

**LoadRunner®**  
*Analysis User's Guide*  
Version 7.02



MERCURY INTERACTIVE

LoadRunner Analysis User's Guide, Version 7.02

© Copyright 2001 by Mercury Interactive Corporation

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

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

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

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

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

Mercury Interactive Corporation  
1325 Borregas Avenue  
Sunnyvale, CA 94089 USA

---

# Table of Contents

<b>Welcome to LoadRunner</b> .....	vii
Online Resources .....	vii
LoadRunner Documentation Set.....	viii
Using the LoadRunner Documentation Set .....	ix
Typographical Conventions.....	xi
<b>Chapter 1: Introducing LoadRunner Analysis</b> .....	1
About the Analysis .....	1
Creating Analysis Sessions .....	2
Starting the Analysis.....	3
Collating Execution Results .....	3
Viewing Summary Data .....	4
Analysis Graphs .....	5
Opening Analysis Graphs.....	7
<b>Chapter 2: Working with LoadRunner Analysis</b> .....	9
Working with Analysis Graphs .....	9
Applying a Filter to a Graph .....	16
Grouping and Sorting Results .....	19
Viewing the Legend .....	21
Viewing the Data as a Spreadsheet and as Raw Data .....	22
Configuring Display Options .....	25
Setting General and Database Options .....	29
Working with Templates.....	33
<b>Chapter 3: Vuser Graphs</b> .....	37
About Vuser Graphs .....	37
Running Vusers Graph .....	38
Vuser Summary Graph .....	39
Rendezvous Graph .....	40

<b>Chapter 4: Error Graphs</b> .....	41
About Error Graphs .....	41
Error Statistics Graph .....	42
Errors per Second Graph .....	43
<b>Chapter 5: Transaction Graphs</b> .....	45
About Transaction Graphs .....	45
Transaction Response Time - Average Graph .....	46
Transactions per Second Graph .....	48
Total Transactions per Second .....	49
Transaction Summary Graph .....	50
Transaction Performance Summary Graph .....	51
Transaction Response Time - Under Load Graph .....	52
Transaction Response Time - Percentile Graph .....	53
Transaction Response Time - Distribution Graph .....	54
<b>Chapter 6: Web Resource Graphs</b> .....	55
About Web Resource Graphs.....	55
Hits per Second Graph .....	56
Throughput Graph .....	57
HTTP Status Code Summary Graph .....	58
HTTP Responses per Second Graph .....	59
Pages Downloaded per Second Graph .....	62
<b>Chapter 7: Web Page Breakdown Graphs</b> .....	65
About Web Page Breakdown Graphs .....	66
Activating the Web Page Breakdown Graphs .....	67
Page Component Breakdown Graph .....	69
Page Component Breakdown (Over Time) Graph .....	71
Page Download Time Breakdown Graph .....	73
Page Download Time Breakdown (Over Time) Graph .....	77
Time to First Buffer Breakdown Graph .....	79
Time to First Buffer Breakdown (Over Time) Graph .....	81
Downloaded Component Size Graph .....	83
<b>Chapter 8: User-Defined Data Point Graphs</b> .....	85
About User-Defined Data Point Graphs.....	85
Data Points (Sum) Graph .....	86
Data Points (Average) Graph .....	87

<b>Chapter 9: System Resource Graphs</b> .....	89
About System Resource Graphs.....	89
Windows Resources Graph .....	90
UNIX Resources Graph.....	94
SNMP Resources Graph .....	96
TUXEDO Resources Graph .....	97
<b>Chapter 10: Network Monitor Graphs</b> .....	99
About Network Monitoring .....	99
Understanding Network Monitoring.....	100
Network Delay Time Graph .....	101
Network Sub-Path Time Graph .....	102
Network Segment Graph .....	103
Verifying the Network as a Bottleneck.....	104
<b>Chapter 11: Firewall Server Monitor Graphs</b> .....	105
About Firewall Server Monitor Graphs .....	105
Check Point FireWall-1 Server Graph .....	106
<b>Chapter 12: Web Server Resource Graphs</b> .....	107
About Web Server Resource Graphs.....	107
Apache Server Graph .....	108
Microsoft Information Internet Server (IIS) Graph .....	110
Netscape Enterprise Server Graph .....	112
<b>Chapter 13: Web Application Server Resource Graphs</b> .....	115
About Web Application Server Resource Graphs.....	115
Web Application Server Resource Graphs.....	116
<b>Chapter 14: Database Server Resource Graphs</b> .....	143
About Database Server Resource Graphs.....	143
SQL Server Graph .....	144
Oracle Graph .....	146
<b>Chapter 15: Streaming Media Graphs</b> .....	149
About Streaming Media Graphs .....	149
RealPlayer Client Graph .....	150
RealPlayer Server Graph .....	152
Windows Media Server Graph .....	153
<b>Chapter 16: Cross Result and Merged Graphs</b> .....	155
About Cross Result and Merged Graphs .....	155
Cross Result Graphs.....	156
Generating Cross Result Graphs .....	158
Merging Graphs .....	159
Creating a Merged Graph.....	161

<b>Chapter 17: Understanding Analysis Reports</b> .....	163
About Analysis Reports.....	164
Viewing Summary Reports .....	165
Creating HTML Reports .....	166
Working with Transaction Reports .....	167
Scenario Execution Report .....	169
Failed Transaction Report .....	170
Failed Vuser Report .....	171
Data Point Report .....	172
Detailed Transaction Report .....	173
Transaction Performance by Vuser Report .....	174
<b>Chapter 18: Managing Results Using TestDirector</b> .....	175
About Managing Results Using TestDirector .....	175
Opening a Connection to a TestDirector Project .....	176
Creating a New Session Using TestDirector .....	178
Opening an Existing Session Using TestDirector .....	180
Saving Sessions to a TestDirector Project .....	181
<b>Chapter 19: Interpreting Analysis Graphs</b> .....	183
Analyzing Performance Under Load .....	183
Breaking Down Transaction Performance Response Time .....	185
Using the Web Page Breakdown Graphs.....	187
Identifying Network Problems .....	188
Identifying Web Server-Related Problems.....	189
Identifying Database and Application Server Problems .....	192
Comparing Scenario Results .....	193
<b>Index</b> .....	195

---

# Welcome to LoadRunner

Welcome to LoadRunner, Mercury Interactive's tool for testing the performance of applications. 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 that are distributed over a network. Using a minimum of hardware resources, these Virtual Users 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 Interactive's Customer Support Web site for updates to LoadRunner online books. The URL for this Web site is <http://support.mercuryinteractive.com>.

**LoadRunner Online 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 Interactive's Customer Support Web site for updates to the *LoadRunner Online 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 Interactive's Customer Support Web site for updates to LoadRunner help files.

**Technical Support Online** uses your default web browser to open Mercury Interactive's Customer Support web site. The URL for this Web site is *<http://support.mercuryinteractive.com>*.

**Support Information** presents the locations of Mercury Interactive's Customer Support Web site and home page, and a list of Mercury Interactive's offices around the world.

**Mercury Interactive on the Web** uses your default Web browser to open Mercury Interactive's home page. The URL for this Web site is *<http://www.mercuryinteractive.com>*.

## LoadRunner Documentation Set

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

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



## Using the LoadRunner Documentation Set

The LoadRunner documentation set consists of one installation guide, a Controller user's guide, an Analysis user's guide, and three guides 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

### Controller User's Guide

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

The *LoadRunner Controller User's Guide (Windows)* 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.

### Analysis User's Guide

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

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

### Guides for Creating Vuser Scripts

The LoadRunner documentation pack has two guides that describe how to create Vuser scripts:

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

<b>For information on</b>	<b>Look here...</b>
Installing LoadRunner	<i>LoadRunner Installation Guide</i>
The LoadRunner testing process	<i>LoadRunner Controller User's Guide (Windows)</i>
Creating Vuser scripts	<i>Creating Vuser Scripts</i> guide
Creating and running scenarios	<i>LoadRunner Controller User's Guide (Windows)</i>
Analyzing test results	<i>LoadRunner Analysis User's Guide</i>

## Typographical Conventions

This book uses the following typographical conventions:

<b>1, 2, 3</b>	Bold numbers indicate steps in a procedure.
►	Bullets indicate options and features.
>	The greater than sign separates menu levels (for example, <b>File &gt; Open</b> ).
<b>Stone Sans</b>	The <b>Stone Sans</b> font indicates names of interface elements in a procedure (for example, “Click the <b>Run</b> button.”).
<i>Italics</i>	<i>Italic</i> text indicates names (for example, names of variables or books).
Helvetica	The Helvetica font is used for examples and strings that are to be typed in literally.
<>	Angle brackets enclose a part of a URL address that needs to be typed in.
[ ]	Square brackets enclose optional parameters.
...	In a line of syntax, an ellipsis indicates that more items of the same format may be included.



# 1

---

## Introducing LoadRunner Analysis

LoadRunner Analysis provides graphs and reports to help you analyze the performance of your system. These graphs and reports summarize the scenario execution.

This chapter describes:

- Creating Analysis Sessions
- Starting the Analysis
- Collating Execution Results
- Viewing Summary Data
- Analysis Graphs
- Opening Analysis Graphs

### About the Analysis

During scenario execution, Vusers generate result data as they perform their transactions. To monitor the scenario performance *during* test execution, use the online monitoring tools described in the *LoadRunner Controller User's Guide (Windows)*. To view a summary of the results *after* test execution, you can use one or more of the following tools:

- The **Vuser log files** contain a full trace of the scenario run for each Vuser. These files are located in the scenario results directory. (When you run a Vuser script in standalone mode, these files are placed in the Vuser script directory.) For more information on Vuser log files, refer to the *Creating Vuser Scripts* guide.

- ▶ The **Controller Output window** displays information about the scenario run. If your scenario run fails, look for debug information in this window. For more information, see the *LoadRunner Controller User's Guide (Windows)*.
- ▶ The **Analysis graphs** help you determine system performance and provide information about transactions and Vusers. You can compare multiple graphs by combining results from several scenarios or merging several graphs into one.
- ▶ The **Graph Data** and **Raw Data** views display the actual data used to generate the graph in a spreadsheet format. You can copy this data into external spreadsheet applications for further processing.
- ▶ The **Report** utilities enable you to view a Summary HTML report for each graph or a variety of Performance and Activity reports.

This chapter provides an overview of the graphs and reports that can be generated through the Analysis.

## Creating Analysis Sessions

When you run a scenario, data is stored in a result file with an *.lrr* extension. The Analysis is the utility that processes the gathered result information and generates graphs and reports.

When you work with the Analysis utility, you work within a *session*. An Analysis session contains at least one set of scenario results (*lrr* file). The Analysis stores the display information and layout settings for the active graphs in a file with an *.lra* extension.

## Starting the Analysis

You can open the Analysis through the LoadRunner program group as an independent application, or directly from the Controller. To open the new Analysis utility as an independent application, choose **Analysis** from the LoadRunner Program Group.

To open the Analysis directly from the Controller, select **Results > Analyze Results**. This option is only available after running a scenario. The Analysis takes the latest result file from the current scenario, and opens a new session using these results. You can also instruct the Controller to automatically open the Analysis after it completes scenario execution by selecting **Results > Auto Load Analysis**.

When creating a new session, the Analysis prompts you for the result file (*.lrr* extension) that you wish to include in the session. To open an existing session, you specify an Analysis Session file (*.lra* extension).

## Collating Execution Results

When you run a scenario, by default all Vuser information is stored locally on each Vuser host. After scenario execution the results are automatically *collated* or consolidated—results from all of the hosts are transferred to the results directory. You disable automatic collation by choosing **Results > Auto Collate Results** from the Controller window, to clear the check mark adjacent to the option. To manually collate results, choose **Results > Collate Results**. If your results have not been collated, the Analysis will automatically collate the results before generating the analysis data. For more information about collating results, see the *LoadRunner Controller User's Guide (Windows)*.

## Viewing Summary Data

In large scenarios, with results exceeding 100 MBs, it can take a while for the Analysis to process the data. While LoadRunner is processing the complete data, you can view a summary of the data collected. To view the summary data, choose **Tools > Options**, and select the **Result Collection** tab. Select **Display summary data while generating complete data** if you want the Analysis to process the complete data graphs while you view the summary data, or select **Generate summary data only** if you do not want LoadRunner to process the complete Analysis data.

The following graphs are not available when viewing summary data only: Rendezvous, Data Point (Sum), and Network Monitor graphs.

---

**Note:** The complete data graphs can be sorted, filtered, and manipulated. The summary graphs contain only general information, such as transaction names and times, and cannot be filtered or manipulated.

---



## Analysis Graphs

The Analysis graphs are divided into the following categories:

- Vusers
- Errors
- Transactions
- Web Resources
- Web Page Breakdown
- User-Defined Data Points
- System Resources
- Network Monitor
- Firewalls
- Web Server Resources
- Web Application Server Resources
- Database Server Resources
- Streaming Media
- Cross Result

**Vuser** graphs display information about Vuser states and other Vuser statistics. For more information, see Chapter 3, “Vuser Graphs.”

**Error** graphs provide information about the errors that occurred during the scenario. For more information, see Chapter 4, “Error Graphs.”

**Transaction** graphs and reports provide information about transaction performance and response time. For more information, see Chapter 5, “Transaction Graphs.”

**Web Resource** graphs provide information about the throughput, hits per second, HTTP responses per second, and downloaded pages per second for Web Vusers. For more information, see Chapter 6, “Web Resource Graphs.”

**Web Page Breakdown** graphs provide information about the size and download time of each Web page component. For more information, see Chapter 7, "Web Page Breakdown Graphs."

**User-Defined Data Point** graphs display information about the custom data points that were gathered by the online monitor. For more information, see Chapter 8, "User-Defined Data Point Graphs."

**System Resource** graphs show statistics relating to the system resources that were monitored during the scenario using the online monitor. This category also includes graphs for SNMP and TUXEDO monitoring. For more information, see Chapter 9, "System Resource Graphs."

**Network Monitor** graphs provide information about the network delays. For more information, see Chapter 10, "Network Monitor Graphs."

**Firewall** graphs provide information about firewall server resource usage. For more information, see Chapter 11, "Firewall Server Monitor Graphs."

**Web Server Resource** graphs provide information about the resource usage for the Netscape, IIS, and Apache Web servers. For more information see Chapter 12, "Web Server Resource Graphs."

**Web Application Server Resource** graphs provide information about the resource usage for various Web application servers such as BroadVision, ColdFusion, SilverStream, ASP, WebLogic, and WebSphere. For more information see Chapter 13, "Web Application Server Resource Graphs."

**Database Server Resource** graphs provide information about SQL Server and Oracle database resources. For more information, see Chapter 14, "Database Server Resource Graphs."

**Streaming Media** graphs provide information about RealPlayer Client, RealPlayer Server, and Windows Media Server resource usage. For more information, see Chapter 15, "Streaming Media Graphs."

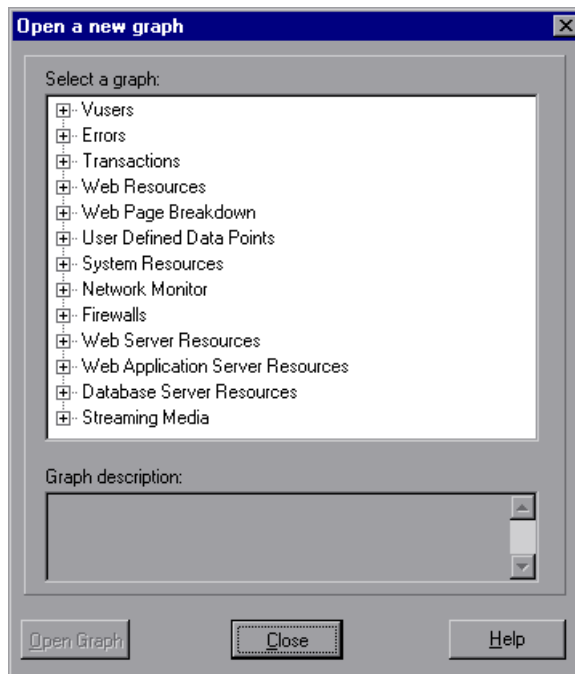
**Cross Result** graphs let you compare the results from several scenario runs. This feature can be useful for benchmarking hardware to determine how a superior system will improve performance. For more information, see Chapter 16, "Cross Result and Merged Graphs."

## Opening Analysis Graphs

By default, LoadRunner displays only the Summary Report in the graph tree view. You can open the Analysis graphs using the Open a New Graph dialog box.

**To open a new graph:**

- 1 Select **Graph > Add Graph**, or click **<New Graph>** in the graph tree view. The Open a New Graph dialog box opens.



- 2 Click the "+" to expand the graph tree, and select a graph. You can view a description of the graph in the Graph Description box.
- 3 Click **Open Graph**. The Analysis generates the graph you selected and adds it to the graph tree view. The graph is displayed in the right pane of the Analysis.

To display an existing graph in the right pane of the Analysis, select the graph in the graph tree view.



# 2

---

## Working with LoadRunner Analysis

The Analysis contains several utilities that allow you to customize result data and view this data in the most beneficial way possible.

This chapter describes:

- ▶ Working with Analysis Graphs
- ▶ Applying a Filter to a Graph
- ▶ Grouping and Sorting Results
- ▶ Viewing the Legend
- ▶ Viewing the Data as a Spreadsheet and as Raw Data
- ▶ Configuring Display Options
- ▶ Setting General and Database Options
- ▶ Working with Templates

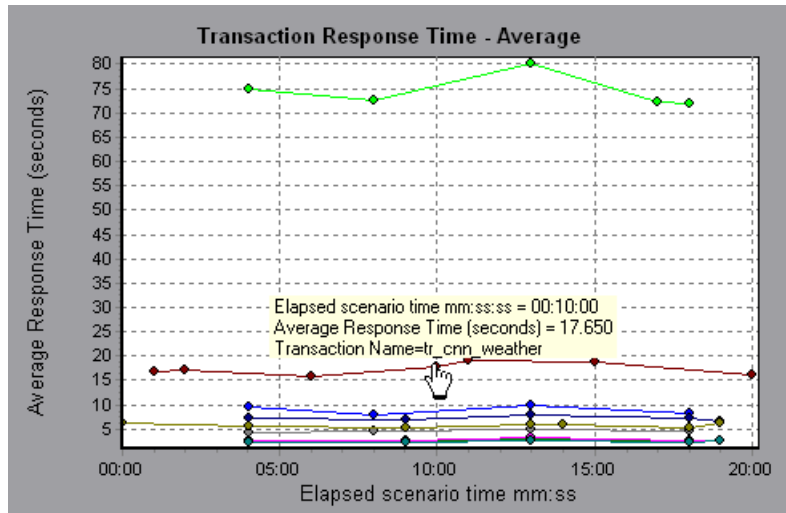
### Working with Analysis Graphs

You can use the following utilities in viewing the Analysis graphs:

- ▶ Determining a Point's Coordinates
- ▶ Performing a Drill Down
- ▶ Enlarging a Section of a Graph
- ▶ Changing the Granularity of the X-axis

## Determining a Point's Coordinates

You can determine the coordinates and values at specific points in a graph. Place the cursor over the point you want to evaluate. The Analysis displays the axis values and other grouping information.



## Performing a Drill Down

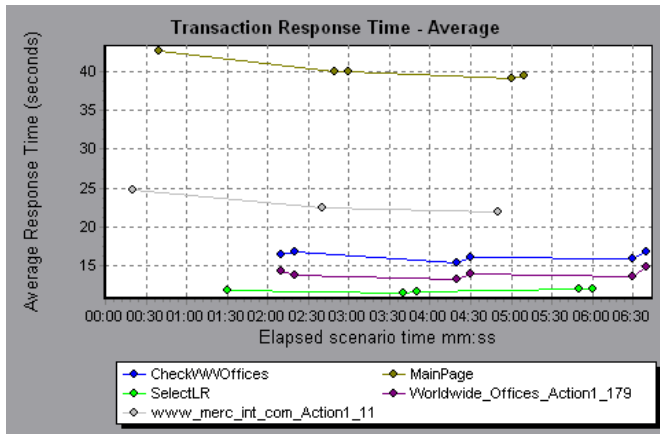
The Analysis provides another filtering mechanism called *drill down*. Drill down lets you focus on a specific measurement within your graph. You select one measurement within the graph and display it according to desired grouping. The available groupings are Vuser ID, Group Name, Vuser End Status, etc. depending on the graph. For example, the **Transaction Response Time - Average** graph shows one line per transaction. To determine the response time for each Vuser, you drill down on one transaction and sort it according to Vuser ID. The graph displays a separate line for each Vuser's transaction response time.

---

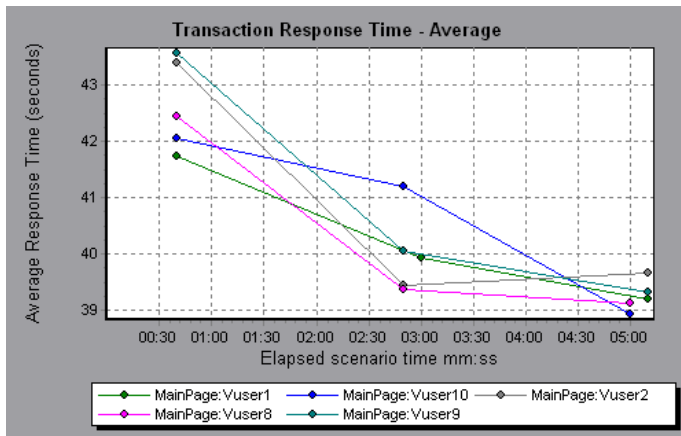
**Note:** The drill down feature is not available for the Web Page Breakdown graph.

---

In the following example, the graph shows a line for each of the five transactions.



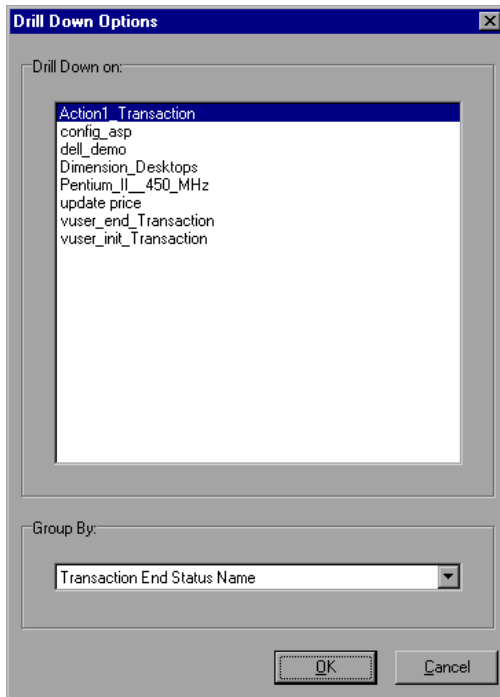
In a drill down for the MainPage transaction per Vuser ID, the graph displays the response time only for the MainPage transaction, one line per Vuser.



You can see from the graph that the response time was longer for some Vusers than for others.

**To drill down in a graph:**

- 1 Right-click on a line, bar, or segment within the graph, and select **Drill Down**. The Drill Down Options dialog box opens listing all of the measurements in the graph.



- 2 Select a measurement for drill down.
- 3 From the Group By box, select a group by which to sort.
- 4 Click **OK**. The Analysis drills down and displays the new graph.
- 5 To undo the last drill down settings, choose **Undo Set Filter/Group By** from the right-click menu.
- 6 To perform additional drill downs, repeat steps 1 to 4.
- 7 To clear all filter and drill down settings, choose **Clear Filter/Group By** from the right-click menu.



## Enlarging a Section of a Graph

Graphs initially display data representing the entire duration of the scenario. You can enlarge any section of a graph to zoom in on a specific period of the scenario run. For example, if a scenario ran for ten minutes, you can enlarge and focus on the scenario events that occurred between the second and fifth minutes.

### To zoom in on a section of the graph:

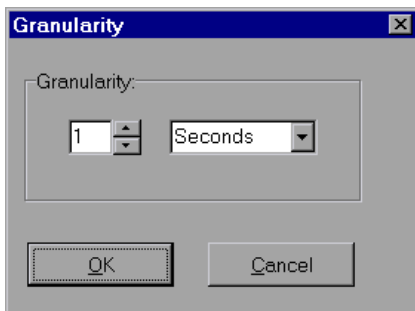
- 1 Click inside a graph.
- 2 Move the mouse pointer to the beginning of the section you want to enlarge, but not over a line of the graph.
- 3 Hold down the left mouse button and draw a box around the section you want to enlarge.
- 4 Release the left mouse button. The section is enlarged.
- 5 To restore the original view, choose **Clear Display Option** from the right-click menu.

## Changing the Granularity of the X-axis

You can make the graphs easier to read and analyze by changing the granularity (scale) of the x-axis. The maximum granularity is half of the graph's time range. To ensure readability and clarity, the Analysis automatically adjusts the minimum granularity of graphs with ranges of 500 seconds or more.

**To change the granularity of a graph:**

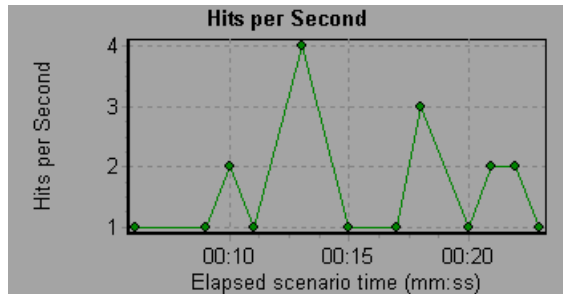
- 1** Click inside a graph.
- 2** Select **View > Set Granularity**, or click the **Set Granularity** button. The Granularity dialog box opens.



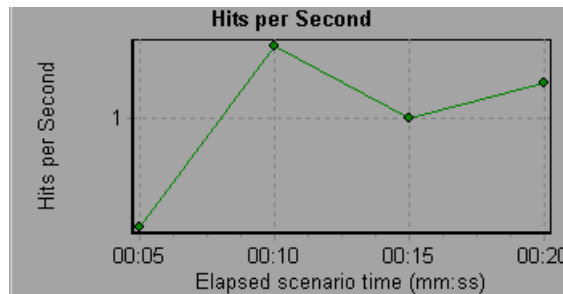
- 3** Enter a new granularity value in milliseconds, minutes, seconds, or hours.
- 4** Click **OK**.

In the following example, the Hits per Second graph is displayed using different granularities. The y-axis represents the number of hits per second within the granularity interval. For a granularity of 1, the y-axis shows the number of hits per second for each one second period of the scenario. For a

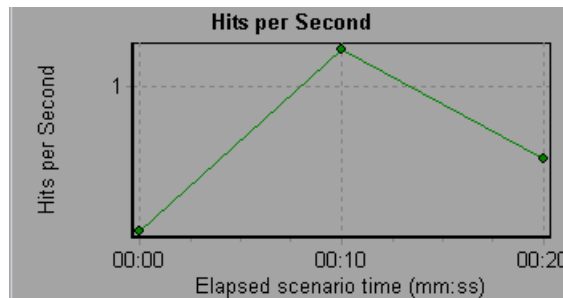
granularity of 5, the y-axis shows the number of hits per second for every five-second period of the scenario.



*Granularity=1*



*Granularity=5*



*Granularity=10*

In the above graphs, the same scenario results are displayed in a granularity of 1, 5, and 10. The lower the granularity, the more detailed the results. For example, using a low granularity as in the upper graph, you see the intervals in which no hits occurred. It is useful to use a higher granularity to study the overall User behavior throughout the scenario.

By viewing the same graph with a higher granularity, you can easily see that overall, there was an average of approximately 1 hit per second.

## Applying a Filter to a Graph

When viewing graphs in an Analysis session, you can specify to display only the desired information. If, by default, all transactions are displayed for the entire length of the scenario, you can filter a graph to show fewer transactions for a specific segment of the scenario. For example, you can display four transactions beginning from five minutes into the scenario and ending three minutes before the end of the scenario.

The filter conditions differ for each type of graph. The filter conditions also depend on your scenario. For example, if you only had one group or one load generator machine in your scenario, the Group Name and Load Generator Name filter conditions do not apply.

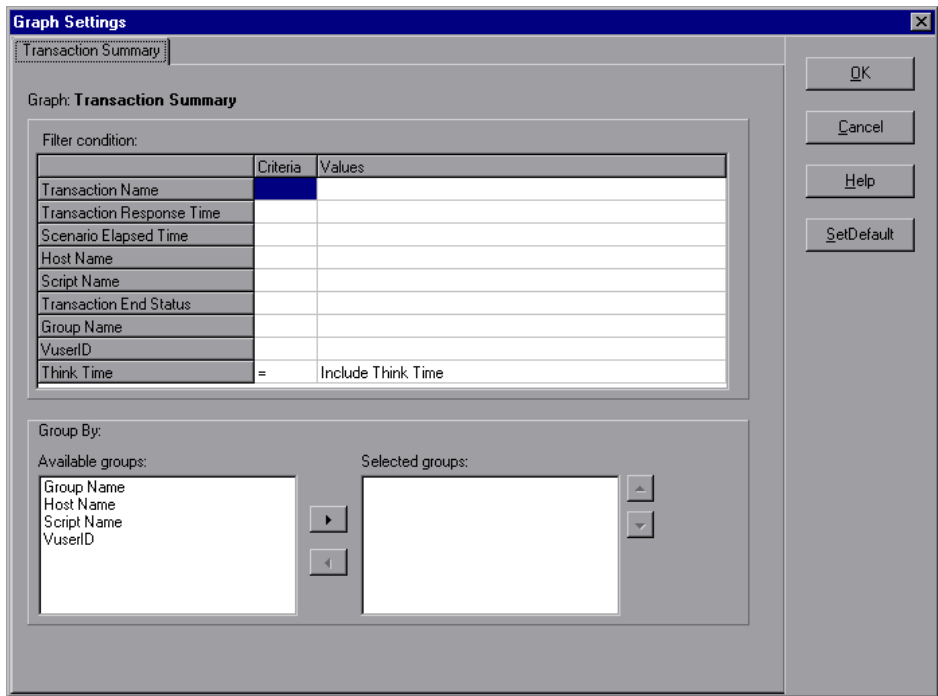
You can also activate global filter conditions that will apply to all graphs in the session. The available global filter conditions are a combination of the filter conditions that apply to the individual graphs.

Note that you can also filter merged graphs. The filter conditions for each graph are displayed on separate tabs.

### **To set a filter condition for an individual graph:**

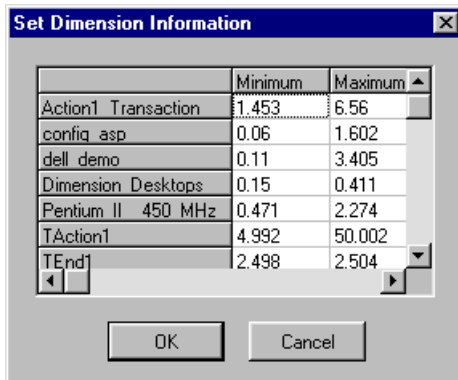
- 1** Select the graph you want to filter by clicking the graph tab or clicking the graph name in the tree view.

**2** Choose **View > Set Filter/Group By**. The Graph Settings dialog box opens.

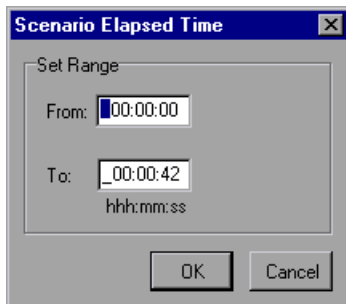


- 3** Click the Criteria box of the condition(s) you want to set, and select either "=" or "<>" (does not equal) from the drop down list.
- 4** Click the Values box of the filter condition you want to set, and select a value from the drop down box. For several filter conditions, a new dialog box opens.

- 5 For the Transaction Response Time filter condition, the Set Dimension Information dialog box opens. Specify a minimum and maximum transaction response time for each transaction.



- 6 For the Scenario Elapsed Time filter condition, the Scenario Elapsed Time dialog box opens. Specify the start and end time for the graph in hours:minutes:seconds format. The time is relative to the start of the scenario.



- 7 For Rendezvous graphs, while setting the *Number of Released Vusers* condition, the Set Dimension Information dialog box opens. Specify a minimum and maximum number of released Vusers.
- 8 For all graphs that measure resources (Web Server, Database Server, etc.), when you set the *Resource Value* condition, the Set Dimension Information dialog box opens displaying a full range of values for each resource. Specify a minimum and maximum value for the resource.

- 9 Click **OK** to accept the settings and close the Graph Settings dialog box.

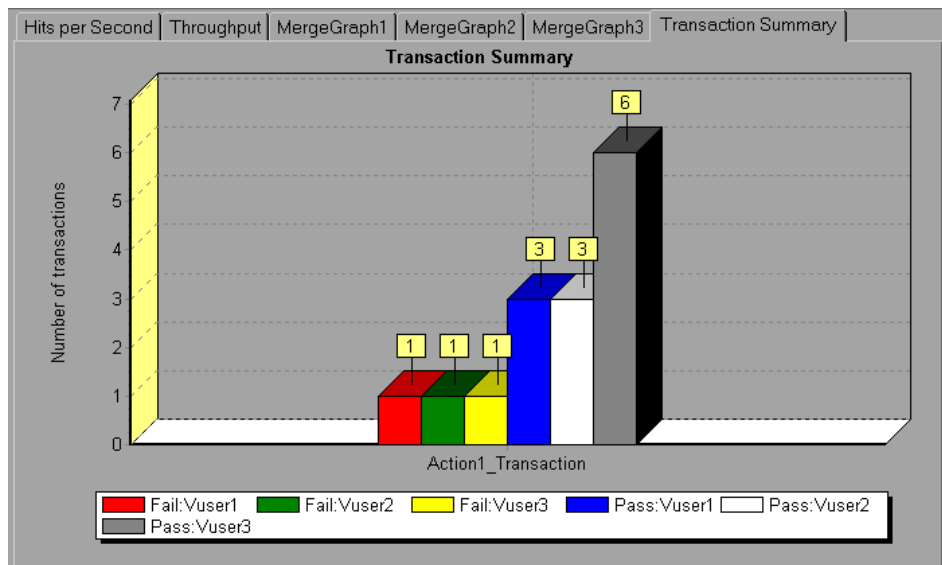
To apply a global filter condition for all graphs in the session (both those displayed and those that have not yet been opened), choose **File > Set Global Filter** or click the **Set Global Filter** button, and set the desired filters.



## Grouping and Sorting Results

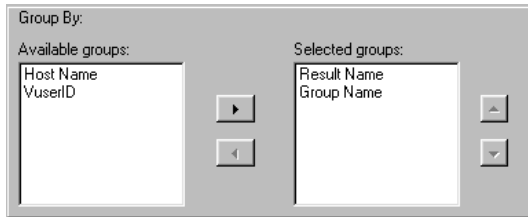
When viewing a graph, you can group the data in several ways. For example, Transaction graphs can be grouped by the Transaction End Status. The Vuser graphs can be grouped by the Scenario Elapsed Time, Vuser End Status, Vuser Status, and VuserID.

You can also sort by more than one group—for example by Vuser ID and then Vuser status. The results are displayed in the order in which the groups are listed. You can change the order in which things are grouped by rearranging the list. The following graph shows the Transaction Summary grouped according to Vusers.



**To sort the graph data according to groups:**

- 1** Select the graph you want to sort by clicking the graph tab or clicking the graph name in the tree view.
- 2** Choose **View > Set Filter/Group By**. The Graph Settings dialog opens.
- 3** In the Available Groups box, select the group by which you want to sort the results.









- 4** Click the right-facing arrow to move your selection to the Selected Groups box.
- 5** To change the order in which the results are grouped, select the group you want to move and click the up or down arrow until the groups are in the desired order.
- 6** To remove an existing grouping, select an item in the Selected Groups box and click the left-facing arrow to move it to the Available Groups box.
- 7** Click **OK** to close the dialog box.



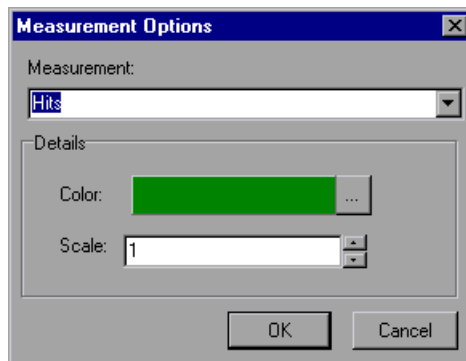
## Viewing the Legend

The Legend tab displays the color, scale, minimum, maximum, average, median, and standard deviation of each measurement appearing in the graph.

Legend   Graph Details   User Notes   Graph Data   Raw Data							
Color	Scale	Measurement	Minimum	Maximum	Average	Median	Std. Deviation
<input checked="" type="checkbox"/> 	1	air_head2_gif_Action1_159	0.176	0.657	0.247	0.19	0.155
<input checked="" type="checkbox"/> 	1	beige_gif_Action1_183	0.167	0.434	0.24	0.194	0.092
<input checked="" type="checkbox"/> 	1	book_gif_Action1_167	0.198	0.812	0.291	0.2	0.2
<input checked="" type="checkbox"/> 	1	bullet1_gif_Action1_234	0.11	0.245	0.164	0.17	0.04
<input checked="" type="checkbox"/> 	1	butt_ast_gif_Action1_207	0.13	0.238	0.177	0.176	0.027
<input checked="" type="checkbox"/> 	1	Click_here_Action1_107	0.187	0.827	0.288	0.201	0.22

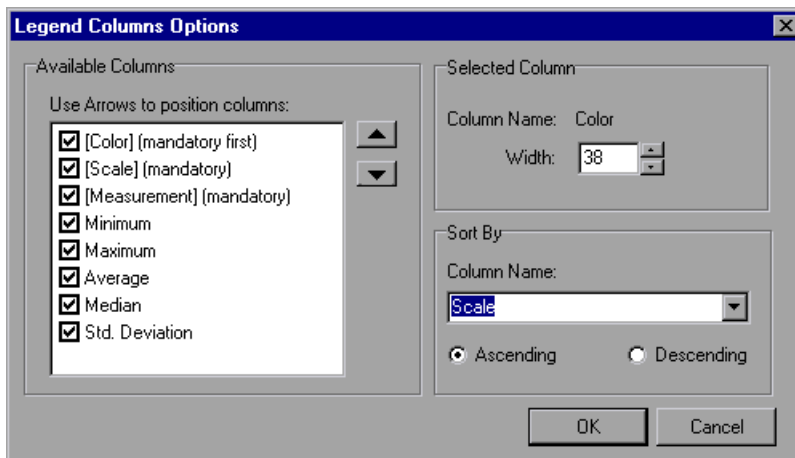
The Legend tab shortcut menu (right-click) has the following additional features:

- **Show/Hide:** Displays or hides a measurement in the graph.
- **Show only selected:** Displays the highlighted measurement only.
- **Show all:** Displays all the available measurements in the graph.
- **Configure measurements:** Opens the Measurement Options dialog box in which you can set the color and scale of the measurement you selected.



- **Animate selected line:** Displays the selected measurement as a flashing line.

- ▶ **Sort by [SELECTED] column:** Sorts the measurements according to the selected column, in ascending or descending order.
- ▶ **Configure columns:** Opens the Legend Columns Options dialog box in which you can choose the columns you want to view, the width of each column, and the way you want to sort the columns.



## Viewing the Data as a Spreadsheet and as Raw Data

The Analysis allows you to view the data in a grid form in two ways:

- ▶ **Spreadsheet:** The graph values displayed in a grid.
- ▶ **Raw Data:** The actual raw data collected during the scenario.

## Spreadsheet View

You can view the graph displayed by the Analysis in spreadsheet format. The **Graph Data** tab below the graph, displays the data in a grid format.

Transaction Name	Minimum	Average	Maximum
air_head2_gif_Action1_159	0.12	0.304	3.105
beige_gif_Action1_183	0.111	0.27	1.462
book_gif_Action1_167	0.19	0.373	3.314
bullet1_gif_Action1_234	0.11	0.184	0.701
butt_ast_gif_Action1_207	0.13	0.184	0.311
Click_here_Action1_107	0.16	0.33	2.124

The first column displays the values of the x-axis. The following columns show the y-axis values for each transaction.

If there are multiple values for the y-axis, as in the Transaction Performance Summary graph (minimum, average, and maximum), all of the plotted values are displayed. If you filter out a transaction, it will not appear in the view.

The Spreadsheet shortcut menu (right-click) has the following additional features:

- **Copy All:** You can copy the spreadsheet to the clipboard in order to paste it into an external spreadsheet program.
- **Save As:** You can save the spreadsheet data to an Excel file. Once you have the data in Excel, you can generate your own customized graphs.

## Raw Data View

You can view the actual raw data collected during test execution for the active graph. The Raw Data view is not available for all graphs.

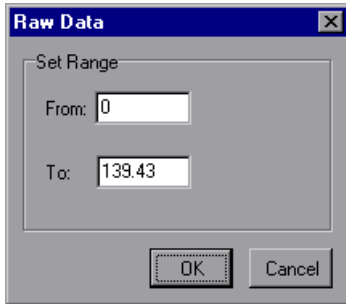
Viewing the raw data can be especially useful for:

- determining specific details about a peak—for example, which Vuser was running the transaction that caused the peak value(s).
- performing a complete export of unprocessed data for your own spreadsheet application.

To display a graph's Raw Data view:



- 1 Choose **View > View Raw Data** or click the **Raw Data** button. The Raw Data dialog box opens.



- 2 Specify a time range—the entire graph (default) or a specific range of time—and click **OK**.
- 3 Click the **Raw Data** tab below the graph. The Analysis displays the raw data in a grid directly below the active graph.

Transaction Name	Transaction Response Time	Scenario Elapsed Time	Transaction End Status Name	VuserID
Dimension_Desktops...	0.25	18	Pass	Vuser1
Pentium_II_450_MHz	2.274	15	Pass	Vuser1
Action1_Transaction	6.5	13	Pass	Vuser1
dell_demo	1.502	18	Pass	Vuser1
Action1_Transaction	4.106	17	Pass	Vuser1
dell_demo	0.17	13	Pass	Vuser1

- 4 To show a different range, repeat the above procedure.

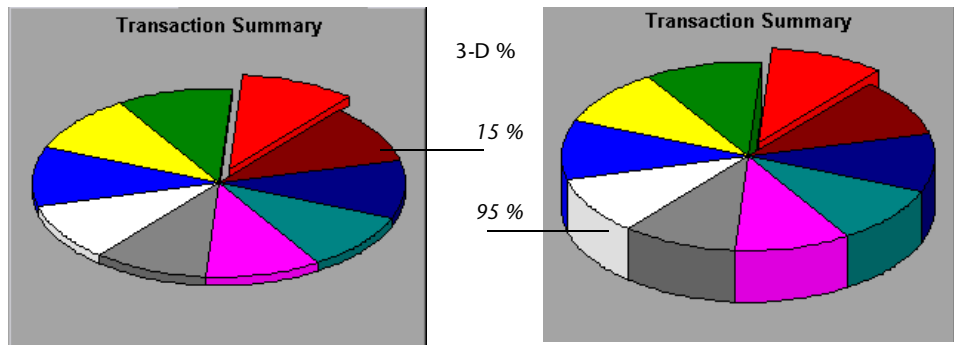
## Configuring Display Options

You can configure both standard and advanced display options for each graph. The standard options let you select the type of graph and the time setting. The advanced options allow you to modify the scale and format of each graph.

### Standard Display Options

The standard display options let you choose the type of graph to display: line, point, bar, or pie graph. Not all options are available for all graphs.

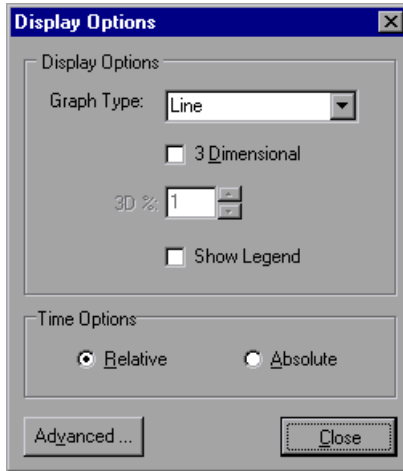
You can also indicate whether the graph should be displayed with a 3 dimensional look. This applies to bar and pie graphs only. You can specify the percent for the 3 dimensional graphs. This percentage indicates the thickness of the bar or pie chart.



The standard display options also let you indicate how to plot the results that are time-based: relative to the beginning of the scenario (default), or absolute time, based on the system clock of the machine.



To open the Display Options dialog box, choose **View > Display Options** or click the **Display Options** button.



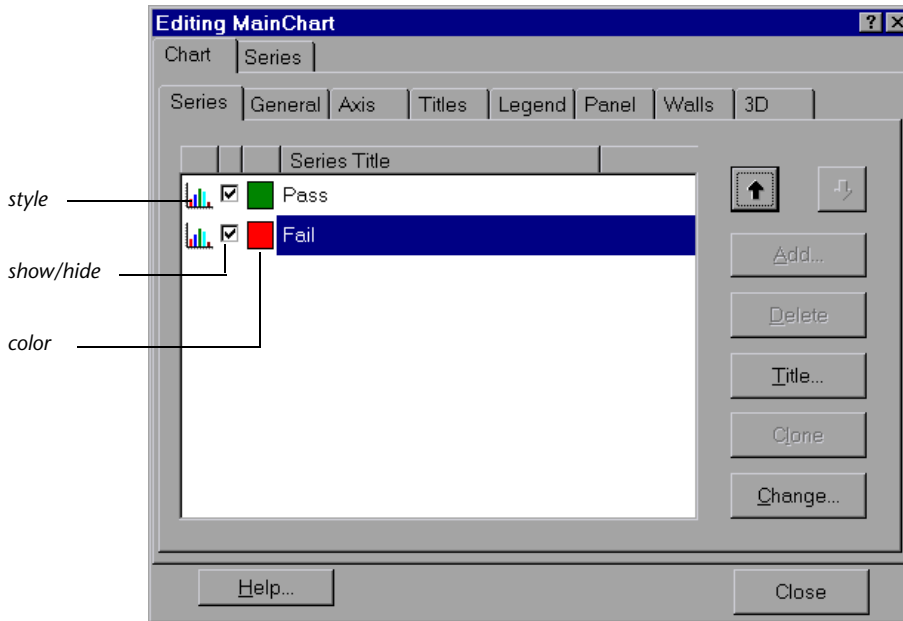
Choose a graph type, specify a 3 dimensional percent, select whether you want a legend to be displayed, and/or choose a time option. Click **Close** to accept the settings and close the dialog box.

### **Advanced Display Options**

The Advanced options let you configure the look and feel of your graph as well as its title and the format of the data. To set the Advanced options, you must first open the Display Options dialog box. Choose **View > Display**



**Options** or click the **Display Options** button. To access the Advanced options, click **Advanced**. The Editing MainChart dialog box opens.



You can customize the graph layout by setting the *Chart* and *Series* preferences. Click the appropriate tab and sub-tab to configure your graph.

## Chart Settings

The Chart settings control the look and feel of the entire graph—not the individual points. You set Chart preferences using the following tabs: Series, General, Axis, Titles, Legend, Panel, Walls, and 3D.

**Series:** displays the graph style, (bar, line, etc.) the hide/show settings, line and fill color, and the title of the series.

**General:** contains options for print preview, export, margins, scrolling, and magnification.

**Axis:** indicates which axes to show as well as their scales, titles, ticks, and position.

**Titles:** allows you to set the title of the graph, its font, background color, border, and alignment.

**Legend:** includes all legend-related settings, such as position, fonts, and divider lines.

**Panel:** shows the background panel layout of the graph. You can modify its color, set a gradient option, or specify a background image.

**Walls:** lets you set colors for the walls of 3-dimensional graphs.

**3D:** contains the 3-dimensional settings, offset, magnification, and rotation angle for the active graph.

### **Series Settings**

The Series settings control the appearance of the individual points plotted in the graph. You set Series preferences using the following tabs: Format, Point, General, and Marks.

**Format:** allows you to set the border color, line color, pattern and invert property for the lines or bars in your graph.

**Point:** displays the point properties. Points appear at various points within your line graph. You can set the size, color and shape of these points.

**General:** contains the type of cursor, the format of the axis values, and show/hide settings for the horizontal and vertical axis.

**Marks:** allows you to display the value for each point in the graph and configure the format of those marks.



## Setting General and Database Options

You use the *General* and *Database* Options to regulate the database containing your test results and optimize the processing time of your Analysis session.

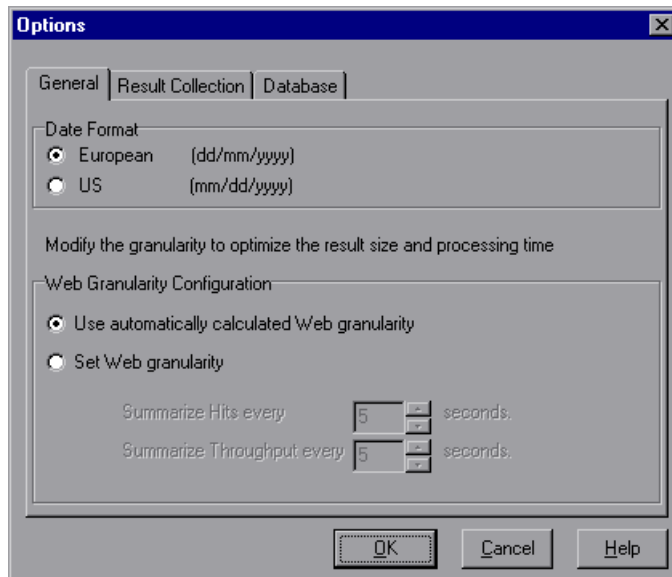
### Setting General Options

User data from long scenario runs (lasting two hours or more) create large databases. To decrease the size of the database and thereby decrease processing time, you can increase the granularity for the two primary Web-related measurements that account for over half of the data in the database—Hits per Second and Throughput.

You can also choose whether you want dates stored and displayed in a European or U.S. format.

To configure General options:

- 1 Select **Tools > Options**. The Options dialog box opens, displaying the General tab.



- 2** In the Date Format box, select **European** to store and display dates in the European format, or **US** to store and display dates in the American format.
- 3** In the Web Granularity Configuration box, select **Use automatically calculated Web granularity** (the default option) to allow the Analysis to determine the best granularity for displaying the results, or **Set Web granularity** to specify a custom granularity for the Hits per Second and Throughput measurements.

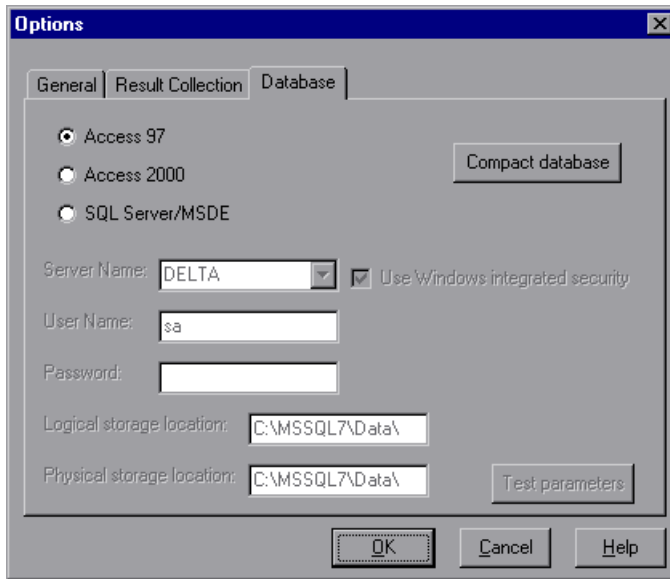
If you choose **Set Web granularity**, select the number of seconds you want the Analysis to wait before it summarizes the Hits and Throughput. By default, the Analysis summarizes the Hits per Second and Throughput measurements every 5 seconds. To reduce the size of the database, increase the granularity. To focus on more detailed results, decrease the granularity.

### Setting Database Options

The Database Options tab lets you specify the database in which to store Analysis session result data. By default, LoadRunner stores Analysis result data in an Access 97 database. If your Analysis result data exceeds two gigabytes, it is recommended that you store it on an SQL server or MSDE machine.

### To configure Database options:

- 1 Select **Tools > Options** to open the Options dialog box. Select the **Database** tab.



- 2 Choose one of the following three options:

- To instruct LoadRunner to save Analysis result data in an Access 97 database format, select **Access 97**.
- To instruct LoadRunner to save Analysis result data in an Access 2000 database format, select **Access 2000**.

When you configure and set up your Analysis session, the database containing the results may become fragmented. As a result, it will use excessive disk space. Click the **Compact Database** button to repair and compress your results and optimize your Access database.

---

**Note:** Long scenarios (duration of two hours or more) will require more time for compacting.

---

- To instruct LoadRunner to save Analysis result data on an SQL Server or MSDE machine, select **SQL Server/MSDE**. Select or enter the name of the machine on which the SQL server or MSDE is running, and enter the master database user name and password. Select **Use Windows-integrated security** in order to use your Windows login, instead of specifying a user name and password.

---

**Note:** By default, the user name “sa” and no password are used for the SQL server.

---

In the Logical storage location box, enter a shared directory on the SQL server/MSDE machine in which you want permanent and temporary database files to be stored. For example, if your SQL server's name is *fly*, enter `\\fly\<Analysis Database>\`. Note that Analysis results stored on an SQL server/MSDE machine can only be viewed on the machine's local LAN.

In the Physical storage location box, enter the real drive and directory path on the SQL server/MSDE machine that correspond to the logical storage location. For example, if the Analysis database is mapped to an SQL server named *fly*, and *fly* is mapped to drive D, enter `D:\<Analysis Database>`.

---

**Note:** If the SQL server/MSDE and Analysis are on the same machine, the logical storage location and physical storage location are identical.

---

Click **Test Parameters** to connect to the SQL server/MSDE machine and verify that the shared directory you specified exists on the server, and that you have write permissions on the shared server directory. If so, the Analysis synchronizes the shared and physical server directories.

---

**Note:** If you store Analysis result data on an SQL server/MSDE machine, you must select **File > Save As** in order to save your Analysis session. To delete your Analysis session, you must select **File > Delete Current Session**.

To open a session stored on an SQL Server/MSDE machine, the machine must be running and the directory you defined must exist as a shared directory.

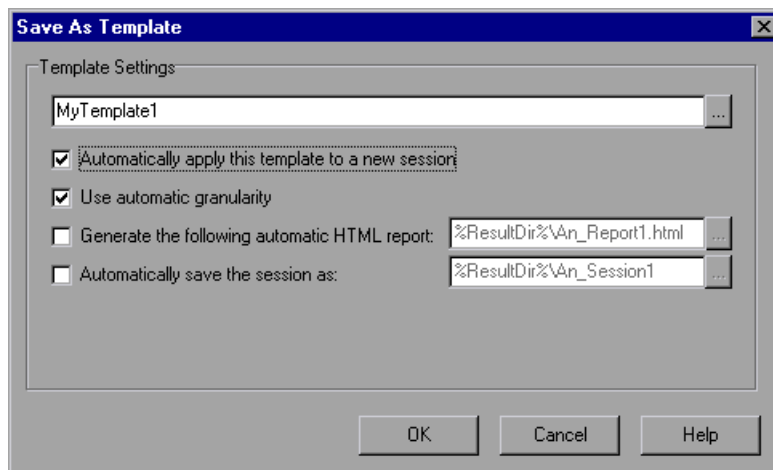
---

## Working with Templates

You can save the filter and display options that you specified for an Analysis session in order to use them later in another session.

**To save the Analysis template that you created:**

- 1 Select **Tools > Templates > Save As Template**. The Save As Template dialog box opens.



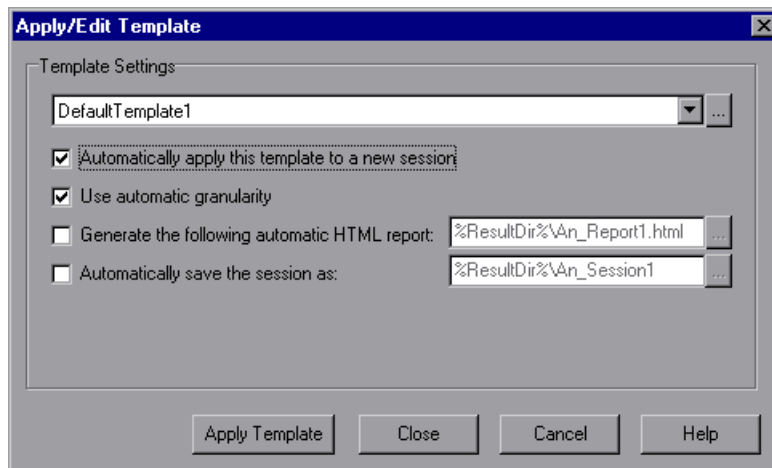
- 2 Enter the name of the template you want to create, or select the template name using the browse button to the right of text box.

- 3 To apply the template to any new session you open, select **Automatically apply this template to a new session**.
- 4 To apply the default Analysis granularity (one second) to the template, select **Use automatic granularity**. For information about setting Analysis granularity, see “Changing the Granularity of the X-axis” on page 13.
- 5 To generate an HTML report using the template, select **Generate the following automatic HTML report**, and specify or select a report name. For information about generating HTML reports, see “Creating HTML Reports” on page 166.
- 6 To instruct the Analysis to automatically save the session using the template you specify, select **Automatically save the session as**, and specify or select a file name.
- 7 Click **OK** to close the dialog box and save the template.

Once you have saved a template, you can apply it to other Analysis sessions.

**To use a saved template in another Analysis session:**

- 1 Select **Tools > Templates > Apply/Edit Template**. The Apply/Edit Template dialog box opens.



- 2 Select the template you want to use from the template selector, or click the browse button to choose a template from a different location.

- 3** To apply the template to any new session you open, select **Automatically apply this template to a new session**.
- 4** To apply the default Analysis granularity (one second) to the template, select **Use automatic granularity**. For information about setting Analysis granularity, see “Changing the Granularity of the X-axis” on page 13.
- 5** To generate an HTML report using the template, select **Generate the following automatic HTML report**, and specify or select a report name. For information about generating HTML reports, see “Creating HTML Reports” on page 166.
- 6** To instruct the Analysis to automatically save the session using the template you specify, select **Automatically save the session as**, and specify or select a file name.
- 7** Click **OK** to close the dialog box and load the template you selected into the current Analysis session.





# 3

---

## Vuser Graphs

After running a scenario, you can check the behavior of the Vusers that participated in the scenario using the following Vuser graphs:

- ▶ Running Vusers Graph
- ▶ Vuser Summary Graph
- ▶ Rendezvous Graph

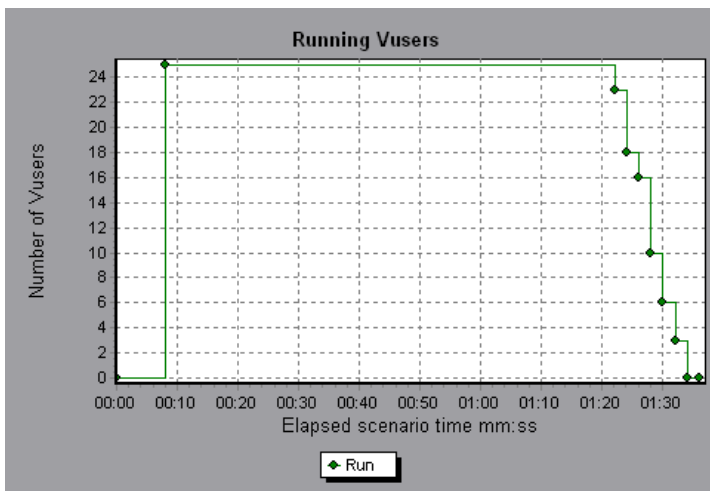
### About Vuser Graphs

During scenario execution, Vusers generate data as they perform transactions. The Vuser graphs let you determine the overall behavior of Vusers during the scenario. They display the Vuser states, the number of Vusers that completed the script, and rendezvous statistics. Use these graphs in conjunction with Transaction graphs to determine the effect of the number of Vusers on transaction response time.

## Running Vusers Graph

The Running Vusers graph displays the number of Vusers that executed Vuser scripts and their status during each second of the test. This graph is useful for determining the Vuser load on your server at any given moment. By default, this graph only shows the Vusers with a *Run* status. To view another Vuser status, set the filter conditions to the desired status. For more information, see Chapter 2, "Working with LoadRunner Analysis."

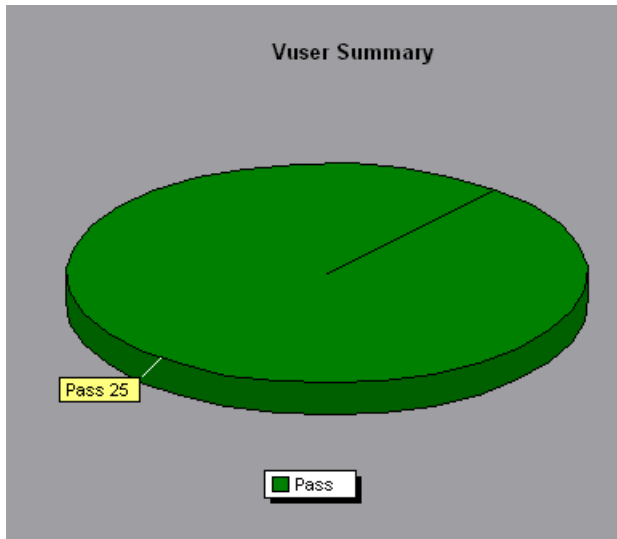
The x-axis represents the elapsed time from the beginning of the scenario run. The y-axis represents the number of Vusers in the scenario.



## Vuser Summary Graph

The Vuser Summary graph displays a summary of Vuser performance. It lets you view the number of Vusers that successfully completed the scenario run relative to those that did not.

This graph may only be viewed as a pie.

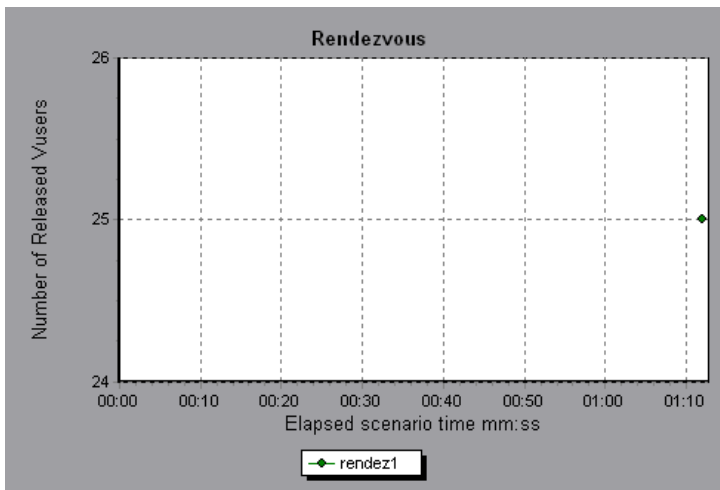


## Rendezvous Graph

The Rendezvous graph indicates when Vusers were released from rendezvous points, and how many Vusers were released at each point.

This graph helps you understand transaction performance times. If you compare the Rendezvous graph to the Transaction Response Time - Average graph, you can see how the load peak created by a rendezvous influences transaction times.

On the Rendezvous graph, the x-axis indicates the time that elapsed since the beginning of the scenario. The y-axis indicates the number of Vusers that were released from the rendezvous. If you set a rendezvous for 60 Vusers, and the graph indicates that only 25 were released, you can see that the rendezvous ended when the timeout expired because all of the Vusers did not arrive.



# 4

---

## Error Graphs

After a scenario run, you can use the error graphs to analyze the errors that occurred during the load test.

This chapter describes:

- Error Statistics Graph
- Errors per Second Graph

### About Error Graphs

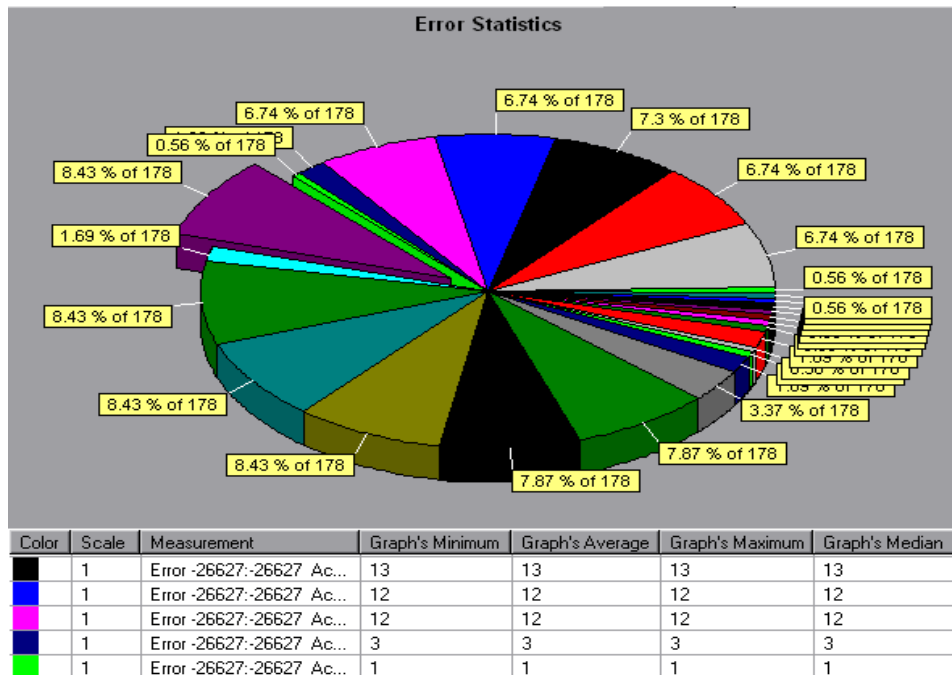
During scenario execution, Vusers may not complete all transactions successfully. The Error graphs let you view information about the transactions that failed, aborted, or ended in errors. Using the Error graphs, you can view a summary of errors that occurred during the scenario and the average number of errors that occurred per second.

## Error Statistics Graph

The Error Statistics graph displays the number of errors that accrued during scenario execution, grouped by error code.

In the graph below, out of a total of 178 errors that occurred during the scenario run, the second error code displayed in the legend occurred twelve times, comprising 6.74% the errors.

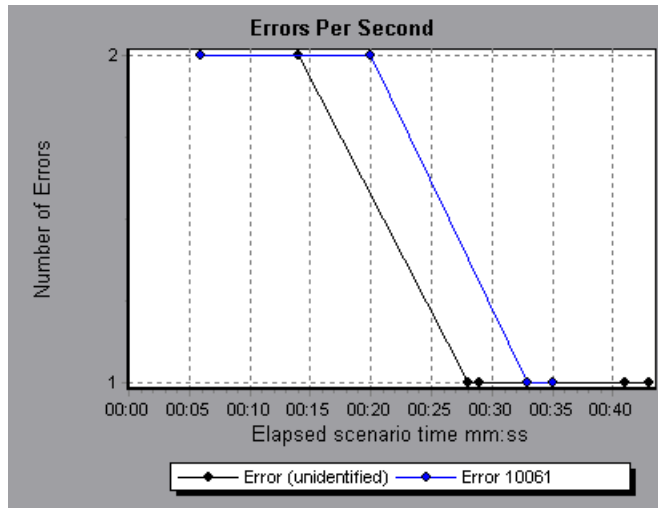
This graph may only be viewed as a pie.



## Errors per Second Graph

The Errors per Second graph displays the average number of errors that occurred during each second of the scenario run, grouped by error code.

The x-axis represents the elapsed time from the beginning of the scenario run. The y-axis represents the number of errors.







# 5

---

## Transaction Graphs

After running a scenario, you can analyze the transactions that were executed during the test using one or more of the following graphs:

- ▶ Transaction Response Time - Average Graph
- ▶ Transactions per Second Graph
- ▶ Total Transactions per Second
- ▶ Transaction Summary Graph
- ▶ Transaction Performance Summary Graph
- ▶ Transaction Response Time - Under Load Graph
- ▶ Transaction Response Time - Percentile Graph
- ▶ Transaction Response Time - Distribution Graph

### About Transaction Graphs

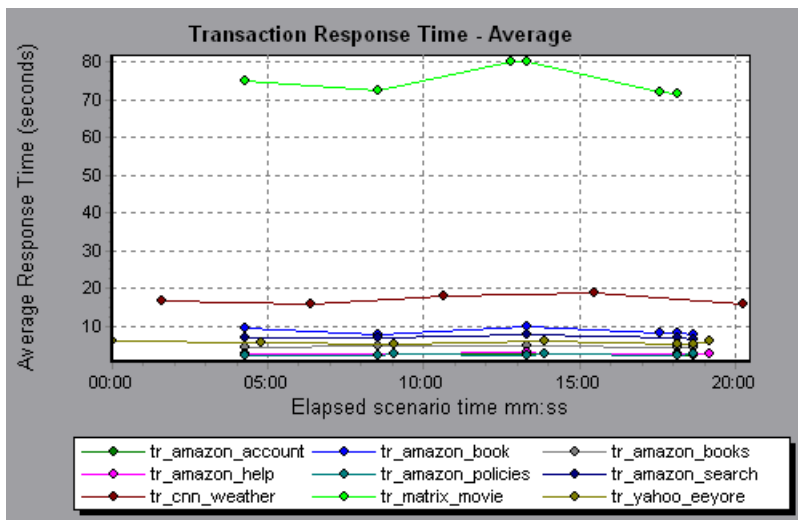
During scenario execution, Vusers generate data as they perform transactions. The Analysis enables you to generate graphs that show the transaction performance and status throughout script execution.

You can use additional Analysis tools such as merging and crossing results to understand your transaction performance graphs. You can also sort the graph information by transactions. For more information about working with the Analysis, see Chapter 2, “Working with LoadRunner Analysis.”

## Transaction Response Time - Average Graph

The Transaction Response Time – Average graph displays the average time taken to perform transactions during each second of the scenario run.

The x-axis represents the elapsed time from the beginning of the scenario run. The y-axis represents the average time (in seconds) taken to perform each transaction.



This graph is displayed differently for each granularity. The lower the granularity, the more detailed the results. However, it may be useful to view the results with a higher granularity to study the overall Vuser behavior throughout the scenario. For example, using a low granularity, you may see intervals when no transactions were performed. However, by viewing the same graph with a higher granularity, you will see the graph for the overall transaction response time. For more information on setting the granularity, see Chapter 2, "Working with LoadRunner Analysis."

---

**Note:** By default, only transactions that passed are displayed.

---

You can compare the Transaction Response Time – Average graph to the Running Vusers graph to see how the number of running Vusers affects the transaction performance time.

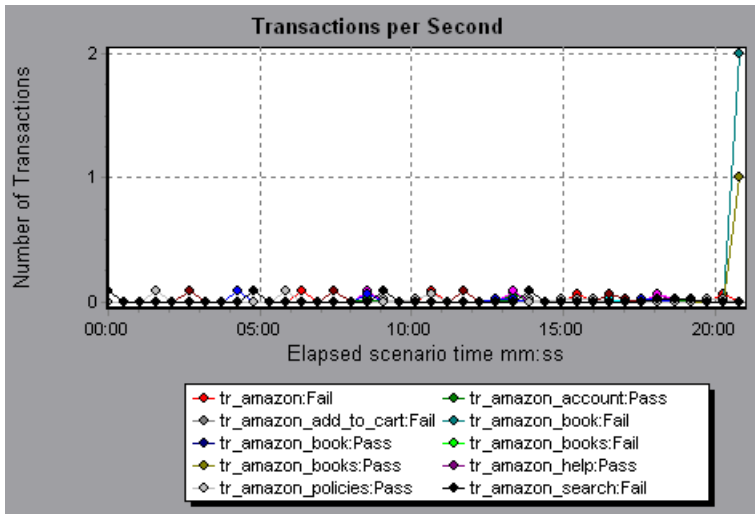
For example, if the Transaction Response Time – Average graph shows that performance time gradually improved, you can compare it to the Running Vusers graph to see whether the performance time improved due to a decrease in the Vuser load.

If you have defined acceptable minimum and maximum transaction performance times, you can use this graph to determine whether the performance of the server is within the acceptable range.

## Transactions per Second Graph

The Transactions per Second graph displays, for each transaction, the number of times it passed, failed, and aborted during each second of a scenario run. This graph helps you determine the actual transaction load on your system at any given moment. You can compare this graph to the Transaction Response Time - Average graph in order to analyze the effect of the number of transactions on the performance time.

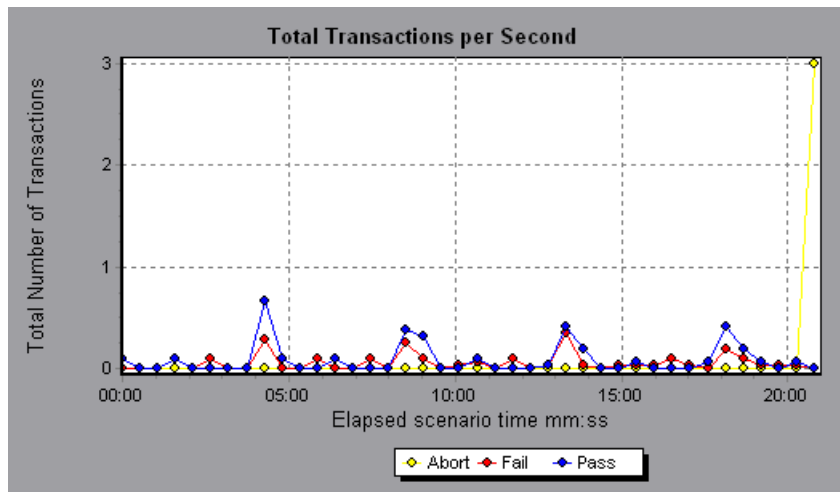
The x-axis represents the elapsed time from the beginning of the scenario run. The y-axis represents the number of transactions performed during the scenario run.



## Total Transactions per Second

The Total Transactions per Second graph displays the total number of transactions that passed, the total number of transactions that failed, and the total number of transactions that were aborted, during each second of a scenario run.

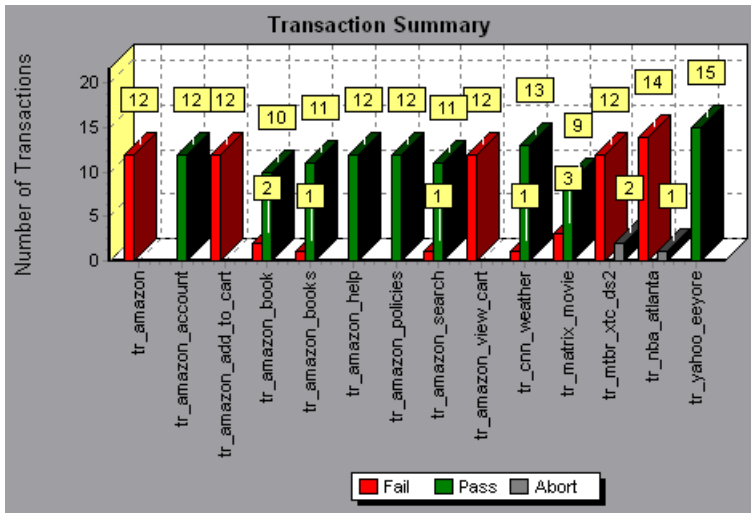
The x-axis represents the elapsed time (in seconds) since the start of the scenario run. The y-axis represents the total number of transactions performed during the scenario run.



## Transaction Summary Graph

The Transaction Summary graph summarizes the number of transactions in the scenario that failed, passed, aborted, and ended in error.

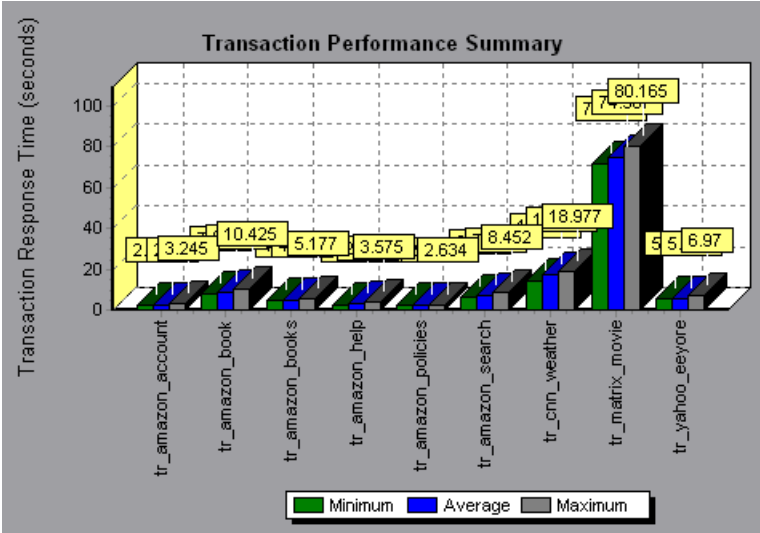
The x-axis specifies the name of the transaction. The y-axis shows the number of transactions performed during the scenario run.



# Transaction Performance Summary Graph

The Transaction Performance Summary graph displays the minimum, maximum and average performance time for all the transactions in the scenario.

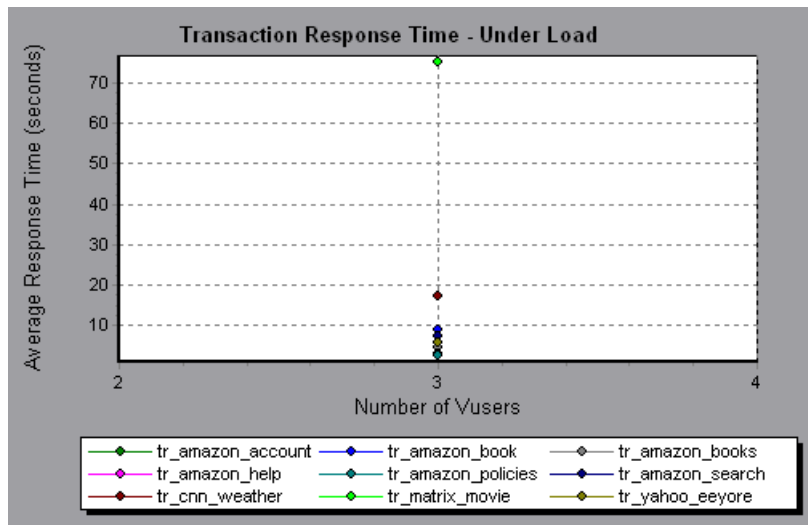
The x-axis specifies the name of the transaction. The y-axis shows the time—rounded off to the nearest second—taken to perform each transaction.



## Transaction Response Time - Under Load Graph

The Transaction Response Time - Under Load graph is a combination of the Running Vusers and Transaction Response Time – Average graphs and indicates transaction times relative to the number of Vusers running at any given point during the scenario. This graph helps you view the general impact of Vuser load on performance time and is most useful when analyzing a scenario with a gradual load. For information about creating a gradual load for a scenario, see the *LoadRunner Controller User's Guide*.

The x-axis indicates the number of running Vusers, and the y-axis indicates the average transaction time in seconds.



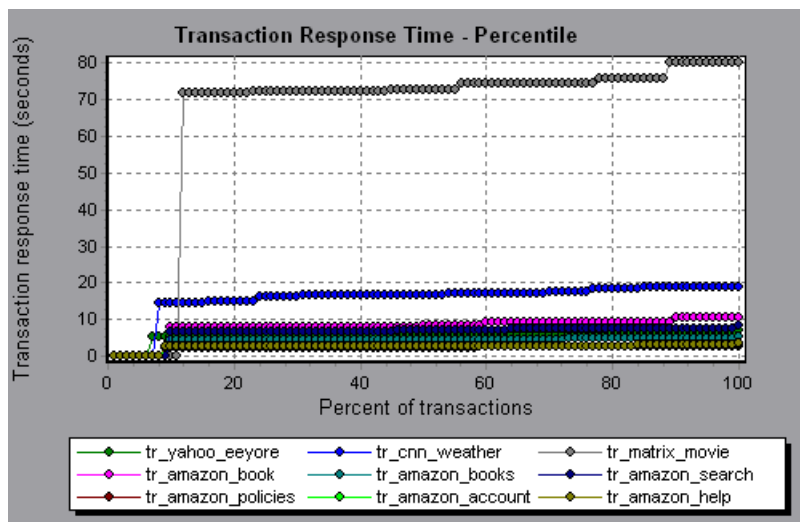


## Transaction Response Time - Percentile Graph

The Transaction Response Time - Percentile graph analyzes the percentage of transactions that were performed within a given time range. This graph helps you determine the percentage of transactions that met the performance criteria defined for your system. In many instances, you need to determine the percent of transactions with an acceptable response time. The maximum response time may be exceptionally long, but if most transactions have acceptable response times, the overall system is suitable for your needs.

The x-axis represents the percentage of the total number of transactions measured during the scenario run. The y-axis represents the time taken to perform the transactions.

In the following graph, fewer than 20 percent of the *tr\_matrix\_movie* transactions had a response time less than 70 seconds.



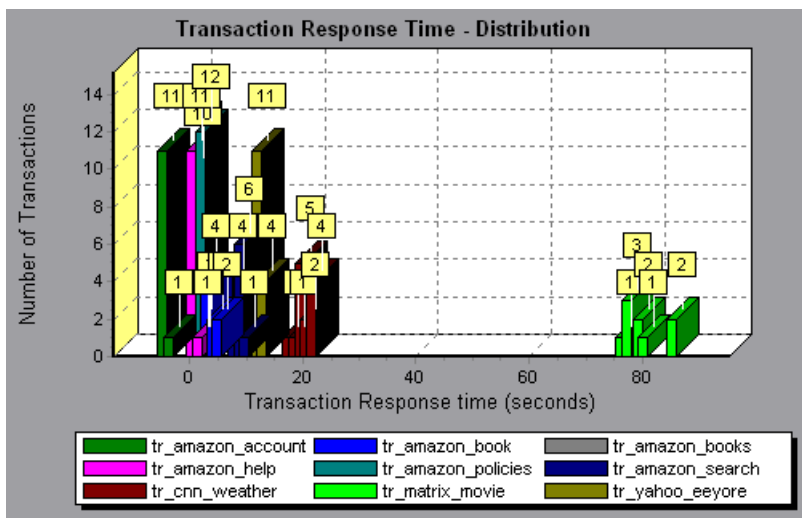
It is recommended to compare the Percentile graph to a graph indicating average response time such as the Transaction Response Time - Average graph. A high response time for several transactions may raise the overall average. However, if the transactions with a high response time occurred less than five percent of the time, that factor may be insignificant.

## Transaction Response Time - Distribution Graph

The Transaction Response Time - Distribution graph displays the distribution of the time taken to perform transactions in a scenario. If you compare it to the Transaction Performance Summary graph, you can see how the average performance was calculated.

The x-axis represents the transaction response time. The y-axis represents the number of transactions executed during the scenario.

In the following graph, most of the transactions had a response time of less than 20 seconds.




---

**Note:** This graph can only be displayed as a bar graph.

---

If you have defined acceptable minimum and maximum transaction performance times, you can use this graph to determine whether the performance of the server is within the acceptable range.

# 6

---

## Web Resource Graphs

After a scenario run, you use the Web Resource graphs to analyze Web server performance.

This chapter describes:

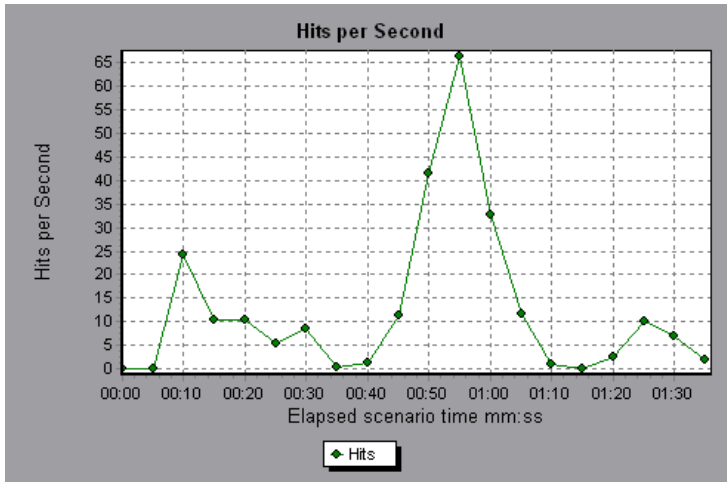
- ▶ Hits per Second Graph
- ▶ Throughput Graph
- ▶ HTTP Status Code Summary Graph
- ▶ HTTP Responses per Second Graph
- ▶ Pages Downloaded per Second Graph

### About Web Resource Graphs

Web Resource graphs provide you with information about the performance of your Web server. You use the Web Resource graphs to analyze the throughput on the Web server, the number of hits per second that occurred during the scenario, the number of HTTP responses per second, the number of downloaded pages per second, and 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.

## Hits per Second Graph

The Hits per Second graph shows the number of HTTP requests made by Users to the Web server during each second of the scenario run. This graph helps you evaluate the amount of load Users generate, in terms of the number of hits. You can compare this graph to the Transaction Response Time - Average graph to see how the number of hits affects transaction performance.



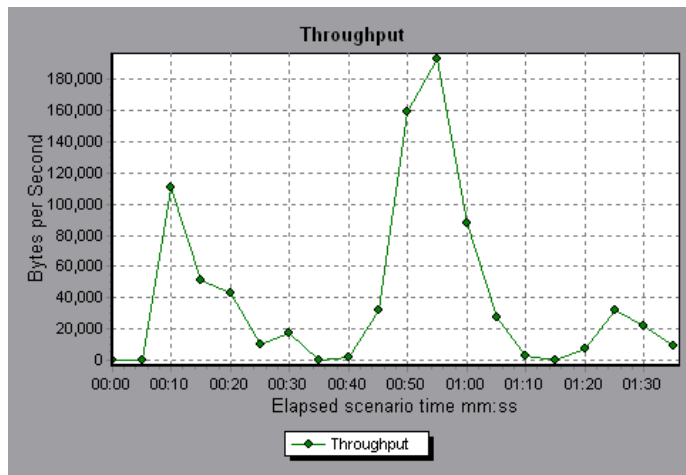
The x-axis represents the elapsed time since the start of the scenario run. The y-axis represents the number of hits on the server. For example, the graph above shows that the most hits per second took place during the fifty-fifth second of the scenario.

## Throughput Graph

The Throughput graph shows the amount of throughput on the server during each second of the scenario run. Throughput is measured in bytes and represents the amount of data that the Vusers received from the server at any given second. This graph helps you evaluate the amount of load Vusers generate, in terms of server throughput. You can compare this graph to the Transaction Response Time - Average graph to see how the throughput affects transaction performance.

The x-axis represents the elapsed time since the start of the scenario run. The y-axis represents the throughput of the server, in bytes.

The following graph shows the highest throughput to be 193,242 bytes during the fifty-fifth second of the scenario.

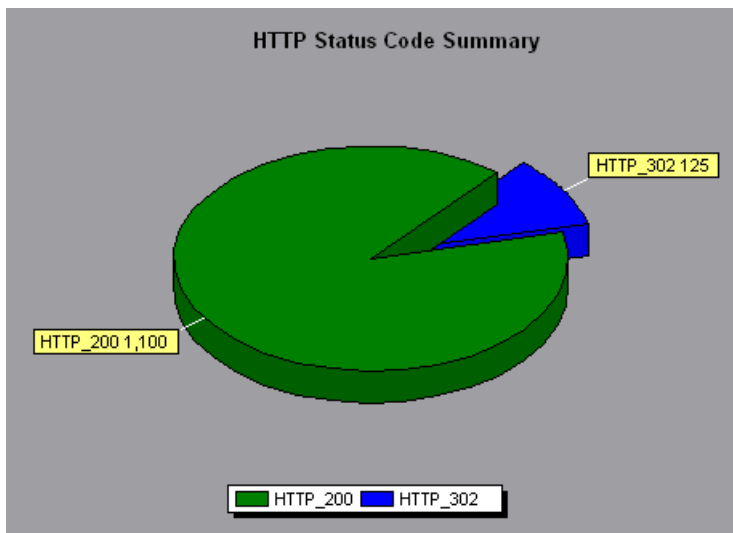


## HTTP Status Code Summary Graph

The HTTP Status Code Summary 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”) returned from the Web server during the scenario run, grouped by status code. Use this graph together with the HTTP Responses per Second Graph to locate those scripts which generated error codes.

This graph may only be viewed as a pie.

The following graph shows that only the HTTP status codes **200** and **302** were generated. Status code **200** was generated 1,100 times, and status code **302** was generated 125 times.

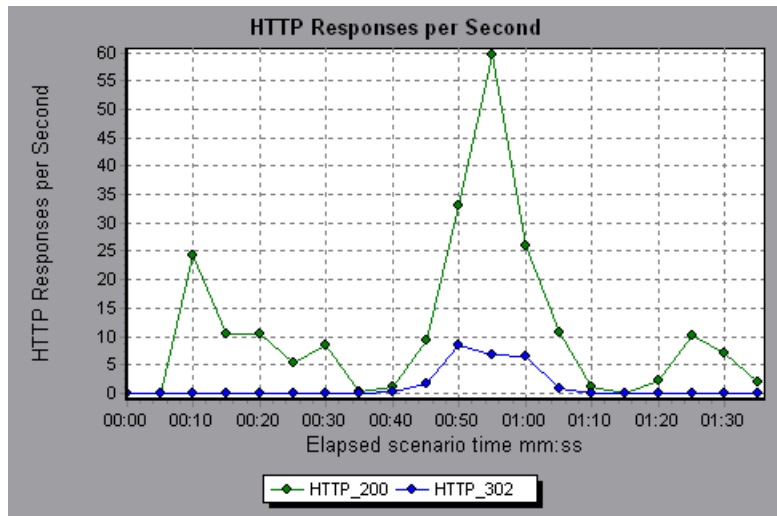


## HTTP Responses per Second Graph

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”) returned from the Web server during each second of the scenario run, 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. For more information on the “Group By” function, see Chapter 2, “Working with LoadRunner Analysis.”

The x-axis represents the time that has elapsed since the start of the scenario run. The y-axis represents the number of HTTP responses per second.

The following graph shows that the greatest number of **200** status codes, 60, was generated in the fifty-fifth second of the scenario run. The greatest number of **302** codes, 8.5, was generated in the fiftieth second of the scenario run.



The following table displays a list of HTTP status codes:

<b>Code</b>	<b>Description</b>
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
402	Payment Required
403	Forbidden
404	Not Found
405	Method Not Allowed
406	Not Acceptable
407	Proxy Authentication Required
408	Request Timeout
409	Conflict



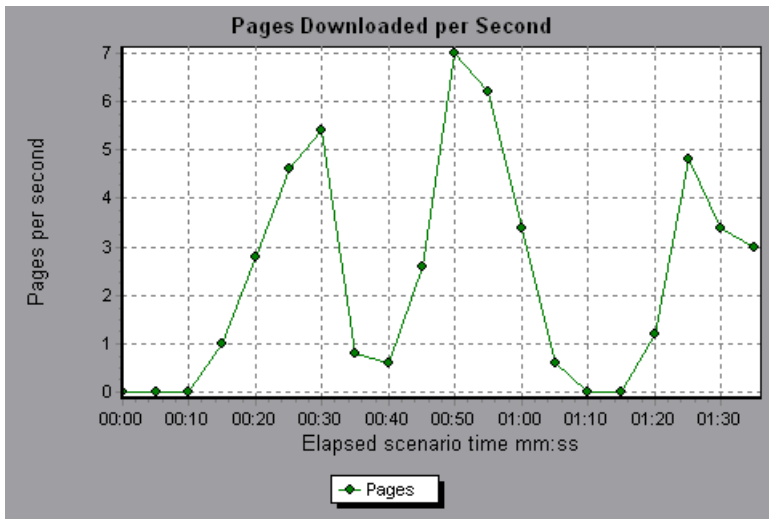
<b>Code</b>	<b>Description</b>
410	Gone
411	Length Required
412	Precondition Failed
413	Request Entity Too Large
414	Request - URI Too Large
415	Unsupported Media Type
416	Requested range not satisfiable
417	Expectation Failed
500	Internal Server Error
501	Not Implemented
502	Bad Gateway
503	Service 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

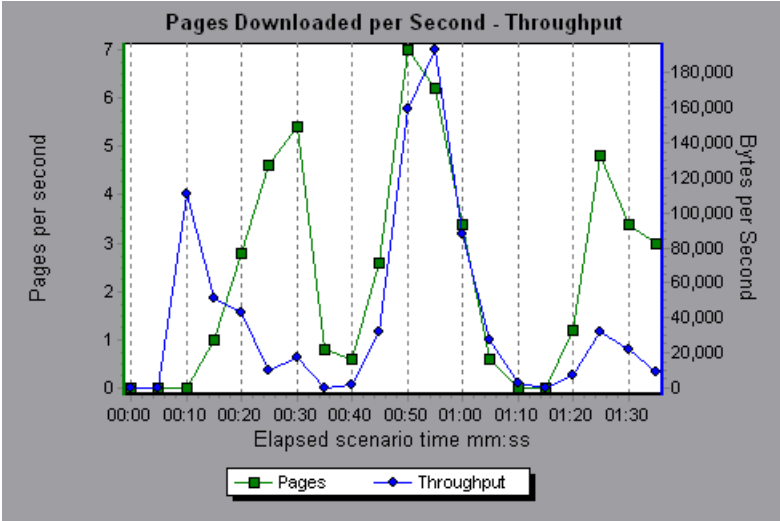
The Pages Downloaded per Second graph shows the number of Web pages (y-axis) downloaded from the server during each second of the scenario run (x-axis). This graph helps you evaluate the amount of load Vusers generate, in terms of the number of pages downloaded.

The following graph shows that the greatest number of pages downloaded per second, about 7, occurred in the fiftieth second of the scenario run.



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. However, 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 only the number of pages.

In the following example, the Throughput graph is merged with the Pages Downloaded per Second graph. It is apparent from the graph that throughput is not completely proportional to the number of pages downloaded per second. For example, between 10 and 25 seconds into the scenario run, the number of pages downloaded per second increased while the throughput decreased.





# 7

---

## Web Page Breakdown Graphs

The Web Page Breakdown graphs enable you to assess whether transaction response times were affected by page content. Using the Web Page Breakdown graphs, you can analyze problematic elements—for example, images that download slowly, or broken links—of a Web site.

This chapter describes:

- ▶ Activating the Web Page Breakdown Graphs
- ▶ Page Component Breakdown Graph
- ▶ Page Component Breakdown (Over Time) Graph
- ▶ Page Download Time Breakdown Graph
- ▶ Page Download Time Breakdown (Over Time) Graph
- ▶ Time to First Buffer Breakdown Graph
- ▶ Time to First Buffer Breakdown (Over Time) Graph
- ▶ Downloaded Component Size Graph

## About Web Page Breakdown Graphs

Web Page Breakdown graphs provide you with performance information for each monitored Web page in your script. You can view the download time of each page in the script and its components, and identify at what point during download time problems occurred. In addition, you can view the relative download time and size of each page and its components. The Analysis displays both average download time and download time over time data.

You correlate the data in the Web Page Breakdown graphs with data in the Transaction Performance Summary and Transaction Response Time - Average graphs in order to analyze why and where problems are occurring, and whether the problems are network- or server-related.

In order for the Analysis to generate Web Page Breakdown graphs, you must enable the component breakdown feature before recording your script.

**To enable the component breakdown feature:**

- 1** From the Controller menu, choose **Tools > Options**.
- 2** Select the **Web Page Breakdown** tab. Select the **Enable Web Page Breakdown** check box.
- 3** From the VuGen menu, choose **Vuser > Run-Time Settings**. Select the **General** tab, and select the **Define each step as a transaction** check box.

---

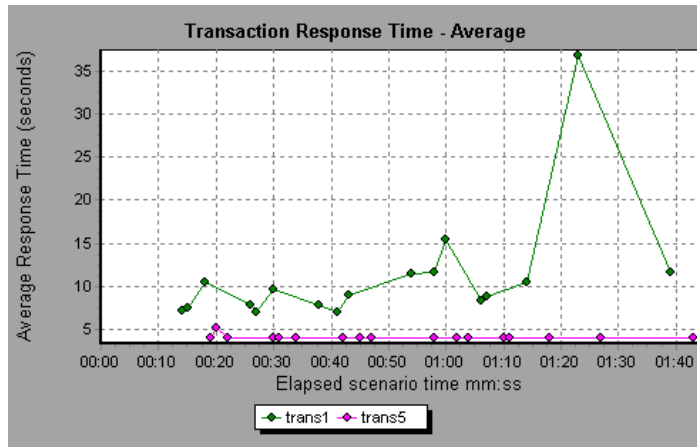
**Note:** It is recommended that, in VuGen, you select **HTML-based script** in the Recording tab of the Recording Options dialog box.

---

For more information on recording Web Vuser scripts, see the *Creating Vuser Scripts* guide.

## Activating the Web Page Breakdown Graphs

The Web Page Breakdown graphs are most commonly used to analyze a problem detected in the Transaction Performance Summary or Transaction Response Time - Average graphs. For example, the Transaction Response Time - Average graph below demonstrates that the average transaction response time for the trans1 transaction was high.



Using the Web Page Breakdown graphs, you can pinpoint the cause of the delay in response time for the trans1 transaction.

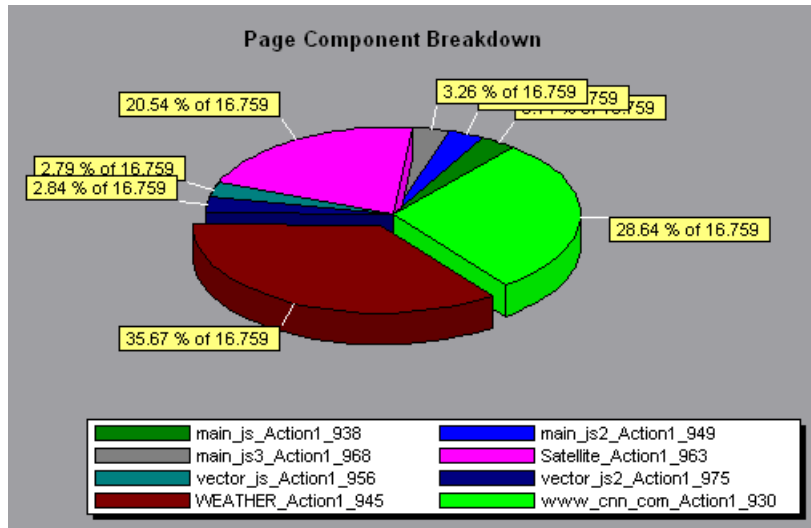
### To view a breakdown of a transaction:

- 1** Right-click trans1 and select **Web Page Breakdown for trans1**. The Web Page Breakdown graph and the Web Page Breakdown Tree appear.
- 2** In the Web Page Breakdown Tree, right-click the problematic page or component you want to break down, and select **Break Down <component name>**. Alternatively, select a page or component in the Select Page to Break Down box. The Web Page Breakdown graph for that page or component appears.
- 3** Select one of the four options:
  - ▶ **Download Time Breakdown:** Displays a table with a breakdown of the selected page or component's download time. The size of each page component (including the component's header) is also displayed. See Page Download Time Breakdown Graph for more information about this display.
  - ▶ **Component Breakdown (Over Time):** Displays the Page Component Breakdown (Over Time) Graph for the selected Web page or component.
  - ▶ **Download Time Breakdown (Over Time):** Displays the Page Download Time Breakdown (Over Time) Graph for the selected Web page or component.
  - ▶ **Network/Server Time Distribution (Over Time):** Displays the Time to First Buffer Breakdown (Over Time) Graph for the selected Web page or component.
- ▶ See the sections below for an explanation of each of these four graphs.

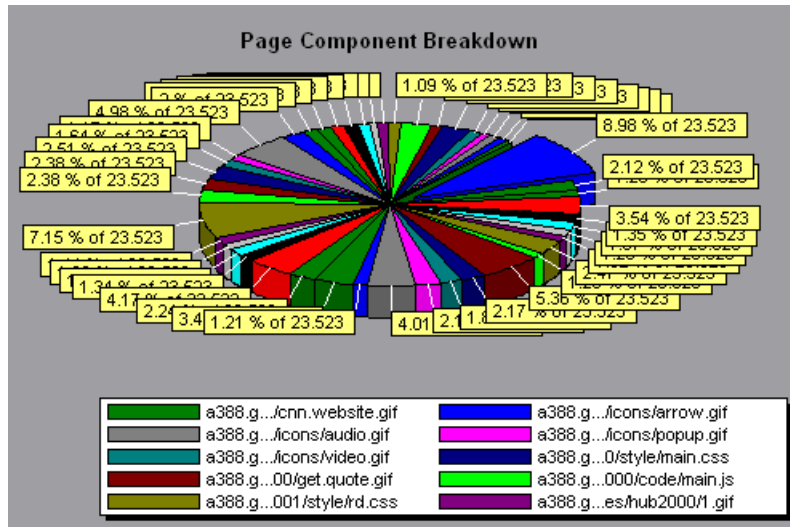


## Page Component Breakdown Graph

The Page Component Breakdown graph displays the average download time (in seconds) for each Web page and its components. For example, the following graph demonstrates that the main cnn.com URL took 28.64% of the total download time, compared to 35.67% for the www.cnn.com/WEATHER component.



To ascertain which components caused the delay in download time, you can break down the problematic URL by double-clicking it in the Web Page Breakdown Tree. Here, the cnn.com/WEATHER component is broken down.



The above graph shows that the main cnn.com/WEATHER component took the longest time to download (8.98% of the total download time). To isolate other problematic components of the URL, it may be helpful to sort the legend according to the average number of seconds taken to download a component. To sort the legend by average, click the **Average** column.

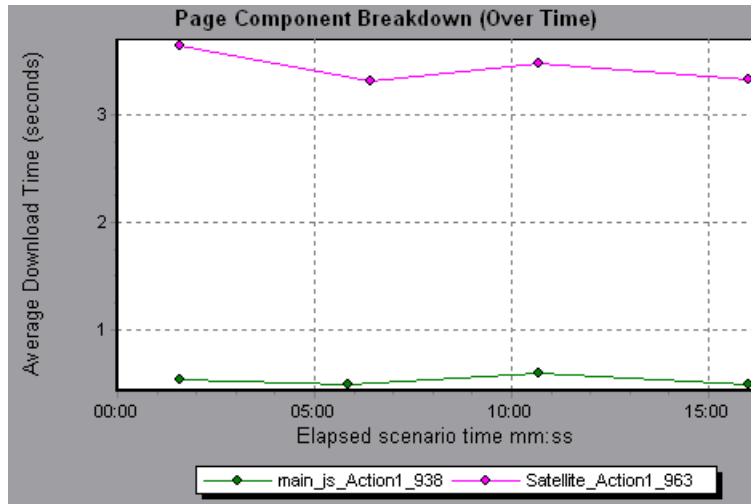
---

**Note:** The Page Component Breakdown graph can only be viewed as a pie.

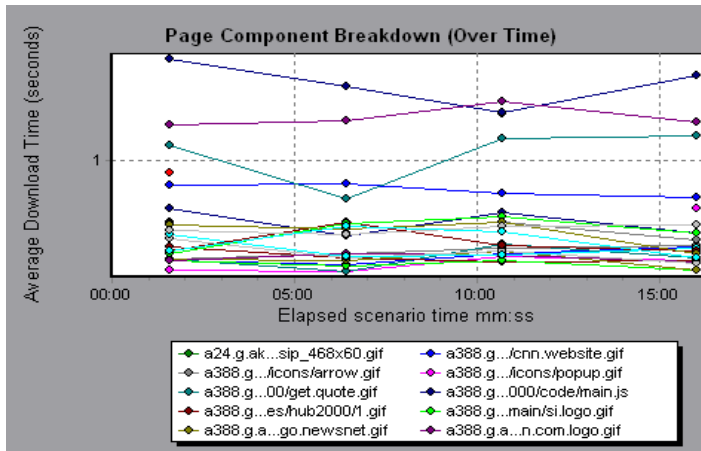
---

## Page Component Breakdown (Over Time) Graph

The Page Component Breakdown (Over Time) graph displays the average response time (in seconds) for each Web page and its components during each second of the scenario run. For example, the following graph demonstrates that the response time for Satellite\_Action1\_963 was significantly greater, throughout the scenario, than the response time for main\_js\_Action1\_938.



To ascertain which components were responsible for the delay in response time, you can break down the problematic component by double-clicking it in the Web Page Breakdown Tree.

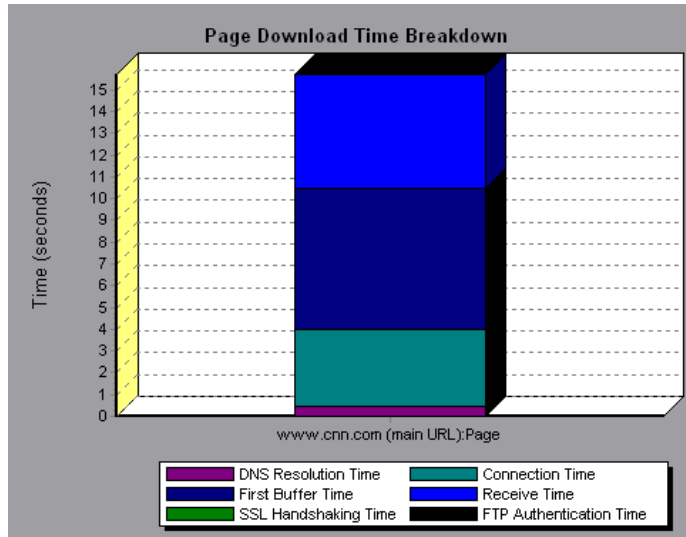


Using the above graph, you can track which components of the main component were most problematic, and at which point(s) during the scenario the problem(s) occurred. To isolate the most problematic components, it may be helpful to sort the legend tab according to the average number of seconds taken to download a component. To sort the legend by average, double-click the **Average** column.

To identify a component in the graph, you can click it. The corresponding line in the legend tab is selected.

## Page Download Time Breakdown Graph

The Page Download Time Breakdown graph displays a breakdown of each page component's download time, enabling you to determine whether slow response times are being caused by network or server errors during Web page download.



The Page Download Time Breakdown graph breaks down each component by DNS resolution time, connection time, time to first buffer, SSL

handshaking time, receive time, and FTP authentication time. These breakdowns are described below:

Name	Description
DNS Resolution	Displays the amount of time needed to resolve the DNS name to an IP address, using the closest DNS server. The DNS Lookup measurement is a good indicator of problems in DNS resolution, or problems with the DNS server.
Connection	Displays the amount of time needed to establish an initial connection with the Web server hosting the specified URL. The connection measurement is a good indicator of problems along the network. It also indicates whether the server is responsive to requests.
First Buffer	Displays the amount of time that passes from the initial HTTP request (usually GET) until the first buffer is successfully received back from the Web server. The first buffer measurement is a good indicator of Web server delay as well as network latency.  <b>Note:</b> Since the buffer size may be up to 8K, the first buffer might also be the time it takes to completely download the element.
SSL Handshaking	Displays the amount of time taken to establish an SSL connection (includes the client hello, server hello, client public key transfer, server certificate transfer, and other—partially optional—stages). After this point, all the communication between the client and server is encrypted. The SSL Handshaking measurement is only applicable for HTTPS communications.

Name	Description
Receive	Displays the amount of time that passes until the last byte arrives from the server and the downloading is complete. The Receive measurement is a good indicator of network quality (look at the time/size ratio to calculate receive rate).
FTP Authentication	Displays the time taken to authenticate the client. With FTP, a server must authenticate a client before it starts processing the client's commands. The FTP Authentication measurement is only applicable for FTP protocol communications.

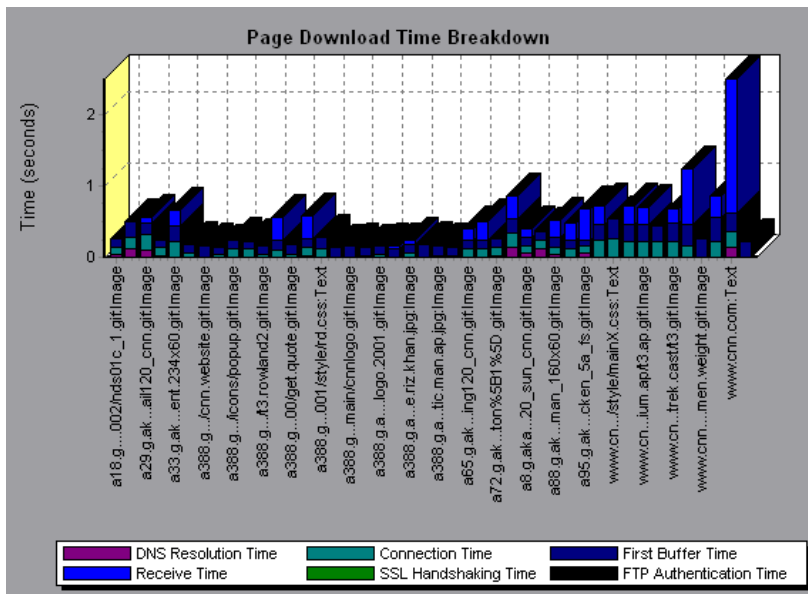
---

**Note:** Each measurement displayed on the page level is the sum of that measurement recorded for each page component. For example, the Connection Time for [www.cnn.com](http://www.cnn.com) is the sum of the Connection Time for each of the page's components.

---

The above graph demonstrates that receive time, connection time, and first buffer time accounted for a large portion of the time taken to download the main [cnn.com](http://cnn.com) URL. If you break the [cnn.com](http://cnn.com) URL down further, you can

isolate the components with the longest download time, and analyze the network or server problems that contributed to the delay in response time.



Breaking down the cnn.com URL demonstrates that for the component with the longest download time (the www.cnn.com component), the receive time accounted for a large portion of the download time.

---

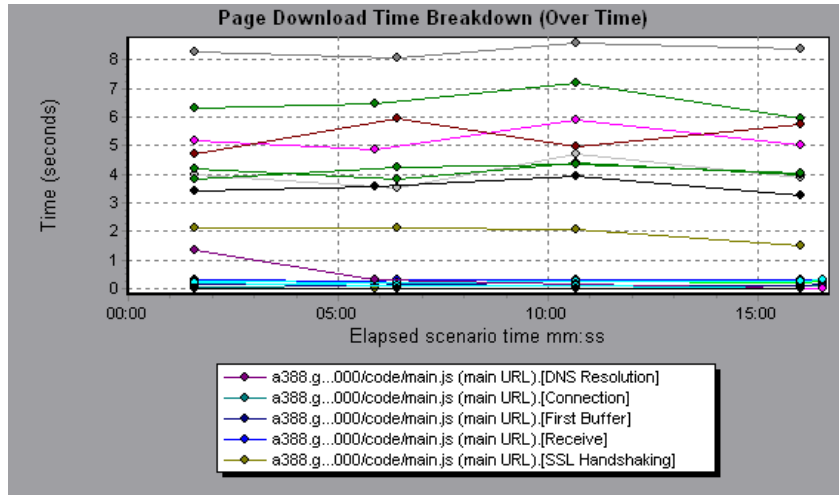
**Note:** When the Page Download Time Breakdown graph is selected from the Web Page Breakdown graph, the x-axis and y-axis are reversed: the x-axis represents the time (in seconds) taken for each step in the download process, and the y-axis specifies the name of the component.

---



## Page Download Time Breakdown (Over Time) Graph

The Page Download Time Breakdown (Over Time) graph displays a breakdown of each page component's download time during each second of the scenario run. This graph enables you to determine at what point during scenario execution network or server problems occurred.




---

**Note:** Each measurement displayed on the page level is the sum of that measurement recorded for each page component. For example, the Connection Time for [www.cnn.com](http://www.cnn.com) is the sum of the Connection Time for each of the page's components.

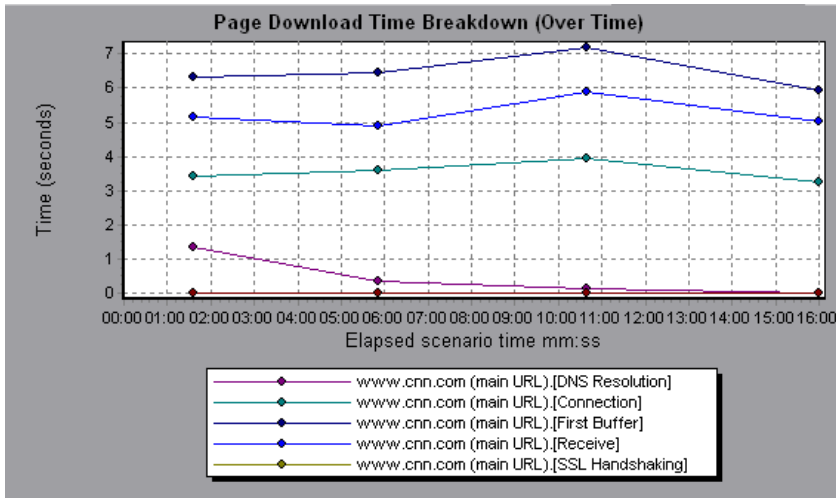
---

To isolate the most problematic components, you can sort the legend tab according to the average number of seconds taken to download a component. To sort the legend by average, double-click the **Average** column.

To identify a component in the graph, you can click it. The corresponding line in the legend tab is selected.

In the example in the previous section, it is apparent that [cnn.com](http://cnn.com) was the most problematic component. If you examine the [cnn.com](http://cnn.com) component, the

Page Download Time Breakdown (Over Time) graph demonstrates that first buffer and receive time remained high throughout the scenario, and that DNS Resolution time decreased throughout the scenario.



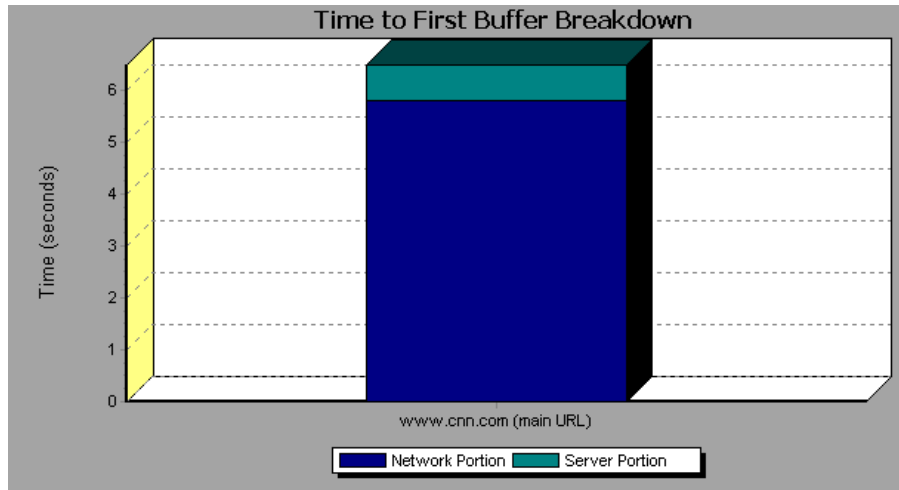
---

**Note:** When the Page Download Time Breakdown (Over Time) graph is selected from the Web Page Breakdown graph, it appears as an area graph.

---

## Time to First Buffer Breakdown Graph

The Time to First Buffer Breakdown graph displays each Web page component's relative server/network time (in seconds) for the period of time until the first buffer is successfully received back from the Web server. If the download time for a component is high, you can use this graph to determine whether the problem is server- or network-related.




---

**Note:** Each measurement displayed on the page level is the sum of that measurement recorded for each page component. For example, the network time for [www.cnn.com](http://www.cnn.com) is the sum of the network time for each of the page's components.

---

Network time is defined as the average amount of time that passes from the moment the first HTTP request is sent until receipt of ACK.

Server time is defined as the average amount of time that passes from the receipt of ACK of the initial HTTP request (usually GET) until the first buffer is successfully received back from the Web server.

In the above graph, it is apparent that network time is greater than server time.

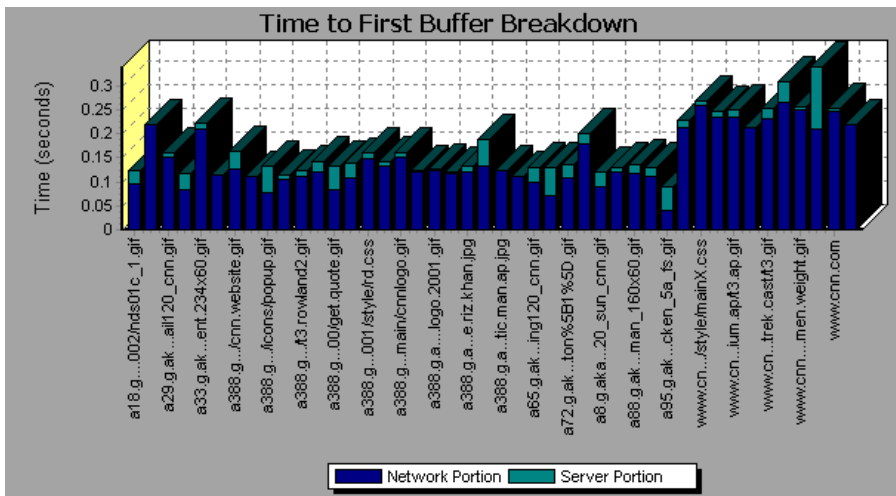
---

**Note:** Because server time is being measured from the client, network time may influence this measurement if there is a change in network performance from the time the initial HTTP request is sent until the time the first buffer is sent. The server time displayed, therefore, is estimated server time and may be slightly inaccurate.

---

The graph can only be viewed as a bar graph.

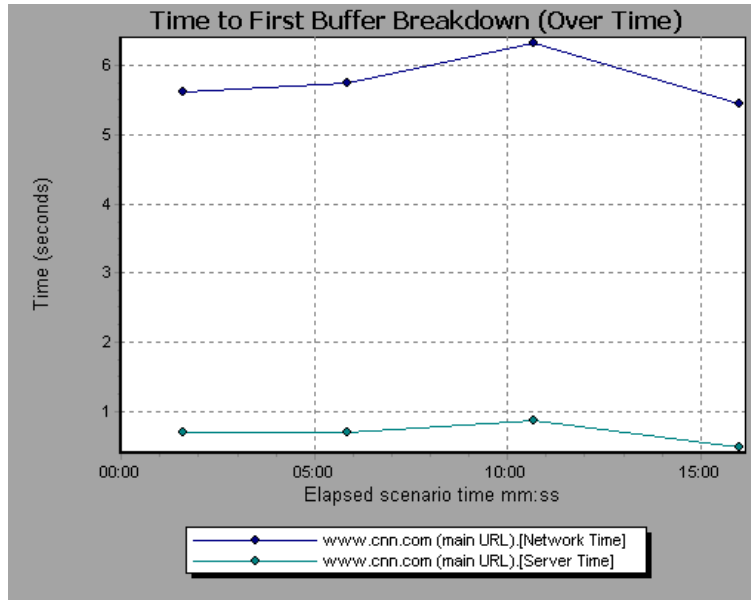
You can break the main cnn.com URL down further to view the time to first buffer breakdown for each of its components.



It is apparent that for the main cnn.com component (the first component on the right), the time to first buffer breakdown is almost all network time.

## Time to First Buffer Breakdown (Over Time) Graph

The Time to First Buffer Breakdown (Over Time) graph displays each Web page component's server and network time (in seconds) during each second of the scenario run, for the period of time until the first buffer is successfully received back from the Web server. You can use this graph to determine when during the scenario run a server- or network-related problem occurred.



Network time is defined as the average amount of time that passes from the moment the first HTTP request is sent until receipt of ACK.

Server time is defined as the average amount of time that passes from the receipt of ACK of the initial HTTP request (usually GET) until the first buffer is successfully received back from the Web server.

---

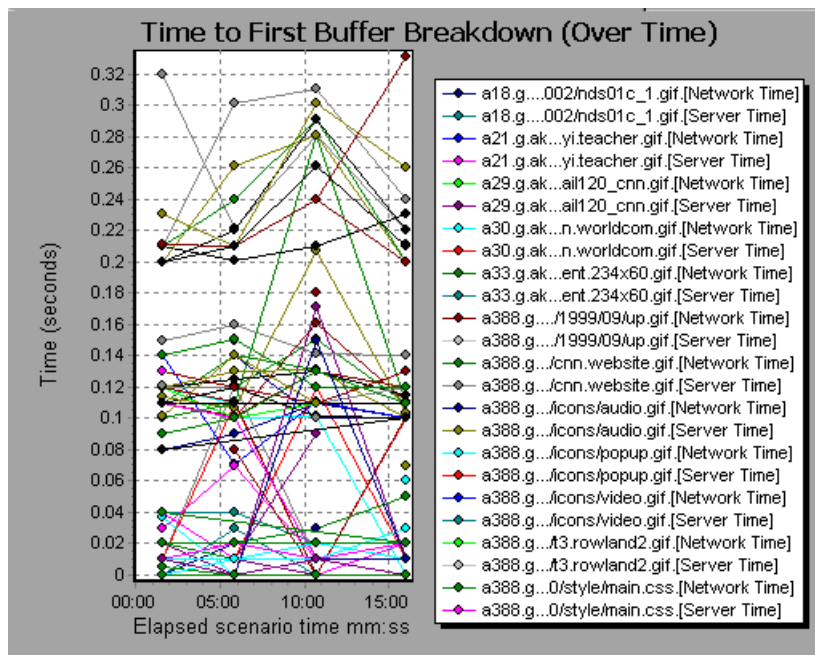
**Note:** Each measurement displayed on the page level is the sum of that measurement recorded for each page component. For example, the network time for [www.cnn.com](http://www.cnn.com) is the sum of the network time for each of the page's components.

---

**Note:** Because server time is being measured from the client, network time may influence this measurement if there is a change in network performance from the time the initial HTTP request is sent until the time the first buffer is sent. The server time displayed, therefore, is estimated server time and may be slightly inaccurate.

---

You can break the main [cnn.com](http://www.cnn.com) URL down further to view the time to first buffer breakdown for each of its components.



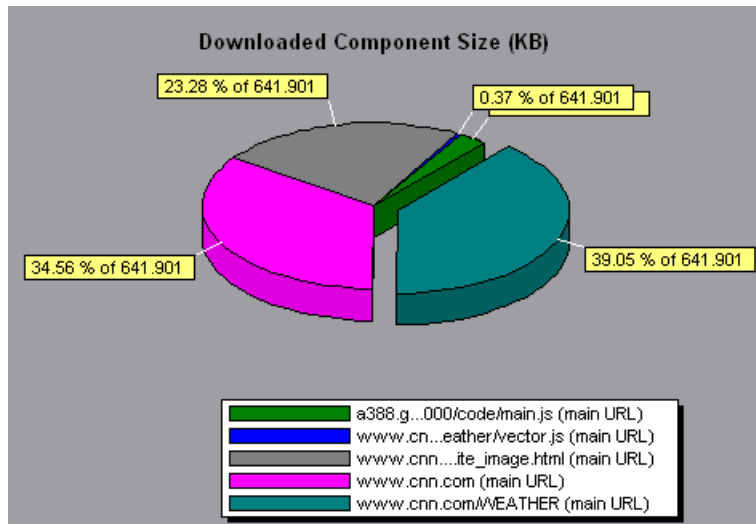
---

**Note:** When the Time to First Buffer Breakdown (Over Time) graph is selected from the Web Page Breakdown graph, it appears as an area graph.

---

## Downloaded Component Size Graph

The Downloaded Component Size graph displays the size of each Web page component. For example, the following graph shows that the `www.cnn.com/WEATHER` component is 39.05% of the total size, whereas the main `cnn.com` component is 34.56% of the total size.

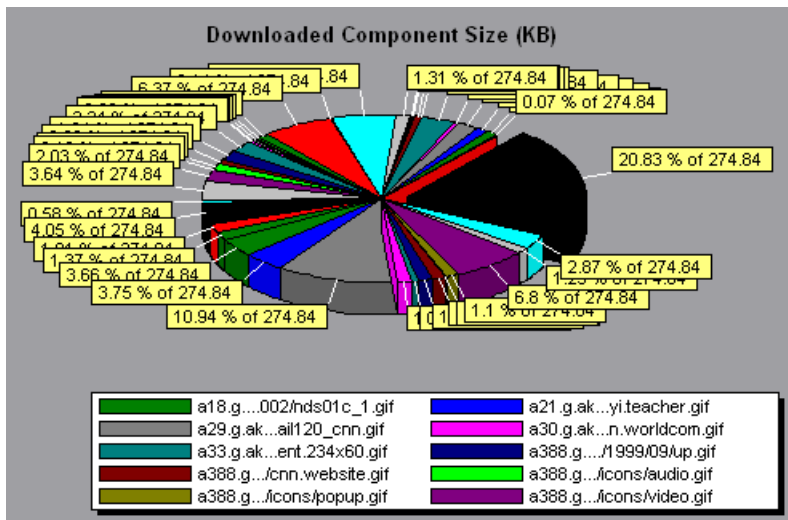


---

**Note:** The Web page size is a sum of the sizes of each of its components.

---

You can break the main cnn.com URL down further to view the size of each of its components.



In the above example, the cnn.com component's size (20.83% of the total size) may have contributed to the delay in its downloading. To reduce download time, it may help to reduce the size of this component.

---

**Note:** The Downloaded Component Size graph can only be viewed as a pie graph.

---



# 8

---

## User-Defined Data Point Graphs

After a scenario run, you can use the User-Defined Data Point graphs to display the values of user-defined data points in the Vuser script.

This chapter describes the:

- ▶ Data Points (Sum) Graph
- ▶ Data Points (Average) Graph

### About User-Defined Data Point Graphs

The User-Defined Data Point graphs display the 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;
}
```

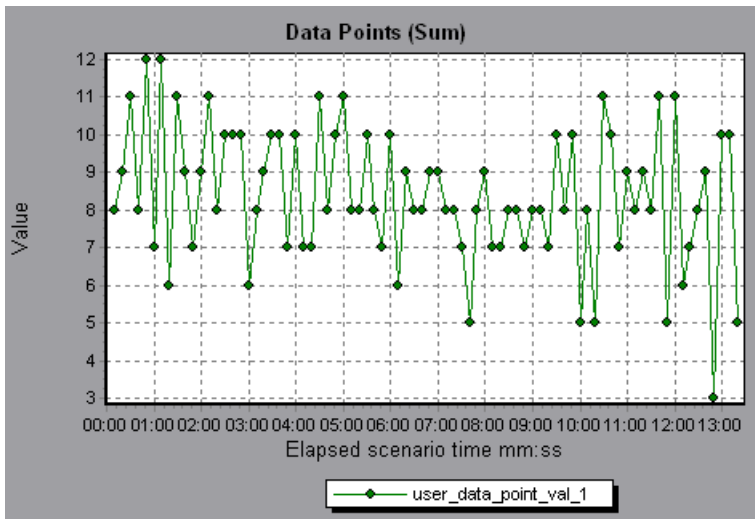
For Vuser protocols that support the graphical script representations such as Web and Oracle NCA, you insert a data point as a User Defined step. Data point information is gathered each time the script executes the function or step. For more information about data points, see the *LoadRunner Online Function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

## Data Points (Sum) Graph

The Data Points (Sum) graph shows the sum values for user-defined data points during the scenario run.

The x-axis represents the number of seconds elapsed since the start time of the run. The y-axis displays the sum values of the recorded data point statements.

In the following example, the *user\_data\_point\_val\_1* data point is shown as a function of the elapsed scenario time.

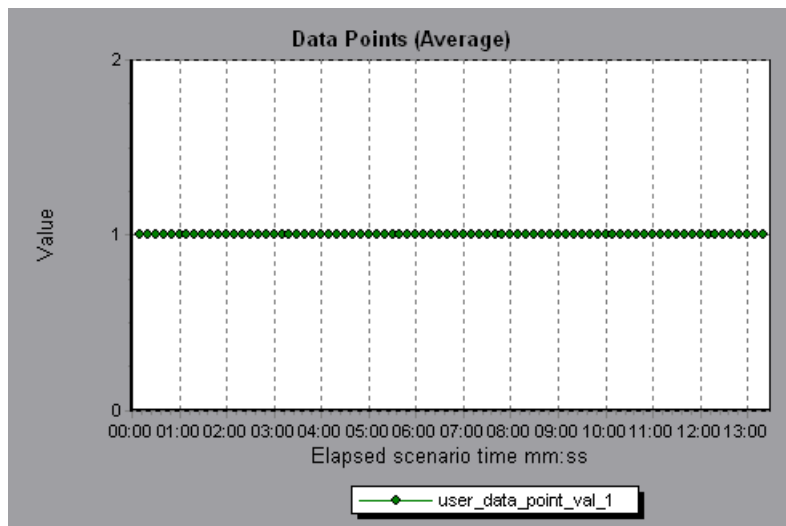


## Data Points (Average) Graph

The Data Points (Average) graph shows the average values that were recorded for user-defined data points during the scenario run.

The x-axis represents the number of seconds elapsed since the start time of the run. The y-axis displays the average values of the recorded data point statements.

In the following example, the *user\_data\_point\_val\_1* data point is shown as a function of the elapsed scenario time.





# 9

---

## System Resource Graphs

After running a scenario, you can check the various system resources that were monitored during the scenario using one or more of the following System Resource graphs:

- Windows Resources Graph
- UNIX Resources Graph
- SNMP Resources Graph
- TUXEDO Resources Graph

### About System Resource Graphs

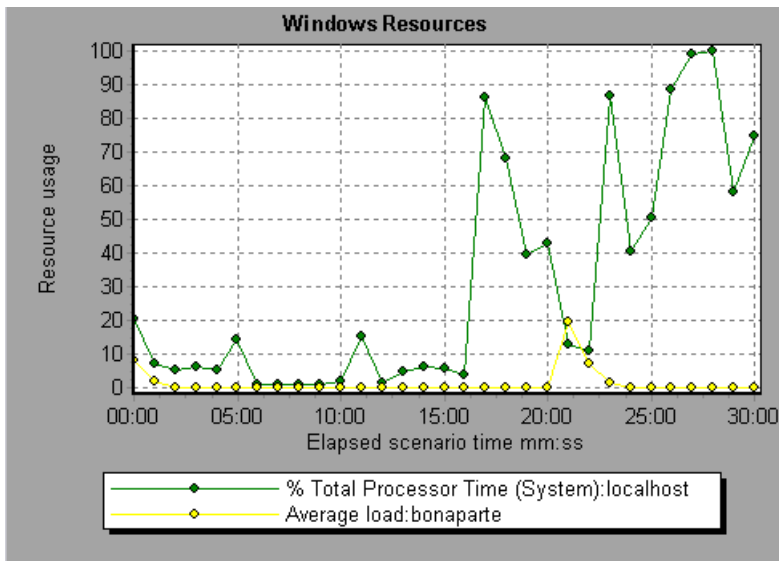
System Resource graphs display the system resource usage measured by the online monitors during the scenario run. These graphs require that you specify the resources you want to measure *before* running the scenario. For more information, see the section on online monitors in the *LoadRunner Controller User's Guide (Windows)*.

System Resource graphs are available for:

- Windows Resources
- UNIX Resources
- SNMP Resources
- TUXEDO Resources

## Windows Resources Graph

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



The following default measurements are available for Windows Resources:

Object	Measurement	Description
<b>System</b>	<b>% Total Processor Time</b>	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.
<b>Processor</b>	<b>% Processor Time (Windows 2000)</b>	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%.
<b>System</b>	<b>File Data Operations/sec</b>	The rate at which the computer issues read and write operations to file system devices. This does not include File Control Operations.

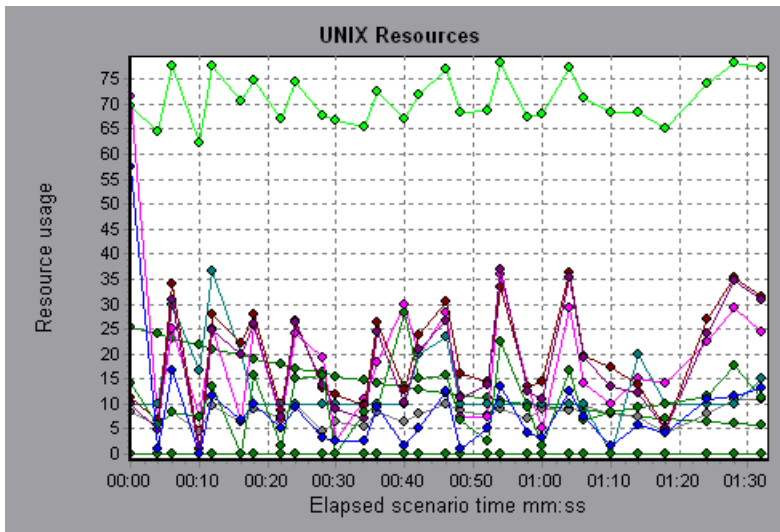
Object	Measurement	Description
System	Processor Queue Length	The instantaneous length of the processor queue in units of threads. This counter is always 0 unless you are also monitoring a thread counter. All processors use a single queue in which threads wait for processor cycles. This length does not include the threads that are currently executing. A sustained processor queue length greater than two generally indicates processor congestion. This is an instantaneous count, not an average over the time interval.
Memory	Page Faults/sec	This is a count of the page faults in the processor. A page fault occurs when a process refers to a virtual memory page that is not in its Working Set in the main memory. A page fault will not cause the page to be fetched from disk if that page is on the standby list (and hence already in main memory), or if it is in use by another process with which the page is shared.
PhysicalDisk	% Disk Time	The percentage of elapsed time that the selected disk drive is busy servicing read or write requests.
Memory	Pool Nonpaged Bytes	The number of bytes in the nonpaged pool, a system memory area where space is acquired by operating system components as they accomplish their appointed tasks. Nonpaged pool pages cannot be paged out to the paging file. They remain in main memory as long as they are allocated.



Object	Measurement	Description
Memory	Pages/sec	The number of pages read from the disk or written to the disk to resolve memory references to pages that were not in memory at the time of the reference. This is the sum of Pages Input/sec and Pages Output/sec. This counter includes paging traffic on behalf of the system cache to access file data for applications. This value also includes the pages to/from non-cached mapped memory files. This is the primary counter to observe if you are concerned about excessive memory pressure (that is, thrashing), and the excessive paging that may result.
System	Total Interrupts/sec	The rate at which the computer is receiving and servicing hardware interrupts. The devices that can generate interrupts are the system timer, the mouse, data communication lines, network interface cards, and other peripheral devices. This counter provides an indication of how busy these devices are on a computer-wide basis. See also Processor:Interrupts/sec.
Objects	Threads	The number of threads in the computer at the time of data collection. Notice that this is an instantaneous count, not an average over the time interval. A thread is the basic executable entity that can execute instructions in a processor.
Process	Private Bytes	The current number of bytes that the process has allocated that cannot be shared with other processes.

## UNIX Resources Graph

The UNIX Resources graph shows the UNIX resources measured during the scenario. 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.



The following default measurements are available for UNIX machines:

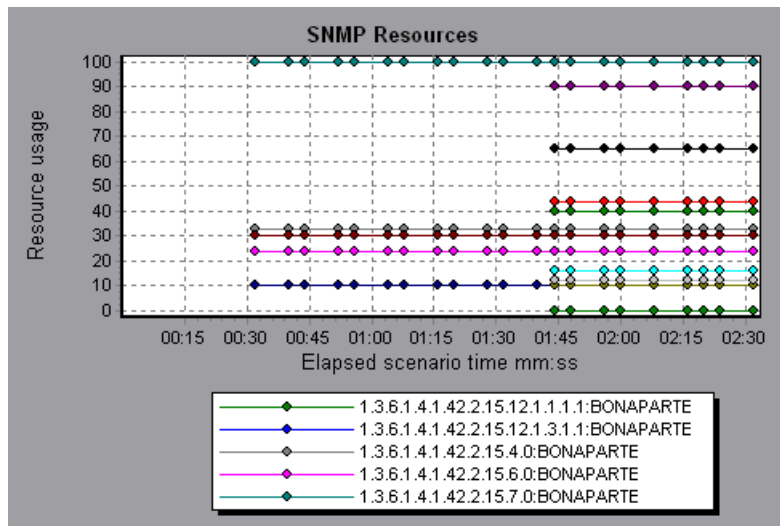
Measurement	Description
Average load	Average number of processes simultaneously in 'Ready' state during the last minute
Collision rate	Collisions per second detected on the Ethernet
Context switches rate	Number of switches between processes or threads, per second
CPU utilization	Percent of time that the CPU is utilized
Disk rate	Rate of disk transfers

<b>Measurement</b>	<b>Description</b>
<b>Incoming packets error rate</b>	Errors per second while receiving Ethernet packets
<b>Incoming packets rate</b>	Incoming Ethernet packets per second
<b>Interrupt rate</b>	Number of device interrupts per second
<b>Outgoing packets errors rate</b>	Errors per second while sending Ethernet packets
<b>Outgoing packets rate</b>	Outgoing Ethernet packets per second
<b>Page-in rate</b>	Number of pages read to physical memory, per second
<b>Page-out rate</b>	Number of pages written to pagefile(s) and removed from physical memory, per second
<b>Paging rate</b>	Number of pages read to physical memory or written to pagefile(s), per second
<b>Swap-in rate</b>	Number of processes being swapped
<b>Swap-out rate</b>	Number of processes being swapped
<b>System mode CPU utilization</b>	Percent of time that the CPU is utilized in system mode
<b>User mode CPU utilization</b>	Percent of time that the CPU is utilized in user mode

## SNMP Resources Graph

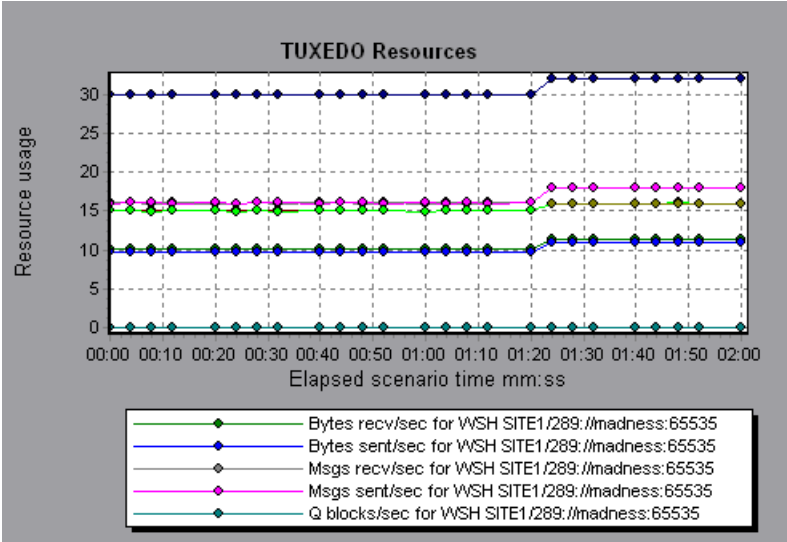
The SNMP Resources graph shows statistics for machines running an SNMP agent, using the Simple Network Management Protocol (SNMP).

The following graph displays SNMP measurements for a machine called *bonaparte*:



# TUXEDO Resources Graph

The TUXEDO Resources graph provides information about the server, load generator machine, workstation handler, and queue in a TUXEDO system.



The following table lists the available TUXEDO monitor measurements:

Monitor	Measurements
Server	<b>Requests per second</b> - How many server requests were handled per second
	<b>Workload per second</b> -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.
Machine	<b>Workload completed per second</b> - The total workload on all the servers for the machine that was completed, per unit time
	<b>Workload initiated per second</b> - The total workload on all the servers for the machine that was initiated, per unit time

Monitor	Measurements
Queue	<b>Bytes on queue</b> - The total number of bytes for all the messages waiting in the queue
	<b>Messages on queue</b> - The total number of requests that are waiting on queue. By default this is 0.
<b>Workstation Handler (WSH)</b>	<b>Bytes received per second</b> - The total number of bytes received by the workstation handler, per unit time
	<b>Bytes sent per second</b> - The total number of bytes sent back to the clients by the workstation handler, per unit time
	<b>Messages received per second</b> - The number of messages received by the workstation handler, per unit time
	<b>Messages sent per second</b> - The number of messages sent back to the clients by the workstation handler, per unit time
	<b>Number of queue blocks per second</b> - The number of times the queue for the workstation handler blocked, per unit time. This gives an idea of how often the workstation handler was overloaded.

# 10

---

## Network Monitor Graphs

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

This chapter describes:

- Understanding Network Monitoring
- Network Delay Time Graph
- Network Sub-Path Time Graph
- Network Segment Graph
- Verifying the Network as a Bottleneck

### About Network Monitoring

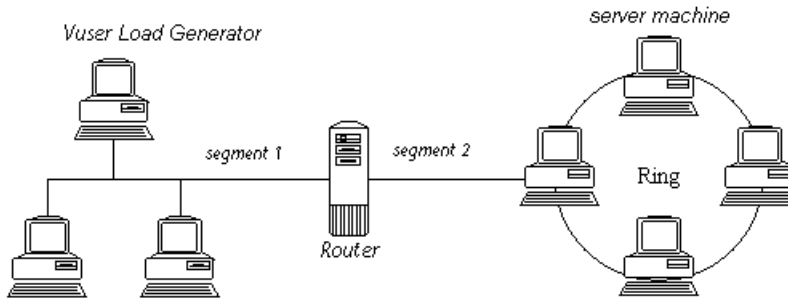
Network configuration is a primary factor in the performance of applications and Web systems. A poorly designed network can slow client activity to unacceptable levels.

In an application, there are many network segments. A single network segment with poor performance can affect the entire application.

Network graphs let you locate the network-related problem so that it can be fixed.

## Understanding Network Monitoring

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



To measure network performance, the Network monitor sends packets of data across the network. When a packet returns, the monitor calculates the time it takes for the packet to go to the requested node and return. The Network Sub-Path Time graph displays the delay from the source machine to each node along the path. The Network Segment graph displays the delay for each segment of the path. The Network Delay Time graph displays the delay for the complete path between the source and destination machines.

Using the Network Monitor graphs, you can determine whether the network is causing a bottleneck. If the problem is network-related, you can locate the problematic segment so that it can be fixed.

In order for the Analysis to generate Network Monitor graphs, you must activate the Network monitor before executing the scenario. In the Network Monitor settings, you specify the path you want to monitor. For information about setting up the Network monitor, see the *LoadRunner Controller User's Guide (Windows)*.

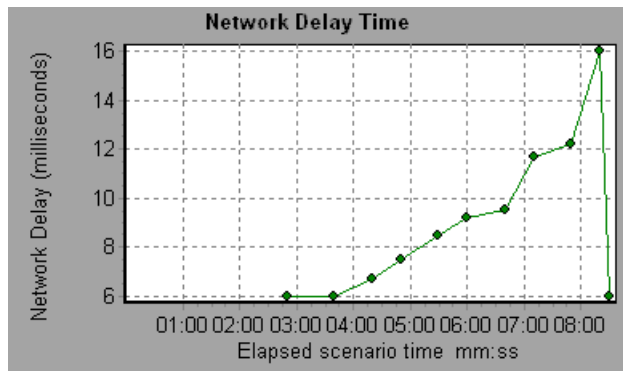


## Network Delay Time Graph

The Network Delay Time graph shows the delays for the complete path between the source and destination machines (for example, the database server and Vuser load generator). The graph maps the delay as a function of the elapsed scenario time.

Each path defined in the Controller is represented by a separate line with a different color in the graph.

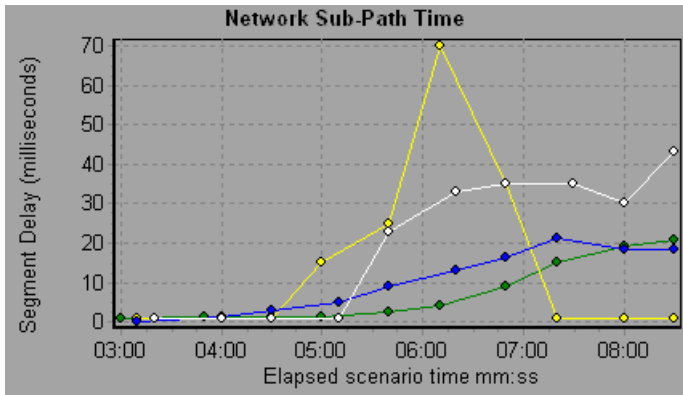
The following graph shows the network delay as a function of the elapsed scenario time. It shows a delay of 16 milliseconds in the 8th minute of the scenario.



## Network Sub-Path Time Graph

The Network Sub-Path Time graph shows the delay for each segment of the path according to the elapsed scenario time. Each segment is displayed as a separate line with a different color.

In the following example, four segments are shown. The graph indicates that one segment caused a delay of 70 milliseconds in the sixth minute.



---

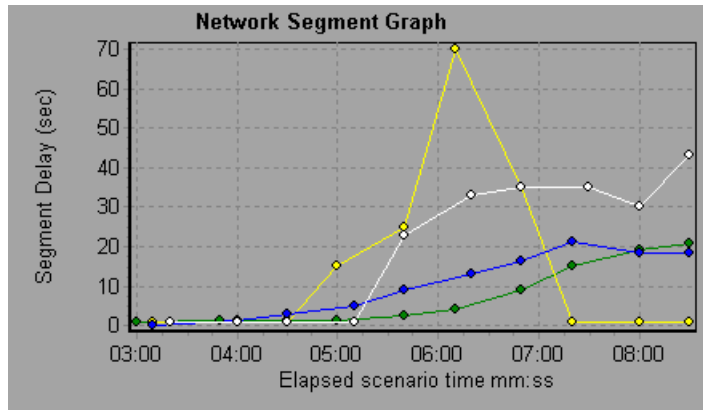
**Note:** The delays from the source machine to each of the nodes are measured concurrently, yet independently. It is therefore possible that the delay from the source machine to one of the nodes could be greater than the delay for the complete path between the source and destination machines.

---

## Network Segment Graph

The Network Segment graph shows the delay for each segment of the path according to the elapsed scenario time. Each segment is displayed as a separate line with a different color.

In the following example, four segments are shown. The graph indicates that one segment caused a delay of 70 seconds in the sixth minute.




---

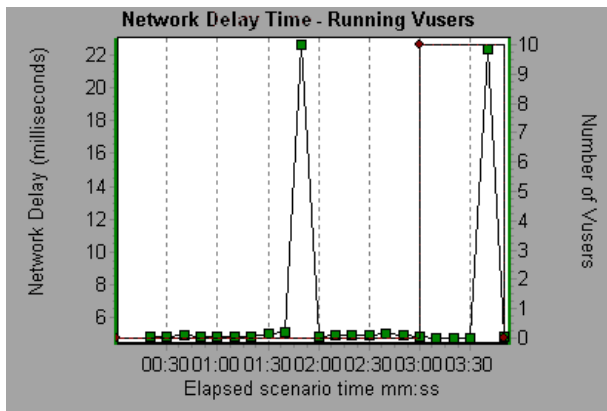
**Note:** The segment delays are measured approximately, and do not add up to the network path delay which is measured exactly. The delay for each segment of the path is estimated by calculating the delay from the source machine to one node and subtracting the delay from the source machine to another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

---

## Verifying the Network as a Bottleneck

You can merge various graphs to determine if the network is a bottleneck. For example, using the Network Delay Time and Running Vusers graphs, you can determine how the number of Vusers affects the network delay. The Network Delay Time graph indicates the network delay during the scenario run. The Running Vusers graph shows the number of running Vusers.

In the following merged graph, the network delays are compared to the running Vusers. The graph shows that when all 10 Vusers were running, a network delay of 22 milliseconds occurred, implying that the network may be overloaded.



# 11

---

## Firewall Server Monitor Graphs

After a scenario run, you can use the Firewall server monitor graphs to analyze Firewall server performance.

This chapter describes the:

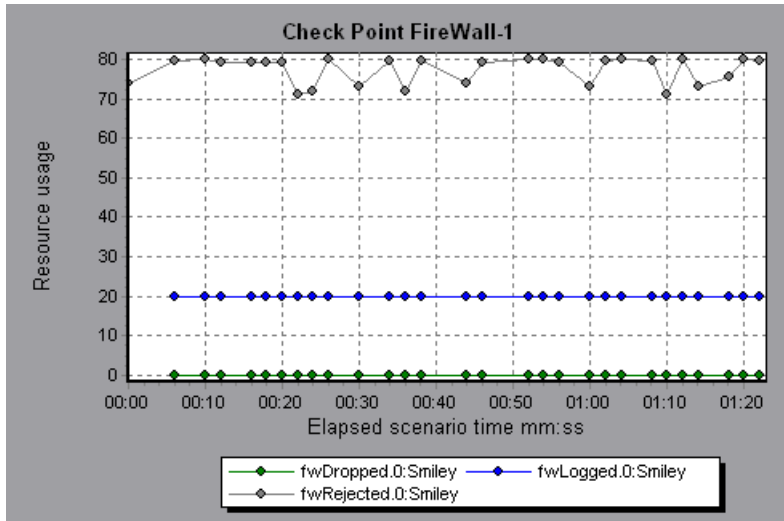
- ▶ Check Point FireWall-1 Server Graph

### About Firewall Server Monitor Graphs

Firewall server monitor graphs provide you with performance information for firewall servers. Note that in order to obtain data for these graphs, you need to activate the Firewall server online monitor before running the scenario. When you set up the online monitor for the Firewall server, you indicate which statistics and measurements to monitor. For more information on activating and configuring Firewall server monitors, see the *LoadRunner Controller User's Guide (Windows)*.

## Check Point FireWall-1 Server Graph

The Check Point FireWall-1 graph shows statistics on Check Point's Firewall server as a function of the elapsed scenario time.



This graph displays the *fwDropped*, *fwLogged*, and *fwRejected* measurements during the first minute and twenty seconds of the scenario. Note that there are differences in the scale factor for the measurements: the scale factor for *fwDropped* is 1, the scale factor for *fwLogged* is 10, and the scale factor for *fwRejected* is 0.0001.

The following measurements are available for the Check Point FireWall-1 server:

Measurement	Description
<b>fwRejected</b>	The number of rejected packets.
<b>fwDropped</b>	The number of dropped packets.
<b>fwLogged</b>	The number of logged packets.

# 12

---

## Web Server Resource Graphs

After a scenario run, you can use Web Server Resource graphs to analyze the performance of your Apache, Microsoft IIS, and Netscape server performance.

This chapter describes:

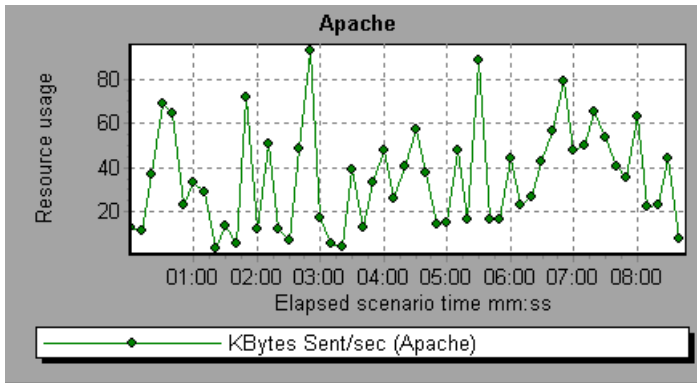
- ▶ Apache Server Graph
- ▶ Microsoft Information Internet Server (IIS) Graph
- ▶ Netscape Enterprise Server Graph

### About Web Server Resource Graphs

Web Server Resource graphs provide you with information about the resource usage of the Apache, Microsoft IIS, and Netscape Web servers. In order to obtain data for these graphs, you need to activate the online monitor for the server and specify which resources you want to measure before running the scenario. For information on activating and configuring the Web Server Resource monitors, see the *LoadRunner Controller User's Guide (Windows)*.

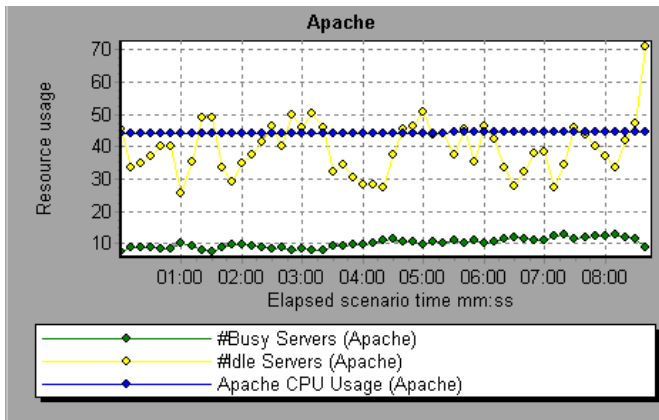
In order to display all the measurements on a single graph, the Analysis may scale them. The Legend tab indicates the scale factor for each resource. To obtain the true value, multiply the scale factor by the displayed value. For example, in the following graph the actual value of *KBytes Sent per second* in

the second minute is 1; 10 multiplied by the scale factor of 1/10 (indicated in the Legend tab below).



## Apache Server Graph

The Apache Server graph shows server statistics as a function of the elapsed scenario time.



In the above graph, the CPU usage remained steady throughout the scenario. At the end of the scenario, the number of idle servers increased. The number of busy servers remained steady at 1 throughout the scenario, implying that the Vuser only accessed one Apache server.



Note that the scale factor for the *Busy Servers* measurement is 1/10 and the scale factor for *CPU usage* is 10.

The following default measurements are available for the Apache server:

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

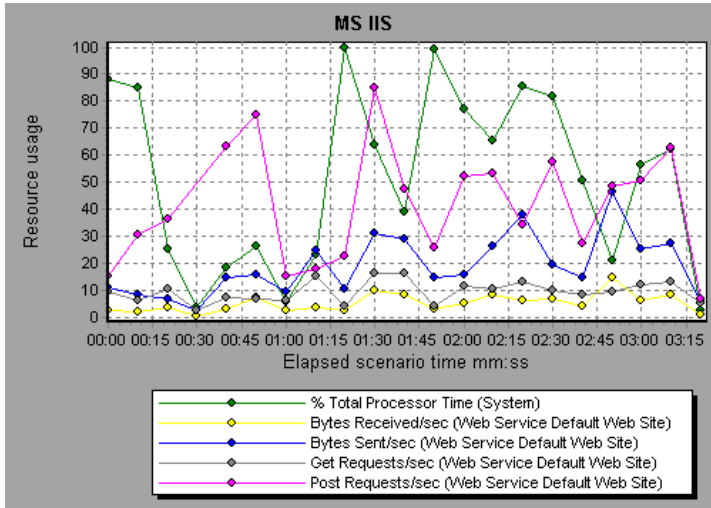
---

**Note:** The Apache monitor connects to the Web server in order to gather statistics, and registers one hit for each sampling. The Apache graph, therefore, always displays one hit per second, even if no clients are connected to the Apache server.

---

## Microsoft Information Internet Server (IIS) Graph

The Microsoft IIS Server graph shows server statistics as a function of the elapsed scenario time.



In the above graph, the *Bytes Received/sec* and *Get Requests/sec* measurements remained fairly steady throughout the scenario, while the *% Total Processor Time*, *Bytes Sent/sec*, and *Post Requests/sec* measurements fluctuated considerably.

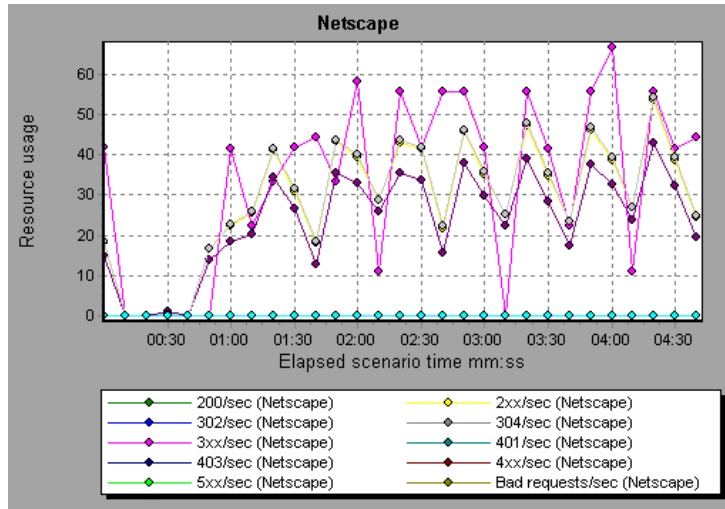
Note that the scale factor for the *Bytes Sent/sec* and *Bytes Received/sec* measurements is 1/100, and the scale factor for the *Post Requests/sec* measurement is 10.

The following default measurements are available for the IIS server:

<b>Object</b>	<b>Measurement</b>	<b>Description</b>
<b>Web Service</b>	<b>Bytes Sent/sec</b>	The rate at which the data bytes are sent by the Web service
<b>Web Service</b>	<b>Bytes Received/sec</b>	The rate at which the data bytes are received by the Web service
<b>Web Service</b>	<b>Get Requests/sec</b>	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.
<b>Web Service</b>	<b>Post Requests/sec</b>	The rate at which HTTP requests using the POST method are made. Post requests are generally used for forms or gateway requests.
<b>Web Service</b>	<b>Maximum Connections</b>	The maximum number of simultaneous connections established with the Web service
<b>Web Service</b>	<b>Current Connections</b>	The current number of connections established with the Web service
<b>Web Service</b>	<b>Current NonAnonymous Users</b>	The number of users that currently have a non-anonymous connection using the Web service
<b>Web Service</b>	<b>Not Found Errors/sec</b>	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.
<b>Process</b>	<b>Private Bytes</b>	The current number of bytes that the process has allocated that cannot be shared with other processes.

## Netscape Enterprise Server Graph

The Netscape Server graph shows server statistics as a function of the elapsed scenario time.



Note that the scale factor for the *302/sec* and *3xx/sec* measurements is 100, and the scale factor for the *Bytes Sent/sec* is 1/100.

The following default measurements are available for the Netscape server:

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

Measurement	Description
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 not return an HTTP status code



# 13

---

## Web Application Server Resource Graphs

After a scenario run, you can use Web Application Server Resource graphs to analyze Web application server performance.

This chapter describes:

- Web Application Server Resource Graphs

### About Web Application Server Resource Graphs

Web Application Server Resource graphs provide you with resource usage information about the following application servers:

- Ariba
- ATG Dynamo
- BroadVision
- ColdFusion
- Fujitsu INTERSTAGE
- Microsoft Active Server Pages (ASP)
- SilverStream
- WebLogic
- WebSphere

In order to obtain data for these graphs, you need to activate the online monitor for the application server and specify which resources you want to measure before running the scenario. For information on activating and configuring the Web Application Server Resource monitors, see the *LoadRunner Controller User's Guide (Windows)*.

When you open a Web Application Server Resource graph, you can filter it to show only the relevant application. When you need to analyze other applications, you can change the filter conditions and display the desired resources.

In order to display all the measurements on a single graph, the Analysis may scale them. The Legend tab indicates the scale factor for each resource. To obtain the true value, multiply the scale factor by the displayed value. For more information, see page 107.

## Web Application Server Resource Graphs

Each Web Application Server Resource graph measures different resources. These are detailed in the following sections.

### Ariba

The following tables describe the default measurements available for the Ariba server:

#### Core Server Performance Counters

Measurement	Description
Total Connections	The cumulative number of concurrent user connections since Ariba Buyer was started.
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.



Measurement	Description
<b>Worker Queue Length</b>	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.
<b>Concurrent Connections</b>	The instantaneous reading of the number of concurrent user connections at the moment this metric is obtained
<b>Total Memory</b>	The instantaneous reading of the memory (in KB) being used by Ariba Buyer at the moment this metric is obtained
<b>Free Memory</b>	The instantaneous reading of the reserved memory (in KB) that is not currently in use at the moment this metric is obtained
<b>Up Time</b>	The amount of time (in hours and minutes) that Ariba Buyer has been running since the previous time it was started
<b>Number of Threads</b>	The instantaneous reading of the number of server threads in existence at the moment this metric is obtained
<b>Number of Cached Objects</b>	The instantaneous reading of the number of Ariba Buyer objects being held in memory at the moment this metric is obtained
<b>Average Session Length</b>	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.
<b>Average Idle Time</b>	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.
<b>Approves</b>	The cumulative count of the number of approves that happened during the sampling period. An Approve consists of a user approving one Approvable.
<b>Submits</b>	The cumulative count of the number of Approvables submitted since previous sampling time

Measurement	Description
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**

The following tables describe the measurements that are available for the ATG Dynamo server:

<b>d3System Measurement</b>	<b>Description</b>
<b>sysTotalMem</b>	The total amount of memory currently available for allocating objects, measured in bytes
<b>sysFreeMem</b>	An approximation of the total amount of memory currently available for future allocated objects, measured in bytes
<b>sysNumInfoMsgs</b>	The number of system global info messages written
<b>sysNumWarningMsgs</b>	The number of system global warning messages written
<b>sysNumErrorMsgs</b>	The number of system global error messages written

<b>d3LoadManagement Measurement</b>	<b>Description</b>
<b>ImIsManager</b>	True if the Dynamo is running a load manager
<b>ImManagerIndex</b>	Returns the Dynamo's offset into the list of load managing entities
<b>ImIsPrimaryManager</b>	True if the load manager is an acting primary manager
<b>ImServicingCMs</b>	True if the load manager has serviced any connection module requests in the amount of time set as the connection module polling interval
<b>ImCMLDRPPort</b>	The port of the connection module agent
<b>ImIndex</b>	A unique value for each managed entity
<b>ImSNMPPort</b>	The port for the entry's SNMP agent
<b>ImProbability</b>	The probability that the entry will be given a new session

<b>d3LoadManagement Measurement</b>	<b>Description</b>
<b>ImNewSessions</b>	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 ImNewSessionOverride.
<b>ImNewSessionOverride</b>	The override set for whether or not a server is accepting new sessions

<b>d3SessionTracking Measurement</b>	<b>Description</b>
<b>stCreatedSessionCnt</b>	The number of created sessions
<b>stValidSessionCnt</b>	The number of valid sessions
<b>stRestoredSessionCnt</b>	The number of sessions migrated to the server
<b>StDictionaryServerStatus</b>	d3Session Tracking

<b>d3DRP Server Measurement</b>	<b>Description</b>
<b>drpPort</b>	The port of the DRP server
<b>drpTotalReqsServed</b>	Total number of DRP requests serviced
<b>drpTotalReqTime</b>	Total service time in msec for all DRP requests
<b>drpAvgReqTime</b>	Average service time in msec for each DRP request
<b>drpNewessions</b>	True if the Dynamo is accepting new sessions

<b>d3DBConnPooling Measurement</b>	<b>Description</b>
<b>dbPoolsEntry</b>	A pooling service entry containing information about the pool configuration and current status
<b>dbIndex</b>	A unique value for each pooling service
<b>dbPoolID</b>	The name of the DB connection pool service
<b>dbMinConn</b>	The minimum number of connections pooled
<b>dbMaxConn</b>	The maximum number of connections pooled
<b>dbMaxFreeConn</b>	The maximum number of free pooled connections at a time
<b>dbBlocking</b>	Indicates whether or not the pool is to block out check outs
<b>dbConnOut</b>	Returns the number of connections checked out
<b>dbFreeResources</b>	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.
<b>dbTotalResources</b>	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

The BroadVision monitor supplies performance statistics for all the servers/services available within the application server.

The following table describes all the servers/services that are available:

Server	Multiple Instances	Description
<b>adm_srv</b>	No	One-To-One user administration server. There must be one.
<b>alert_srv</b>	No	Alert server handles direct IDL function calls to the Alert system.
<b>bvconf_srv</b>	No	One-To-One configuration management server. There must be one.
<b>cmsdb</b>	Yes	Visitor management database server.
<b>cntdb</b>	Yes	Content database server.
<b>deliv_sntp_d</b>	Yes	Notification delivery server for e-mail type messages. Each instance of this server must have its own ID, numbered sequentially starting with "1".
<b>deliv_comp_d</b>	No	Notification delivery completion processor.
<b>extdbacc</b>	Yes	External database accessor. You need at least one for each external data source.
<b>genericdb</b>	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.
<b>hostmgr</b>	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.
<b>g1_ofbe_srv</b>	No	Order fulfillment back-end server.

Server	Multiple Instances	Description
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. Please note that for some services the number of counters for the same group can be different.

Counter groups:

- BV\_DB\_STAT
- BV\_SRV\_CTRL
- BV\_SRV\_STAT
- NS\_STAT
- BV\_CACHE\_STAT
- JS\_SCRIPT\_CTRL
- JS\_SCRIPT\_STAT

### **BV\_DB\_STAT**

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

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

### **BV\_SRV\_CTRL**

- SHUTDOWN



**NS\_STAT**

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

- Bind
- List
- New
- Rebind
- Rsvl
- Unbind

**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 NT Task Manager will report 25%. The value reported by this statistic is consistent with "% Processor Time" on the Windows NT 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.
- **IdIQ**
- **JOB**
- **LWP** - Number of light-weight processes (threads).
- **RSS** - Resident memory size of server process (in Kilobytes).
- **STIME** - System start time.
- **SESS** - Current count of sessions.
- **SYS** - Accumulated system mode CPU time (seconds).
- **THR** - Current count of threads.
- **USR** - Accumulated user mode CPU time (seconds).
- **VSZ** - Virtual memory size of server process (in kilobytes). If a process is growing in size, it probably has a memory leak. If it is an Interaction Manager process, the culprit is most likely a component or dynamic object (though Interaction Manager servers do grow and shrink from garbage collection during normal use).

### **BV\_CACHE\_STAT**

Monitors the request cache status.

The available counters for each request are:

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

## Cache Metrics

Cache metrics are available for the following items:

- **AD**
- **ALERTSCHED** - Notification schedules are defined in the BV\_ALERTSCHED and BV\_MSGSCHED tables. They are defined by the One-To-One Command Center user or by an application.
- **CATEGORY\_CONTENT**
- **DISCUSSION** - The One-To-One discussion groups provide moderated system of messages and threads of messages aligned to a particular topic. Use the Discussion group interfaces for creating, retrieving and deleting individual messages in a discussion group. To create, delete, or retrieve discussion groups, use the generic content management API. The BV\_DiscussionDB object provides access to the threads and messages in the discussion group database.
- **EXT\_FIN\_PRODUCT**
- **EDITORIAL** - Using the Editorials content module, you can point cast and community cast personalized editorial content, and sell published text on your One-To-One site. You can solicit editorial content, such as investment reports and weekly columns, from outside authors and publishers, and create your own articles, reviews, reports, and other informative media. In addition to text, you can use images, sounds, music, and video presentations as editorial content.
- **INCENTIVE** - Contains sales incentives
- **MSGSCHED** - Contains the specifications of visitor-message jobs. Notification schedules are defined in the BV\_ALERTSCHED and BV\_MSGSCHED tables. They are defined by the One-To-One Command Center user or by an application.
- **MSGSCRIPT** - Contains the descriptions of the JavaScripts that generate visitor messages and alert messages. Contains the descriptions of the JavaScripts that generate targeted messages and alert messages. Use the Command Center to add message script information to this table by selecting the Visitor Messages module in the Notifications group. For more information, see the Command Center User's Guide.

- **PRODUCT** - BV\_PRODUCT contains information about the products that a visitor can purchase.
- **QUERY** - BV\_QUERY contains queries.
- **SCRIPT** - BV\_SCRIPT contains page scripts.
- **SECURITIES**
- **TEMPLATE** - The Templates content module enables you to store in the content database any BroadVision page templates used on your One-To-One site. Combining BroadVision page templates with BroadVision dynamic objects in the One-To-One Design Center application is one way for site developers to create One-To-One Web sites. If your developers use these page templates, you can use the Command Center to enter and manage them in your content database. If your site doesn't use BroadVision page template, you will not use this content module.

### JS\_SCRIPT\_CTRL

- CACHE
- DUMP
- FLUSH
- METER
- TRACE

### JS\_SCRIPT\_STAT

- ALLOC
- ERROR
- FAIL
- JSPERR
- RELEASE
- STOP
- SUCC
- SYNTAX

## ColdFusion

The following default measurements are available for Allaire's ColdFusion server:

Measurement	Description
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 Out/sec	The number of bytes per second returned 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.

## Fujitsu INTERSTAGE

The following default measurements are available for the Fujitsu INTERSTAGE server:

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

Measurement	Description
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

### Microsoft Active Server Pages (ASP)

The following default measurements are available for Microsoft Active Server Pages (ASP):

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.

Measurement	Description
<b>Errors During Script Run-Time</b>	The number of failed requests due to run-time errors.
<b>Sessions Current</b>	The current number of sessions being serviced.
<b>Transactions/sec</b>	The number of transactions started per second.

### SilverStream

The following default measurements are available for the SilverStream server:

Measurement	Description
<b>#Idle Sessions</b>	The number of sessions in the Idle state.
<b>Avg. Request processing time</b>	The average request processing time.
<b>Bytes Sent/sec</b>	The rate at which data bytes are sent from the Web server.
<b>Current load on Web Server</b>	The percentage of load utilized by the SilverStream server, scaled at a factor of 25.
<b>Hits/sec</b>	The HTTP request rate.
<b>Total sessions</b>	The total number of sessions.
<b>Free memory</b>	The total amount of memory in the Java Virtual Machine currently available for future allocated objects.
<b>Total memory</b>	The total amount of memory in the Java Virtual Machine.
<b>Memory Garbage Collection Count</b>	The total number of times the JAVA Garbage Collector has run since the server was started.
<b>Free threads</b>	The current number of threads not associated with a client connection and available for immediate use.

Measurement	Description
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.

---

**Note:** The SilverStream monitor connects to the Web server in order to gather statistics, and registers one hit for each sampling. The SilverStream graph, therefore, always displays one hit per second, even if no clients are connected to the SilverStream server.

---

## WebLogic

The following default measurements are available for the WebLogic server:

### ► Server Table

The Server Table lists all WebLogic 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. <i>Up</i> implies that the agent can contact the server. <i>Down</i> implies that the agent cannot contact the server.
ServerLoginEnable	This value is true if client logins are enabled on the server.
ServerMaxHeapSpace	The maximum heap size for this server, in KB
ServerHeapUsedPct	The percentage of heap space currently in use on the server



Measurement	Description
<b>ServerQueueLength</b>	The current length of the server execute queue
<b>ServerQueueThroughput</b>	The current throughput of execute queue, expressed as the number of requests processed per second
<b>ServerNumEJBDeployment</b>	The total number of EJB deployment units known to the server
<b>ServerNumEJBBeansDeployed</b>	The total number of EJB beans actively deployed on the server

► **Listen Table**

The Listen Table is the set of protocol, IP address, and port combinations on which servers are listening. There will be multiple entries for each server: one for each protocol, ipAddr, port) combination. If clustering is used, the clustering-related MIB objects will assume a higher priority.

Measurement	Description
<b>ListenPort</b>	Port number.
<b>ListenAdminOK</b>	True if admin requests are allowed on this (protocol, ipAddr, port); otherwise false
<b>ListenState</b>	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 server, and servlets. There are multiple entries in this table for each server. There may also be multiple entries for each path on a server. If clustering is used, the clustering-related MIB objects will assume a higher priority.

Measurement	Description
CPType	The type of CP element: Java, WebLogic, servlet. A Java CPTYPE means the cpElement is one of the elements in the normal Java classpath. A WebLogic CPTYPE means the cpElement is one of the elements in weblogic.class.path. A servlet CPTYPE means the cpElement is one of the elements in the dynamic servlet classpath.
CPIIndex	The position of an element within its path. The index starts at 1.

**WebSphere**

The following measurements are available for the WebSphere server:

► **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
<b>BeanCreates</b>	The number of beans created. Applies to an individual bean that is either 'stateful' or 'entity'
<b>EntityBeanCreates</b>	The number of entity beans created
<b>BeanRemoves</b>	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'
<b>EntityBeanRemoves</b>	The number of entity beans removed
<b>StatefulBeanCreates</b>	The number of stateful beans created
<b>StatefulBeanRemoves</b>	The number of stateful bean removed
<b>BeanPassivates</b>	The number of bean passivates pertaining to a specific bean. Applies to an individual bean that is either 'stateful' or 'entity'
<b>EntityBeanPassivates</b>	The number of entity bean passivates
<b>StatefulBeanPassivates</b>	The number of stateful bean passivates
<b>BeanActivates</b>	The number of bean activates pertaining to a specific bean. Applies to an individual bean that is either 'stateful' or 'entity'
<b>EntityBeanActivates</b>	The number of entity bean activates
<b>StatefulBeanActivates</b>	The number of stateful bean activates
<b>BeanLoads</b>	The number of times the bean data was loaded. Applies to entity
<b>BeanStores</b>	The number of times the bean data was stored in the database. Applies to entity

Measurement	Description
<b>BeanInstantiates</b>	The number of times a bean object was created. This applies to an individual bean, regardless of its type.
<b>StatelessBeanInstantiates</b>	The number of times a stateless session bean object was created
<b>StatefulBeanInstantiates</b>	The number of times a stateful session bean object was created
<b>EntityBeanInstantiates</b>	The number of times an entity bean object was created
<b>BeanDestroys</b>	The number of times an individual bean object was destroyed. This applies to any bean, regardless of its type
<b>StatelessBeanDestroys</b>	The number of times a stateless session bean object was destroyed
<b>StatefulBeanDestroys</b>	The number of times a stateful session bean object was destroyed
<b>EntityBeanDestroys</b>	The number of times an entity bean object was destroyed
<b>BeansActive</b>	The average number of instances of active beans pertaining to a specific bean. Applies to an individual bean that is either 'stateful' or 'entity'
<b>EntityBeansActive</b>	The average number of active entity beans
<b>StatefulBeansActive</b>	The average number of active session beans
<b>BeansLive</b>	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.
<b>StatelessBeansLive</b>	The average number of stateless session bean objects that are instantiated but not yet destroyed
<b>StatefulBeansLive</b>	The average number of stateful session bean objects that are instantiated but not yet destroyed

Measurement	Description
<b>EntityBeansLive</b>	The average number of entity bean objects that are instantiated but not yet destroyed
<b>BeanMethodRT</b>	The average method response time for all methods defined in the remote interface to this bean. Applies to all beans
<b>BeanMethodActive</b>	The average number of methods being processed concurrently. Applies to all beans
<b>BeanMethodCalls</b>	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
<b>BeanObjectPoolContainer</b>	The pool of a specific bean type
<b>BeanObject</b>	The pool specific to a home
<b>NumGet</b>	The number of calls retrieving an object from the pool
<b>NumGetFound</b>	The number of calls to the pool that resulted in finding an available bean
<b>NumPuts</b>	The number of beans that were released to the pool
<b>NumPutsDiscarded</b>	The number of times releasing a bean to the pool resulted in the bean being discarded because the pool was full
<b>NumDrains</b>	The number of times the daemon found the pool was idle and attempted to clean it

Measurement	Description
DrainSize	The average number of beans discarded by the daemon during a clean
BeanPoolSize	The average number of beans in the pool

► **OrbThreadPool**

These are resources related to the ORB thread pool that is on the server.

Measurement	Description
ActiveThreads	The average number of active threads in the pool
TotalThreads	The average number of threads in the pool
PercentTimeMaxed	The average percent of the time that the number of threads in the pool reached or exceeded the desired maximum number
ThreadCreates	The number of threads created
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

Measurement	Description
<b>ConnectionAllocates</b>	The number of times a connection was allocated
<b>ConnectionWaiters</b>	The average number of threads waiting for a connection
<b>ConnectionWaitTime</b>	The average time, in seconds, of a connection grant
<b>ConnectionTime</b>	The average time, in seconds, that a connection is in use
<b>ConnectionPercentUsed</b>	The average percentage of the pool that is in use
<b>ConnectionPercentMaxed</b>	The percentage of the time that all connections are in use

► **TransactionData**

These are resources that pertain to transactions.

Measurement	Description
<b>NumTransactions</b>	The number of transactions processed
<b>ActiveTransactions</b>	The average number of active transactions
<b>TransactionRT</b>	The average duration of each transaction
<b>BeanObjectCount</b>	The average number of bean object pools involved in a transaction
<b>RolledBack</b>	The number of transactions rolled back
<b>Committed</b>	The number of transactions committed
<b>LocalTransactions</b>	The number of transactions that were local
<b>TransactionMethodCount</b>	The average number of methods invoked as part of each transaction
<b>Timeouts</b>	The number of transactions that timed out due to inactivity timeouts
<b>TransactionSuspended</b>	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
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.

<b>Measurement</b>	<b>Description</b>
<b>SessionsCreated</b>	The number of sessions created on the server
<b>SessionsActive</b>	The number of currently active sessions
<b>SessionsInvalidated</b>	The number of invalidated sessions. May not be valid when using sessions in the database mode
<b>SessionLifetime</b>	Contains statistical data of sessions that have been invalidated. Does not include sessions that are still alive



# 14

---

## Database Server Resource Graphs

After running a scenario, you can use Database Server Resource graphs to analyze the resource usage of your SQL Server and Oracle databases.

This chapter describes:

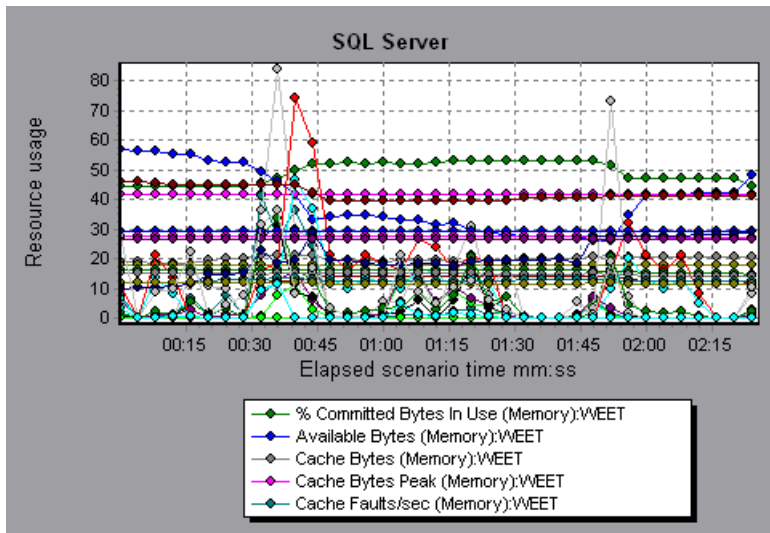
- SQL Server Graph
- Oracle Graph

### About Database Server Resource Graphs

The Database Server Resource graphs show statistics for several database servers. Currently SQL server and Oracle databases are supported. These graphs require that you specify the resources you want to measure *before* running the scenario. For more information, see the section on online monitors in the *LoadRunner Controller User's Guide (Windows)*.

## SQL Server Graph

The SQL Server graph shows the standard Windows resources on the SQL server machine.



The following table describes the default counters that can be monitored on version 6.5 of the SQL Server:

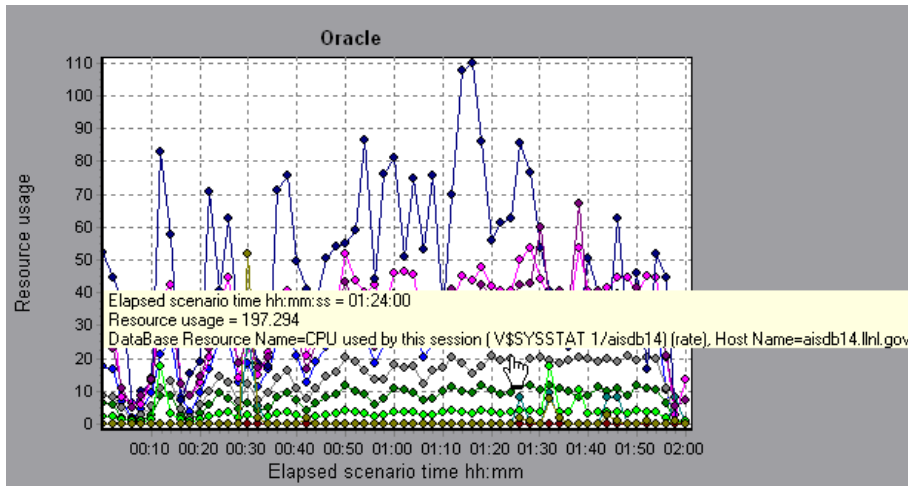
Measurement	Description
<b>% Total Processor Time (NT)</b>	The average percentage of time that all the processors on the system are busy executing non-idle threads. On a multi-processor system, if all processors are always busy, this is 100%, if all processors are 50% busy this is 50% and if 1/4th of 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.
<b>Cache Hit Ratio</b>	The percentage of time that a requested data page was found in the data cache (instead of being read from disk)

Measurement	Description
<b>I/O - Batch Writes/sec</b>	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.
<b>I/O - Lazy Writes/sec</b>	The number of 2K pages flushed to disk per second by the Lazy Writer
<b>I/O - Outstanding Reads</b>	The number of physical reads pending
<b>I/O - Outstanding Writes</b>	The number of physical writes pending
<b>I/O - Page Reads/sec</b>	The number of physical page reads per second
<b>I/O - Transactions/sec</b>	The number of Transact-SQL command batches executed per second
<b>User Connections</b>	The number of open user connections
<b>% Processor Time (Win 2000)</b>	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%.

## Oracle Graph

The Oracle graph displays information from Oracle V\$ tables: Session statistics, V\$SESSTAT, and system statistics, V\$SYSSTAT.

In the following Oracle graph, the V\$SYSSTAT resource values are shown as a function of the elapsed scenario time.



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

Measurement	Description
<b>Logons current</b>	The total number of current logons
<b>Opens of replaced files</b>	The total number of files that needed to be reopened because they were no longer in the process file cache
<b>User calls</b>	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.
<b>SQL*Net roundtrips to/from client</b>	The total number of Net8 messages sent to, and received from, the client
<b>Bytes sent via SQL*Net to client</b>	The total number of bytes sent to the client from the foreground process(es)
<b>Opened cursors current</b>	The total number of current open cursors
<b>DB block changes</b>	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.
<b>Total file opens</b>	The total number of file opens being performed by the instance. Each process needs a number of files (control file, log file, database file) in order to work against the database.





# 15

---

## Streaming Media Graphs

After a scenario run, you can use the Streaming Media graphs to analyze RealPlayer Client, RealPlayer Server, and Windows Media Server performance.

This chapter describes the:

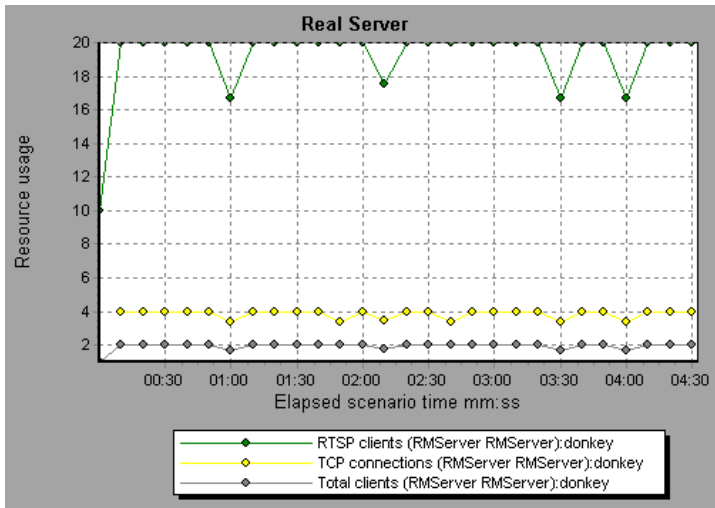
- RealPlayer Client Graph
- RealPlayer Server Graph
- Windows Media Server Graph

### About Streaming Media Graphs

Streaming Media Resource graphs provide you with performance information for the RealPlayer Client, RealPlayer Server, and Windows Media Server machines. Note that in order to obtain data for Streaming Media Resource graphs, you need to install the RealPlayer Client and activate the online monitor for the RealPlayer Server or Windows Media Server before running the scenario. When you set up the online monitor for the RealPlayer Server or Windows Media Server, you indicate which statistics and measurements to monitor. For more information on installing and configuring the Streaming Media Resource monitors, see the *LoadRunner Controller User's Guide (Windows)*.

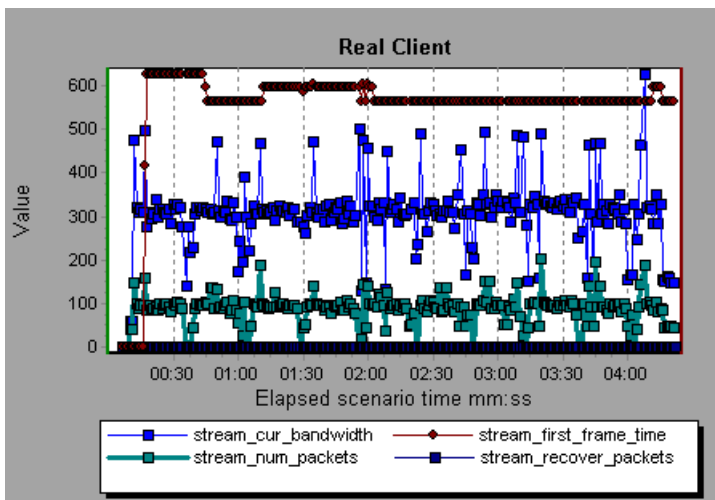
In order to display all the measurements on a single graph, the Analysis may scale them. The Legend tab indicates the scale factor for each resource. To obtain the true value, multiply the scale factor by the displayed value. For

example, in the following graph the actual value of *RTSP Clients* two minutes into the scenario is 200; 20 multiplied by the scale factor of 10.



## RealPlayer Client Graph

The Real Client graph shows statistics on the RealPlayer client machine as a function of the elapsed scenario time.



This graph displays the *Total Number of Packets*, *Number of Recovered Packets*, *Current Bandwidth*, and *First Frame Time* measurements during the first four and a half minutes of the scenario. Note that the scale factor is the same for all of the measurements.

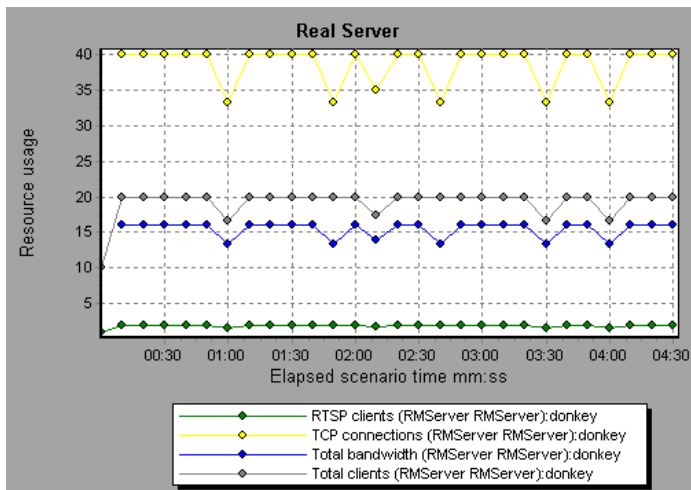
The following default measurements are available for the RealPlayer Client:

Measurement	Description
<b>Current Bandwidth (Kbits/sec)</b>	The number of kilobytes in the last second
<b>Buffering Event Time (sec)</b>	The average time spent on buffering
<b>Network Performance</b>	The ratio (percentage) between the current bandwidth and the actual bandwidth of the clip
<b>Percentage of Recovered Packets</b>	The percentage of error packets that were recovered
<b>Percentage of Lost Packets</b>	The percentage of packets that were lost
<b>Percentage of Late Packets</b>	The percentage of late packets
<b>Time to First Frame Appearance (sec)</b>	The time for first frame appearance (measured from the start of the replay)
<b>Number of Buffering Events</b>	The average number of all buffering events
<b>Number of Buffering Seek Events</b>	The average number of buffering events resulting from a seek operation
<b>Buffering Seek Time</b>	The average time spent on buffering events resulting from a seek operation
<b>Number of Buffering Congestion Events</b>	The average number of buffering events resulting from network congestion
<b>Buffering Congestion Time</b>	The average time spent on buffering events resulting from network congestion

Measurement	Description
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

## RealPlayer Server Graph

The Real Server graph shows RealPlayer server statistics as a function of the elapsed scenario time.



In this graph, the number of *RTSP Clients* remained steady during the first four and a half minutes of the scenario. The *Total Bandwidth* and number of *Total Clients* fluctuated slightly. The number of *TCP Connections* fluctuated more significantly.

Note that the scale factor for the *TCP Connections* and *Total Clients* measurements is 10, and the scale factor for *Total Bandwidth* is 1/1000.

The following default measurements are available for the RealPlayer Server:

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

## Windows Media Server Graph

The Windows Media Server graph shows the Windows Media server statistics as a function of the elapsed scenario time.

The following default measurements are available for the RealPlayer Server:

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

Measurement	Description
<b>Aggregate Send Rate</b>	The total, aggregate rate (bytes/sec) of stream transmission
<b>Connected Clients</b>	The number of clients connected to the server
<b>Connection Rate</b>	The rate at which clients are connecting to the server
<b>Controllers</b>	The number of controllers currently connected to the server
<b>HTTP Streams</b>	The number of HTTP streams being streamed
<b>Late Reads</b>	The number of late read completions per second
<b>Pending Connections</b>	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.
<b>Stations</b>	The number of station objects that currently exist on the server
<b>Streams</b>	The number of stream objects that currently exist on the server
<b>Stream Errors</b>	The cumulative number of errors occurring per second

# 16

---

## Cross Result and Merged Graphs

The Analysis utility lets you compare results and graphs to determine the source of a problem.

This chapter discusses:

- Cross Result Graphs
- Generating Cross Result Graphs
- Merging Graphs
- Creating a Merged Graph

### About Cross Result and Merged Graphs

Comparing results is essential for determining bottlenecks and problems. You use Cross Result graphs to compare the results of multiple scenario runs. You create Merged graphs to compare results from different graphs within the same scenario run.

## Cross Result Graphs

Cross Result graphs are useful for:

- benchmarking hardware
- testing software versions
- determining system capacity

If you want to benchmark two hardware configurations, you run the same scenario with both configurations and compare the transaction response times using a single Cross Result graph.

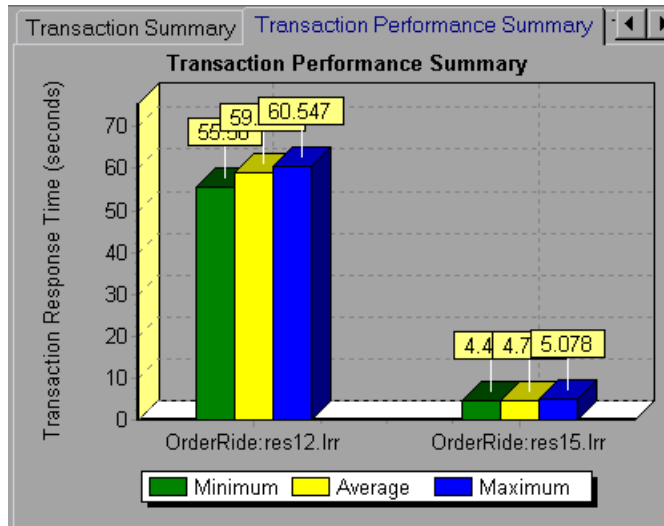
Suppose that your vendor claims that a new software version is optimized to run quicker than a previous version. You can verify this claim by running the same scenario on both versions of the software, and comparing the scenario results.

You can also use Cross Result graphs to determine your system's capacity. You run scenarios using different numbers of Vusers running the same script. By analyzing Cross Result graphs, you can determine the number of users that cause unacceptable response times.

In the following example, two scenario runs are compared by crossing their results, *res12*, and *res15*. The same script was executed twice—first with 100 Vusers and then with 50 Vusers. In the first run, the average transaction time was approximately 59 seconds. In the second run, the average time was



4.7 seconds. It is apparent that the system works much slower with a greater load.



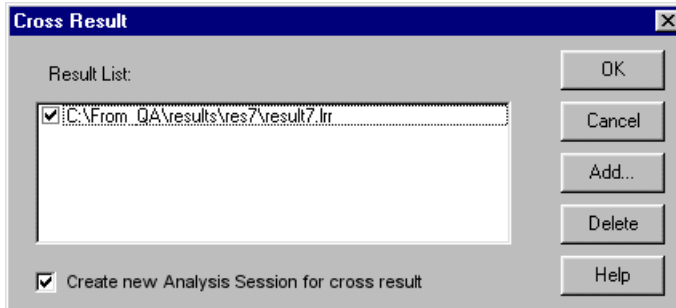
The Cross Result graphs have an additional filter and group by category: *Result Name*. The above graph is filtered to the *OrderRide* transaction for results *res12*, and *res15*, grouped by *Result Name*.

## Generating Cross Result Graphs

You can create a Cross Result graph for two or more result sets.

To generate a Cross Result graph:

- 1 Choose **File > Cross With Result**. The Cross Results dialog box opens.



- 2 Click **Add** to add an additional result set to the **Result List**. The Select Result Files for Cross Results dialog box opens.
- 3 Locate a results directory and select its result file (*.lrr*). Click **OK**. The scenario is added to the Result List.
- 4 Repeat steps 2 and 3 until all the results you want to compare are in the Result List.
- 5 When you generate a Cross Result graph, by default it is saved as a new Analysis session. To save it in an existing session, clear the **Create New Analysis Session for Cross Result** check box.
- 6 Click **OK**. The Analysis processes the result data and asks for a confirmation to open the default graphs.

After you generate a Cross Result graph, you can filter it to display specific scenarios and transactions. You can also manipulate the graph by changing the granularity, zoom, and scale. For more information, see Chapter 2, “Working with LoadRunner Analysis.”

## Merging Graphs

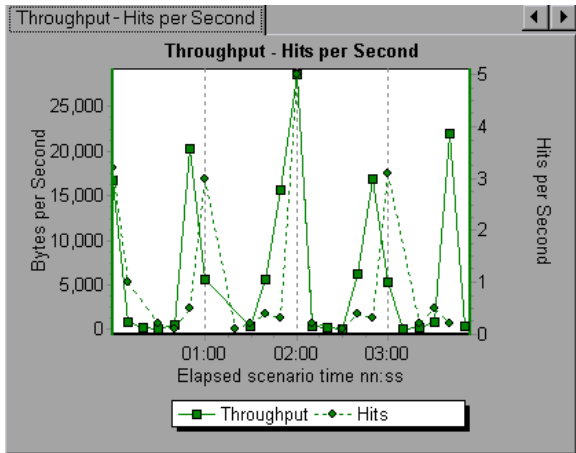
The Analysis lets you merge the results of two graphs from the same scenario into a single graph. The merging allows you to compare several different measurements at once. For example, you can make a merged graph to display the network delay and number of running Vusers, as a function of the elapsed time.

In order to merge graphs, their x-axis must be the same measurement. For example, you can merge Web Throughput and Hits per second, because the common x-axis is Scenario Elapsed Time. The drop-down list only shows the active graphs with an x-axis common with the current graph. The Analysis provides three types of merging:

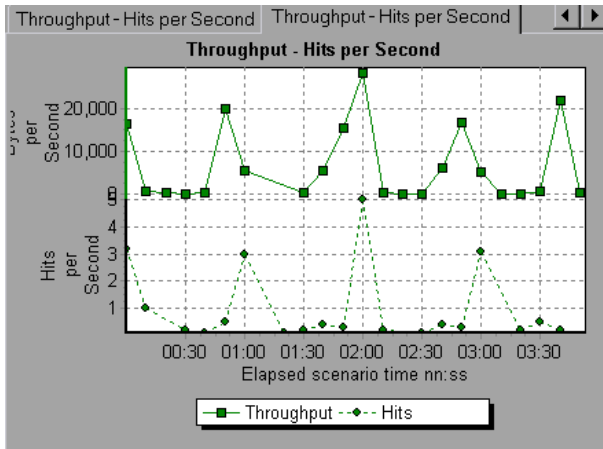
- Overlay
- Tile
- Correlate

**Overlay:** Superimpose the contents of two graphs that share a common x-axis. The left y-axis on the merged graph shows the current graph's values. The right y-axis shows the values of the graph that was merged. There is no limit to the number of graphs that you can overlay. When you overlay two graphs, the y-axis for each graph is displayed separately to the right and left of the graph. When you overlay more than two graphs, the Analysis displays a single y-axis, scaling the different measurements accordingly.

In the following example, the Throughput and Hits per Second graph are overlaid with one another.

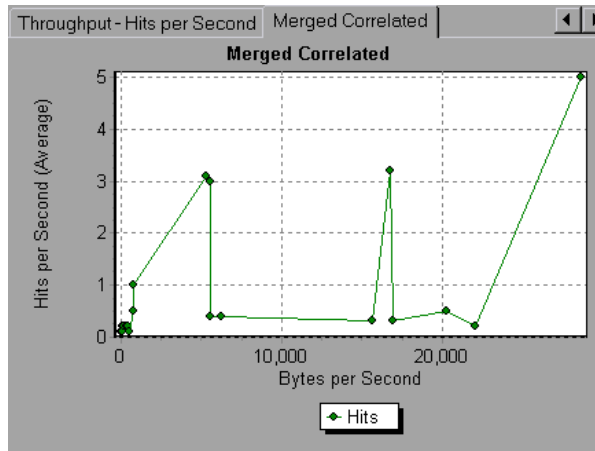


**Tile:** View contents of two graphs that share a common x-axis in a tiled layout, one above the other. In the following example the Throughput and Hits per Second graph are tiled one above the other.



**Correlate:** Plot the y-axis of two graphs against each other. The active graph's y-axis becomes the x-axis of the merged graph. The y-axis of the graph that was merged, becomes the merged graph's y-axis.

In the following example, the Throughput and Hits per Second graph are correlated with one another. The x-axis displays the Bytes per Second (the Throughput measurement) and the y-axis shows the Hits per Second.



## Creating a Merged Graph

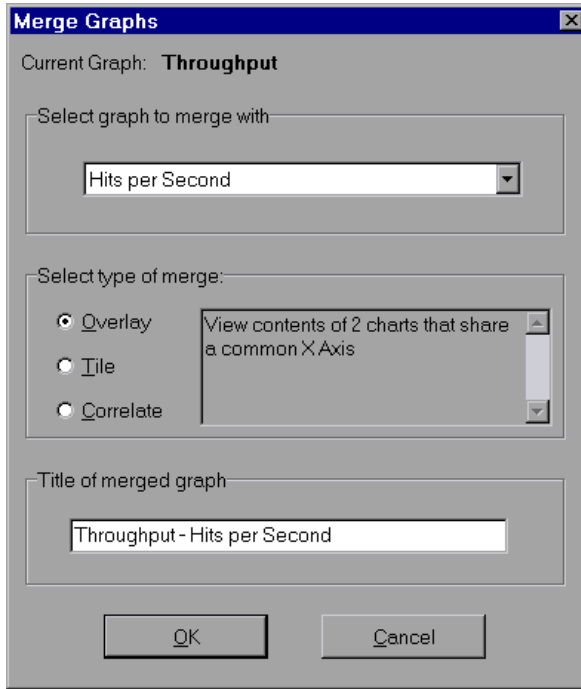
You can merge all graphs with a common x-axis.

**To create a merged graph:**

- 1 Select a graph in the tree view or click on its tab to make it active.



- 2 Choose **View > Merge Graphs** or click the **Merge Graphs** button. The Merge Graphs dialog box opens and displays the name of the active graph.



- 3 Select a graph with which you want to merge your active graph. Only the graphs with a common x-axis to the active graph are available.
- 4 Select the type of merge: **Overlay**, **Tile**, or **Correlate**.
- 5 Specify a title for the merged graph. By default, the Analysis combines the titles of the two graphs being merged.
- 6 Click **OK**.
- 7 Filter the graph just as you would filter any ordinary graph.

# 17

---

## Understanding Analysis Reports

After running a scenario, you can use the Analysis reports to analyze the performance of your application.

This chapter describes:

- Viewing Summary Reports
- Creating HTML Reports
- Working with Transaction Reports
- Data Point Report
- Failed Transaction Report
- Failed Vuser Report
- Data Point Report
- Detailed Transaction Report
- Transaction Performance by Vuser Report

## About Analysis Reports

After running a scenario, you can view reports that summarize your system's performance. The Analysis provides the following reporting tools:

- Summary report
- HTML reports
- Transaction reports

The Summary report provides general information about the scenario run. You can view the Summary report at any time from the Analysis window.

You can instruct the Analysis to create an HTML report. The Analysis creates an HTML report for each one of the open graphs.

Transaction reports provide performance information about the transactions defined within the Vuser scripts. These reports give you a statistical breakdown of your results and allow you to print and export the data.



## Viewing Summary Reports

The Summary report provides general information about scenario execution. This report is always available from the tree view or as a tab in the Analysis window. It lists statistics about the scenario run and provides links to the following graphs: Running Vusers, Throughput, Hits Per Second, Transaction Summary, and Transaction Response Time - Average. At the bottom of the page, the Summary report displays a table containing the scenario's run-time data. Included in this data is a 90 Percent column, indicating the maximum response time for ninety percent of the transactions.

The screenshot shows the 'Analysis Summary' report for a scenario named 'Sanity\_Analysis\Scenario1Scenario1.js'. The session results are located at 'F:\Results\Sessions\v\Jres\_amazon\res\_amazon1.js', and the duration is 20 minutes and 49 seconds. The statistics summary includes: Maximum Running Vusers (3), Total Throughput (bytes) (50,222,795), Throughput (bytes/second) (Average: 40,210), Total Hits (12,217), and Hits per Second (Average: 10). The transaction summary table shows two transactions: 't\_amazon\_account' and 't\_amazon\_book'. The table includes columns for Transaction Name, Minimum, Average, Maximum, 90 Percent, Pass, Fail, and Abort. Summary statistics at the top of the table indicate: Total Passed: 73, Total failed: 73, and Total aborted: 3.

Transaction Name	Minimum	Average	Maximum	90 Percent	Pass	Fail	Abort
t_amazon_account	2.143	2.56	3.245	2.68	12	0	0
t_amazon_book	7.871	8.825	10.425	10.31	10	2	0

## Creating HTML Reports

The Analysis lets you create HTML reports for your scenario run. It creates a separate report for each one of the open graphs and a Summary report. The Summary report is identical to the Summary report that you access from the Analysis window. The report also provides a link to an Excel file containing the graph data.

The screenshot shows a web browser displaying an HTML report. The report is titled "Analysis Summary" and is for the period "07/00/2001 10:05:0". It provides a summary of the scenario run, including the scenario name, results in session, and duration. The report is divided into sections: "Statistics Summary" and "Transaction Summary".


**Statistics Summary**

- Maximum Running Users: 3
- Total Throughput (bytes): 50,222,495
- Throughput (bytes/second): Average: 40,210
- Total Hits: 12,217
- Hits per Second: Average: 10

**Transaction Summary**

Transaction Name	Minimum	Average	Maximum	90 Percent	Pass	Fail
transaction_name	2.143	2.56	3.245	2.66	12	0

To create HTML reports:

- 1 Open all graphs that you want to be included in the report.
- 2  Choose **Reports > HTML Report** or click the **Create HTML Report** button. The Select Report Filename and Path dialog box opens.

- 3 Specify a path and file name for the HTML report and click **OK**. The Analysis saves a Summary report called by the file name in the selected folder, and the rest of the graphs in a folder with the same name as the file name. When you create an HTML report, the Analysis opens your default browser and displays the Summary report.
- 4 To view an HTML report for one of the graphs, click on its link in the left frame.
- 5 To copy the HTML reports to another location, be sure to copy the filename and the folder with the same name. For example, if you named your HTML report *test1*, copy *test1.html* and the folder *test1* to the desired location.

## Working with Transaction Reports

LoadRunner's Transaction reports are divided into the following categories:

- Activity
- Performance

**Activity** reports provide information about the number of Vusers and the number of transactions executed during the scenario run. The available Activity reports are *Scenario Execution*, *Failed Transaction*, and *Failed Vusers*.

**Performance** reports analyze Vuser performance and transaction times. The available Performance reports are *Data Point*, *Detailed Transaction*, and *Transaction Performance by Vuser*.

In order to view a report, you must generate it from the Analysis window. LoadRunner reports are displayed in a Report Viewer. You can print, save, or export the data using the viewer.

### Selecting and Displaying Reports

The Analysis provides several built-in reports which contain detailed summaries about the scenario, the transactions and Vusers.

**To display a report:**

- 1 Open the desired Analysis session file (.lra extension), or LoadRunner result file (.lrr extension), if it is not already open.

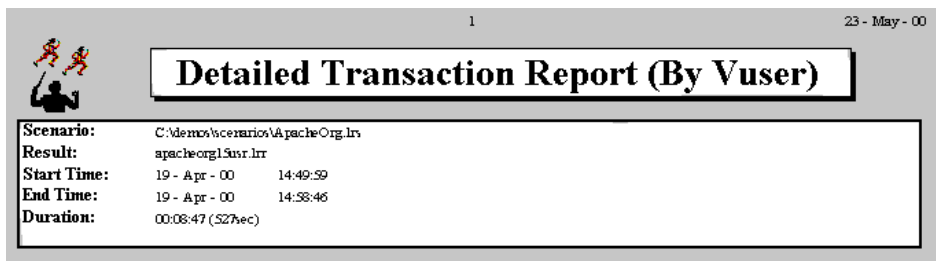
- 2 From the Reports menu choose a report. The report is generated and displayed. You can display multiple copies of the same report.

## The Report Viewer

Each report is displayed in its own report viewer. Each viewer contains a header and a toolbar.

### Report Header

The header displays general run-time information.



The report header contains the following information:

**Title:** The name of the report.

**Scenario:** The name of the scenario described in the report.

**Result:** The pathname of the scenario results directory.

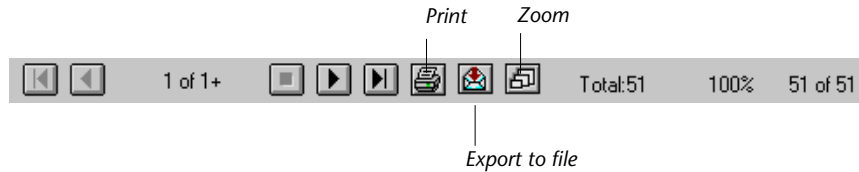
**Start time:** The time at which the Run Scenario command was executed.

**End time:** The time at which the scenario script was terminated.

**Duration:** The total run time of the scenario.

## Report Viewer Toolbar

Each report viewer has a toolbar that lets you perform operations on displayed reports.



The report viewer toolbar contains the following buttons:



**Zoom** : Toggles between an actual size, full page, and magnified views of the report.



**Print** : Prints the displayed report.



**Export to file** : Exports the displayed information to a text file.

If there are multiple values for the y-axis, as in the Transaction Performance by Vuser graph (min, average, and max), all of the plotted values are displayed.

## Scenario Execution Report

The Scenario Execution report is an Activity report that provides details about major events that occurred during the scenario run. This includes information on every Vuser, such as when it was ready to run and for how long it ran.

<b>Group:</b> g1						
<i>Vuser</i>	<i>Host</i>	<i>Ready At</i>	<i>Running At</i>	<i>Duration</i>	<i>Termination Status</i>	
Vuser1	10.1.1.30	14:52:58	14:53:30	00:05:43 (343sec)	Abort	
Vuser2	10.1.1.30	14:52:58	14:53:30	00:05:43 (343sec)	Abort	
Vuser3	10.1.1.30	14:52:58	14:53:30	00:05:43 (343sec)	Abort	
Vuser4	10.1.1.30	14:52:58	14:53:30	00:05:43 (343sec)	Abort	
Vuser5	10.1.1.30	14:52:58	14:53:30	00:05:43 (343sec)	Abort	

### Summary

*Vusers:* 5  
*Passed:* 0      *Failed:* 0      *Error:* 0      *Stopped:* 5

## Failed Transaction Report

The Failed Transaction report is an Activity report that provides details about the beginning time, end time, and duration of the failed, but completed transaction.

**Group: g1**

Vuser: Vuser1

<i>Transaction</i>	<i>Start time</i>	<i>End time</i>	<i>Duration</i>
Apache_Server	14:58:36.117	14:58:37.319	00:00:01.202
Surf_Apache	14:58:33.043	14:58:37.319	00:00:04.276

Vuser: Vuser2

<i>Transaction</i>	<i>Start time</i>	<i>End time</i>	<i>Duration</i>
Apache_Server	14:58:35.265	14:58:37.328	00:00:02.063
Surf_Apache	14:58:24.810	14:58:37.328	00:00:12.518

## Failed Vuser Report

The Failed Vuser report is an Activity report that provides details about all Vusers that were in the ERROR, ABORTED, or DONE:FAILED states during the scenario execution. The *Ready At* and *Running At* times are relative to the computer's system clock.

<b>Group:</b> g2						
<i>Vuser</i>	<i>Host</i>	<i>Ready At</i>	<i>Running At</i>	<i>Duration</i>	<i>Termination Status</i>	
Vuser1	localhost	14:53:26	14:54:42	00:05:14 (314sec)	Abort	
Vuser2	localhost	14:53:26	14:54:42	00:05:14 (314sec)	Abort	
Vuser3	localhost	14:53:26	14:54:42	00:05:14 (314sec)	Abort	
Vuser4	localhost	14:53:26	14:54:42	00:05:14 (314sec)	Abort	
Vuser5	localhost	14:53:26	14:54:42	00:05:14 (314sec)	Abort	

**Summary**

<i>Vusers:</i>	5			
<i>Failed:</i>	0	<i>Error:</i>	0	<i>Stopped:</i> 5

In this scenario, all five Vusers were aborted.

## Data Point Report

LoadRunner enables you to record your own data for analysis. You instruct LoadRunner to record the value of an external function or variable, also known as a *data point*, during the scenario run. Using the gathered data, LoadRunner creates a graph and report for the data point.

The data point is set by including an **lr\_user\_data\_point** function (**user\_data\_point** for GUI Vusers) in your Vuser script. For more information, refer to the *LoadRunner Online Function Reference*.

The **Data Point** graph shows the value of the data point during the scenario run. The *x-axis* represents the number of seconds elapsed since the start time of the run. The *y-axis* displays the value of each recorded data point statement.

The **Data Point** report is a Performance report that lists the name of the data point, its value, and the time its value was recorded. The values are displayed for each Group and Vuser.

<b>Group:</b> Group1		<b>Data Point</b>	<b>Time</b>
<b>Vuser id:</b>	1		
	memory	19.00	13:37:16
	memory	1.00	13:37:20
	memory	9.00	13:37:32
	memory	1.00	13:37:36
	memory	1.00	13:37:40
<b>Vuser id:</b>	2		
	memory	6.00	13:37:05
	memory	8.00	13:37:20
	memory	9.00	13:37:32
	memory	1.00	13:37:36
	memory	1.00	13:37:40



## Detailed Transaction Report

The Detailed Transaction (by Vuser) report is a Performance report that provides a list of all transactions executed by each Vuser during a scenario. The report provides details about the execution time of each transaction per Vuser.

**Group:** zorb

Vuser\_id: 1

<i>Transaction</i>	<i>Start time</i>	<i>End time</i>	<i>Duration</i>	<i>Think Time</i>	<i>Wasted Time</i>	<i>Result</i>
end_section	14:07:59.029	14:08:00.045	00:00:01.016	00:00:01.016	00:00:00.000	Pass
General	14:05:27.748	14:06:06.482	00:00:38.618	00:00:21.715	00:00:00.116	Pass
General	14:06:06.482	14:07:02.701	00:00:37.104	00:00:21.716	00:00:19.115	Pass
General	14:07:02.701	14:07:59.029	00:00:37.212	00:00:21.715	00:00:19.116	Pass
mc_run	14:05:37.310	14:05:41.764	00:00:04.453	00:00:02.609	00:00:00.001	Pass
mc_run	14:06:16.045	14:06:20.248	00:00:04.202	00:00:02.609	00:00:00.001	Pass
mc_run	14:07:12.264	14:07:15.920	00:00:03.655	00:00:02.609	00:00:00.001	Pass

The following values are reported:

**Start time:** the system time at the beginning of the transaction

**End time:** the actual system time at the end of the transaction, including the think time and wasted time.

**Duration:** the duration of the transaction in the following format: hrs:minutes:seconds:milliseconds. This value includes think time, but does not include wasted time.

**Think time:** the Vuser's think time delay during the transaction.

**Wasted time:** the LoadRunner internal processing time not attributed to the transaction time or think time. (primarily RTE Vusers)

**Results:** the final transaction status, either Pass or Fail.

## Transaction Performance by Vuser Report

The Transaction Performance Summary by Vuser report is a Performance report that displays the time required by each Vuser to perform transactions during the scenario. The report indicates if the transaction was successful and what the minimum, maximum, and average times were for each Vuser. This report is useful when you have several different types of Vusers in a scenario and you want to characterize performance for each type.

**Transaction:** Apache\_Home

**Group:** g1

			<u>P e r f o r m a n c e (sec)</u>				
<i>Vuser</i>	<i>Pass</i>	<i>Fail</i>	<i>Min</i>	<i>Avg</i>	<i>Max</i>	<i>STD</i>	
Vuser1	40	0	0.22	1.06	3.80	0.99	
Vuser2	37	0	0.22	1.37	7.40	1.60	
Vuser3	40	1	0.22	1.08	5.67	1.17	
Vuser4	39	0	0.23	1.04	3.91	0.97	
Vuser5	39	1	0.22	0.95	2.92	0.75	
<b>Total:</b>	<b>5</b>	<b>195</b>	<b>2</b>	<b>0.22</b>	<b>1.10</b>	<b>7.40</b>	

# 18

---

## Managing Results Using TestDirector

The integration of the Analysis with TestDirector lets you manage results and sessions using TestDirector projects. TestDirector helps you organize and manage all scripts, scenarios, and results using a repository.

This chapter describes:

- ▶ Opening a Connection to a TestDirector Project
- ▶ Creating a New Session Using TestDirector
- ▶ Opening an Existing Session Using TestDirector
- ▶ Saving Sessions to a TestDirector Project

For detailed information on using TestDirector to manage your sessions, refer to the *TestDirector User's Guide*.

### About Managing Results Using TestDirector

The Analysis works together with TestDirector, Mercury Interactive's test management tool. You can store both sessions and results in a TestDirector repository and organize them into unique groups. You can also retrieve results from the TestDirector repository, and save the Analysis sessions with your specific graph settings, locally.

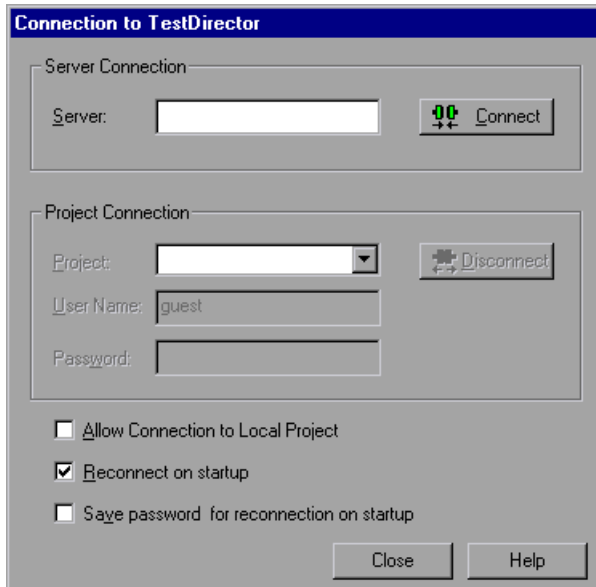
You can instruct the Analysis to automatically connect to the TestDirector server upon startup.

## Opening a Connection to a TestDirector Project

Before you can work with a TestDirector project, you must open a connection to the server hosting the project. After connecting to the TestDirector server, you still have complete access to the regular file system.

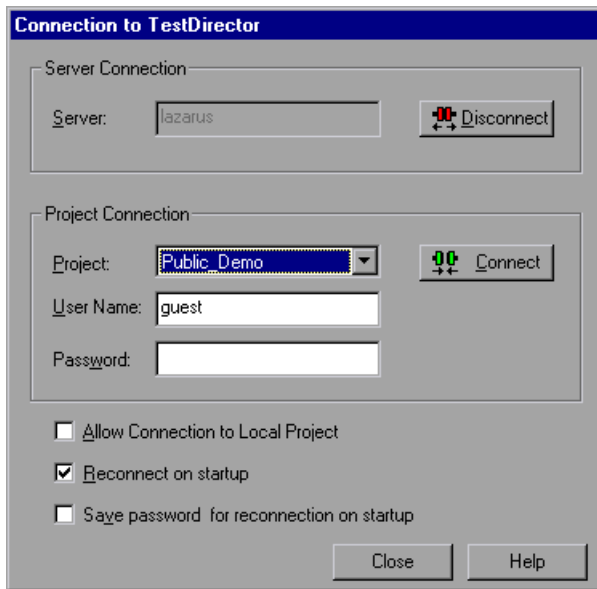
**To open a connection to a TestDirector project:**

- 1 Choose **Tools > TestDirector Connection** from the Analysis window.



- 2 In the Server box, type in the name of the TestDirector server and then click **Connect**. A message appears indicating that the Analysis is connecting to the server.

- 3 In the Project Connection box, select one of the available projects.



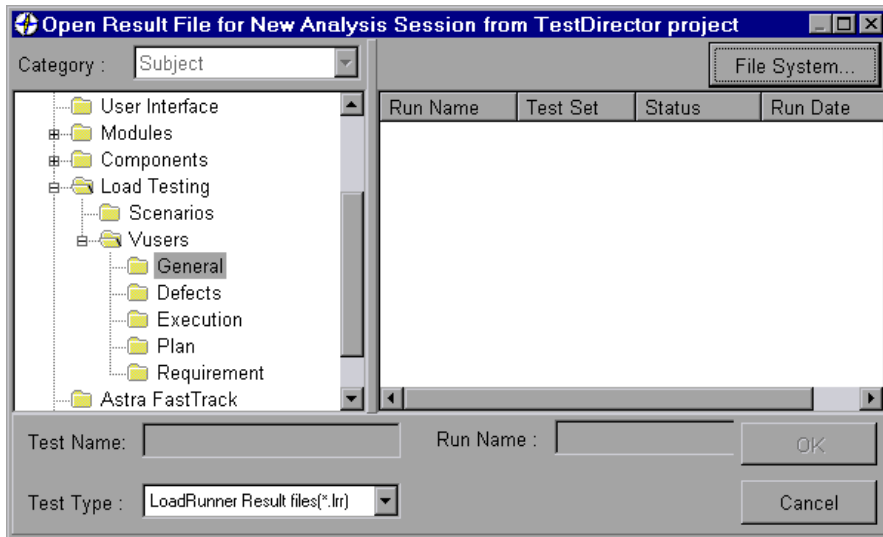
- 4 Provide the user login and password.
- 5 Select the **Reconnect on Startup** check box to automatically open the connection to the TestDirector server and project when you start the Analysis.
- 6 If you select Reconnect on Startup, you can save the specified password to automate the login process. Select the **Save Password for Reconnection on Startup** check box.
- 7 Click **Connect**. A message appears indicating that you are connecting to a project.
- 8 Click **Close** to close the Connection to TestDirector dialog box.

## Creating a New Session Using TestDirector

You can create a new Analysis session using a result file (.lrr extension) located in the TestDirector repository or file system.

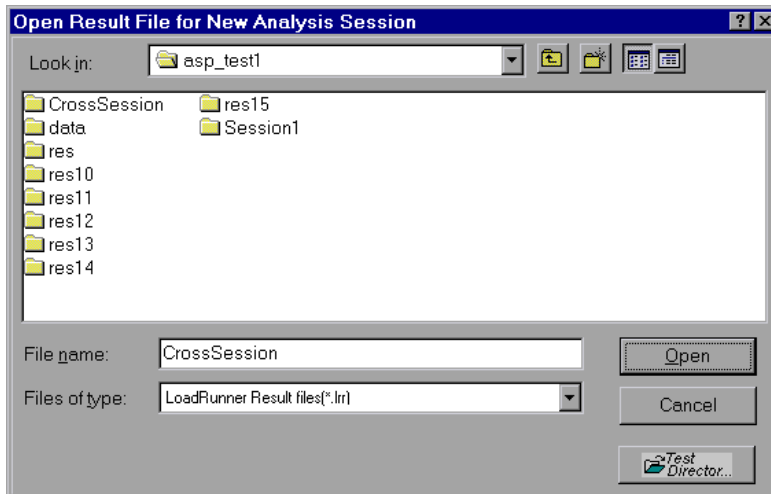
To create a new session using results from the TestDirector repository:

- 1 Connect to the TestDirector server (see “Opening a Connection to a TestDirector Project,” on page 176.)
- 2 Choose **File > New** or click the **Create New Analysis Session** button. The Open Result File for New Analysis Session from TestDirector Project dialog box opens.



- 3 Select a result file from the project.
- 4 Click **OK**. The Analysis loads the results.

- 5 To select a session file from the standard file system, click the **File System** button in the upper right corner of the dialog box. The Open Result File for New Session dialog box opens.



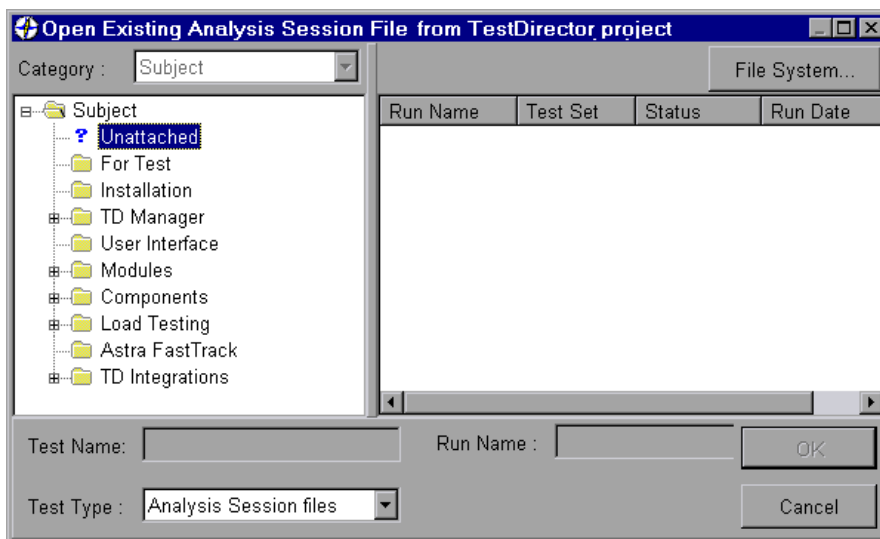
To return to the TestDirector dialog box, click the **TestDirector** button in the bottom right of the dialog box.

## Opening an Existing Session Using TestDirector

You can open an existing Analysis session through the standard file system or the TestDirector repository.

To open a session from the TestDirector repository:

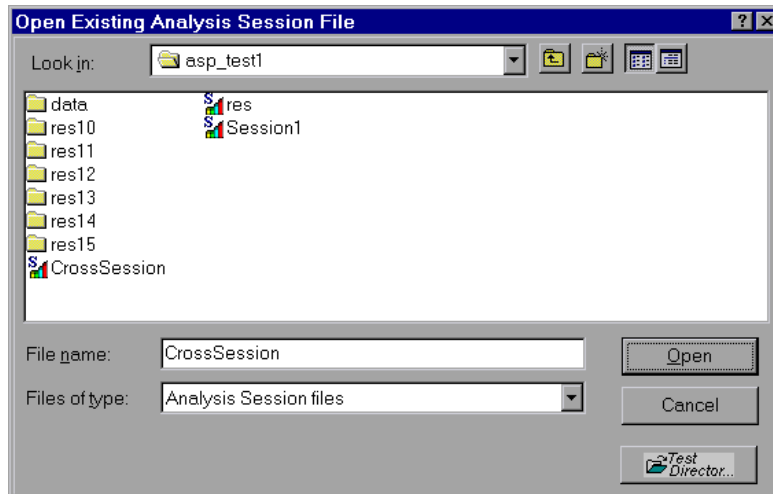
- 1 Open a connection to the TestDirector server (see “Opening a Connection to a TestDirector Project,” on page 176).
- 2 Choose **File > Open** or click the **Open Existing Analysis Session** button. The Open Existing Analysis Session File from TestDirector Project dialog box opens.



- 3 Select a session from the project.
- 4 Click **OK**. The Analysis loads the scenario. The name of the session appears in the banner. The session's graphs are displayed in the Tree view.



- 5 To select a session file from the standard file system, click the **File System** button in the upper right corner of the dialog box. The Open Existing Analysis Session File dialog box opens.



To return to the TestDirector dialog box, click the **TestDirector** button in the bottom right of the dialog box.

## Saving Sessions to a TestDirector Project

You can save a new analysis session, or one that previously existed as a local file, into a TestDirector project.

### To save a session to the TestDirector project:

- 1 Connect to the TestDirector server (see “Opening a Connection to a TestDirector Project,” on page 176). Note that you must first close your current session.
- 2 Select **File > Save** and save the session in the TestDirector data directory.



# 19

---

## Interpreting Analysis Graphs

LoadRunner Analysis graphs present important information about the performance of your scenario. Using these graphs, you can identify and pinpoint bottlenecks in your application and determine what changes are needed to improve its performance.

This chapter presents examples of:

- ▶ Analyzing Performance Under Load
- ▶ Breaking Down Transaction Performance Response Time
- ▶ Using the Web Page Breakdown Graphs
- ▶ Identifying Network Problems
- ▶ Identifying Web Server-Related Problems
- ▶ Identifying Database and Application Server Problems
- ▶ Comparing Scenario Results

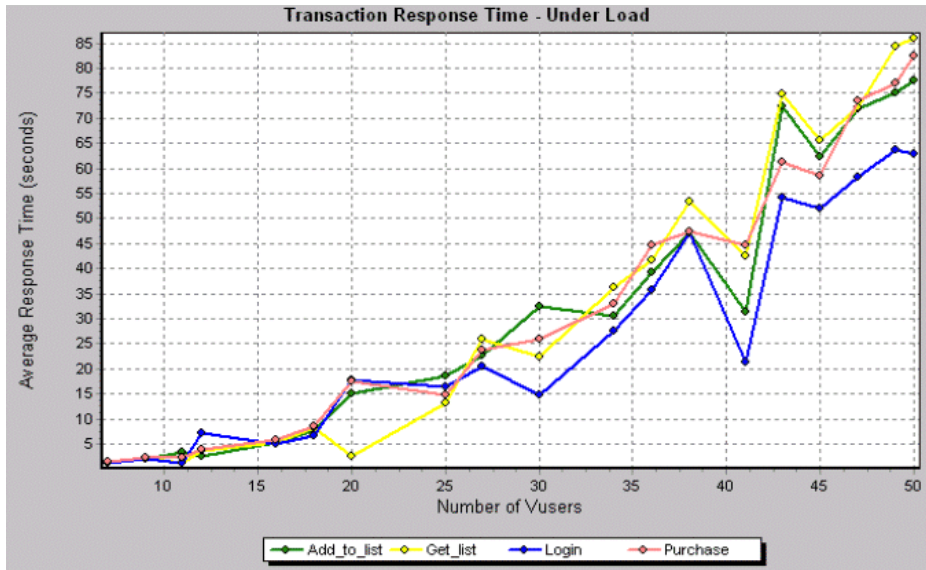
### Analyzing Performance Under Load

The **Transaction Response Time - Under Load** graph should be your starting point in analyzing a scenario run, since it correlates between the average transaction response time and the number of running Vusers.

**Question 1:** What happened to response time as the number of Vusers increased? Did your application scale to handle the maximum number of concurrent Vusers you specified?

**Answer:** The Transaction Response Time - Under Load graph demonstrates what happened to response time as the number of users increased during

the scenario. In the example below, the application could not sustain more than 50 concurrent Vusers. Even with fewer than 50 Vusers, the average transaction response time was higher than the maximum acceptable transaction time defined by the user.



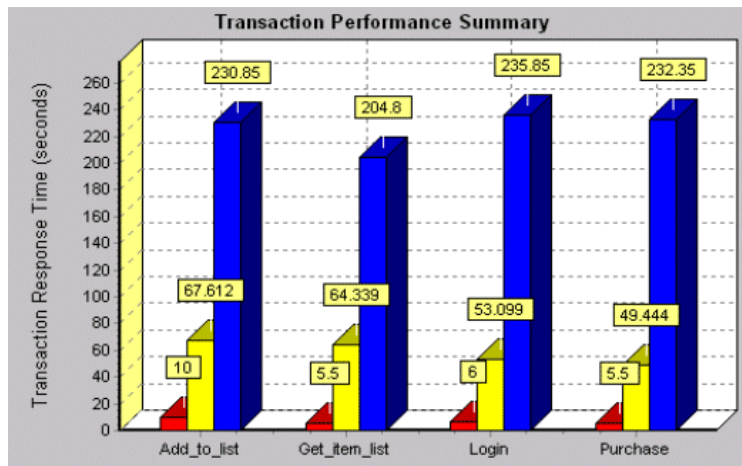
In order to pinpoint the problem and understand why transaction response time was high in this scenario, it is necessary to drill down to—and analyze the performance of—the different components of your application.

## Breaking Down Transaction Performance Response Time

You can drill down on the **Transaction Performance Summary** graph in order to view response times for the sub-transactions that comprise each transaction. Note that this is possible only if you pre-defined the sub-transactions you want to examine in your script.

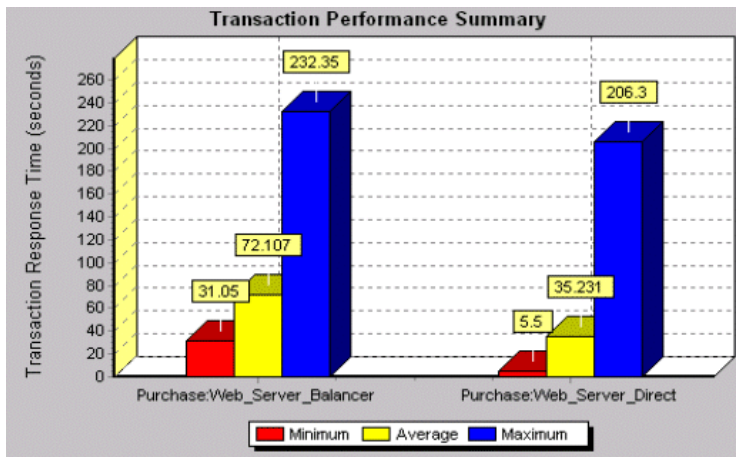
**Question 2:** Which sub-transactions were responsible for the slow transaction response time?

**Answer:** The Transaction Performance Summary graph displays the response times for each transaction in the scenario.



## LoadRunner Analysis User's Guide

You can drill down on each transaction to view response times for each sub-transaction. For example, if you drill down on the Purchase transaction in the above graph, the Analysis displays its two sub-transactions.

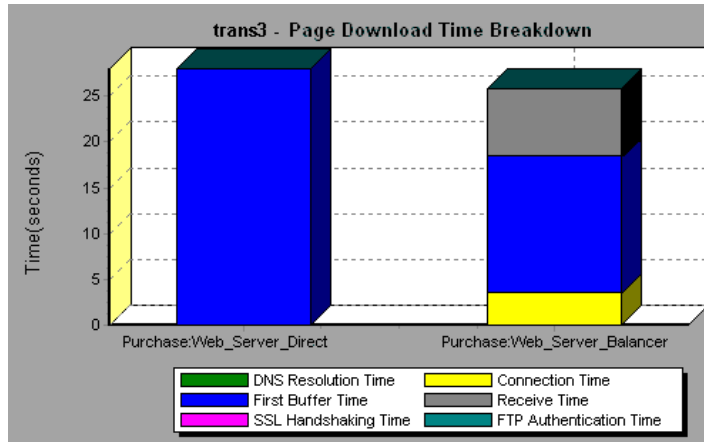


In the above example, the Web\_Server\_Balancer sub-transaction was performed using a load balancer, whereas the Web\_Server\_Direct sub-transaction was performed without using a load balancer. The graph demonstrates that the same transaction took twice as long when it was run through the load balancer than when it was run directly through the Web server.

This indicates that there was a problem with the load balancer that caused a performance bottleneck. The problem should be fixed before you re-run your load test.

## Using the Web Page Breakdown Graphs

Using the **Web Page Breakdown graphs**, you can drill down further to view the download time for each page component in a transaction. When you locate a problematic component, you can view a breakdown of the component's download time in the Page Download Time Breakdown graph.



The Page Download Time Breakdown graph indicates the DNS resolution time, connection time, time to first buffer, SSL handshaking time, receive time, and FTP authentication time. Using these measurements, you can identify whether a problem is network- or server-related. For more information regarding these measurements, see “Page Download Time Breakdown Graph” on page 73.

The above graph demonstrates that the Receive time was unusually high, indicating that network problems occurred during the scenario run. To determine what network problems caused the performance bottleneck, it is necessary to examine the Network Monitor graphs.

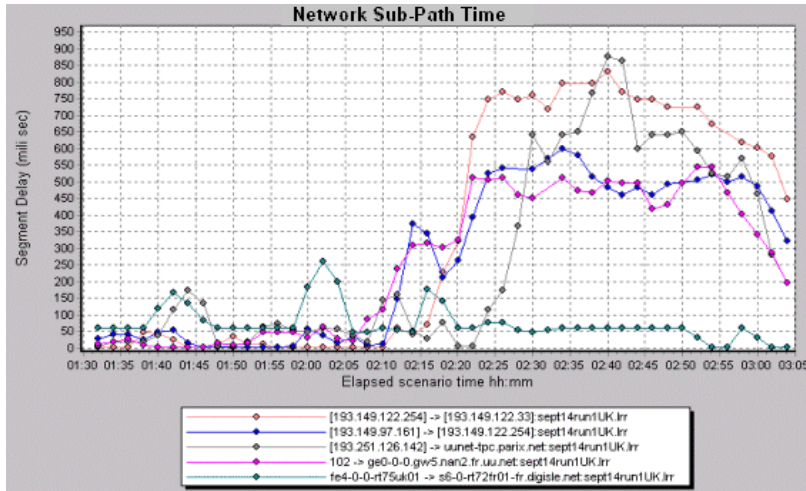
## Identifying Network Problems

You use the **Network Sub-Path Time** graph to identify where network bottlenecks occurred. Network bottlenecks can usually be identified when the load increment is considerable and is not significantly affecting any of the server side components, as in the case of informational sites using many static Web pages.

**Question 3:** Which network segments were responsible for the network bottleneck?

**Answer:** The Network monitor records a transaction's route along the Internet, enabling you to determine which network segments caused the bottleneck. The Network monitor is unlike standard trace route programs because it uses unique technology that enables it to cross firewalls.

In the following example, the Vuser load created an average delay of 600 milliseconds for each network segment. As a result, the network time for each transaction exceeded three seconds—a considerably long period of time.





Using the above graph, you can pinpoint the network segments that were causing the bottleneck. Once you have fixed the problematic segments, it is recommended that you re-run your scenario to verify that the network problems were in fact resolved.

## Identifying Web Server-Related Problems

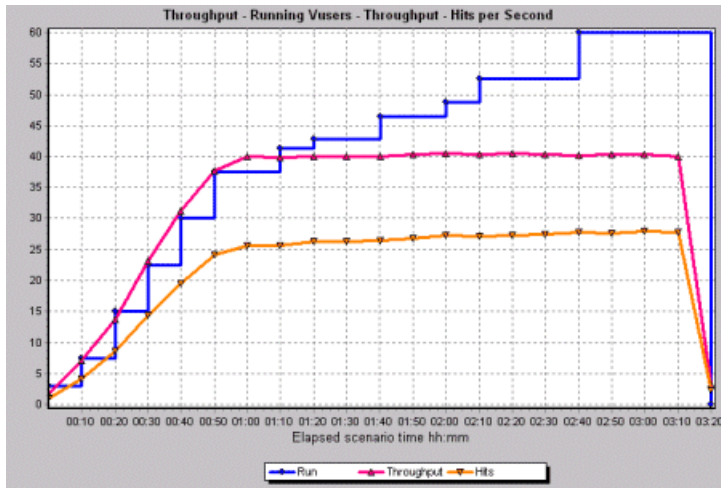
You can identify whether a server-related problem exists by analyzing the Page Download Time Breakdown graph, or by viewing the **Hits per Second** and **Throughput** graphs.

**Question 4:** Did the number of hits per second increase as the number of Vusers increased? Did the Web server throughput increase as the Vuser load increased?

**Answer:** The Hits per Second graph demonstrates how many hits per second were generated on the application's Web server during each second of the scenario run. The Throughput graph displays the amount of data that the Vusers received from the Web server during scenario execution. As Vusers are added during the load test, the number of hits per second and throughput should increase until the system's maximum threshold is reached.

In the merged Running Vusers, Throughput, and Hits per Second graph below, however, it is apparent that as the number of Vusers increased, neither the throughput nor the number of hits per second increased. This indicates that there was a bottleneck in the system architecture which

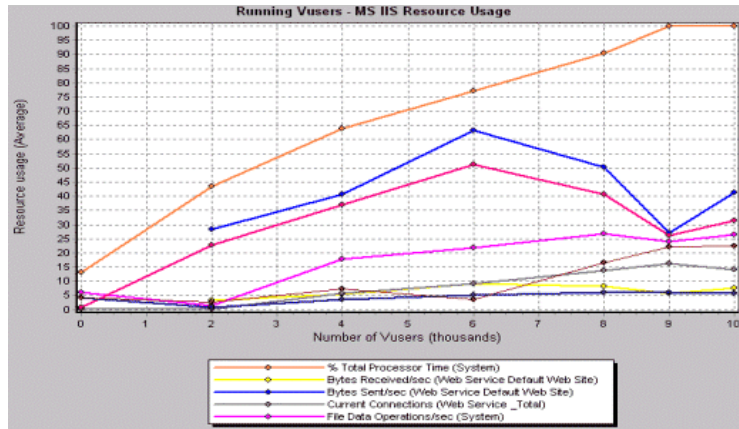
prevented the Web server from responding as expected to the gradual increase in requests.



The bottleneck in the system architecture could be caused by network problems, Web server inadequacies, faulty database server design, or application server problems. In the above case, the network problems were resolved before this scenario run, so it is clear that the performance bottleneck was caused by a problem with the Web server, application server, or database server.

Using the appropriate Web server graph, you can identify the problematic Web server measurements and isolate the cause of server degradation. For example, the graph below which correlates between the number of running Vusers and the resource usage on the Microsoft IIS server, demonstrates that the Web server's processing time (CPU) was the cause of slow server performance. As the number of Vusers increased, the percentage of CPU

resource usage increased, while the number of requests the server could handle—the Bytes Received measurement—remained extremely low.



In twenty percent of load tests, Web servers are found to be the cause of performance bottlenecks. The bottlenecks are usually the result of poor server configuration and insufficient resources. For example, poorly written code and DLLs can use nearly all of a computer's processor time (CPU) and create a bottleneck on the Web server. Likewise, physical memory constraints and mismanagement of server memory can easily cause a Web server bottleneck. It is therefore advisable to check both the CPU and physical memory of your Web server before exploring other possible causes of poor Web server performance.

For information about other useful Web server measurements, see the *LoadRunner Controller User's Guide*.

### HTTPS Problems

Excessive use of HTTPS and other security measures can quickly exhaust server resources and contribute to system bottlenecks. For example, when HTTPS is implemented on the Web server during a load test, system resources are quickly exhausted by a relatively small load. This is caused by secured socket layer (SSL) resource-intensive operations.

Continuously open connections can also drain server resources. Unlike browsers, servers providing SSL services typically create numerous sessions with large numbers of clients. Caching the session identifiers from each

transaction can quickly exhaust the server's resources. In addition, the "keep-alive" enhancement features of most Web browsers keep connections open until they are explicitly terminated by the client or server. As a result, server resources may be wasted as large numbers of idle browsers remain connected to the server.

The performance of secured Web sites can be improved by:

- ▶ Fine-tuning the SSL and HTTPS services according to the type of application
- ▶ Using SSL hardware accelerators, such as SSL accelerator appliances and cards
- ▶ Changing the level of security according to the level of sensitivity of the data (i.e., changing the key length used for public-key encryption from 1,024 to 512 bits)
- ▶ Avoiding the excessive use of SSL and redesigning those pages that have low levels of data sensitivity to use regular HTTPS

## Identifying Database and Application Server Problems

**Question 5:** If bottlenecks persist after the Web server and network problems have been resolved, what other system components should be examined?

**Answer:** If bottlenecks are not being caused by Web server or network problems, they may be related to database server problems. Dynamic Web sites that rely heavily on database operations are particularly at risk for performance problems. As a result, 25 percent of Web site performance problems can be traced to malfunctioning database operations.

The most common database problems are inefficient index design, fragmented databases, out-of-date statistics, and faulty application design. Database system performance can therefore be improved by using smaller result sets, automatically updating data, optimizing indexes, frequently compacting data, implementing a query or a lock on time-outs, using shorter transactions, and avoiding application deadlocks.

To analyze whether a problem is database server-related, use the database server—SQL or Oracle—monitors and graphs.

For information about useful database server measurements, see the *LoadRunner Controller User's Guide*.

### Application Server Problems

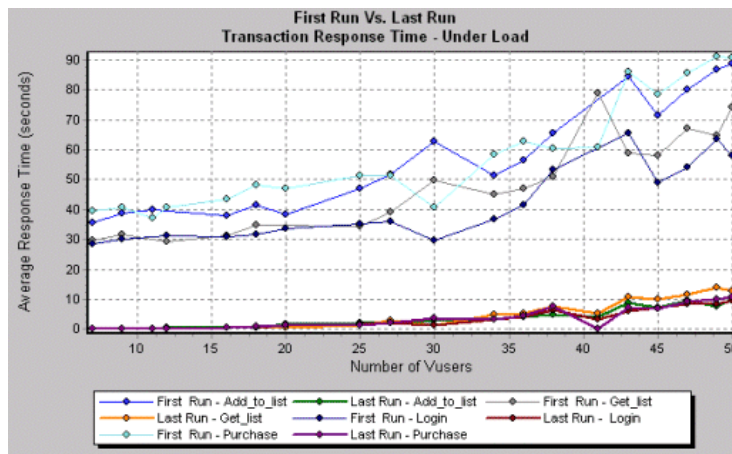
If bottlenecks are not being caused by database server problems, they may be related to application server problems. To analyze whether a problem is application server-related, use the appropriate application server—Ariba, ATG Dynamo, BroadVision, ColdFusion, Fujitsu INTERSTAGE, Microsoft ASP, SilverStream, WebLogic, or WebSphere—monitor and graph.

For information about application server measurements, see Chapter 13, “Web Application Server Resource Graphs.”

## Comparing Scenario Results

Each time the system is fine tuned and another performance bottleneck is resolved, the same load test should be run again in order to verify that the problem has been fixed and that no new performance bottlenecks have been created. After performing the load test several times, you can compare the initial results to the final results.

The following graph displays a comparison of the initial load test and final load test for the scenario described in the previous sections.



The first load test demonstrates the application's performance in its initial state, before any load testing was performed. From the graph, you can see that with approximately 50 Vusers, response time was almost 90 seconds, indicating that the application suffered from severe performance problems.

Using the analysis process described in the previous sections, it was possible to determine that there was a problem with the load balancer, the network, the Web server, the database server, and the application server. In this particular case, the results led the customer to look at the application code and site architecture, and decide to add a middleware server between the application server and the database to cache requests and take some of the load off the database.

As a result of these site architectural changes, the transaction response time for the same business process, with the same number of users, was less than 10 seconds in the last load test performed. Using the Analysis, therefore, the customer was able to increase site performance tenfold.

---

# Index

## A

Acrobat Reader vii  
Activity reports 167  
advanced display settings  
    chart 27  
    series 28  
Analysis  
    overview 1–7  
    sessions 2  
    working with 9–35  
Apache graph 108  
application server graphs 193  
Ariba graph 116  
ATG Dynamo graph 119

## B

Books Online vii  
breaking down transactions 68  
BroadVision graph 122

## C

chart settings 27  
Check Point FireWall-1 graph 106  
ColdFusion graph 129  
collating results 3  
comparing scenario runs 193  
connecting to a TestDirector server 176  
Connection time 74  
Context Sensitive Help viii  
coordinates of a point 10  
Cross Result graphs 155–162

## D

Data Point  
    report 172  
Data Points (Average) graph 87  
Data Points (Sum) graph 86  
Database options 30  
database server graphs 192  
Detailed Transaction report 173  
display options  
    advanced 26  
    standard 25  
DNS Resolution time 74  
documentation set viii  
Downloaded Component Size graph 83  
drill down 10

## E

enlarging graphs 13  
Error graphs 41–43  
Error Statistics graph 42  
Errors per Second graph 43  
Excel file  
    exporting to 23  
    viewing 166

## F

Failed Transaction report 170  
Failed Vuser report 171  
filtering graphs 16  
Firewall Server Monitor graphs 105–106  
First Buffer time 74  
FTP Authentication time 75  
Fujitsu INTERSTAGE graph 129

### G

General options 29

granularity 13

graph

  Apache 108

  Ariba 116

  ATG Dynamo 119

  BroadVision 122

  Check Point FireWall-1 106

  ColdFusion 129

  Data Points (Average) 87

  Data Points (Sum) 86

  Downloaded Component Size 83

  Error Statistics 42

  Errors per Second 43

  Fujitsu INTERSTAGE 129

  Hits per Second 56

  HTTP Responses per Second Graph 59

  HTTP Status Code Summary Graph 58

  Microsoft Active Server Pages  
  (ASP) 130

  Microsoft IIS 110

  Netscape 112

  Network Delay Time 101

  Network Segment 103

  Network Sub-Path Time 102

  Oracle 146

  Page Component Breakdown 69

  Page Component Breakdown (Over  
  Time) 71

  Page Download Time Breakdown 73

  Page Download Time Breakdown  
  (Over Time) 77

  Pages Downloaded per Second  
  Graph 62

  RealPlayer Client 150

  ReaPlayer Server 152

  Rendezvous 40

  Response Time-Under Load 52

  Running Vusers 38

  SilverStream 131

  SNMP Resources 96

  SQL Server 144

  Throughput 57

graph (cont'd)

  Time to First Buffer Breakdown 79

  Time to First Buffer Breakdown (Over  
  Time) 81

  Total Transactions per second 49

  Transaction Distribution 54

  Transaction Performance Summ. 51

  Transaction Response Time -  
  Average 46

  Transaction Summary 50

  TransactionPercentile 53

  Transactions per second 48

  TUXEDO Resources 97

  Vuser Summary 39

  WebLogic 132

  WebSphere 134

  Windows Media Server 153

  Windows Resources 90

graph types

  Database Server Resources 143–147

  Errors 41–43

  Firewall Server Monitor 105–106

  Network Monitor 99–104

  Streaming Media Resources 149–154

  System Resources 89–98

  Transaction 45–54

  User-Defined Data Points 85–87

  Vuser 37–40

  Web Application Server  
  Resources 115–141

  Web Page Breakdown 65–79

  Web Resources 55–63

  Web Server Resources 107–113

graphs

  application server 193

  database server 192

  Hits per Second 189

  Network Sub-Path Time 188

  Throughput 189

  Transaction Performance  
  Summary 185

  Transaction Response Time - Under  
  Load 183

  Web Page Breakdown 187



graphs, working with  
 background 28  
 crossing results 155–162  
 display options 25  
 merging 159  
 overlaying, superimposing 159

**H**

Hits per Second graph 56, 189  
 HTML reports 166  
 HTTP Responses per Second graph 59  
 HTTP Status Code Summary graph 58  
 HTTPS 191

**L**

legend 21  
 legend preferences 28  
 LoadRunner Analysis 1  
 LoadRunner Analysis Reports 164  
 lr\_user\_data\_point 85

**M**

marks in a graph 28  
 Merging graphs 159  
 Microsoft Active Server Pages (ASP)  
 graph 130  
 Microsoft IIS graph 110

**N**

Netscape graph 112  
 Network Delay Time graph 101  
 Network Monitor graphs 99–104  
 Network Segment graph 103  
 Network Sub-Path Time graph 188  
 Network Sub-PathTime graph 102

**O**

Online Function Reference vii  
 Oracle graph 146  
 output window 2  
 overlay graphs 159

**P**

packets 100  
 Page Component Breakdown (Over Time)  
 graph 71  
 Page Component Breakdown graph 69  
 Page Download Time Breakdown (Over  
 Time) graph 77  
 Page Download Time Breakdown graph 73  
 Pages Downloaded per Second graph 62  
 Performance reports 167

**R**

raw data 22  
 RealPlayer Client graph 150  
 RealPlayer Server graph 152  
 Receive time 75  
 Rendezvous graph 40  
 report  
 Data Point 172  
 Detailed Transaction 173  
 Failed Transaction 170  
 Failed Vuser 171  
 Scenario Execution 169  
 Transaction Performance by  
 Vuser 174  
 report viewer 168  
 reports 163–174  
 Activity and Performance 167  
 displaying 167  
 HTML 166  
 summary 165  
 Running Vusers graph 38

**S**

scale factor 107, 149  
 scale of graph 13  
 Scenario Execution report 169  
 security problems 191  
 sessions 2  
 SilverStream graph 131  
 SNMP Resources graph 96  
 spreadsheet view 22  
 SQL Server graph 144  
 SSL Handshaking time 74

## LoadRunner Analysis User's Guide

Streaming Media Resource graphs 149–154  
summary data, viewing 4  
Summary report 165  
superimposing graphs 159  
Support Information viii  
Support Online viii

### T

templates  
    applying 34  
    saving 33  
TestDirector  
    connecting to a server 176  
    integration 175–181  
    opening a new session 178  
    opening an existing session 180  
    saving sessions to a project 181  
three-dimensional properties 28  
Throughput graph 57, 189  
Time to First Buffer Breakdown (Over Time)  
    graph 81  
Time to First Buffer Breakdown graph 79  
tool options  
    setting database options 30  
    setting general options 29  
Total Transactions per Second graph 49  
Transaction graphs 45–54  
Transaction Performance by Vuser  
    report 174  
Transaction Performance Summ. graph 51  
Transaction Performance Summary  
    graph 185  
Transaction Response Time - Under Load  
    graph 183  
Transaction Response Time graphs 46–54  
    Average 46  
    Distribution 54  
    Percentile 53  
    Under Load 52  
Transaction Summary graph 50  
transaction, breakdown 68  
Transactions per Second graph 48  
TUXEDO Resources graph 97

### U

user-defined data point 85  
User-Defined Data Point graphs 85–87

### V

Vuser graphs 37–40  
Vuser Summary graph 39

### W

Web Application Server graphs 115–141  
Web Page Breakdown graphs 65–79, 187  
    activating 67  
Web Resource graphs 55–63  
Web Server Resource graphs 107–113  
WebLogic graph 132  
WebSphere graph 134  
Windows Media Server graph 153  
Windows Resources graph 90

### X

x-axis interval 13

### Z

zoom 13





MERCURY INTERACTIVE

**Mercury Interactive Corporation**

1325 Borregas Avenue  
Sunnyvale, CA 94089 USA

**Main Telephone:** (408) 822-5200

**Sales & Information:** (800) TEST-911

**Customer Support:** (877) TEST-HLP

**Fax:** (408) 822-5300

**Home Page:** [www.mercuryinteractive.com](http://www.mercuryinteractive.com)

**Customer Support:** [support.mercuryinteractive.com](http://support.mercuryinteractive.com)



L RANUG7 . 02' 0 1