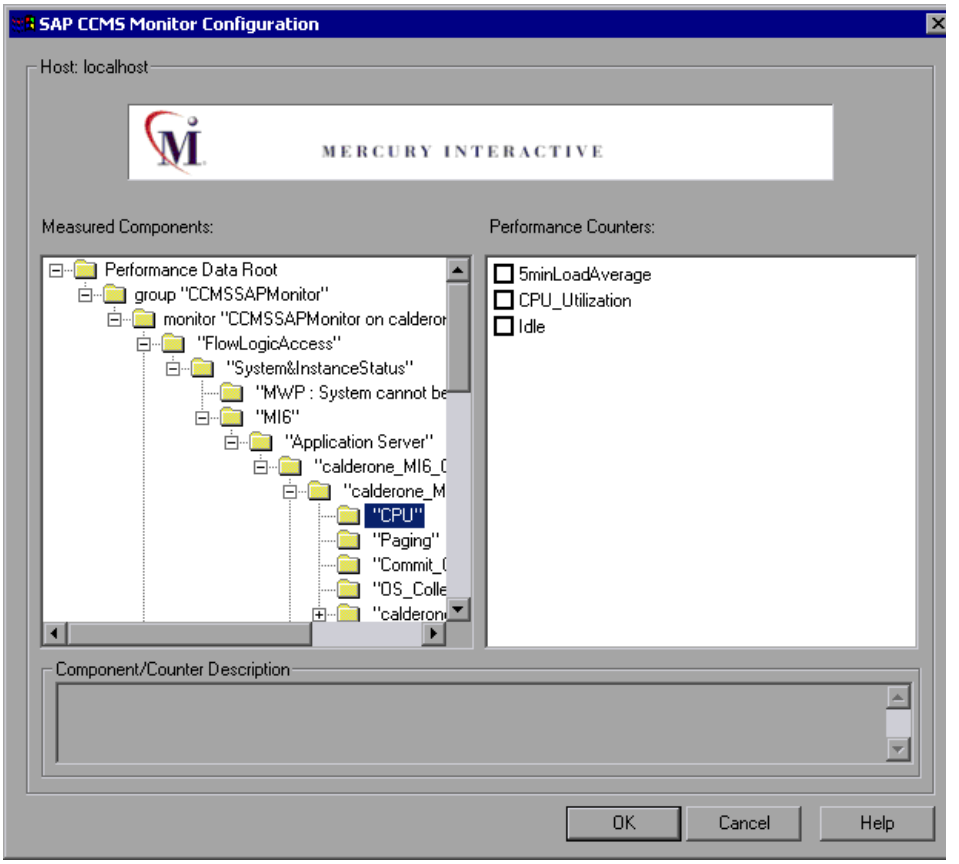
Configuring the SAP CCMS Monitor

1 The Configuring SAP CCMS Monitor dialog box opens.



Enter the SAP CCMS configuration values, as described in “Understanding the Configuring SAP CCMS Monitor Dialog Box” on page 323.

2 Click **OK**. The SAP CCMS Monitor Configuration dialog box opens, displaying the available measurements.



Note: Due to the large amount of metrics that are retrieved, it may take several minutes to display the performance counters tree. However, once the tree has been successfully retrieved, it will be cached to a file, so that the next time you retrieve metrics from the same server and username, the wait time will be greatly reduced.

---

3 Browse the Measured Components tree, and select performance counters, as described in "Understanding the SAP CCMS Monitor Configuration Dialog Box" on page 324, and then click **OK**.

For more information on the available measurements, refer to the SAP CCMS documentation.

4 In the Controller, click **OK** in the SAP CCMS Monitor Configuration dialog box, and in the SAP CCMS dialog box, to activate the monitor.

5 In the Tuning Console, click **OK** in the Select Measurements to Monitor dialog box, and in the Monitors Configuration dialog box, to activate the monitor.

Note: The minimum recommended SiteScope monitor refresh rate is 30 seconds. If you choose a lower refresh rate, the Controller or Tuning Console may not get all the data in time.

Understanding the Configuring SAP CCMS Monitor Dialog Box

The Configuring SAP CCMS Monitor dialog box enables you to configure the SAP CCMS monitor.

Application Server: Enter the address of the SAP server you want to monitor.

SAP Client: Enter the Client to use for connecting to SAP. A default client of 800 is typically used.

System Number: Enter the System number for the SAP server. A default system number of 00 is typically used.

Authorization User Name: Enter the Username required to connect to the SAP server.

Authorization Password: Enter the Password required to connect to the SAP server.

SAP Router String: If your connection is being made through a router, enter a router address string. You can find the router address using the SAP Logon tool from the SAP Client software. Open the Logon console, select the server you want to monitor and then select Properties to view the router address.

Update every: Enter how frequently (in seconds) the monitor should check the SAP server. SAP CCMS metrics are generally updated once every five minutes.

Understanding the SAP CCMS Monitor Configuration Dialog Box

The SAP CCMS dialog box lets you select the SAP CCMS resources to monitor.

Host: Displays the name of the monitored machine.

Measured Components: Displays the available components. Browse the tree and select the component you want to monitor. A description of the highlighted component appears in the Component/Counter Description box.

Performance Counters: Check the required performance counters. A description of the selected counter appears in the Component/Counter Description box.

Component/Counter Description: Displays a description of the selected component or counter.

Siebel Web Server Resource Monitoring

The Siebel Web Server monitor displays statistics about the resource usage of a Siebel Web Server during the scenario or session step run.

To obtain data on the Siebel Web Server, you need to enable the Siebel Web Server online monitor (from the Controller or Tuning Console) before executing the scenario or session step, and indicate which statistics and measurements you want to monitor. You select these counters using the Siebel Web Server dialog box.

Setting up the Monitoring Environment

Before monitoring a Siebel Web Server, perform the following:

➤ Ensure that the Siebel Web server plug-in is installed and configured to enable the display of the statistics you want to monitor. This may require that stats page sections be enabled by editing the **eapps.cfg** file for the Siebel server. For more information, refer to the Siebel documentation.

➤ Ensure that SiteScope has been installed on a server. SiteScope is the application that is used to monitor the Siebel server. Although you can install SiteScope on the Controller or Tuning Console machine, we recommend installing it on a dedicated server.

For Tuning Console only:

➤ In the System Topology window, add an ERP/CRM Server element representing the server running Siebel, and specify the server where SiteScope is installed.

➤ In the Element Properties tab's Product field, choose **Siebel Web Server**.

Adding a Monitored Machine to the Controller

1. Click the Siebel Web Server graph in the graph tree, and drag it into the right pane of the Run view.
2. Right-click the graph and select **Add Measurements**, or choose **Monitors** > **Add Measurements**. The Siebel Web Server dialog box opens.
3. In the Monitored Server Machines section, click **Add**. The Add Machine dialog box opens.

4 In the **Monitored Machine Information** section, enter the server name or IP address of the machine you want to monitor, and select the platform on which the machine runs.

In the **SiteScope Server Information** section, enter the SiteScope machine name, port (default: 8888), and version.

Click **OK**. The Siebel Web Server dialog box is redisplayed.

5 In the **Resource Measurements** section of the Siebel Web Server dialog box, click **Add**.

Continue with Configuring the Siebel Web Server Monitor below.

Adding a Monitored Machine to the Tuning Console

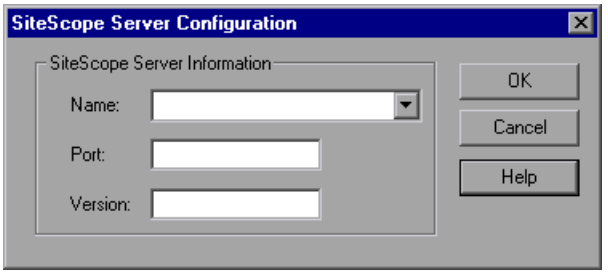
1 Click **Monitors** to open the Monitors Configuration dialog box.

2From the Server list, choose the ERP/CRM server running Siebel Web Server.

3 Click **Add Monitor** and select **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the ERP-CRM Server category, select **Siebel Web Server**, and then click **Add**.

If the SiteScope server has not been configured yet, the SiteScope Server Configuration dialog box opens.

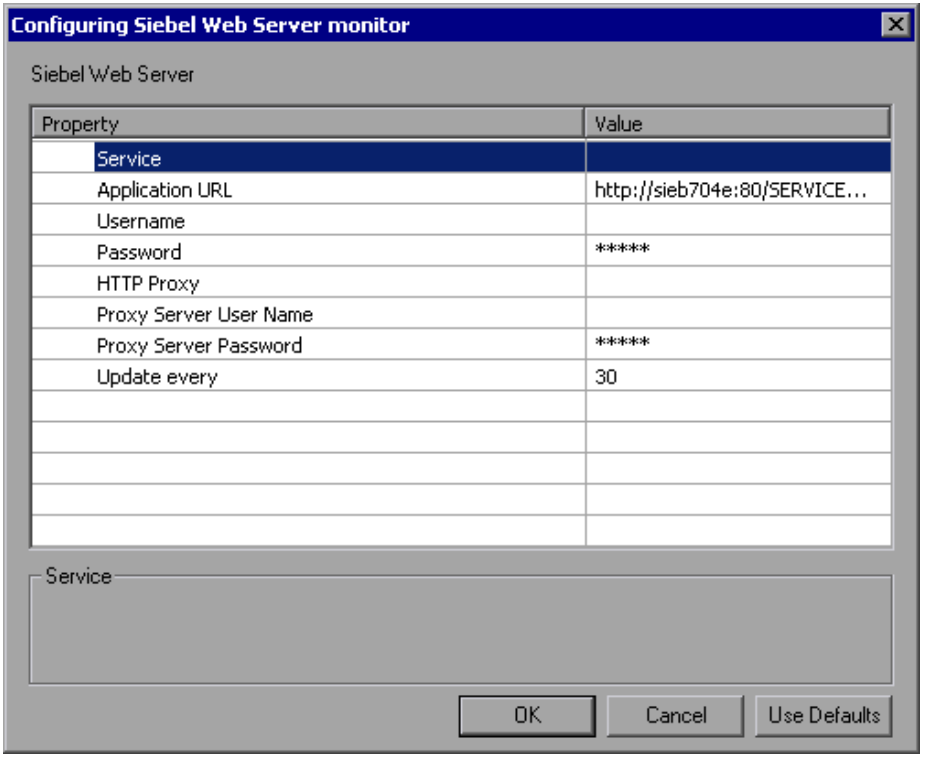


5 Enter the SiteScope server information as described in “Understanding the SiteScope Server Configuration Dialog Box” on page 43 and click **OK**.

Continue with Configuring the Siebel Web Server Monitor below.

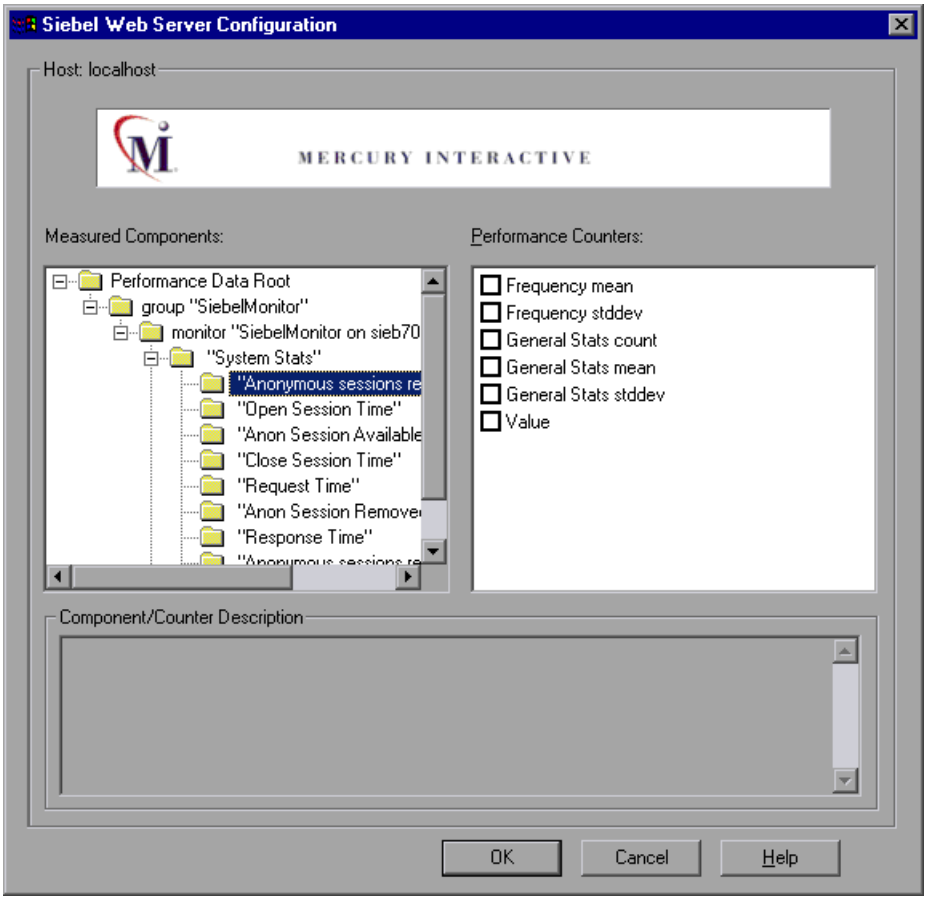
Configuring the Siebel Web Server Monitor

1 The Configuring Siebel Web Server Monitor dialog box opens.



Enter the Siebel Web Server information, as described in “Understanding the Configuring Siebel Web Server Monitor Dialog Box” on page 329.

2 Click **OK**. The Siebel Web Server Configuration dialog box opens, displaying the available measurements.



3 Browse the Measured Components tree, and select performance counters, as described in "Understanding the Siebel Web Server Configuration Dialog Box" on page 330, and then click **OK**.

For a description of the available measurements, see "Siebel Web Server Performance Counters" on page 330.

4 In the Controller, click **OK** in the Siebel Web Server Configuration dialog box, and in the Siebel Web Server dialog box, to activate the monitor.

5 In the Tuning Console, click **OK** in the Select Measurements to Monitor dialog box, and in the Monitors Configuration dialog box, to activate the monitor.

Note: The minimum recommended SiteScope monitor refresh rate is 30 seconds. If you choose a lower refresh rate, the Controller or Tuning Console may not get all the data in time.

Understanding the Configuring Siebel Web Server Monitor Dialog Box

The Configuring Siebel Web Server Monitor dialog box enables you to configure the Siebel Web Server monitor.

Service: Enter the name of a valid Siebel virtual directory (for example, callcenter or sales), one whose URL is an entry point defined in the Siebel SWSE configuration file (eapps.cfg).

Application URL: Displays the default URL of the web plug-in server stats page for the application you want to monitor. For example, http://siebelsrv/<service>/_stats.swe. If the Siebel web server is configured to support verbose mode, and you want to include information on "Locks" and "Current Operations Processing," you can add verbose=high, for example: http://siebelsrv/<service>/_stats.swe?verbose=high.

Username: Enter the user name to access the web server stats page.

Password: Enter the password to accessing the web server stats page.

HTTP Proxy: If you are using a proxy to access the Siebel server, enter the proxy server to use including the port (for example, **proxy.sitescope.com:8080**).

Proxy Server User Name: If the proxy server requires authorization, enter the proxy user name.

Proxy Server Password: If the proxy server requires authorization, enter the proxy password.

Update every: Enter how frequently the monitor should check the Database server.

Understanding the Siebel Web Server Configuration Dialog Box

The Siebel Web Server Configuration dialog box lets you select the Siebel Web Server resources to monitor.

Host: Displays the name of the monitored machine.

Measured Components: Displays the available components. Browse the tree and select the component you want to monitor. A description of the highlighted component appears in the Component/Counter Description box.

Performance Counters: Check the required performance counters. A description of the selected counter appears in the Component/Counter Description box.

Component/Counter Description: Displays a description of the selected component or counter.

Siebel Web Server Performance Counters

The following table shows the default counters that can be measured:

System Statistics

Measurement	Description
Anonymous sessions requested from the pool	The number of anonymous sessions requested from the pool.
Open Session Time	The time users experience logging on to the system.
Anon Session Available	The number of anonymous sessions available in the pool.

Measurement	Description
Close Session Time	The time users experience logging off the system.
Request Time	The time taken to process the user request.
Anon Session Removed	The number of anonymous sessions removed from the pool.
Response Time	The time taken to respond to a user request.
Anonymous sessions returns to the pool	The number of anonymous sessions returned to the pool.

Application Statistics

Measurement	Description
Session Lifespan	The duration during which a client session ran in the Siebel system.

Siebel Server Manager Resource Monitoring

The Siebel Server Manager monitor displays statistics about the resource usage of a Siebel Server Manager during the scenario or session step run.

To monitor Siebel Server Manager performance, you first install the Siebel Server Manager client on the SiteScope machine. You must then enable the Siebel Server Manager online monitor (from the Controller or Tuning Console) before executing the scenario or session step, and indicate which statistics and measurements you want to monitor. You select these counters using the Siebel Server Manager dialog box.

Setting up the Monitoring Environment

Before you set up the monitor, perform the following:

- Ensure that SiteScope (the application that is used to monitor the Siebel Server Manager) has been installed on a server. You can install SiteScope on the Controller or Tuning Console machine, but we recommend installing it on a dedicated server.
- On the machine where SiteScope is installed, configure SiteScope to monitor the required Siebel Server Manager machine. For more information, see "Configuring the Siebel Server Manager Client on the SiteScope Machine" on page 332.
- Verify that SiteScope is collecting the required data from the servers it is monitoring. From the SiteScope Panel, select the monitor group polling the Siebel Server Manager machine, and check that the monitor displays a list of server measurements in the Status column.

Configuring the Siebel Server Manager Client on the SiteScope Machine

1. Verify connectivity to the Siebel SWSE page by opening the following URL from the machine where Sitescope is installed:

   http://<your_siebel_server>/callcenter/_stats.swe

2. In the SiteScope main panel, select an existing group or create a new group.
3. In the **Add to Group** section, click **Monitor** and select **Siebel Server Manager** from the list of monitors.

4 Click **Choose server**, and enter the name of the Siebel Server in the **Application server** field, the Enterprise Server in the **Enterprise server** field, and the Gateway Server in the **Gateway server** field.

5 If necessary, enter the user name and password for the Siebel Server Manager client.

6 Enter the path to the Siebel Server Manager in the **Path to Script** field.

7 Click **Browse**, select your desired counters, and then click **Choose** Counters.

8 In the **Title** field, enter a name for the monitor.

Note: When you assign a name to a monitor, include the server name in the monitor name. This avoids any confusion as to which host the monitor belongs. For example, SiebelManager on sieb07.

Do not use "\" in the Title field.

---

9 Click **Add Monitor**.

Troubleshooting the Siebel Server Manager Monitor

The Siebel Server Manager monitor uses a Siebel command line utility (srvrmgr) to gather it's statistics. If you are having trouble getting the Siebel Server Manager monitor to work, run this command from the Siebel Server Manager client:

```
srvrmgr /s <server> /g <gateway> /e <enterprise> /u <user> /p <pw>
```

If this command works from the command line, but SiteScope has trouble executing the command, open **/sitescope/templates.applications/commandline.siebel,** and verify that you can run the following command from the command line:

```
CONNECT_COMMAND:$PATH$/srvrmgr /g $GATEWAY$ /e $ENTERPRISE$
/s $SERVERS$ /u $USERNAME$ /p $PASSWORD$
```

Note: Note that on a Windows 2000 Advanced Server platform this command must be changed to:

```
CONNECT_COMMAND:$PATH$\srvrmgr.exe /g $GATEWAY$ /e
$ENTERPRISE$ /s $SERVERS$ /u $USERNAME$ /p $PASSWORD$
```

Adding a Monitored Machine to the Controller

1. Click the Siebel Server Manager graph in the graph tree, and drag it into the right pane of the Run view.
2. Right-click the graph and select **Add Measurements**, or choose **Monitors** > **Add Measurements**. The Siebel Server Manager dialog opens.
3. In the **Monitored Server Machines** section, click **Add.** The Add Machine dialog box opens.
4. Enter the server name or IP address of the machine you want to monitor, select the platform on which the machine runs, and click **OK**. The Siebel Sever Manager dialog is redisplayed.
5. In the **Resource Measurements** section of the Siebel Server Manager dialog box, click **Add.**

   Continue with Configuring the Siebel Server Manager Monitor below.

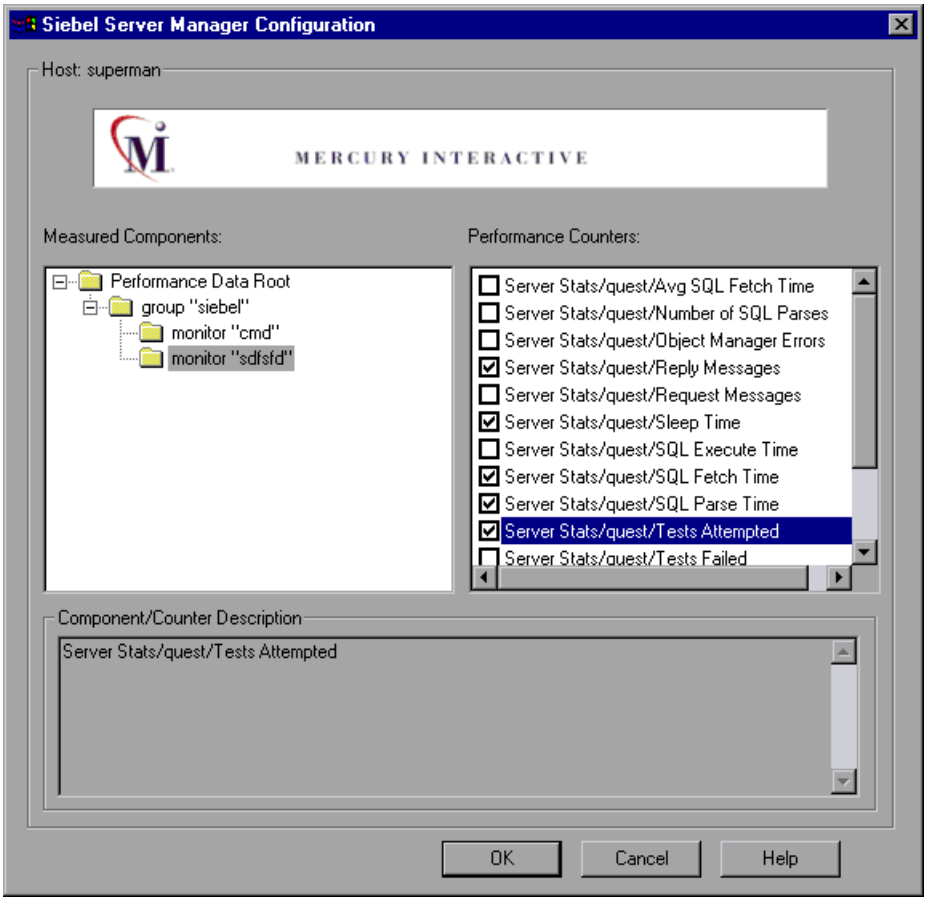
Adding a Monitored Machine to the Tuning Console

1. Click **Monitors** to open the Monitors Configuration dialog box.
2. From the Server list, choose the ERP/CRM server running Siebel Web Server.
3. Click **Add Monitor** and select **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.
4. In the left section of the dialog box, expand the ERP-CRM Server category, select **Siebel Server Manager**, and then click **Add**.

   Continue with Configuring the Siebel Server Manager Monitor below.

Configuring the Siebel Server Manager Monitor

1 The Siebel Server Manager Configuration dialog box opens, displaying the available measurements.



2 Browse the Measured Components tree, select the required performance counters, as described in "Understanding the Siebel Server Manager Configuration Dialog Box" on page 336, and then click **OK**.

For a description of the available measurements, see "Siebel Server Manager Performance Counters" on page 337.

3 In the Controller, click **OK** in the Siebel Server Manager Configuration dialog box, and in the Siebel Server Manager dialog box to activate the monitor.

4 In the Tuning Console, click **OK** in the Select Measurements to Monitor dialog box, and in the Monitors Configuration dialog box, to activate the monitor.

Note: The minimum recommended SiteScope monitor refresh rate is 30 seconds. If you choose a lower refresh rate, the Controller or Tuning Console may not get all the data in time.

Limitations

The SiteScope monitor has the following limitations when it retrieves information from a Siebel Server Manager:

- The component counters (for example, Average SQL Time for <component>) are updated with the aggregated Siebel data only at the end of a user session.
- The SiteScope monitor consumes very high CPU resources (approximately 40%).

Understanding the Siebel Server Manager Configuration Dialog Box

The Siebel Server Manager Configuration dialog box enables you to select the Siebel Server Manager resources to monitor.

Host: Displays the name of the monitored machine.

Measured Components: Displays the available components. Browse the tree and select the component you want to monitor. A description of the highlighted component appears in the Component/Counter Description box.

Performance Counters: Select performance counters.

Component/Counter Description: Displays a description of the selected component or counter.

Siebel Server Manager Performance Counters

The following table shows the default counters that can be measured:

Measurement	Description
Average Connect Time	The average connection time.
Average Reply Size	The average size of a user reply.
Average Request Size	The average size of a user request.
Average Requests Per Session	The average number of user requests per session.
Average Response Time	The average amount of time that it takes the server to respond to a request.
Average Think Time	The average amount of think time taken to respond to a request.
Avg SQL Execute Time	The average SQL execute time.
Avg SQL Fetch Time	The average SQL fetch time.
Avg SQL Parse Time	The average SQL parse time.
CPU Time	The CPU time used in the work process.
Elapsed Time	The total amount of elapsed time.
Num of DBConn Retries	The number of database connection retries.
Num of DLRbk Retries	The number of DLRbk retries.
Num of Exhausted Retries	The total number of retries that expired.
Number of SQL Executes	The total number of SQL executes.

Measurement	Description
Number of SQL Fetches	The total number of SQL fetches.
Number of SQL Parses	The total number of SQL parses.
Number of Sleeps	The number of sleeps.
Object Manager Errors	The total number of object manager errors.
Reply Messages	The total number of reply messages.
Request Messages	The total number of request messages.
SQL Execute Time	The total SQL execute time.
SQL Fetch Time	The total SQL fetch time.
SQL Parse Time	The total SQL parse time.
Sleep Time	The total sleep time.
Tests Attempted	The number of tests attempted.
Tests Failed	The number of tests that failed.
Tests Successful	The number of tests that were successful.
Total Reply Size	The total reply size, measured in bytes.
Total Request Size	The total request size, measured in bytes.
Total Response Time	The total response time.
Total Tasks	The total number of tasks.
Total Think Time	The total think time.

PeopleSoft (Tuxedo) Resource Monitoring

The PeopleSoft (Tuxedo) monitor displays statistics about the resource usage of a PeopleSoft (Tuxedo) server during the scenario or session step run.

To obtain data for this graph, you need to enable the PeopleSoft (Tuxedo) online monitor (from the Controller or Tuning Console) and select the default measurements you want to display, before running the scenario or session step. You select these counters using the PeopleSoft (Tuxedo) dialog box.

Setting up the Monitoring Environment

If Tuxedo 7.1 or later is installed, you can monitor more than one PeopleSoft (Tuxedo) application server at a time. If Tuxedo 6.5 or earlier is installed, you can monitor only one PeopleSoft (Tuxedo) application server at a time.

Before you set up the monitor, perform the following:

➤ Ensure that a Tuxedo workstation client (not a native client) is installed on the Controller or Tuning Console machine. Use a Tuxedo 6.x client if a Tuxedo 6.x server is used, and Tuxedo 7.1 or later client if a Tuxedo 7.1 or later server is used. If you use a Tuxedo 6.5 or earlier server, you can still use a Tuxedo 7.1 or later client in order to monitor it, provided that you set the WSINTOPPRE71 environment variable to “yes”.

Note: A Tuxedo workstation client communicates with the application server over the network, and is not required to run the Tuxedo application server on the same machine. A native client can only communicate with the Tuxedo application server if it is part of the relevant Tuxedo domain.

➤ Define the Tuxedo environment variables on the Controller or Tuning Console machine—set the TUXDIR variable to the Tuxedo installation directory (for example, V:\environ\32\Tuxedo8.0), and add the Tuxedo **bin** directory to the PATH variable.

➤ Ensure that the workstation listener (WSL) process is running. This enables the application server to accept requests from workstation clients. Note that the address and port number used to connect to the application server must match those dedicated to the WSL process.

Note: For information on configuring the WSL, refer to the BEA Tuxedo Web site (http://edocs.beasys.com/tuxedo/tux81/rf5/rf5101.htm#1534543).

Adding a Monitored Machine to the Controller

1. Click the PeopleSoft (Tuxedo) graph in the graph tree, and drag it into the right pane of the **Run** view.
2. Right-click the graph and select **Add Measurements**, or choose **Monitors** > **Add Online Measurement**. The PeopleSoft (Tuxedo) dialog box opens.
3. In the **Monitored Server Machines** section, click **Add.** The Add Machine dialog box opens.
4. Enter the server name or IP address of the Tuxedo machine you want to monitor, select the platform on which the machine runs, and click **OK**.
5. In the **Resource Measurements** section of the PeopleSoft (Tuxedo) dialog box, click **Add**.

   Continue with Configuring the Siebel Server Manager Monitor below.

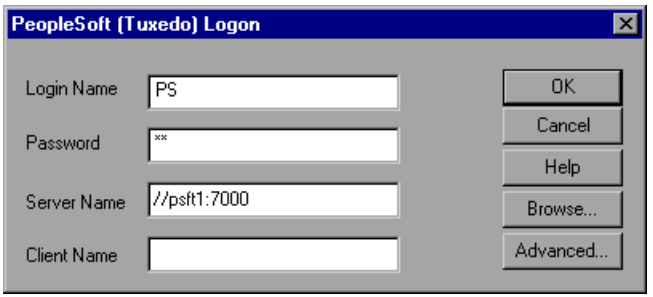
Adding a Monitored Machine to the Tuning Console

1. Click **Monitors** to open the Monitors Configuration dialog box.
2. From the Server list, choose the server running PeopleSoft.
3. Click **Add Monitor** and select **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.
4. In the left section of the dialog box, expand the ERP/CRM Server category, select **PeopleSoft (Tuxedo)**, and click **Add**.

   Continue with Configuring the Siebel Server Manager Monitor below.

Configuring the PeopleSoft (Tuxedo) Monitor

1 The PeopleSoft (Tuxedo) Logon dialog box opens.



Enter the following information:

Login Name: Enter the login name used to access the PeopleSoft (Tuxedo) server. The default is PS.

Password: Enter the password for the login name. The default is PS.

Server Name: Enter the name of the PeopleSoft (Tuxedo) server in the format of //<machine name>:<port number>. The default port is 7000. Alternatively, you can specify the IP address or the hexadecimal format used by old versions of Tuxedo. Note that you cannot use quotation marks.

Client: Enter the name of the client machine.

Note: If you are using PeopleSoft 7.x, you can determine the logon information from the **Logon** section of the **tpinit.ini** file in the recorded script's directory.

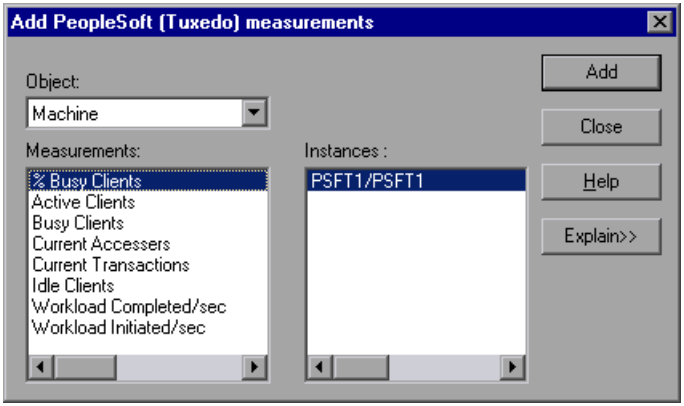
---

2 To authenticate the PeopleSoft (Tuxedo) monitor, click **Advanced**, and enter the authentication data as a hexadecimal string (beginning with "0x") in the **Data** field. The authentication data value can be obtained from the **tpinit.ini** file of an existing Tuxedo script.

Note: If you are using Tuxedo 6.5 or below, the monitor can only connect to one application server during a Controller scenario. Once it connects to an application server, that server is the only one used by the monitor until the Controller is closed. This applies even when all of the counters are deleted from the monitor.

---

3 Click **OK**. The Add PeopleSoft (Tuxedo) Measurements dialog box opens.



4 For each measurement, select an object, measurement, and instance, and then click **Add**, as described in "Understanding the Add PeopleSoft (Tuxedo) Measurements Dialog Box" on page 343.

For a description of the available measurements, see "PeopleSoft (Tuxedo) Performance Counters" on page 344.

5 When you have finished adding measurements, click **Close**.

6 In the Controller, click **OK** in the PeopleSoft (Tuxedo) dialog box to activate the monitor.

7 In the Tuning Console, click **OK** in the Select Measurements to Monitor dialog box, and in the Monitors Configuration dialog box, to activate the monitor.

Understanding the Add PeopleSoft (Tuxedo) Measurements Dialog Box

The Add PeopleSoft (Tuxedo) Measurements dialog box enables you to select the PeopleSoft (Tuxedo) resources to monitor.

Object: Select the object being monitored on the specified machine.

Measurements: Select a resource measurement to monitor. Select multiple measurements using the CTRL key. For an explanation of each measurement, click **Explain**.

Instances: If multiple instances of the selected measurement are running, select one or more instances to monitor for the selected measurement.

Explain: Displays a description of the selected measurement.

PeopleSoft (Tuxedo) Performance Counters

The following table describes the default counters that can be measured. It is recommended to pay particular attention to the following measurements: % Busy Clients, Active Clients, Busy Clients, Idle Clients, and all the queue counters for the APPQ/PSAPPSRV queue.

Monitor	Measurements
Machine	**% Busy Clients** - The per cent of active clients currently logged in to the Tuxedo application server which are waiting for a response from the application server.
	Active Clients - The total number of active clients currently logged in to the Tuxedo application server.
	Busy Clients - The total number of active clients currently logged in to the Tuxedo application server that are waiting for a response from the application server.
	Current Accessers - Number of clients and servers currently accessing the application either directly on this machine or through a workstation handler on this machine.
	Current Transactions - Number of in use transaction table entries on this machine.
	Idle Clients - The total number of active clients currently logged in to the Tuxedo application server that are not waiting for a response from the application server.
	Workload Completed/second - The total workload on all the servers for the machine that was completed, per unit time.
	Workload Initiated/second - The total workload on all the servers for the machine that was initiated, per unit time.

Monitor	Measurements
Queue	**% Busy Servers** - The per cent of active servers currently handling Tuxedo requests.
	Active Servers - The total number of active servers either handling or waiting to handle Tuxedo requests.
	Busy Servers - The total number of active servers currently busy handling Tuxedo requests.
	Idle Servers - The total number of active servers currently waiting to handle Tuxedo requests.
	Number Queued - The total number of messages which have been placed on the queue.
Server	**Requests/second** - How many server requests were handled per second
	Workload/second -The workload is a weighted measure of the server requests. Some requests could have a different weight than others. By default, the workload is always 50 times the number of requests.
Workstation Handler (WSH)	**Bytes Received/sec** - The total number of bytes received by the workstation handler, per second.
	Bytes Sent/sec - The total number of bytes sent back to the clients by the workstation handler, per second.
	Messages Received/sec - The number of messages received by the workstation handler, per second.
	Messages Sent/sec - The number of messages sent back to the clients by the workstation handler, per second.
	Number of Queue Blocks/sec - The number of times the queue for the workstation handler blocked, per second. This gives an idea of how often the workstation handler was overloaded.

PeopleSoft Ping Monitoring

The PeopleSoft Ping script monitor displays the data that a user sees on the PeopleSoft Ping page in the **PeopleTools\utilities** folder of the PeopleSoft 8.42 installation. This data is sampled by using a canned lightweight monitoring script, and displayed by the Mercury Tuning Console in the PeopleSoft Ping graph.

Note: The PeopleSoft Ping default sampling rate is every five seconds. To change the default sampling rate, select **After the previous iteration ends** in the **General: Pacing** section of the run-time settings, and enter a sampling rate.

To obtain data on the PeopleSoft server:

1 Define the PeopleSoft 8.42 server in the System Topology window.

For more information on defining a system topology, refer to "Creating a Topology" in the *Mercury Tuning Module Console User's Guide*.

2 Add a step to your session.

In the **Select Scripts from Template** section, choose **PeopleSoft Ping** from the **PeopleSoft** canned scripts.

In the **Execute script against the following servers** section, select the PeopleSoft servers against which you want the script to run.

In the **Script Parameters** section, specify the server, username, and password where PeopleSoft is running.

3 Add the scripts (that you want to run) to the step that already contains the PeopleSoft Ping script.

In the **Add Step** dialog box, click the **Add...** button to the right of the Custom Scripts pane, add the PeopleSoft scripts that you want to use, and click **OK**. The script names are added to the Custom Scripts pane.

4 Assign one Vuser to run in this script. If you set more than one Vuser, you create unnecessary load on the server.

In the **Add Step** dialog box, click the **Add and Configure** button. The new step is displayed in the session tree using the **ps84_ping_wrapper** script, and the Step Configuration dialog box opens. Enter "1" in the Number of Vusers box, and click **Close**.

5 Execute the session step against a PeopleSoft 8.4 server.

The x-axis of the graph represents the time that has elapsed since the start of the session step run. The y-axis represents the resource usage.

12

J2EE Performance Monitoring

The J2EE performance monitor provides complete insight into the J2EE components on the application server (Servlets, JSPs, EJBs, JNDI, JDBC, and DB SQL calls).

This chapter describes:

- About J2EE Performance Monitoring
- Setting up the Monitoring Environment
- Configuring the J2EE Monitor on the Client Machine
- Examples of Modifying Application Server Configurations
- Troubleshooting the J2EE Monitor

Note: For information on setting up the J2EE Transaction Breakdown Diagnostics monitor, refer to the "J2EE Diagnostics Module" chapter in the *LoadRunner Controller User's Guide*.

About J2EE Performance Monitoring

The J2EE monitor provides the following information for each J2EE component:

- Average response time per method/query
- Number of method calls per second

With such coverage of the J2EE architecture, users can get an overview of the entire activity within the system. They can very easily correlate the end user response time with the Web server activity (Servlets and JSPs data), application server activity (JNDI and EJB's), and back-end activity of database requests (JDBC methods and SQL queries).

The J2EE Monitor allows LoadRunner users to analyze J2EE component metrics during a scenario run by using an agent which is installed on the application server to collect information on the J2EE components. These measurements are sent from the application server back to the LoadRunner Controller through a Web server contained in the J2EE monitor. The J2EE Monitor supports the leading applications servers, such as: IBM WebSphere, BEA WebLogic, Oracle 9iAS and JBoss. For information about the supported application servers, refer to the "Support Matrix" on page 351.

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

Setting up the Monitoring Environment

To monitor J2EE objects, you must first install and activate the J2EE monitor on the application server machine. After installing the J2EE monitor, you can configure the initial settings (hooking mechanism, operation mode, JDBC, and EJB information retrieval) on the J2EE monitor application server.

This section describes the following installation and configuration topics:

- Installing the J2EE Monitor on the Application Server
- Initial J2EE Monitor Configuration Settings
- Configuring JDBC Information Retrieval
- Configuring the EJB Information Retrieval

Installing the J2EE Monitor on the Application Server

To monitor J2EE objects, you must first install and activate the J2EE monitor on the application server machine. You then configure the J2EE monitor on the client machine by selecting the counters you want the monitor to measure.

You can monitor Java 2 Platform, Enterprise Edition (J2EE) objects on a WebLogic, WebSphere, Oracle 9iAS, or JBoss application server during a scenario or session step run using the J2EE performance monitor.

Support Matrix

Application Server	Version	Platform
WebLogic	4.x; 5.x; 6.x; 7.0; 8.1	Windows; Solaris; AIX
WebSphere	3.x; 4.x	Windows; Solaris; AIX
Oracle 9iAS	1.0.2.2	Windows; Solaris; AIX
JBoss	2.4.x; 3.04	Windows; Solaris; AIX

To install the J2EE monitor on the application server:

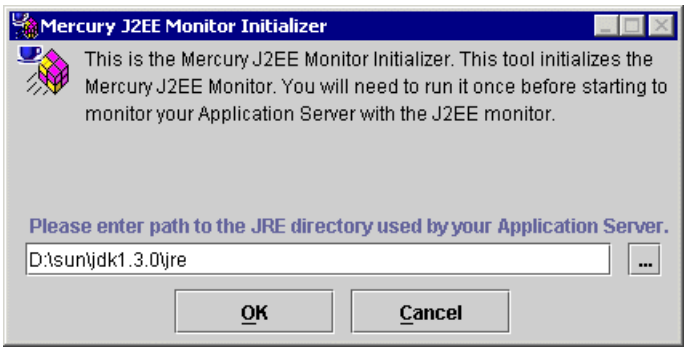
1 Create a home directory on the application server machine—for example, J2EEMonitor, and unzip the installation file **<LoadRunner CD>\Add-ins\J2EE\jmonitor_<platform>.jar** file into that directory.

If you do not have WinZip to unzip the installation file, use the following command line to extract the installation file:

<JDK>\bin\jar.exe -xf <installation file>

UNIX scripts extracted from the jar file may lose their execute permissions. To fix this, open the J2EEMonitor Home Directory, and change the permissions using the command line: chmod +x *.sh.

2 Open the Mercury J2EE Monitor Initializer by double-clicking **<J2EEMonitor Home Directory>\classes\sipatool.jar**.



On a UNIX platform, or if the **.jar** extension in your system is not associated with the Java runtime environment, run **sipatool.jar** as follows:

From the **<J2EEMonitor Home Directory>\classes** directory, type java -jar sipatool.jar.

Note: If the Mercury J2EE Monitor Initializer is not displayed, run **sipatool.jar** as follows:

From the **<J2EEMonitor Home Directory>\classes** directory, type <Java path>\ -jar sipatool.jar -nogui. Use the full path of the Java process of the application server JDK.

---

3 In the Mercury J2EE Monitor Initializer, enter the path to the application server Java home directory, and click **OK** to run the tool.

4 Add -Xbootclasspath/p:<J2EEMonitor Home Directory>\classes\boot to the application server command line arguments.

Refer to "Examples of Modifying Application Server Configurations", on page 360 to see syntax for WebLogic, WebSphere, Oracle 9iAS, or JBoss application servers.

Initial J2EE Monitor Configuration Settings

The J2EE monitor application server installation configured the hooking mechanism, operation mode, JDBC, and EJB information retrieval.

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

Operation mode: The J2EE monitor uses the Auto Discovery operating mode. In this mode, the system automatically discovers the J2EE components (Servlet, JSP, JNDI, EJB and JDBC) that actually participate in the business process.

JDBC information retrieval: The JDBC information retrieval setting determines which data to return from the JDBC call. By default, the J2EE monitor aggregates the measured data according to the JDBC operation, for example: SELECT,UPDATE,CREATE. To modify this configuration, refer to "Configuring JDBC Information Retrieval" on page 354.

EJB information retrieval: The EJB information retrieval setting determines which data to return from the EJB call. By default, the J2EE monitor is not configured to measure container methods, (e.g., ejbPassivate(), ejbCreate()).

To modify this configuration, refer to "Configuring the EJB Information Retrieval" on page 354.

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

Configuring JDBC Information Retrieval

To configure JDBC information retrieval:

1 Open **<J2EEMonitor Home Directory>\dat\monitor.properties**.

2 In the property monitor.jdbc.mode, enter one of the following:

- "1" to measure the JDBC the method calls, like any other (non-JDBC) measured method calls.
- "2" to aggregate the measured data according to the JDBC operation, for example: SELECT,UPDATE,CREATE.
- "3" to aggregate the measured data according to specific SQL statement (including the operation, the table(s) it acted on, and other parameters of this statement).

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

Configuring the EJB Information Retrieval

To configure EJB information retrieval to include container methods:

1 Open **<J2EEMonitor Home Directory>\dat\java_monitor.ini**.

2 In the EJB_CONFIG section of the file, change the hook_files=auto_detect setting to the following:

```
hook_files=auto_detect_container
```

Configuring the J2EE Monitor on the Client Machine

To monitor J2EE performance, you must select the counters you want the J2EE monitor to measure. You select these counters using the Controller or Tuning Console's J2EE Monitor Configuration dialog box.

Before configuring the J2EE monitor:

In Auto Discovery mode (the J2EE monitor's default operating mode), the system discovers which methods of the components (Servlet, JSP, JNDI, EJB and JDBC) are participating in your business process and measures those objects only.

To start the Auto Discovery process, start the application server, and run the Vuser script that you intend to use in your load test against the application server. This provides the Controller or Tuning Console with a list of measurements that will be available for monitoring.

Note: The next time you run the same script, you don't need to run a Vuser before selecting the methods and counters you want to monitor.

Adding a Monitored Machine to the Controller

1. Click the J2EE graph in the graph tree, and drag it into the right pane of the Run view.
2. Right-click the graph and select **Add Measurements**, or choose **Monitors** > **Add Measurements**. The J2EE dialog box opens.
3. In the **Monitored Server Machines** section, click **Add.** The Add Machine dialog box opens.
4. Enter the server name or IP address of the machine you want to monitor. Select the platform on which the machine runs, and click **OK**.
5. In the **Resource Measurements** section of the J2EE dialog box, click **Add**.

   Continue with Configuring the J2EE Monitor below.

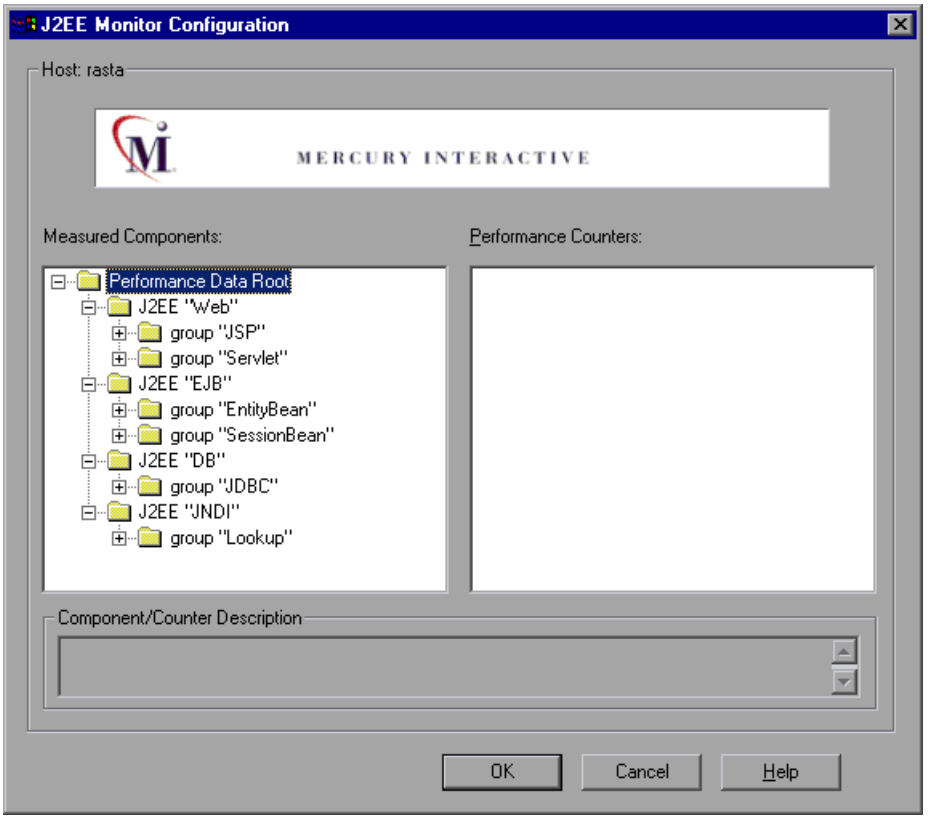
Adding a Monitored Machine to the Tuning Console

1. Click **Monitors** to open the Monitors Configuration dialog box.
2. Select the server whose monitors you want to configure from the Server list box.
3. Click **Add Monitor** and select **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.
4. In the left section of the dialog box, expand the Java Technology category, select **J2EE**, and then click **Add**.

   Continue with Configuring the J2EE Monitor below.

Configuring the J2EE Monitor

1 The J2EE Monitor Configuration dialog box opens, displaying the available J2EE counters.



Browse the Measured Components tree, and select the methods and counters you want to monitor, as described in "Understanding the J2EE configuration dialog box" on page 359.

For a description of the available measurements, see "J2EE Performance Counters" on page 359.

Note: The size of a measurement name that can be displayed in the Analysis is limited to 255 characters. If a measurement name exceeds this limit, the counter name is truncated, and given a unique ID (UID). If you monitor different events or make cross result graphs on the same counter, the UID will stay the same.

The measurement name is truncated as follows:
standard prefix/**counter truncated name**<**UID**>/monitored event

For example:
/DB/JDBC/weblogic.jdbc.rmi.SerialPreparedStatement/int executeUpdate()/**INSERT INTO orders (orderid _ userid _ orderdate _ shipaddr1 _ shipaddr2 _ shipcity _ shipstate _ shipzip _ shipcountry _ billaddr1 _ billaddr2 _ b** <**1**> /Average Response Time

The full measurement name appears in the Measurement Description box.

---

2. In the Controller, click **OK** in the J2EE Monitor Configuration dialog box, and in the J2EE dialog box, to activate the monitor.
3. In the Tuning Console, click **OK** in the J2EE Monitor Configuration dialog box, and in the Select Measurements to Monitor dialog box, to activate the monitor.

Understanding the J2EE configuration dialog box

The J2EE dialog box lets you select the J2EE measurements to monitor.

Host: Displays the name of the monitored machine.

Measured Components: Displays the available components. Browse the tree and select the component you want to monitor. A description of the highlighted component appears in the Component/Counter Description box.

Performance Counters: Check the required performance counters. A description of the selected counter appears in the Component/Counter Description box.

Component/Counter Description: Displays a description of the selected component or counter.

J2EE Performance Counters

The following counters can be monitored for each method:

Measurement	Description
Average Response Time	The average response time, in milliseconds, of the J2EE object being monitored.
Method Calls per Second	The number of J2EE object method calls per second.

Examples of Modifying Application Server Configurations

When you installed Mercury's J2EE monitor files on your application server, you already configured it to run with J2EE monitor support. This section provides examples modifying the configuration of the following application servers:

- WebLogic - Version 4.x-5.x
- WebLogic - Version 6.x
- WebLogic - Version 7.x
- WebLogic - Version 8.1
- WebSphere Server - Version 3.x
- WebSphere Server - Version 4.x
- Oracle 9iAS Server
- JBoss 2.4.x-3.04 Server

Note: It is important to set the environment variables in the order in which they appear below.

If the J2EE monitor is installed on your application server machine, perform the following file modifications on the original application server batch file, that is, the one supplied with the application server before the J2EE installation modifications. Note that the WebSphere application server does not use a batch file. Follow the instructions for the relevant WebSphere version as described in the following sections.

WebLogic Server

Each WebLogic server is configured differently.

WebLogic - Version 4.x-5.x

To configure the WebLogic 4.x-5.x server:

1 Copy the **<WebLogic Home>\startWeblogic.cmd** file into **<WebLogic Home>\startWeblogicMercury.cmd** so that the file is backed up.

2 Open the **<WebLogic Home>\startWeblogicMercury.cmd** file.

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

For Windows platforms:

```
set MERC_MONITOR_HOME=<J2EEMonitor Home Directory>
set JAVA_CLASSPATH=%JAVA_CLASSPATH%;
%MERC_MONITOR_HOME%\dat;
%MERC_MONITOR_HOME%\classes\xerces.jar
```

For UNIX platforms (csh):

```
MERC_MONITOR_HOME <J2EEMonitor Home Directory>
JAVACLASSPATH=$JAVACLASSPATH:
$MERC_MONITOR_HOME/classes/xerces.jar
```

In the same section of the file, add the following parameter to the Java command line:

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

Example:

```
%JAVA_HOME%\bin\java -ms64m -mx64m -
-Xbootclasspath/p:%MERC_MONITOR_HOME%\classes\boot
-Dweblogic.class.path=%WEBLOGIC_CLASSPATH% -Dweblogic.home=.
-Djava.security.manager
-Djava.security.policy==.\weblogic.policy weblogic.Server
```

4 Run the <**WebLogic Home**>**startWeblogicMercury.cmd** file.

WebLogic - Version 6.x

To configure the WebLogic 6.x server:

1 Copy the <**WebLogic Home**>**config**\<**domain name**>**startWeblogic.cmd** file into <**WebLogic Home**>**config**\<**domain name**>**startWeblogicMercury.cmd** so that the file is backed up.

2 Open the <**WebLogic Home**>**config**\<**domain name**>**startWeblogicMercury.cmd** file.

3 Just before the java command line used to start the server add the following variables:

For Windows platforms:

```
set MERC_MONITOR_HOME=<J2EEMonitor Home Directory>
set CLASSPATH=%CLASSPATH%;%MERC_MONITOR_HOME%\dat;
%MERC_MONITOR_HOME%\classes\xerces.jar
```

For UNIX platforms:

```
MERC_MONITOR_HOME=<J2EEMonitor Home Directory>
CLASSPATH=$CLASSPATH:$MERC_MONITOR_HOME/dat:
$MERC_MONITOR_HOME/classes/xerces.jar
```

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

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

Example:

```
"%JAVA_HOME%\bin\java" -hotspot -ms64m -mx64m
-Xbootclasspath/p:%MERC_MONITOR_HOME%\classes\boot
-classpath %CLASSPATH% -Dweblogic.Domain=mydomain
-Dweblogic.Name=myserver "-Dbea.home=f:\bea" "
-Djava.security.policy==f:\bea\wlserver6.0/lib/weblogic.policy"
-Dweblogic.management.password=%WLS_PW% weblogic.Server
```

5 Run the <**WebLogic Home**>**config**\<**domain name**>\ **startWeblogicMercury.cmd** file.

WebLogic - Version 7.x

To configure the WebLogic 7.x server:

1 Copy the <**WebLogic Home**>**server****bin****startwls.cmd** file into <**WebLogic Home**>**server****bin****startwlsMercury.cmd** so that the file is backed up.

2 Open the <**WebLogic Home**>**server****bin****startwlsMercury.cmd** file.

3 Just before the java command line used to start the server add the following variables:

For Windows platforms:

```
set MERC_MONITOR_HOME=<J2EEMonitor Home Directory>
set CLASSPATH=%CLASSPATH%;%MERC_MONITOR_HOME%\dat;
%MERC_MONITOR_HOME%\classes\xerces.jar
```

For UNIX platforms:

```
MERC_MONITOR_HOME=<J2EEMonitor Home Directory>
CLASSPATH=$CLASSPATH:$MERC_MONITOR_HOME/dat:
$MERC_MONITOR_HOME/classes/xerces.jar
```

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

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

Example:

```
"%JAVA_HOME%\bin\java" -hotspot -ms64m -mx64m
-Xbootclasspath/p:%MERC_MONITOR_HOME%\classes\boot
-classpath %CLASSPATH% -Dweblogic.Domain=mydomain
-Dweblogic.Name=myserver "-Dbea.home=f:\bea" "
-Djava.security.policy==f:\bea\wlserver7.0/lib/weblogic.policy"
-Dweblogic.management.password=%WLS_PW% weblogic.Server
```

4. Copy the <**domain name**>**startWeblogic.cmd** file into <**domain name**>**startWeblogicMercury.cmd** so that the file is backed up.
5. Open the <**domain name**>**startWeblogicMercury.cmd** file.
6. Find the call to the Weblogic server. For example:
   call D:\bea\weblogic700\server\bin\startWLS.cmd
7. Change the call from **startWeblogic.cmd** to **startWeblogicMercury.cmd**, and save the file.

WebLogic - Version 8.1

To configure the WebLogic 8.1 server:

1. Copy the <**WebLogic Home**>**samples****domains**\<**domain name**>**startWeblogic.cmd** file into <**WebLogic Home**>**samples****domains**\<**domain name**>**startWeblogicMercury.cmd** so that the file is backed up.
2. Open the <**WebLogic Home**>**samples****domains**\<**domain name**>**startWeblogicMercury.cmd** file.
3. Just before the java command line used to start the server add the following variables:

   For Windows platforms:

```
set MERC_MONITOR_HOME=<J2EEMonitor Home Directory>
set CLASSPATH=%CLASSPATH%;%MERC_MONITOR_HOME%\dat;
%MERC_MONITOR_HOME%\classes\xerces.jar
```

For UNIX platforms:

```
MERC_MONITOR_HOME=<J2EEMonitor Home Directory>
CLASSPATH=$CLASSPATH:$MERC_MONITOR_HOME/dat:
$MERC_MONITOR_HOME/classes/xerces.jar
```

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

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

Example:

```
"%JAVA_HOME%\bin\java" -hotspot -ms64m -mx64m
-Xbootclasspath/p:%MERC_MONITOR_HOME%\classes\boot
-classpath %CLASSPATH% -Dweblogic.Domain=mydomain
-Dweblogic.Name=myserver "-Dbea.home=f:\bea" "
-Djava.security.policy==f:\bea\wlserver8.1/lib/weblogic.policy"
-Dweblogic.management.password=%WLS_PW% weblogic.Server
```

4. Copy the **<domain name>\startWeblogic.cmd** file into **<domain name>\startWeblogicMercury.cmd** so that the file is backed up.
5. Open the **<domain name>\startWeblogicMercury.cmd** file.
6. Find the call to the Weblogic server. For example:
   call D:\bea\weblogic8\server\bin\startWLS.cmd
7. Change the call from **startWeblogic.cmd** to **startWeblogicMercury**, and save the file.

Note: WebLogic 8.1 supports SUN JVM only. JRockit JVM is available with WebLogic 8.1 using the J2EE Diagnostics monitor only.

WebSphere Server - Version 3.x

By default, the WebSphere 3.x application server runs on Windows as an automatic service, upon machine startup. Since Mercury does not currently support LoadRunner J2EE monitoring on a WebSphere server run as an automatic service, you must change the default WebSphere server startup to **manual**.

To change the default WebSphere 3.x server startup:

1. Select **Start** > **Settings** > **Control Panel**.
2. Double-click **Services**.
3. Select **IBM WS AdminServer**, and click the **Stop** button.
4. Double-click **IBM WS AdminServer**, and select the **Manual** Startup Type.
5. Click **OK** to save your settings and close the dialog box.

You can now start the WebSphere Server from **<WebSphere Home>\AppServer\bin\debug\adminserver.bat**, instead of using the automatic service.

To add LoadRunner J2EE monitor support to the WebSphere 3.x server:

1. Make a backup copy of the **<WebSphere Home>\AppServer\bin\debug\adminserver.bat** file.
2. Open the **<WebSphere Home>\AppServer\bin\debug\adminserver.bat** file.
3. Add the following environment variables at the end of the 'SET_CP' section:

For Windows platforms:

```
set MERC_MONITOR_HOME=<J2EEMonitor Home Directory>
set CLASSPATH=%CLASSPATH%;%MERC_MONITOR_HOME%\dat;
%MERC_MONITOR_HOME%\classes\xerces.jar
```

For UNIX platforms:

```
MERC_MONITOR_HOME=<J2EEMonitor Home Directory>
CLASSPATH=$CLASSPATH:$MERC_MONITOR_HOME/dat:
$MERC_MONITOR_HOME/classes/xerces.jar
export MERC_MONITOR_HOME
```

4. Run the **adminserver.bat** file.
5. Open the WebSphere Advanced Administrative Console, and select **View** > **Topology**.
6. Expand the WebSphere Administrative Domain tree by selecting **<server machine name>** > **Default Server**.
7. Select the **General** tab in the Application Server:Default Server window.
8. Add -Xbootclasspath/p:%MERC_MONITOR_HOME%\classes\boot to the command line Arguments box, and click **Apply**.

   If you are working with a WebSphere 3.0 Server with JDK1.1.7 IBM, double-click on **Environment**. Type _CLASSLOAD_HOOK in the Variable Name box, and jdkhook in the Value box. Click the **Add**, **OK**, and **Apply** buttons.
9. For Windows 2000/NT or Solaris, open the Environment Editor dialog box from the General tab, and add the following variables to the Environment box:

   For Windows 2000/NT:

   ```
   name=CLASSPATH
   value=<J2EEMonitor Home Directory>\dat
   ```

   For Solaris:

   ```
   name=CLASSPATH
   value=<J2EEMonitor Home Directory>/dat
   ```

   Click **OK** to close the Environment Editor dialog box.
10. Close the WebSphere Advanced Administrative Console.
11. Close and restart the **adminserver.bat** file.

WebSphere Server - Version 4.x

You can start the WebSphere 4.x server using the startServerBasic.bat file or the startServer.bat file.

To configure the WebSphere 4.x server:

1. Ensure that the WebSphere Administrative Server is running, and start the Administrator Console.
2. In the WebSphere Administrative Domain tree, expand the Nodes, Hostname, and Application Servers subtrees, and select the Default Server (or the application server you wish to use with J2EE monitor).
3. Right-click the Default Server, select Properties from the menu, and click the General tab.
4. For Windows 2000/NT or Solaris, open the Environment Editor dialog box from the General tab, and add the following variables to the Environment box:

   For Windows 2000/NT:

   ```
   name=CLASSPATH
   value=<J2EEMonitor Home Directory>\dat;
   <J2EEMonitor Home Directory>\classes\xerces.jar
   ```

   For Solaris:

   ```
   name=CLASSPATH
   value=<J2EEMonitor Home Directory>/dat:
   <J2EEMonitor Home Directory>/classes/xerces.jar
   ```

   Click **OK** to close the Environment Editor dialog box.

5 Click the Advanced JVM Settings tab and select Advanced JVM settings. In the Command line arguments field, add the following value for Windows 2000/NT, Solaris, and AIX:

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

6 Click the **OK** and **Apply** buttons to save the changes for the Application server. You can now start and stop your WebSphere server using the LoadRunner J2EE Monitor.

Oracle 9iAS Server

1 Edit the file **env.cmd** (**env.sh** in UNIX platforms) as follows:

- the JAVA_HOME environment variable should point to the location of the Java Virtual machine used to run the application server.
- the DETECTOR_INS_DIR environment variable should point to the location of the monitor installation.
- the APP_SERVER_DRIVE environment variable should specify the drive hosting the application server installation (e.g., D:). Do not modify this variable on UNIX Platforms.
- the APP_SERVER_ROOT environment variable should specify the application server root directory.

2 Run the **oc4jMonitor.cmd** (**oc4jMonitor.sh** on UNIX platforms).

JBoss 2.4.x-3.04 Server

1. Make a backup copy of <**JBoss Home**>**run.bat** (**run.sh** on UNIX platforms) file into <**JBoss Home**>**runMercury.bat** (**runMercury.sh** for UNIX).
2. Open the <**JBoss Home**>**runMercury.bat** file (**runMercury.sh** on UNIX).

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

   For Windows platforms:

   ```
   set MERC_MONITOR_HOME=<J2EEMonitor Home Directory>
   set CLASSPATH=%CLASSPATH%;%MERC_MONITOR_HOME%\dat;
   %MERC_MONITOR_HOME%\classes\xerces.jar
   ```

   For UNIX platforms:

   ```
   MERC_MONITOR_HOME=<J2EEMonitor Home Directory>
   CLASSPATH=$CLASSPATH:$MERC_MONITOR_HOME/dat:
   $MERC_MONITOR_HOME/classes/xerces.jar
   ```

   In the same section of the file add the following parameter to the command line:

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

   Example:

   ```
   %JAVA_HOME%\bin\java -ms64m -mx64m  -Xbootclass-
   path/p:%MERC_MONITOR_HOME%\classes\boot
   -Dweblogic.class.path=%WEBLOGIC_CLASSPATH% -Dweblogic.home=.
   -Djava.security.manager
   -Djava.security.policy==.\weblogic.policy weblogic.Server
   ```

3. Run the <**JBoss Home**>**runMercury.bat** file (**runMercury.sh** on UNIX platforms).

Troubleshooting the J2EE Monitor

Changing the Default Port

The J2EE monitor communicates with LoadRunner, by default, using port 2004. If this port has already been taken, you can select another port as follows:

1 On the application server machine, open **<J2EEMonitor Home Directory>\dat\monitor.properties** and change the port number specified in the property: webserver.monitor.port

2 On the LoadRunner machine, open **<LoadRunner Home Directory>\dat\monitors\xmlmonitorshared.ini** and change the port number specified in section “mon_j2ee” under the key "DefaultPort".

Initialization Errors

If you are getting application server initialization errors such as: "UnsupportedClassVersionError", "NoSuchMethodError" or "NoClassDefFoundError", there might be a conflict between the JDK version specified using the Mercury J2EE Monitor Initializer, and the actual JDK version used in application server launch.

Make sure that you selected the correct JDK that is currently being used by the application server. Note that if you switched the application server to work with a different JDK, you must run the Mercury J2EE Monitor Initializer again.

13

Application Component Monitoring

Using LoadRunner's Application Component monitors, you can monitor the Microsoft COM+ server and the Microsoft .NET CLR server during a scenario run in order to isolate server performance bottlenecks.

This chapter describes:

- About Application Component Monitoring
- Microsoft COM+ Server Monitoring
- Microsoft .NET CLR Monitoring

About Application Component Monitoring

Application Component monitors provide you with information about the resource usage of the Microsoft COM+ server and Microsoft .NET CLR server during scenario execution. In order to obtain this data, you need to activate the online monitor for the server and specify which resources you want to measure before executing the scenario.

The procedures for selecting monitor measurements and configuring the monitors vary according to server type. The following sections contain specific configuration instructions for each server type.

Microsoft COM+ Server Monitoring

The Microsoft COM+ monitor is an Application Component monitor that provides performance information for the Microsoft COM+ server. Before monitoring a Microsoft COM+ server, you install the Microsoft COM+ Server Monitor Probe on the server machine. You can then specify which measurements and resources you want the Microsoft COM+ monitor to measure. You select these counters using the Controller's monitor configuration dialog box.

Setting up the Monitoring Environment

To monitor the Microsoft COM+ server performance, you must first install the Microsoft COM+ Server Monitor Probe on the server machine. You can then specify which measurements and resources you want the Microsoft COM+ monitor to measure. You select these counters using the Controller's monitor configuration dialog box.

For more information on installing the Microsoft COM+ server add-in, refer to the *LoadRunner Installation Guide*.

Configuring the Microsoft COM+ Monitor Over a Firewall

Before running the Microsoft COM+ Monitor over a firewall:

➤ Make sure that the MI Listener is installed on any machine (including the Controller machine) outside of the firewall. Refer to the *LoadRunner Installation Guide* for installation instructions.

➤ Configure the firewall agent on the server machine. For more information refer to the chapter "Running Vusers Over a Firewall" in the *LoadRunner Controller User's Guide* or *Mercury Tuning Module Console User's Guide*.

➤ Specify the correct connection string on the client machine, as described in step 4 of "Configuring the Microsoft COM+ Server Monitor" on page 375.

Configuring the Microsoft COM+ Server Monitor

You select measurements to monitor the COM+ server using the Microsoft COM+ Performance Monitors dialog box.

To configure the COM+ monitor:

1. Click the Microsoft COM+ graph in the graph tree, and drag it into the right pane of the Run view.
2. Right-click the graph and select **Add Measurements**, or choose **Monitors** > **Add Measurements**. The Microsoft COM+ dialog box opens.
3. In the **Monitored Server Machines** section, click **Add.** The Add Machine dialog box opens.

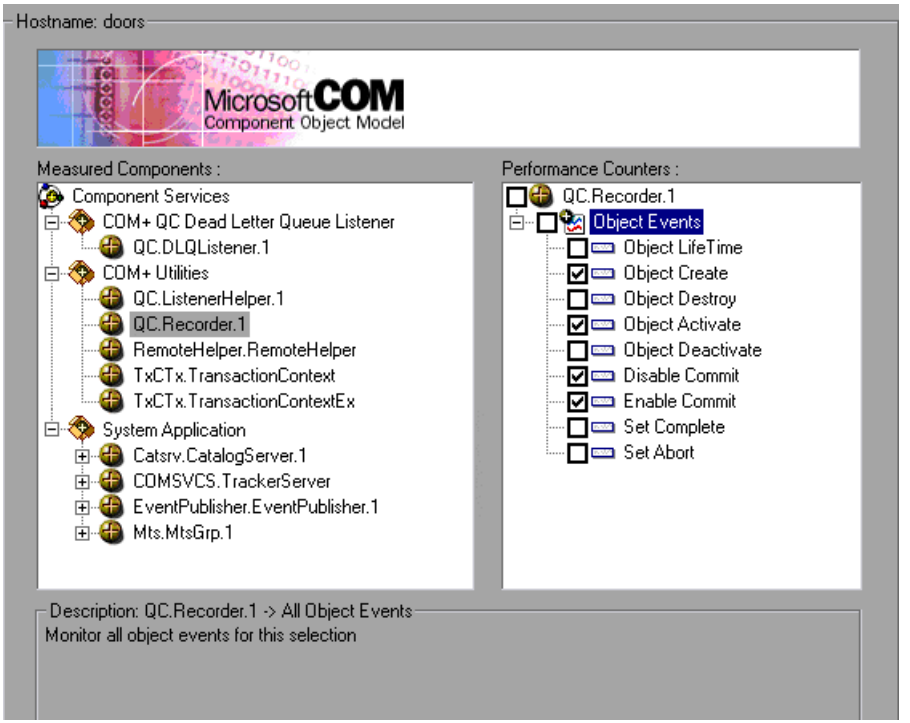
   For regular monitoring without a firewall, enter the server name or IP address of the machine you want to monitor, select the platform on which the machine runs, and click **OK**.
4. To connect to the monitor over a firewall, enter the server name or IP address of the machine you want to monitor, according to the following format, and click **OK**:

   <MI Listener machine>:<server machine key>

   where **server machine key** is the unique key that you chose when configuring the firewall Agent on the server machine.

   For example: 12.12.12.3:serverid
5. In the **Resource Measurements section** of the Microsoft COM+ dialog box, click **Add** to select the measurements that you want to monitor.

The Microsoft COM+ Performance Monitors dialog box opens displaying the available measurements.



6 Browse the Measured Components tree, and check the required performance counters, as described in "Understanding the COM+ Monitor Configuration Dialog Box," on page 377.

For a description of the available measurements, see "Microsoft COM+ Performance Counters," on page 377.

7 Click **OK** in the Microsoft COM+ Performance Monitors dialog box, and in the Microsoft COM+ dialog box, to activate the Microsoft COM+ monitor.

Note: The data sampling rate for the COM+ monitor is fixed and cannot be modified using the Controller **Tools** > **Options**> **Monitors** dialog.

Understanding the COM+ Monitor Configuration Dialog Box

The COM+ Server dialog box lets you select the items to monitor on the Microsoft COM+ server.

Hostname: Displays the name of the monitored machine.

Measured Components: Displays the available components. Browse the tree and select the component you want to monitor. A description of the highlighted component appears in the Component/Counter Description box.

Performance Counters: Check the required performance counters. A description of the selected counter appears in the Component/Counter Description box.

Component/Counter Description: Displays a description of the selected component or counter.

Microsoft COM+ Performance Counters

The following tables describe the default counters that can be measured:

Authentication Metrics

Measurement	Description
Authenticate	Frequency of successful method call level authentication. When you set an authentication level for an application, you determine what degree of authentication is performed when clients call into the application.
Authenticate Failed	Frequency of failed method call level authentication.

Application Events

Measurement	Description
Activation	Frequency of application activation or startup.
Shutdown	Frequency of application shutdown or termination.

Thread Events

Measurement	Description
Thread Start	Rate at which single-threaded apartment (STA) thread for application have been started.
Thread Terminate	Rate at which single-threaded apartment (STA) thread for application have been terminated.
Work Enque	Event sent if a work is queued in single thread apartment object (STA). Note: These events are not signaled/sent in Windows Server 2003 and later.
Work Reject	Event sent if a work is rejected from single thread apartment object (STA). Note: These events are not signaled/sent in Windows Server 2003 and later.

Transaction Events

Measurement	Description
Transaction Duration	Duration of COM+ transactions for selected application.
Transaction Start	Rate at which transactions have started.
Transaction Prepared	Rate at which transactions have completed the prepare phase of the two-phase protocol.
Transaction Aborted	Rate at which transactions have been aborted.
Transaction Commit	Rate at which transactions have completed the commit protocol.

Object Events

Measurement	Description
Object Life Time	Duration of object existence (from instantiation to destruction).
Object Create	Rate at which new instances of this object are created.
Object Destroy	Rate at which instances of the object are destroyed.
Object Activate	Rate of retrieving instances of a new JIT-activated object.
Object Deactivation	Rate of freeing JIT-activated object via SetComplete or SetAbort.
Disable Commit	Rate of client calls to DisableCommit on a context. DisableCommit declares that the object's transactional updates are inconsistent and can't be committed in their present state.
Enable Commit	Rate of client calls to EnableCommit on a context. EnableCommit declares that the current object's work is not necessarily finished, but that its transactional updates are consistent and could be committed in their present form.
Set Complete	Rate of client calls to SetComplete on a context. SetComplete declares that the transaction in which the object is executing can be committed, and that the object should be deactivated on returning from the currently executing method call.
Set Abort	Rate of client calls to SetAbort on a context. SetAbort declares that the transaction in which the object is executing must be aborted, and that the object should be deactivated on returning from the currently executing method call.

Method Events

Measurement	Description
Method Duration	Average duration of method.
Method Frequency	Frequency of method invocation.
Method Failed	Frequency of failed methods (i.e. methods that return error HRESULT codes).
Method Exceptions	Frequency of exceptions thrown by selected method.

Microsoft .NET CLR Monitoring

The Microsoft .NET CLR monitor provides performance information for applications running in the Microsoft .NET CLR environment. During a scenario run, you can monitor the resources to isolate performance bottlenecks. You can specify which measurements and resources you want the monitor to measure.

Before monitoring a .NET server, you need to install the Microsoft .NET Server Monitor Probe on the server machine. You also need to ensure that Microsoft .NET Framework has been installed on the server machine. You can then specify which measurements and resources you want the Microsoft .NET CLR monitor to measure. You select these counters using the Controller's monitor configuration dialog box.

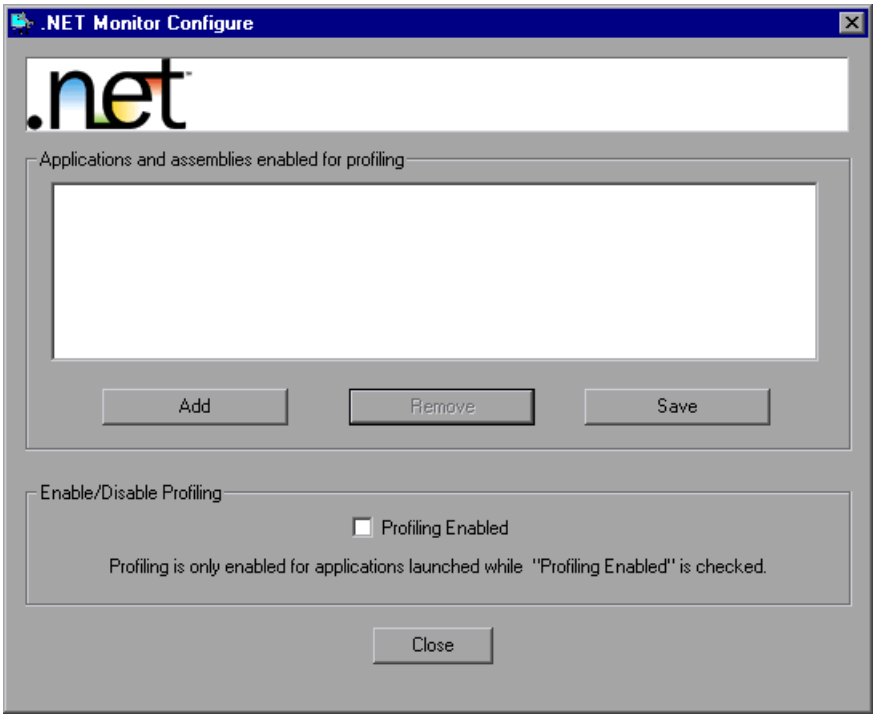
Setting Up the Monitoring Environment

To monitor Microsoft .NET server performance, you must first install the Microsoft .NET Server Monitor Probe on the server machine, and select the components (applications and assemblies) for .NET profiling.

For more information on installing the Microsoft .NET server add-in, refer to the *LoadRunner Installation Guide*.

To select components using the .NET Monitor:

1 On the .NET server machine, select the **.NET Monitor Configure** program from the Windows **Start** menu. The .NET Monitor Configure window opens.



2 Click **Add,** select a .NET executable or assembly file to profile, and click **Open**. The selection is displayed in the .NET Monitor Configure window.

3 Click **Save**.

4 Enable or disable profiling and click **Close**.

Understanding the .NET Monitor Configure Window

Applications and assemblies enabled for profiling: Displays the applications and assemblies that are selected for .NET profiling.

Add: Click this button to select an application for profiling.

Remove: Click this button to remove the selected application.

Save: Click this button to save the displayed applications and profiling.

Enable/Disable Profiling

Profiling enabled: Select this option to enable profiling. Profiling may slow down system performance. Use this option to enable or disable .NET profiling.

To ensure that changes to this option take effect immediately, reboot the computer, or log out and log back in.

Configuring the Microsoft .NET CLR .NET Monitor Over a Firewall

Before running the Microsoft .NET CLR Monitor over a firewall:

➤ Make sure that the MI Listener is installed on any machine (including the Controller machine) outside of the firewall. Refer to the *LoadRunner Installation Guide* for installation instructions.

➤ Configure the firewall agent on the server machine. For more information refer to the chapter "Running Vusers Over a Firewall" in the *LoadRunner Controller User's Guide* or *Mercury Tuning Module Console User's Guide*.

➤ Specify the correct connection string on the client machine, as described in step 3 of "Configuring the Microsoft .NET CLR Monitor" on page 383.

Configuring the Microsoft .NET CLR Monitor

You select monitor counters using the .NET Monitoring Configuration window.

To select measurements to monitor:

1. Click the Microsoft .NET CLR graph in the graph tree, and drag it into the right pane of the Run view.
2. Right-click the graph and choose **Add Measurements**, or choose **Monitors** > **Add Measurements.** The Microsoft .NET CLR dialog box opens.
3. In the **Monitored Server Machines** section, click **Add.** The Add Machine dialog box opens.
4. Enter the server name or IP address of the machine you want to monitor. Select the platform on which the machine runs, and click **OK**.

Note: To connect to the monitor over a firewall, enter the server name or IP address of the machine you want to monitor, according to the following format, and click **OK**:

<MI Listener machine>:<server machine key>

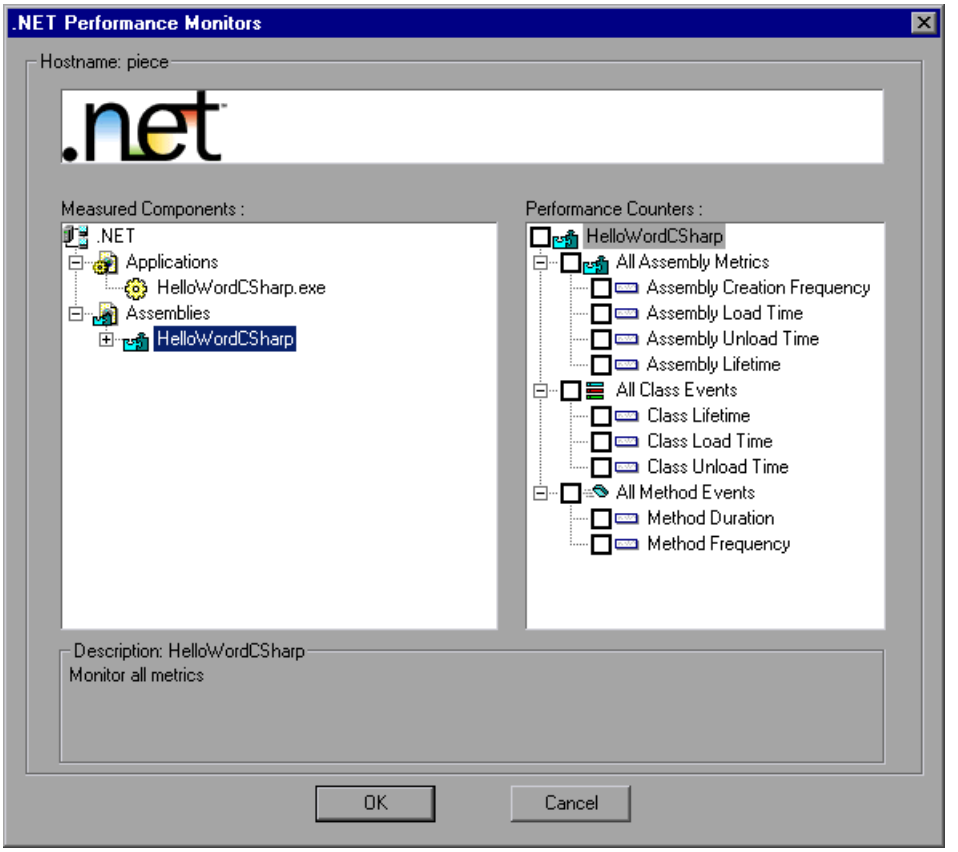
where **server machine key** is the unique key that you chose when configuring the firewall Agent on the server machine.

For example: 12.12.12.3:serverid

---

5 In the **Resource Measurements** section of the Microsoft .NET CLR dialog box, select the server running .NET and click **Add**.

The .NET Performance Monitors dialog box opens displaying the available measurements.



6 Browse the Measured Components tree, and select performance counters, as described in "Understanding the .NET Performance Monitors Window," on page 385.

For a description of the available measurements, see "Microsoft .NET CLR Performance Counters," on page 386.

7 Click **OK** in the .NET Performance Monitors dialog box, and in the Microsoft .NET CLR dialog box, to activate the Microsoft .NET CLR monitor.

Note: The data sampling rate for the .NET monitor is fixed (five seconds) and cannot be modified using the Controller **Tools** > **Options** > **Monitors** dialog.

Understanding the .NET Performance Monitors Window

Hostname: Displays the name of the host on which the profiled application is running.

Applications and Assemblies: Displays the applications and assemblies included in the selected application or assembly. Select the components to measure. The available counters are displayed in the **Performance Counters** box.

Descriptions for each counter are displayed in the Description area at the bottom of the .NET Monitoring Configuration window. For more information, see "Microsoft .NET CLR Performance Counters" on page 386.

To select monitoring measures for classes, expand the application and select the class.

Counters: Displays the available measurement counters. Select the counters to measure.

Tip: To select all available measures, select the **Select All Method Metrics** option from any level in the **Performance Counters** box.

Microsoft .NET CLR Performance Counters

You can monitor .NET counters at the application, assembly, class, and method levels. Measurements that take place before the application is fully loaded (such as Assembly Load Time, that measures the time it takes to load an assembly) will not be measured.

The following tables describe the counters that can be measured at each level. All durations are reported in seconds, and all frequencies are reported per five-second polling periods. For example, if 20 events occur in a 5 second polling period, the frequency is 4.

Application Level

Measurement	Description
Application Lifetime	Monitors the duration of the application in seconds.
Exception Frequency	Monitors the number of exceptions per second, in the five second polling period.
JIT (Just In Time) Duration	Monitors the time, in seconds, it takes for the JIT to compile code.
Thread Creation Frequency	Monitors the number of threads that are created in a polling period.
Thread Lifetime	Monitors the duration of threads.
Domain Creation Frequency	Monitors the number of domain creations in a polling period. (Domains protect areas of code. All applications run in a domain which keeps them encapsulated, so that they cannot interfere with other applications outside the domain).
Domain Load Time	Monitors the time it takes to load a domain. (Domains protect areas of code. All applications run in a domain which keeps them encapsulated, so that they cannot interfere with other applications outside the domain).
Domain Unload Time	Monitors the time it takes to unload a domain. (Domains protect areas of code. All applications run in a domain which keeps them encapsulated, so that they cannot interfere with other applications outside the domain).

Measurement	Description
Domain Lifetime	Monitors the duration of a domain. (Domains protect areas of code. All applications run in a domain which keeps them encapsulated, so that they cannot interfere with other applications outside the domain).
Module Creation Frequency	Monitors the number of modules that get created in a polling period. (Modules are groups of assemblies that make up a DLL or EXE.)
Module Load Time	Monitors the time it takes to load a module. (Modules are groups of assemblies that make up a dll or exe.)
Module Unload Time	Monitors the time it takes to unload a module. (Modules are groups of assemblies that make up a dll or exe.)
Module Lifetime	Monitors the duration of a module. (Modules are groups of assemblies that make up a dll or exe.)
Garbage Collection Duration	Monitors the duration between the start and stop of Garbage Collection.
Garbage Collection Frequency	Monitors the number of breaks for Garbage Collections in a polling period.
Unmanaged Code Duration	Monitors the duration of the calls to unmanaged code.
Unmanaged Code Frequency	Monitors the number of calls to unmanaged code.

Assembly Level

Measurement	Description
Assembly Creation Frequency	Monitors the number of assembly creations in a polling period. (Assemblies hold the .NET byte code and metadata).
Assembly Load Time	Monitors the time it takes to load an assembly. (Assemblies hold the .NET byte code and metadata).
Assembly Unload Time	Monitors the time it takes to unload an assembly. (Assemblies hold the .NET byte code and metadata).
Assembly Lifetime	Monitors the duration of an assembly. (Assemblies hold the .NET byte code and metadata).

Class Level

Measurement	Description
Class Lifetime	Monitors the duration of a class.
Class Load Time	Monitors the time it takes to load a class.
Class Unload Time	Monitors the time it takes to unload a class.

Method Level

At the method level, the measured time is per method, exclusive of other methods, calls to unmanaged code, and garbage collection time.

Measurement	Description
Method Duration	Monitors the duration of a method.
Method Frequency	Monitors the number of methods called in a polling period.

14

Application Deployment Solutions

Using LoadRunner's Application Deployment Solution monitor, you can isolate server performance bottlenecks by monitoring the Citrix MetaFrame XP or 1.8 server during a scenario or session step run.

This chapter describes:

- About Application Deployment Solutions Monitoring
- Citrix MetaFrame XP Monitoring

About Application Deployment Solutions Monitoring

LoadRunner's Citrix MetaFrame XP monitor provides you with information about the application deployment usage of the Citrix MetaFrame XP and 1.8 servers during a scenario or session step execution. The Citrix Monitor allows you to monitor the server performance statistics from Citrix MetaFrame Servers. You can monitor multiple parameters (counters) with a single monitor instance. This allows you to watch server loading for performance, availability, and capacity planning.

To obtain performance data, you need to activate the online monitor for the server and specify which resources you want to measure before executing the scenario or session step.

Citrix MetaFrame XP Monitoring

The Citrix MetaFrame XP monitor displays statistics about the resource usage on the Citrix MetaFrame XP or 1.8 server machine during the scenario or session step run.

To obtain data for this graph, you need to activate the MetaFrame XP monitor on the application server machine before executing the scenario or session step, enable the counters you want to monitor on the Citrix server, and specify which measurements and resources you want the Citrix monitor to measure. You select these counters using the Controller or Tuning Console's Citrix MetaFrame XP dialog box.

Setting up the Monitoring Environment

To monitor the Citrix server performance, you must first activate the Citrix MetaFrame XP monitor on the application server machine and enable the counters you want to monitor on the Citrix server.

Note: The port you use to monitor a Citrix MetaFrame server through a firewall depends on the configuration of your server.

Before setting up the Citrix MetaFrame Server monitor:

1. Make sure that Citrix MetaFrame Server has been installed and is running on a computer. If the computer running Citrix MetaFrame Server is running Windows 2000, make sure that the Remote Registry service is running on it.
2. Make sure that the computer on which you are running LoadRunner has administrator privileges on the machine running Citrix.
3. From the Controller or Tuning Console machine, map a network drive to the Citrix server machine. This ensures that the required authentication is provided to the Controller or Tuning Console to access the resource counters.
4. Launch PerfMon from the Controller or Tuning Console machine to enable the counters on the Citrix server. This allows you to monitor the same counters for the ICA Session object on the Citrix monitor.

5 You can configure the Citrix monitor to view ICA Session object counters only if at least one session is being run on the Citrix server. If no "real" user has opened a connection with the Citrix server, you need to first initialize or run a Citrix Vuser against the server, and only then configure the Citrix Monitor and add the ICA Session counters. If you configure the Citrix monitor without first initializing or running a Citrix Vuser (or connecting to the Citrix server as a "real" user), you will not be able to view the ICA Session object.

6 **Tuning Console only:** In the System Topology window, add an Application Deployment element representing the server running Citrix. Assign the following values to the element's settings:

- **O/S:** The version of Windows running on the Citrix machine
- **Product**: Citrix MetaFrame
- **Version:** XP

Note: Measurements that monitor instances are valid for the currently running Citrix session only. If you run this scenario or session step again, you will need to reconfigure the measurements that are instance-oriented.

To monitor the different instances, ensure that the server login and logout procedures are recorded in the **Vuser_init** and **Vuser_end** sections respectively, and not in the Action section of the script. For more information, refer to the *Mercury Virtual User Generator User's Guide*.

Adding a Monitored Machine to the Controller

1 Click the Citrix MetaFrame XP graph in the graph tree, and drag it into the right pane of the Run view. The Citrix MetaFrame XP dialog box opens.

2 Right-click the graph and select **Add Measurements**, or choose **Monitors** > **Add Measurements**.

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

4 Enter the server name or IP address of the machine you want to monitor. Select the platform on which the machine runs, and click **OK**.

5 In the **Resource Measurements** section of the dialog box, select the default measurements that you want to monitor.

For a description of the available measurements, see "Citrix MetaFrame Performance Counters" on page 395.

6 To select additional measurements, click **Add**.

Continue with Configuring the Citrix MetaFrame XP Monitor below.

Adding a Monitored Machine to the Tuning Console

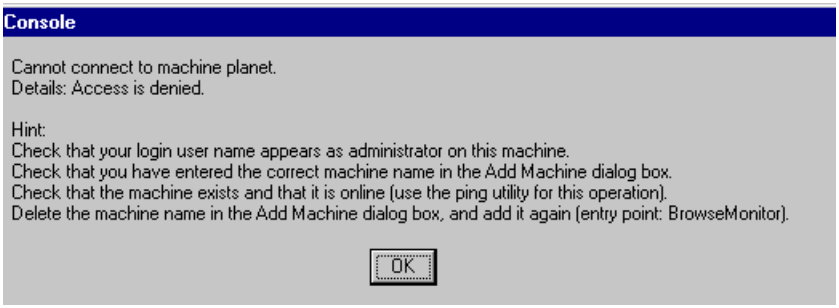
1 Click **Monitors** to open the Monitors Configuration dialog box.

2 Select the server whose monitors you want to configure from the Server list box.

3 Click **Add Monitor**, and then check **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left section of the dialog box, expand the Application Deployment category, select **Citrix MetaFrame XP**, and then click **Add**.

Note: If Mercury Tuning Console displays the following error message:

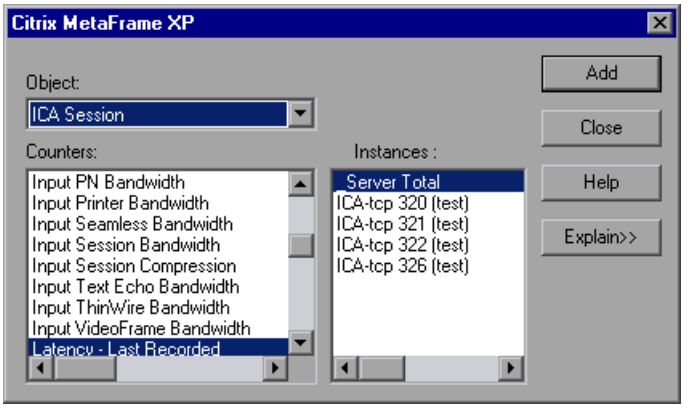


it means that the computer on which you are running Mercury Tuning Console does not have administrator privileges on the Citrix machine. To remedy this, close the Mercury Tuning Console session, acquire administrator privileges on the Citrix machine, and reopen the Mercury Tuning Console session. Then repeat this procedure, starting again with step 1 above.

Continue with Configuring the Citrix MetaFrame XP Monitor below.

Configuring the Citrix MetaFrame XP Monitor

1 The Citrix MetaFrame XP dialog box opens displaying the Citrix object, its counters, and instances opens.



For each measurement, select an object, counter, and instance, and then click **Add**, as described in "Understanding the Citrix MetaFrame XP Dialog Box" on page 395.

For a description of the available measurements, see "Citrix MetaFrame Performance Counters" on page 395.

Note: If the dialog box freezes after clicking **Add**, you may need to rebuild the localhost cache on the Citrix server machine. For more information, refer to Document IDs CTX003648 and CTX759510 in the Citrix Knowledge Base (http://knowledgebase.citrix.com/cgi-bin/webcgi.exe?New,KB=CitrixKB).

---

2 Add all the desired resources to the list, and click **Close**.

3 In the Controller, click **OK** in the Citrix MetaFrame dialog box to activate the monitor.

4 In the Tuning Console, click **OK** in the Select Measurements to Monitor dialog box to activate the monitor.

Understanding the Citrix MetaFrame XP Dialog Box

The Ariba dialog box lets you select the items to monitor on the Citrix MetaFrame XP or 1.8 server.

Object: Select the object being monitored on the specified machine. The Object listbox allows you to display three types of counters: Citrix MetaFrame XP, Citrix IMA Networking, and ICA Session.

Counters: Select a resource counter to monitor. Select multiple counters using the CTRL key. For an explanation of each counter, click **Explain**.

Instances: If multiple instances of the selected counter are running, select one or more instances to monitor for the selected counter.

Explain: Displays a description of the selected counter.

Citrix MetaFrame Performance Counters

The following table describes some of the counters that can be measured.

Non-Virtual Counters

Measurement	Description
% Disk Time	The percentage of elapsed time that the selected disk drive is busy servicing read or write requests.
% Processor Time	The percentage of time that the processor is executing a non-Idle thread. This counter is a primary indicator of processor activity. It is calculated by measuring the time that the processor spends executing the thread of the Idle process in each sample interval, and subtracting that value from 100%. (Each processor has an Idle thread which consumes cycles when no other threads are ready to run). It can be viewed as the percentage of the sample interval spent doing useful work. This counter displays the average percentage of busy time observed during the sample interval. It is calculated by monitoring the time the service was inactive, and then subtracting that value from 100%.

Measurement	Description
File data Operations/sec	The rate that the computer is issuing Read and Write operations to file system devices. It does not include File Control Operations.
Interrupts/sec	The average number of hardware interrupts the processor is receiving and servicing in each second. It does not include DPCs, which are counted separately. This value is an indirect indicator of the activity of devices that generate interrupts, such as the system clock, the mouse, disk drivers, data communication lines, network interface cards and other peripheral devices. These devices normally interrupt the processor when they have completed a task or require attention. Normal thread execution is suspended during interrupts. Most system clocks interrupt the processor every 10 milliseconds, creating a background of interrupt activity. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.
Output Session Line Speed	This value represents the line speed from server to client for a session in bps.
Input Session Line Speed	This value represents the line speed from client to server for a session in bps.
Page Faults/sec	A count of the Page Faults in the processor. A page fault occurs when a process refers to a virtual memory page that is not in its Working Set in main memory. A Page Fault will not cause the page to be fetched from disk if that page is on the standby list, and hence already in main memory, or if it is in use by another process with whom the page is shared.

Measurement	Description
Pages/sec	The number of pages read from the disk or written to the disk to resolve memory references to pages that were not in memory at the time of the reference. This is the sum of Pages Input/sec and Pages Output/sec. This counter includes paging traffic on behalf of the system Cache to access file data for applications. This value also includes the pages to/from non-cached mapped memory files. This is the primary counter to observe if you are concerned about excessive memory pressure (that is, thrashing), and the excessive paging that may result.
Pool Nonpaged Bytes	The number of bytes in the Nonpaged Pool, a system memory area where space is acquired by operating system components as they accomplish their appointed tasks. Nonpaged Pool pages cannot be paged out to the paging file, but instead remain in main memory as long as they are allocated.
Private Bytes	The current number of bytes this process has allocated that cannot be shared with other processes.
Processor Queue Length	The instantaneous length of the processor queue in units of threads. This counter is always 0 unless you are also monitoring a thread counter. All processors use a single queue in which threads wait for processor cycles. This length does not include the threads that are currently executing. A sustained processor queue length greater than two generally indicates processor congestion. This is an instantaneous count, not an average over the time interval.
Threads	The number of threads in the computer at the time of data collection. Notice that this is an instantaneous count, not an average over the time interval. A thread is the basic executable entity that can execute instructions in a processor.

Measurement	Description
Latency – Session Average	This value represents the average client latency over the life of a session.
Latency – Last Recorded	This value represents the last recorded latency measurement for this session.
Latency – Session Deviation	This value represents the difference between the minimum and maximum measured values for a session.
Input Session Bandwidth	This value represents the bandwidth from client to server traffic for a session in bps.
Input Session Compression	This value represents the compression ratio for client to server traffic for a session.
Output Session Bandwidth	This value represents the bandwidth from server to client traffic for a session in bps.
Output Session Compression	This value represents the compression ratio for server to client traffic for a session.
Output Session Linespeed	This value represents the line speed from server to client for a session in bps.

Virtual Channel Counters

Measurement	Description
Input Audio Bandwidth	This value represents the bandwidth from client to server traffic on the audio mapping channel. This is measured in bps.
Input Clipboard Bandwidth	This value represents the bandwidth from client to server traffic on the clipboard mapping channel. This is measured in bps.
Input COM1 Bandwidth	This value represents the bandwidth from client to server traffic on the COM1 channel. This is measured in bps.

Measurement	Description
Input COM2 Bandwidth	This value represents the bandwidth from client to server traffic on the COM2 channel. This is measured in bps.
Input COM Bandwidth	This value represents the bandwidth from client to server traffic on the COM channel. This is measured in bps.
Input Control Channel Bandwidth	This value represents the bandwidth from client to server traffic on the ICA control channel. This is measured in bps.
Input Drive Bandwidth	This value represents the bandwidth from client to server traffic on the client drive mapping channel. This is measured in bps.
Input Font Data Bandwidth	This value represents the bandwidth from client to server traffic on the local text echo font and keyboard layout channel. This is measured in bps.
Input Licensing Bandwidth	This value represents the bandwidth from server to client traffic on the licensing channel. This is measured in bps.
Input LPT1 Bandwidth	This value represents the bandwidth from client to server traffic on the LPT1 channel. This is measured in bps.
Input LPT2 Bandwidth	This value represents the bandwidth from client to server traffic on the LPT2 channel. This is measured in bps.
Input Management Bandwidth	This value represents the bandwidth from client to server traffic on the client management channel. This is measured in bps.
Input PN Bandwidth	This value represents the bandwidth from client to server traffic on the Program Neighborhood channel. This is measured in bps.
Input Printer Bandwidth	This value represents the bandwidth from client to server traffic on the printer spooler channel. This is measured in bps.

Measurement	Description
Input Seamless Bandwidth	This value represents the bandwidth from client to server traffic on the Seamless channel. This is measured in bps.
Input Text Echo Bandwidth	This value represents the bandwidth from client to server traffic on the local text echo data channel. This is measured in bps.
Input Thinwire Bandwidth	This value represents the bandwidth from client to server traffic on the Thinwire (graphics) channel. This is measured in bps.
Input VideoFrame Bandwidth	This value represents the bandwidth from client to server traffic on the VideoFrame channel. This is measured in bps.
Output Audio Bandwidth	This value represents the bandwidth from server to client traffic on the audio mapping channel. This is measured in bps.
Output Clipboard Bandwidth	This value represents the bandwidth from server to client traffic on he clipboard mapping channel. This is measured in bps.
Output COM1 Bandwidth	This value represents the bandwidth from server to client traffic on the COM1 channel. This is measured in bps.
Output COM2 Bandwidth	This value represents the bandwidth from server to client traffic on the COM2 channel. This is measured in bps.
Output COM Bandwidth	This value represents the bandwidth from server to client traffic on the COM channel. This is measured in bps.
Output Control Channel Bandwidth	This value represents the bandwidth from server to client traffic on the ICA control channel. This is measured in bps.
Output Drive Bandwidth	This value represents the bandwidth from server to client traffic on the client drive channel. This is measured in bps.

Measurement	Description
Output Font Data Bandwidth	This value represents the bandwidth from server to client traffic on the local text echo font and keyboard layout channel. This is measured in bps.
Output Licensing Bandwidth	This value represents the bandwidth from server to client traffic on the licensing channel. This is measured in bps.
Output LPT1 Bandwidth	This value represents the bandwidth from server to client traffic on the LPT1 channel. This is measured in bps.
Output LPT2 Bandwidth	This value represents the bandwidth from server to client traffic on the LPT2 channel. This is measured in bps.
Output Management Bandwidth	This value represents the bandwidth from server to client traffic on the client management channel. This is measured in bps.
Output PN Bandwidth	This value represents the bandwidth from server to client traffic on the Program Neighborhood channel. This is measured in bps.
Output Printer Bandwidth	This value represents the bandwidth from server to client traffic on the printer spooler channel. This is measured in bps.
Output Seamless Bandwidth	This value represents the bandwidth from server to client traffic on the Seamless channel. This is measured in bps.
Output Text Echo Bandwidth	This value represents the bandwidth from server to client traffic on the local text echo data channel. This is measured in bps.
Output Thinwire Bandwidth	This value represents the bandwidth from server to client traffic on the Thinwire (graphics) channel. This is measured in bps.
Output VideoFrame Bandwidth	This value represents the bandwidth from server to client traffic on the VideoFrame channel. This is measured in bps.

15

Middleware Performance Monitoring

Using LoadRunner's Middleware Performance monitors, you can monitor the Tuxedo and the IBM WebSphere MQ servers during a scenario or session step run and isolate server performance bottlenecks.

This chapter describes:

- About Middleware Performance Monitoring
- Tuxedo Monitoring
- IBM WebSphere MQ Monitoring

About Middleware Performance Monitoring

A primary factor in a transaction's response time is the middleware performance usage. LoadRunner's Middleware Performance monitors provide you with information about the middleware performance usage of the Tuxedo and IBM WebSphere MQ servers during a scenario or session step execution. To obtain performance data, you need to activate the online monitor for the server and specify which resources you want to measure before executing the scenario or session step.

The Tuxedo monitor can monitor the server, load generator machine, workstation handler, and queue in a Tuxedo system. To run the Tuxedo monitor, you must install the Tuxedo client libraries on the machine you want to monitor.

The IBM WebSphere MQ monitor is used to monitor channel and queue performance counters on an IBM WebSphere MQ (version 5.x) Server.

The procedures for selecting monitor measurements and configuring the monitors vary according to server type. The following sections contain specific configuration instructions for each server type.

Tuxedo Monitoring

The Tuxedo monitor allows you to measure and view your Tuxedo server performance. It provides information about the host machine, workstation handler, and queue in a Tuxedo system.

To obtain data for this graph, you need to enable the Tuxedo monitor (from the Controller or Tuning Console) and select the default measurements you want to display, before running the scenario or session step.

Note: If Tuxedo 7.1 or higher is installed on the Controller or Tuning Console machine, more than one Tuxedo application server can be monitored at a time. However, if Tuxedo 6.5 or below is installed on the Controller or Tuning Console machine, only one Tuxedo application server can be monitored at a time.

Setting up the Tuxedo Monitor

Before you set up the monitor, perform the following:

➤ Ensure that a Tuxedo workstation client (not a native client) is installed on the Controller or Tuning Console machine. Use a Tuxedo 6.x client if a Tuxedo 6.x server is used, and Tuxedo 7.1 or above client if a Tuxedo 7.1 or above server is used. If you use a Tuxedo 6.5 or earlier server, you can still use a Tuxedo 7.1 or later client to monitor it, provided that you set the WSINTOPPRE71 environment variable to “yes”.

Note: A Tuxedo workstation client communicates with the application server over the network, and is not required to run the Tuxedo application server on the same machine. A native client can only communicate with the Tuxedo application server if it is part of the relevant Tuxedo domain.

➤ Define the Tuxedo environment variables on the Controller or Tuning Console machine—set the TUXDIR variable to the Tuxedo installation directory (for example, V:\environ\32\Tuxedo8.0), and add the Tuxedo **bin** directory to the PATH variable.

➤ Ensure that the workstation listener (WSL) process is running. This enables the application server to accept requests from workstation clients. Note that the address and port number used to connect to the application server must match those dedicated to the WSL process.

Note: For information on configuring the WSL, refer to the BEA Tuxedo Web site (http://edocs.beasys.com/tuxedo/tux81/rf5/rf5101.htm#1534543).

Adding a Monitored Machine to the Controller

1. Click the Tuxedo graph in the graph tree, and drag it into the right pane of the Run view.
2. Right-click the graph and select **Add Measurements**, or choose **Monitors** > **Add Measurements**. The Tuxedo dialog box opens.
3. In the **Monitored Server Machines** section, click **Add.** The Add Machine dialog box opens.
4. Enter the server name or IP address of the machine you want to monitor. Select the platform on which the machine runs, and click **OK**.
5. In the **Resource Measurements** section of the Tuxedo dialog box, click **Add**.

Continue with Configuring the Tuxedo Monitor below.

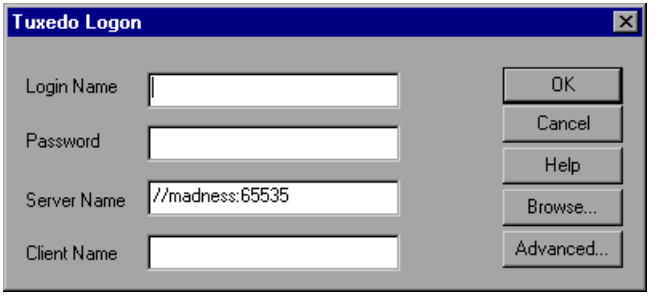
Adding a Monitored Machine to the Tuning Console

1. Click **Monitors** to open the Monitors Configuration dialog box.
2. Select the server whose monitors you want to configure from the Server list box.
3. Click **Add Monitor** and select **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.
4. In the left section of the dialog box, expand the Middleware Component category, select **Tuxedo**, and then click **Add**.

   Continue with Configuring the Tuxedo Monitor below.

Configuring the Tuxedo Monitor

1. The Tuxedo Logon dialog box opens.



Enter the following information:

Login Name: Enter your login name.

Password: Enter your password.

Server Name: Enter the name of the server. The format of the server name is //<machine name>:<port number>. Alternatively, you can specify the IP address or the hexadecimal format used by old versions of Tuxedo. Note that you cannot use quotation marks.

Client Name: Enter the name of the client machine. If a Tuxedo server was previously monitored, its name is displayed in the Server Name box.

Browse: Enables you to navigate to the tpinit.ini file of a recorded Tuxedo script.

Note: Logon information is located in the Logon section of the **tpinit.ini** file in the recorded script's directory. It is recommended that you use the Browse button and select the **tpinit.ini** file from a recorded script, rather than enter the values manually. You can also determine the client name from the **lrt_tpinitialize** statement in the recorded script.

In the following example of a **tpinit.ini** file, the Tuxedo monitor was configured for a server named psft1 using port 7000, and a client named bankapp. The logon user name was PS and the password was PS.

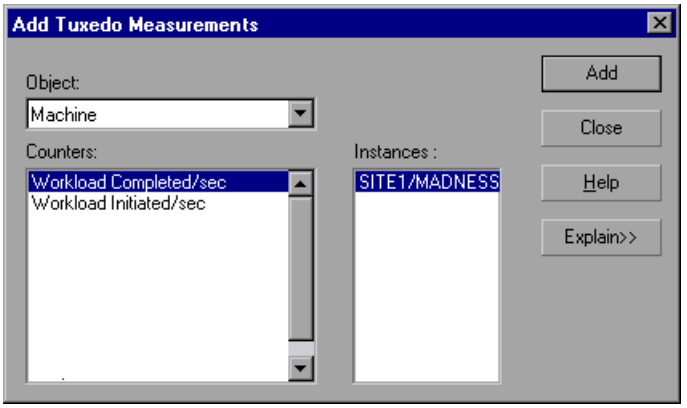
```
[Logon]
LogonServername=//psft1:7000
LogonUsrName=PS
LogonCltName=bankapp
LogonGrpName=
LogonPasswd=PS
LogonData=
```

2 To authenticate the Tuxedo monitor, click **Advanced**, and enter the authentication data as a hexadecimal string (beginning with "0x") in the data box. The authentication data value can be obtained from the **tpinit.ini** file of an existing Tuxedo script.

Note: If you are using Tuxedo 6.5 or below, the monitor can only connect to one application server during a Controller or Tuning Console session. Once it connects to an application server, the server is the only one used by the monitor until the Controller or Tuning Console is closed. This applies even when all of the counters are deleted from the monitor.

---

3 Click **OK**. The Add Tuxedo Measurements dialog box opens.



4 For each measurement, select an object, counter, and instance, and then click **Add**, as described in "Understanding the Tuxedo Dialog Box" on page 408.

For a description of the available measurements, see "Tuxedo Performance Counters" on page 409.

5 Add all the desired objects to the list, and click **Close**.

6 In the Controller, click **OK** in the Tuxedo dialog box to activate the monitor.

7 In the Tuning Console, click **OK** in the Select Measurements to Monitor dialog box to activate the monitor.

Understanding the Tuxedo Dialog Box

The Tuxedo dialog box lets you select the resources to monitor on the Tuxedo client.

Object: Select the object being monitored on the specified machine.

Counters: Select a resource counter to monitor. Select multiple counters using the CTRL key. For an explanation of each counter, click **Explain**.

Instances: If multiple instances of the selected counter are running, select one or more instances to monitor for the selected counter.

Explain: Displays a description of the selected counter.

Tuxedo Performance Counters

The following table lists the available Tuxedo monitor measurements. It is recommended to pay particular attention to the following measurements: % Busy Clients, Active Clients, Busy Clients, Idle Clients, and all the queue counters for relevant queues:.

Monitor	Measurements
Machine	**% Busy Clients** - The per cent of active clients currently logged in to the Tuxedo application server which are waiting for a response from the application server.
	Active Clients - The total number of active clients currently logged in to the Tuxedo application server.
	Busy Clients - The total number of active clients currently logged in to the Tuxedo application server that are waiting for a response from the application server.
	Current Accessers - Number of clients and servers currently accessing the application either directly on this machine or through a workstation handler on this machine.
	Current Transactions - Number of in use transaction table entries on this machine.
	Idle Clients - The total number of active clients currently logged in to the Tuxedo application server that are not waiting for a response from the application server.
	Workload Completed/second - The total workload on all the servers for the machine that was completed, per unit time.
	Workload Initiated/second - The total workload on all the servers for the machine that was initiated, per unit time.

Monitor	Measurements
Queue	**% Busy Servers** - The per cent of active servers currently handling Tuxedo requests.
	Active Servers - The total number of active servers either handling or waiting to handle Tuxedo requests.
	Busy Servers - The total number of active servers currently busy handling Tuxedo requests.
	Idle Servers - The total number of active servers currently waiting to handle Tuxedo requests.
	Number Queued - The total number of messages which have been placed on the queue.
Server	**Requests/second** - How many server requests were handled per second
	Workload/second -The workload is a weighted measure of the server requests. Some requests could have a different weight than others. By default, the workload is always 50 times the number of requests.
Workstation Handler (WSH)	**Bytes Received/sec** - The total number of bytes received by the workstation handler, per second.
	Bytes Sent/sec - The total number of bytes sent back to the clients by the workstation handler, per second.
	Messages Received/sec - The number of messages received by the workstation handler, per second.
	Messages Sent/sec - The number of messages sent back to the clients by the workstation handler, per second.
	Number of Queue Blocks/sec - The number of times the queue for the workstation handler blocked, per second. This gives an idea of how often the workstation handler was overloaded.

IBM WebSphere MQ Monitoring

The IBM WebSphere MQ monitor displays statistics about the resource usage on the IBM WebSphere MQ server during the scenario or session step run.

To use the IBM WebSphere MQ monitor you must first install the IBM WebSphere MQ client on the Controller or Tuning Console machine and configure the server environment to monitor events. The IBM WebSphere MQ monitor connects to the IBM WebSphere MQ server (via the MQ Client Connection installed on the Controller or Tuning Console machine). In MQ Client environments, the client machine connects to an MQ Server instance, and uses the Server's resources as if they were local to the client machine.

You can then enable the MQ online monitor (from the Controller or Tuning Console) and select the default measurements you want to display, before running the scenario or session step. You select these measurements using the IBM WebSphere MQ Add Measurements dialog box.

Note: The LoadRunner IBM WebSphere MQ monitor supports machines running the IBM MQ Server (version 5.2) on Windows platforms only.

To monitor the IBM WebSphere MQ server, the Windows user must be part of the Administration Group of the IBM WebSphere MQ server.

Setting up the Monitoring Environment

Before you set up the monitor, perform the following:

➤ Ensure that an IBM WebSphere MQ Client Connection (version 5.21 only) is installed on the Controller or Tuning Console machine.

For additional information on installing the IBM WebSphere MQ Server/Client, refer to the WebSphere MQ family Web site (http://www-3.ibm.com/software/ts/mqseries/library/manuals/index.htm).

➤ Configure the IBM WebSphere server to monitor events.

The LoadRunner MQ Monitor retrieves event messages from two standard MQSeries queues only:

➤ SYSTEM.ADMIN.PERFM.EVENT – performance events, such as "queue depth high"

➤ SYSTEM.ADMIN.CHANNEL.EVENT – channel events, such as "channel stopped"

Events must be enabled for the queue manager (and in many cases, on the applicable object, as well). Performance events are enabled by setting attributes for the queue on the MQ Server. Channel events are enabled by default, and cannot be disabled.

Note: The IBM WebSphere MQ monitor does not retrieve data from a queue manager after the queue manager has been restarted.

To enable performance events for the Queue Manager:

1 Use the following MQSC command: ALTER QMGR PERFMEV(ENABLED).

2 Set the following attributes for the queue:

Measurement	Set Event Attributes
Event - Queue Depth High	• QDEPTHHI(integer) – where integer is a value expressed as a percentage of maximum messages allowed, and is in the range of 0 to 100 inclusive. • QDPHIEV(action) – where action is the word "ENABLED" or "DISABLED", enabling or disabling the generation of the event, respectively.
Event - Queue Depth Low	To enable the event for a queue, the following attributes of the queue must be set: • QDEPTHLO(integer) – where integer is a value expressed as a percentage of maximum messages allowed, and is in the range of 0 to 100 inclusive. • QDPLOEV(action) – where action is the word "ENABLED" or "DISABLED", enabling or disabling the generation of the event, respectively.

Measurement	Set Event Attributes
Event - Queue Full	• QDEPTHHI(integer) – where integer is a value expressed as a percentage of maximum messages allowed, and is in the range of 0 to 100 inclusive. • QDPMAXEV(action) – where action is the word "ENABLED" or "DISABLED", enabling or disabling the generation of the event, respectively.
Event - Queue Service Interval High	• QSVCINT(integer) – where integer is a value expressed as milliseconds, in the range of 0 and 999,999,999, inclusive. Note: this value is shared with Queue Service Interval OK. • QSVCIEV(type) – where type is the word "HIGH", "OK", or "NONE", enabling service interval high events, enabling service interval ok events, or disabling the generation of the event, respectively.
Event - Queue Service Interval OK	• QSVCINT(integer) – where integer is a value expressed as milliseconds, in the range of 0 and 999,999,999, inclusive. Note: this value is shared with Queue Service Interval High. • QSVCIEV(type) – where type is the word "HIGH", "OK", or "NONE", enabling service interval high events, enabling service interval ok events, or disabling the generation of the event, respectively.

Note: If you encounter an MQ Server error message (starting with the characters MQRC_), refer to the WebSphere MQ family support Web site (http://www-306.ibm.com/software/integration/mqfamily/support/).

After you have installed the MQ Client on the Controller or Tuning Console, and configured the server environment to monitor events, you can specify which resources you want to measure.

Adding a Monitored Machine to the Controller

1. Click the IBM WebSphere MQ graph in the graph tree, and drag it into the right pane of the Run view.
2. Right-click the graph and select **Add Measurements**, or choose **Monitors** > **Add Measurements**. The IBM WebSphere MQ dialog box opens.
3. In the **Monitored Server Machines** section, click **Add.** The Add Machine dialog box opens.
4. Enter the server name or IP address of the machine you want to monitor. The format of the server name is <machine name>:<port number>. Select the platform on which the machine runs, and click **OK**.
5. In the Resource Measurements section of the IBM WebSphere MQ dialog box, click **Add**.

   Continue with Configuring the IBM WebSphere MQ Monitor below.

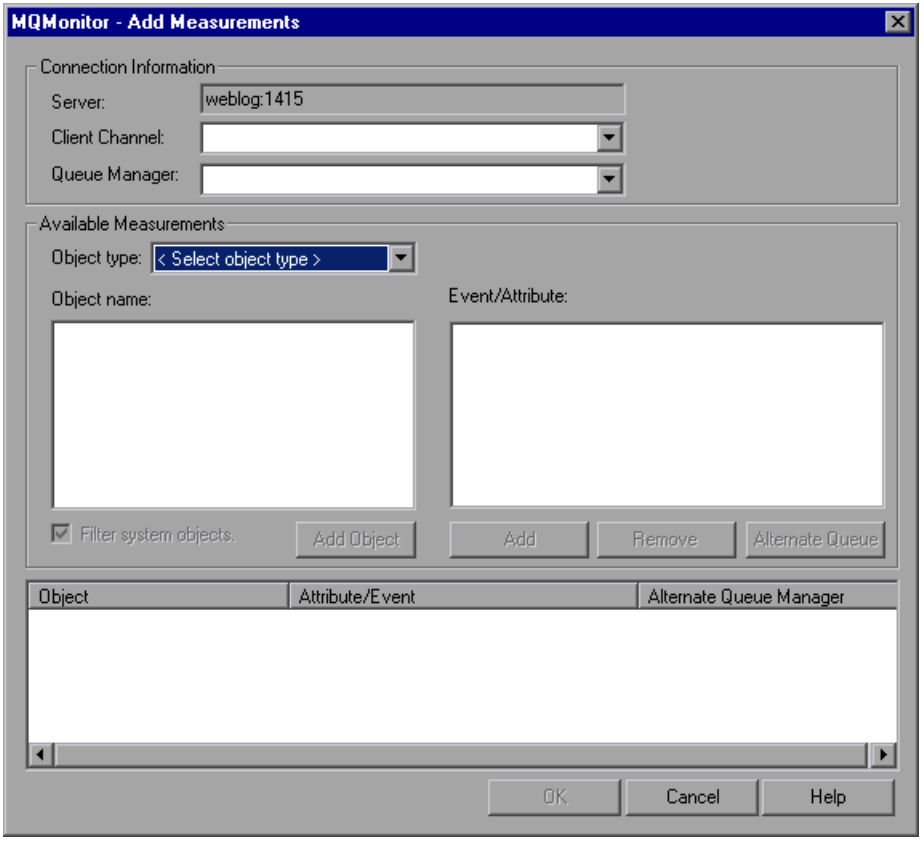
Adding a Monitored Machine to the Tuning Console

1. Click **Monitors** to open the Monitors Configuration dialog box.
2. Select the server whose monitors you want to configure from the Server list box.
3. Click **Add Monitor** and select **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.
4. In the left section of the dialog box, expand the Middleware Component category, select **IBM WebSphere MQ**, and then click **Add**.

   Continue with Configuring the IBM WebSphere MQ Monitor below.

Configuring the IBM WebSphere MQ Monitor

1 The MQMonitor Add Measurements dialog box opens.



In the **Connections Information** section, enter the name of the channel through which a client connection is made to an MQ Server, and the name of the queue manager to be monitored.

Note: A queue manager can only be accessed by one Controller or Tuning Console or monitoring application at any one time.

User entries for any text box are limited to 48 characters.

---

2 In the **Available Measurements** section, select an object type.

A list of previously added objects of the selected object type appear in the Object name list. A list of attributes or events applicable to the selected object type appear in the Events/Attributes list.

The names of monitored objects, event/attribute selected, and alternate queue managers, are listed in the monitored objects pane.

3 By default, only user-defined objects are displayed in the Object name list. To show all objects, clear the **Filter System Objects** check box. You can modify the filter settings, in the **<LoadRunner_installation>\dat\monitors\mqseries.cfg** file.

4 Select an object or add a new object to the Object name list. To add a new object name, click **Add Object**. In the Add Object Name dialog box, enter the name of an object to be monitored and click **OK**. The dialog box closes and the name of the object appears in the Object name list.

5 Select the attributes or events to be measured from the Attribute/Event box. The list of attributes or events is applicable to the selected object type.

For a list of the available performance counters, see page 419.

Note: To enable the event for a queue, ensure that the attributes for the queue have been set. For more information, refer to "Setting up the Monitoring Environment" on page 411.

---

6 If the event configured for monitoring is from a remote queue manager (other than the one identified in the queue manager field of the IBM WebSphere MQ Add Measurements dialog box), click **Alternate Queue**. Enter the name of an alternate queue manager in the Alternate Queue dialog box, and click **OK**.

Note: When you add an alternate queue manager, this becomes the default queue manager for any events that you subsequently add. To return to the queue manager to which you are connected, enter that name in the Alternate Queue Manager dialog box.

---

7 To add the object measurements to the monitored objects list, click **Add**. The name of the object, it's events and attributes, and any alternate queue managers, are listed in the monitored objects pane.

8 To remove a monitored object event or attribute, select the object measurement in the monitored objects pane, and click **Remove**. The entry is deleted from the monitored objects list.

9 Add all the desired counters to the monitored objects list, and click **OK**.

10 In the Controller, click **OK** in the IBM WebSphere MQ dialog box to activate the monitor.

11 In the Tuning Console, click **OK** in the Select Measurements to Monitor dialog box to activate the monitor.

Understanding the IBM WebSphere MQ Dialog Box

The IBM WebSphere MQ dialog box lets you select the items to monitor on the IBM WebSphere MQ server.

Connection Information

- **Server:** The name of the server you are monitoring.
- **Client Channel:** Enter the name of the channel through which a client connection is made to an MQ Server.

Note: You can set up a specific channel on an MQ Server instance, or use the default "SYSTEM.DEF.SVRCONN" channel. If the client channel is undefined, the MQ Server will be inaccessible via client connections (the MQ Monitor will not work, as it will not be able to connect to the queue manager which it is supposed to monitor).

➤ **Queue Manager:** Enter the name of the queue manager to be monitored.

Note: The monitor is not restricted to monitoring only the queue manager to which it is connected. You can configure multiple queue managers to write to the event queue of a central queue manager for centralized monitoring (this applies to Events only, not polled object attributes). All events contain a queue manager attribute identifying their source.

Available Measurements

➤ **Object Type:** Select an object type from either Channel or Queue.

➤ **Object Name:** Enter a name for object you want to monitor.

➤ **Event/Attribute:** Select the events and attributes you want to monitor.

➤ **Filter System Objects**: Select to enable the system objects filter.

➤ **Add Object**: Enables you to add a new object name to the Object name list.

➤ **Add:** Enables you to add an Event or Attribute to an object.

➤ **Remove:** Enables you to remove a monitored object event or attribute from the Object name list.

➤ **Alternate Queue:** Enter the name of an alternate queue manager if the event is from a remote queue manager.

Monitored Object list

A list of monitored objects, including the object's name, events and attributes, and alternate queue manager.

IBM WebSphere MQ Performance Counters

The following tables list the available IBM WebSphere MQ monitor measurements:

Queue Performance Counters

Measurement	Description
Event - Queue Depth High (events per second)	An event triggered when the queue depth reaches the configured maximum depth.
Event - Queue Depth Low (events per second)	An event triggered when the queue depth reaches the configured minimum depth.
Event - Queue Full (events per second)	An event triggered when an attempt is made to put a message on a queue that is full.
Event - Queue Service Interval High (events per second)	An event triggered when no messages are put to or retrieved from a queue within the timeout threshold.
Event - Queue Service Interval OK (events per second)	An event triggered when a message has been put to or retrieved from a queue within the timeout threshold.
Status - Current Depth	Current count of messages on a local queue. This measurement applies only to local queues of the monitored queue manager.
Status - Open Input Count	Current count of open input handles. Input handles are opened so that an application may "put" messages to a queue.
Status - Open Output Count	Current count of open output handles. Output handles are opened so that an application may "get" messages from a queue.

Channel Performance Counters

Measurement	Description
Event - Channel Activated (events per second)	Event generated when a channel, waiting to become active but inhibited from doing so due to a shortage of queue manager channel slots, becomes active due to the sudden availability of a channel slot.
Event - Channel Not Activated (events per second)	Event generated when a channel, attempts to become active but inhibited from doing so due to a shortage of queue manager channel slots.
Event - Channel Started (events per second)	Event generated when a channel is started.
Event - Channel Stopped (events per second)	Event generated when a channel is stopped, regardless of source of stoppage.
Event - Channel Stopped by User (events per second)	Event generated when a channel is stopped by a user.
Status - Channel State	The current state of a channel. Channels pass through several states from STOPPED (inactive state) to RUNNING (fully active state). Channel states range from 0 (STOPPED) to 6 (RUNNING).
Status - Messages Transferred	The count of messages that have been sent over the channel. If no traffic is occurring over the channel, this measurement will be zero. If the channel has not been started since the queue manager was started, no measurement will be available.
Status - Buffer Received	The count of buffers that have been received over the channel. If no traffic is occurring over the channel, this measurement will be zero. If the channel has not been started since the queue manager was started, no measurement will be available.

Measurement	Description
Status - Buffer Sent	The count of buffers that have been sent over the channel. If no traffic is occurring over the channel, this measurement will be zero. If the channel has not been started since the queue manager was started, no measurement will be available.
Status - Bytes Received	The count of bytes that have been received over the channel. If no traffic is occurring over the channel, this measurement will appear as zero. If the channel has not been started since the queue manager was started, no measurement will be available.
Status - Bytes Sent	The count of bytes that have been sent over the channel. If no traffic is occurring over the channel, this measurement will appear as zero. If the channel has not been started since the queue manager was started, no measurement will be available.

16

Application Traffic Management Monitoring

Using Mercury Tuning Console's Application Traffic Management monitor, you can monitor the BIG-IP load balancing device during a session step run and isolate server performance bottlenecks.

This chapter describes:

- About Application Traffic Management Monitoring
- F5 BIG-IP Monitoring

About Application Traffic Management Monitoring

Mercury Tuning Console's F5 BIG-IP monitor provides you with information about the content of event logs and other data from F5 BIG-IP load balancing device using SNMP during a session step execution. In order to obtain performance data, you need to activate the online monitor for the server and specify which resources you want to measure before executing the session step.

F5 BIG-IP Monitoring

To use the F5 BIG-IP monitor, you must first configure the F5 BIG-IP monitor on the SiteScope machine, and then select the counters you want the F5 BIG-IP monitor to measure. You select these counters using the Tuning Console's F5 BIG-IP dialog box.

Setting Up the Monitoring Environment

Before you set up the monitor, perform the following:

➤ Make sure that SiteScope has been installed on a server. SiteScope is the application that is used to monitor the F5 BIG-IP server. Although you can install SiteScope on the Tuning Console machine, we recommend installing it on a dedicated server.

➤ On the machine where SiteScope is installed, configure SiteScope to monitor the required F5 BIG-IP machines. For more information on configuring the SiteScope server, refer to the SiteScope User Guide (http://www.freshwater.com/SiteScope/UGtoc.htm).

Note: When you assign a name to a monitor, include the server name in the monitor name. This avoids any confusion as to which host the monitor belongs.

➤ Verify that SiteScope is collecting the required data from the servers it is monitoring. From the SiteScope Panel, select the monitor group polling the F5 BIG-IP server machines, and check that the monitor displays a list of server measurements in the Status column.

Configuring the F5 BIG-IP Monitor on the SiteScope Machine

1. Open the SiteScope Add Monitors to Group page, and click **F5 BIG-IP Monitor**.
2. Click **Choose Server**.

3 In the server field, enter the name or IP address of the F5 BIG-IP server that you want to monitor.

4 Enter the community for the SNMP object. The default community is public.

5 Enter the appropriate value in the Retry Delay field. The default value is 1 second.

6 Enter the appropriate value in the Timeout field. The default value is 5 seconds.

7 Click **Browse Counters** and select your desired counters.

8 Click **Choose Counters**, and then click **Add Monitor**.

Configuring the F5 BIG-IP Monitor on the Tuning Console Machine

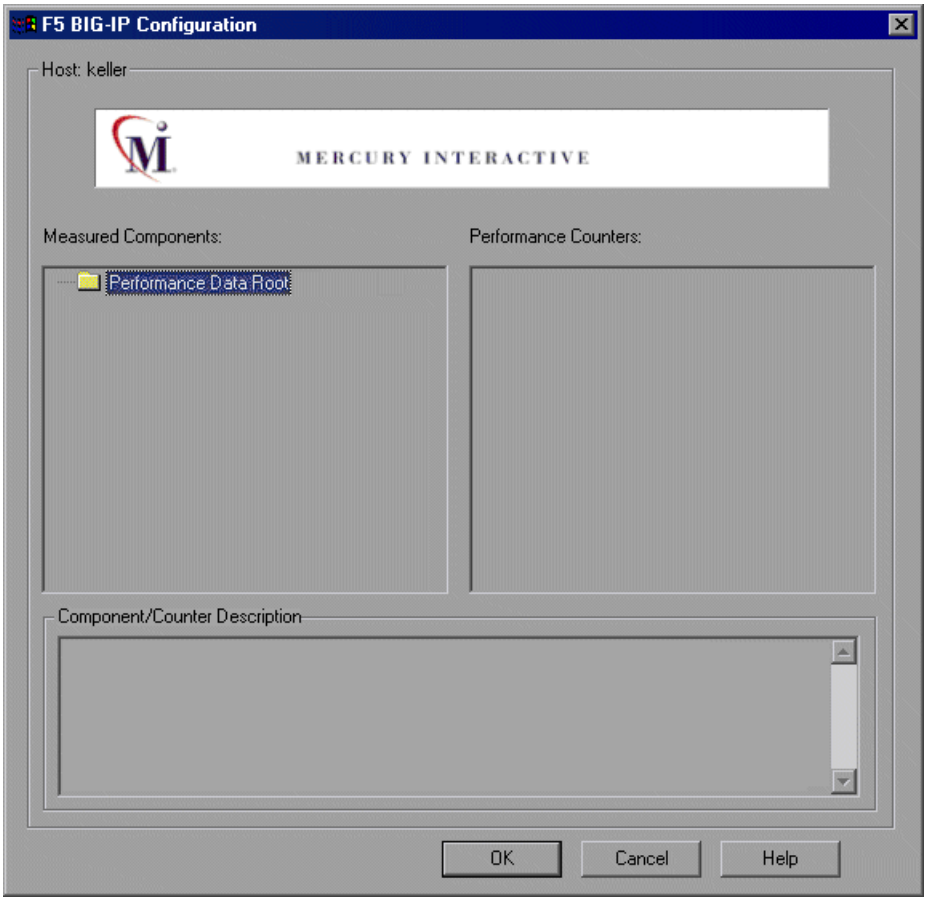
1 Click **Monitors** to open the Monitors Configuration dialog box.

2 From the Server list, choose the server running F5 BIG-IP.

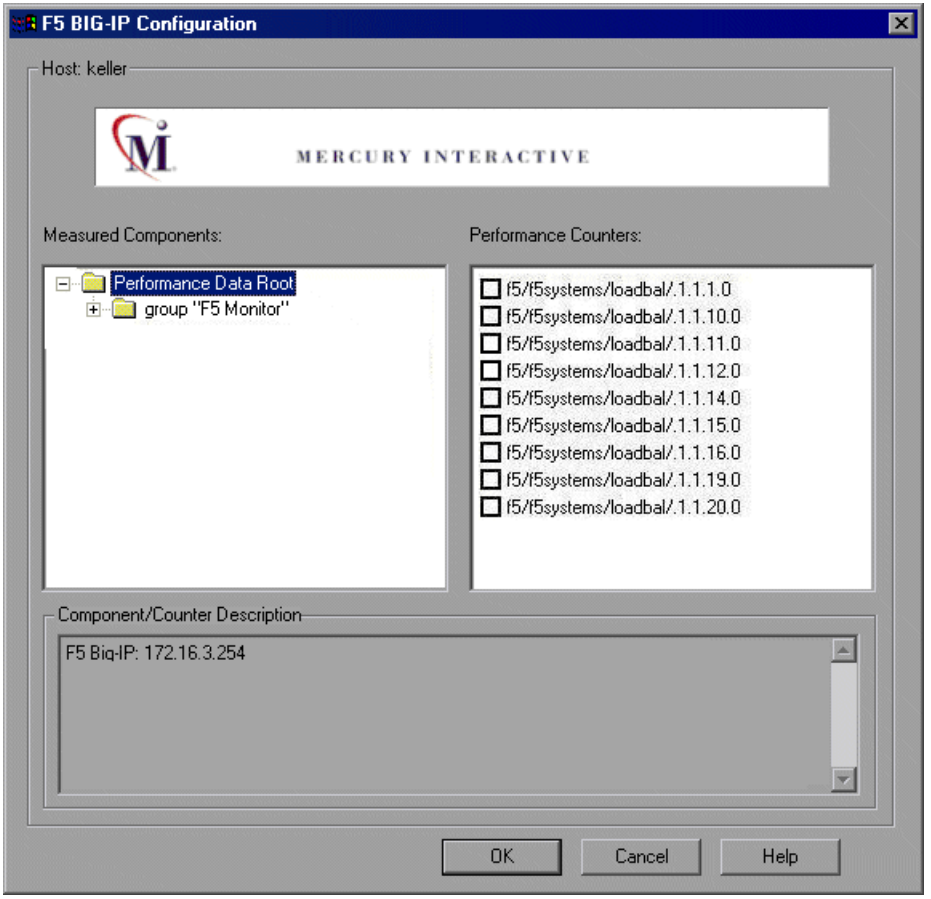
3 Click **Add Monitor**, and then check **Show All Available Monitors**. The Select Measurements to Monitor dialog box opens displaying all the available monitors.

4 In the left pane, expand the Load Balancer category, click **F5 BIG-IP**, and click **Add**.

The F5 BIG-IP Monitor Configuration dialog box is displayed.



5 In the Measured Components pane, locate the F5 BIG-IP measurement that you are monitoring and click it. The performance counters that F5 BIG-IP is monitoring on the selected component are displayed in the Performance Counters pane.



The following table shows the default counters that can be measured:

Measurement
pktsin
pksout
concur
portdeny
uptime
droppedin
droppedout
MemoryUsed

6 Check the required performance counters in the Performance Counters pane.

7 When you have selected the performance counters for the F5 BIG-IP measurements you are monitoring, click **OK** to close the F5 BIG-IP Configuration dialog box. The **Select Measurements to Monitor** dialog box appears with the selected F5 BIG-IP measurements in the **Selected Measurements** pane.

8 Click **OK** in the **Select Measurements to Monitor** dialog box, and click **OK** in the Monitors Configuration dialog box, to activate the F5 BIG-IP monitor.

Note: For troubleshooting tips and limitations, see the section on "Troubleshooting Server Resource Monitors" in the *Mercury Tuning Module Console User's Guide*.

17

Infrastructure Resources Monitoring

Using LoadRunner's Application Infrastructure Resources monitor, you can monitor network client resources for FTP, POP3, SMTP, IMAP, and DNS Vusers during a scenario or session step run and isolate client performance bottlenecks.

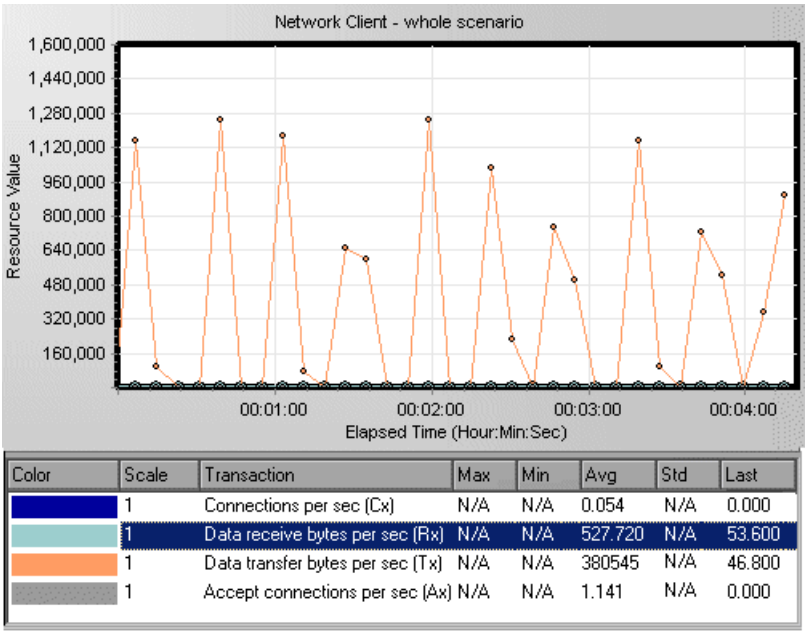
This chapter describes:

➤ Network Client Monitoring

Network Client Monitoring

The Network Client graph shows statistics for FTP, POP3, SMTP, IMAP, and DNS Vusers on the network client machine as a function of the elapsed scenario or session step time.

The x-axis represents the time that has elapsed since the start of the scenario or session step run. The y-axis represents the resource value.



Color	Scale	Transaction	Max	Min	Avg	Std	Last
	1	Connections per sec (Cx)	N/A	N/A	0.054	N/A	0.000
	1	Data receive bytes per sec (Rx)	N/A	N/A	527.720	N/A	53.600
	1	Data transfer bytes per sec (Tx)	N/A	N/A	380545	N/A	46.800
	1	Accept connections per sec (Ax)	N/A	N/A	1.141	N/A	0.000

Configuring the Network Client monitor

On the Controller:

You can view the Network Client online monitor graph by dragging it from the Infrastructure Resources Graph section in the graph tree into the right pane of the Run view. The graph appears in the graph view area.

On the Tuning Console:

1. Right-click a graph and select **Open a New Graph**. The Open a New Graph dialog box opens.
2. Click the "+" next to **Infrastructure Resources** to expand the graph tree, and select the **Network Client** graph.
3. Click **Open Graph**. The graph appears in the graph view area.

Network Client Performance Counters

The following table describes the Network Client measurements that are monitored:

Measurement	Description
Pings per sec	Number of pings per second
Data transfer bytes per sec	Number of data bytes transferred per second
Data receive bytes per sec	Number of data bytes received per second
Connections per sec	Number of connections per second
Accept connections per sec	Number of connections accepted per seconds
SSL Connections per sec	Number of SSL connections per second
SSL Data transfer bytes per sec	Number of SSL data bytes transferred per second
SSL Data receive bytes per sec	Number of SSL data bytes received per second
SSL Accept connections per sec	Number of SSL connections accepted per seconds

18

Security Monitoring

When you run certain security scripts, you can use LoadRunner's security graphs to view information about the simulated attacks on the server.

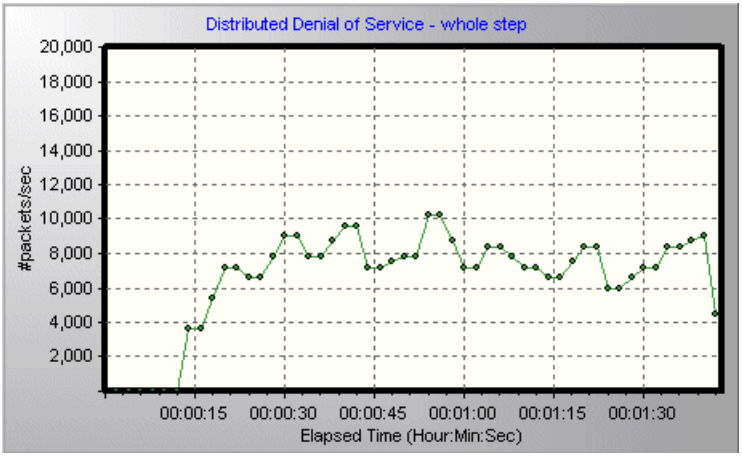
This chapter describes:

➤ Distributed Denial of Service Graph

Distributed Denial of Service Graph

The Distributed Denial of Service graph displays the number of packets per second sent to the specified target to cause denial of service.

The x-axis represents the elapsed time. The y-axis represents the number of packets sent per second.



Note: To obtain data for this graph, you need to execute one of the following canned security scripts: SYN FLOOD DDOS Attack or UDP Echo DDOS Attack.

19

Troubleshooting Online Monitors

LoadRunner monitors allow you to view the performance of the scenario or session step during execution.

The following sections describe several tips and known issues relating to the online monitors.

- Troubleshooting Server Resource Monitors
- Troubleshooting the Network Delay Monitor
- Network Considerations

Troubleshooting Server Resource Monitors

To monitor resources on a server machine, you must be able to connect to that machine. If monitoring is unsuccessful and LoadRunner cannot locate the specified server, make sure that the specified server is available. Perform a "ping" operation by typing ping <server_name> from the Controller or Tuning Console machine command line.

Once you verify that the machine is accessible, check this table for additional tips on troubleshooting the monitor.

Problem	Solution
Cannot monitor a Windows machine on a different domain, or "access denied."	To gain administrative privileges to the remote machine, perform the following from the command prompt: %net use \\<MachineName>/ user:[<Domain>\<RemoteMachineUsername>] At the password prompt, enter the password for the remote machine.

Problem	Solution
Cannot monitor an NT/Win 2000 machine (An error message is issued: "computer_name not found" or "Cannot connect to the host")	The NT/Win 2000 machine you want to monitor only enables monitoring for users with administrator privileges. To allow monitoring for non-admin users, you must grant read permission to certain files and registry entries (Microsoft tech-note number Q158438.) The required steps are: **a.** Using Explorer or File Manager, give the user READ access to: %windir%\system32\PERFCxxx.DAT %windir%\system32\PERFHxxx.DAT where xxx is the basic language ID for the system—for example, 009 for English. These files may be missing or corrupt. If you suspect this; expand these files off of the installation cd. **b.** Using REGEDT32, give the user READ access to: HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Perflib and all sub keys of that key. **c.** Using REGEDT32, give the user at least READ access to: HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\SecurePipeServers\winreg
Some Win 2000 counters cannot be monitored from an NT machine.	Run the Controller or Tuning Console on a Win 2000 machine.
Some Windows default counters are generating errors	Remove the problematic counters and add the appropriate ones using the "Add Measurement" dialog box.
You cannot get performance counters for the SQL server (version 6.5) on the monitored machine.	There is a bug in SQL server version 6.5. As a workaround, give read permission to the following registry key at the monitored machine (use regedt32): HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\MSSQLServer (Microsoft tech-note number Q170394)

Problem	Solution
The selected measurements are not displayed in the graph.	Ensure that the display file and online.exe are registered. To register the monitor dll's, without performing a full installation, run the **set_mon.bat** batch file located in **LoadRunner\bin**.
When monitoring a Windows machine, no measurements appear in the graph.	Check the built-in Windows Performance Monitor. If it is not functional, there may be a problem with the communication setup.
When monitoring a UNIX machine, no measurements appear in the graph.	Ensure that an **rstatd** is running on the UNIX machine (Refer to Chapter 4, "System Resource Monitoring.").
Cannot monitor one of the following Web servers: MS IIS, MS ASP, or ColdFusion	Refer to problem above, "Cannot monitor a Windows machine."
Cannot monitor the WebLogic (JMX) server	Open the <**LoadRunnerroot folder**>**\dat\monitors\WebLogicMon.ini** file, and search for: [WebLogicMonitor] JVM=javaw.exe Change javaw.exe to java.exe. A window containing trace information opens.

Troubleshooting the Network Delay Monitor

If monitoring is unsuccessful and LoadRunner cannot locate the source or destination machines, make sure that the specified machines are available to your machine. Perform a "ping" operation. At the command line prompt, type:

```
ping server_name
```

To check the entire network path, use the trace route utility to verify that the path is valid.

For Windows, type tracert <server_name>.

For UNIX, type traceroute <server_name>.

If the monitoring problem persists once you verify that the machines are accessible and that the network path is valid, perform the following procedures:

1. If you are using the TCP protocol, run **<LoadRunner root folder>\bin\webtrace.exe** from the source machine to determine whether the problem is related to the Controller or Tuning Console, or the WebTrace technology on which the Network Delay monitor is based. If you are using the UDP or ICMP protocols, the problem must be related to the Controller or Tuning Console and not WebTrace, since these protocols are not WebTrace technology-based.
2. If you receive results by running **webtrace.exe**, the problem is related to the Controller or Tuning Console. Verify that the source machine is not a UNIX machine, and contact Mercury's Customer Support with the following information:
   - the Controller or Tuning Console log file, **drv_log.txt**, located in the **temp** directory of the Controller or Tuning Console machine.
   - the **traceroute_server** log file, located on the source machine. Note that in LoadRunner 7.02, this information is located in **<LR root folder>\dat\<the latest mdrv log>**. In LoadRunner 7.5, this information is located in **<LR root folder>\bin\traceroute_server.log**.

➤ the debug information located in the **TRS_debug.txt** and **WT_debug.txt** files in the path directory. These files are generated by adding the following line to the [monitors_server] section of the **<LoadRunner root folder>\dat\mdrv.dat** file, and rerunning the Network monitor:

ExtCmdLine=-traceroute_debug path

3 If you do not receive results by running **webtrace.exe**, the problem is related to the WebTrace technology, on which the Network Delay monitor is based. Perform the following procedures on the source machine:

➤ Verify that the **packet.sys** file (the Webtrace driver) exists in the **WINNT\system32\drivers** directory.

➤ Check whether a driver (such as "Cloud" or "Sniffer") is installed on top of the network card driver. If so, remove it and run WebTrace again.

➤ Verify that there are administrator permissions on the machine.

➤ Using ipconfig /all, check that only one IP address is assigned to the network card. WebTrace does not know how to handle multiple IP addresses assigned to the same card (IP spoofing).

➤ Check the number of network cards installed. Run webtrace –devlist to receive a list of the available network cards.

➤ If there is more than one card on the list, run webtrace -dev <dev_name> <destination>, where <dev_name> is one of the network card names shown in the list. If you discover that WebTrace is binding to the wrong card, you can use webtrace set_device <dev_name> to set a registry key that instructs WebTrace to use a specified card instead of the default one.

➤ Verify that the network card is of the Ethernet type.

➤ Contact Mercury's Customer Support with the output of webtrace.exe –debug (for example, webtrace.exe –debug www.merc-int.com) and ipconfig /all on the machine.

Network Considerations

If you notice extraordinary delays on the network, refer to one of the following sections to increase the performance:

- Network Bandwidth Utilization
- Ethernet-bus Based Networks
- Working on a WAN or Heavily Loaded LAN

Network Bandwidth Utilization

In most load-testing scenario or session steps, the network card has little impact on scenario or session step performance. Network cards are manufactured to handle the bandwidth of the physical network layer. Packets are transferred over an Ethernet at a rate that complies with IEEE 803.x standards. If the network becomes a bottleneck, the issue is not the brand of the network card, but rather the bandwidth limitations on the physical layer (--i.e. Ethernet, FDDI, ATM, Ethernet Token-ring, etc.).

That is, instead of load testing over a T10 line, upgrade your line to DS3 (45Mbps), or T100 (100Mbps).

Below are a few tips that will help qualify the need to upgrade the network:

1) Run the performance monitor on the Vuser load generators. As the number of Vusers increases, check the network byte transfer rate for saturation. If a saturation point has been reached, do not run any more Vusers without upgrading the network—otherwise performance of Vusers will degrade. Degradation is exponential in networking environments.

2) Run the performance monitor on the server machine. Run many Vusers on several load generator machines. Check the kernel usage and network transfer rate for saturation. If saturation is reached with less than the desired Vuser load, upgrade the network.

3) Every network has a different Maximum Transmission Unit or MTU, which is set by the network administrator. The MTU is the largest physical packet size (in bytes) that a network can transmit. If a message is larger than the MTU, it is divided into smaller packets before being sent.

If clients and servers are passing large data sets back and forth, instruct the network administrator to increase the MTU to yield better bandwidth utilization. Ideally, you want the MTU to be the same as the smallest MTU of all the networks between your machine and a message's final destination.

If you send a message that is larger than one of the MTUs, it will be broken up into fragments, slowing transmission speeds. If the MTU is too high, it may cause unintended degradation. Trial and error is the only sure way of finding the optimal MTU, but there are some guidelines that can help. For example, most Ethernet networks have an MTU of 1500.

If the desired MTU reduces performance, upgrade the network or reduce the MTU to improve performance.

Ethernet-bus Based Networks

The following guidelines apply to Ethernet-bus based networks:

Networks with only 2 active machines communicating yield a maximum of 90% bandwidth utilization.

Networks with 3 active machines communicating yield a maximum of approximately 85% bandwidth utilization.

As the number of active machines on the network increases, the total bandwidth utilization decreases.

Working on a WAN or Heavily Loaded LAN

When you work with LoadRunner on a WAN or heavy loaded LAN, you may notice some unusual LoadRunner behavior, which indicates network problems. The Output window may contain messages about retries, lost packets, or message mismatch. This is because some of the messages from the Controller or Tuning Console may not be reaching the LoadRunner agent. To solve this problem, you should reduce the network traffic or improve the network bandwidth.

The following steps may help reduce network traffic:

- Click the **Run-Time Settings** button and select the **General: Log** node. Clear the **Enable logging** check box.
- Initialize all users before running them. Run them only after initialization is completed.

20

Working with Server Monitor Counters

When you configure the System Resource, Microsoft IIS, Microsoft ASP, ColdFusion, and SQL Server monitors, you are presented with a list of default counters that you can measure on the server you are monitoring. Using the procedure described below, you can create a new list of default counters by including additional counters, or deleting existing counters.

In addition, there are specific counters that are especially useful for determining server performance and isolating the cause of a bottleneck during an initial stress test on a server.

The following sections describe:

- Changing a Monitor's Default Counters
- Useful Counters for Stress Testing

Changing a Monitor's Default Counters

You can change the default counters for the System Resource, Microsoft IIS, Microsoft ASP, or SQL Server monitors by editing the **res_mon.dft** file found in the **LoadRunner\dat** directory.

To change the default counters:

1. Open a new scenario or session and click the **Run** tab (in the Controller) or **Session** tab (in the Tuning Console).
2. For each of the monitors, select the counters you want to measure.
3. Save the scenario or session and open the scenario or session **.lrs** file with an editor.

4 Copy the MonItemPlus section of the each counter you selected into the **res_mon.dft** file.

5 Count the number of new counters in the file and update the **ListCount** parameter with this number.

Useful Counters for Stress Testing

Certain counters are especially useful for determining server performance and isolating the cause of a bottleneck during an initial stress test on a server.

The following is a list of counters that are useful for monitoring Web server performance:

Object	Counter
Web Service	Maximum Connections
Web Service	Bytes Total/sec
Web Service	Current NonAnonymous Users
Web Service	Current Connections
Web Service	Not Found Errors
Active Server Pages	Requests/sec
Active Server Pages	Errors/sec
Active Server Pages	Requests Rejected
Active Server Pages	Request Not Found
Active Server Pages	Memory Allocated
Active Server Pages	Requests Queued
Active Server Pages	Errors During Script Run Time
Memory	Page Faults/sec
Server	Total Bytes/sec
Process	Private Bytes/Inetinfo

The following is a list of counters that are useful for monitoring SQL Server performance:

Object	Counter
SQLServer	User Connections
SQLServer	Cache Hit Ratio
SQLServer	Net-Network Reads/sec
SQLServer	I/O-Lazy Writes/sec
SQLServer-Locks	Total Blocking Locks
PhysicalDisk	Disk Queue Length

The following is a list of counters that are useful for monitoring both Web and SQL server performance:

Object	Counter
Processor	% Total Processor Time
PhysicalDisk	% Disk Time
Memory	Available Bytes
Memory	Pool Nonpaged Bytes
Memory	Pages/sec
Memory	Committed Bytes
System	Total Interrupts/sec
Object	Threads
Process	Private Bytes:_Total

Note: The % Disk Time counter requires that you run the diskperf -y utility at the command prompt and reboot your machine.

Index

D

E

F

H

I

J

L

M

N

O

P

R

S

T

U

V

W

Host Resolution Functions Copyright Agreement