HP LoadRunner

for the Windows operating systems

Software Version: 11.00 Patch 02

Analysis User Guide

Document Release Date: February 2011

Software Release Date: February 2011

Legal Notices

Warranty

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

The information contained herein is subject to change without notice.

Restricted Rights Legend

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

Copyright Notices

Trademark Notices

Java is a registered trademark of Oracle and/or its affiliates..

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

Oracle® is a registered US trademark of Oracle Corporation, Redwood City, California.

UNIX® is a registered trademark of The Open Group.

Documentation Updates

The title page of this document contains the following identifying information:

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

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

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

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

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

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

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

Support

Visit the HP Software Support web site at:

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

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

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

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

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract. To register for an HP Passport ID, go to:

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

To find more information about access levels, go to:

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

Table of Contents

PART II: ANALYSIS GRAPHS

PART III: ANALYSIS REPORTS

PART IV: WORKING WITH DIAGNOSTICS

Welcome to LoadRunner Analysis

Welcome to the *HP LoadRunner Analysis User Guide*. This guide describes how to use the LoadRunner Analysis graphs and reports in order to analyze system performance.

You use Analysis after running a load test scenario in the HP LoadRunner Controller or HP Performance Center.

HP LoadRunner, a tool for performance testing, stresses your entire application to isolate and identify potential client, network, and server bottlenecks.

HP Performance Center implements the capabilities of LoadRunner on an enterprise level.

How This Guide Is Organized

This guide contains the following parts:

Part I **Working with Analysis**

Introduces LoadRunner Analysis, and describes how you work with Analysis graphs.

Part II **Analysis Graphs**

Lists the different types of Analysis graphs and explains how to interpret them.

Part III **Analysis Reports**

Explains Analysis reports and describes how create a report in Word.

Part IV **Working with Diagnostics**

Explains how to use the Analysis graphs to identify and pinpoint performance problems in Siebel, Oracle, SAP, J2EE, and .NET environments.

Who Should Read This Guide

This guide is for the following users of LoadRunner:

- Performance Engineers
- Project Manager

Readers of this guide should be moderately knowledgeable about enterprise application development and highly skilled in enterprise system and database administration.

Documentation Library Guides

The Documentation Library consists of the following guides and references, available online, in PDF format, or both. PDFs can be read and printed using Adobe Reader, which can be downloaded from the Adobe Web site (http://www.adobe.com).

Using this Documentation Library explains how to use the Documentation Library and how it is organized.

Accessing the Documentation

You can access the documentation as follows:

- From the **Start** menu, click **Start** > **LoadRunner** > **Documentation** and select the relevant document.
- From the **Help** menu, click **Documentation Library** to open the merged help.

Getting Started Documentation

- **Readme.** Provides last-minute news and information about LoadRunner. You access the Readme from the **Start** menu.
- **HP LoadRunner Quick Start** provides a short, step-by-step overview and introduction to using LoadRunner. To access the Quick Start from the Start menu, click **Start** > **LoadRunner** > **Quick Start.**
- **HP LoadRunner Tutorial.** Self-paced printable guide, designed to lead you through the process of load testing and familiarize you with the LoadRunner testing environment. To access the tutorial from the Start menu, click **Start** > **LoadRunner** > **Tutorial.**

LoadRunner Guides

- **HP Virtual User Generator User Guide.** Describes how to create scripts using VuGen. The printed version consists of two volumes, Volume I - *Using VuGen* and Volume II - *Protocols*, while the online version is a single volume. When necessary, supplement this user guide with the online *HP LoadRunner Online Function Reference*.
- **HP LoadRunner Controller User Guide.** Describes how to create and run LoadRunner scenarios using the LoadRunner Controller in a Windows environment. Also describes how to set up the server monitor environment and configure LoadRunner monitors for monitoring data generated during a scenario.
- **HP LoadRunner Analysis User Guide.** Describes how to use the LoadRunner Analysis graphs and reports after running a scenario to analyze system performance.
- **HP LoadRunner Installation Guide.** Explains how to install LoadRunner and additional LoadRunner components, including LoadRunner samples.

LoadRunner References

- **LoadRunner Function Reference.** Gives you online access to all of LoadRunner's functions that you can use when creating Vuser scripts, including examples of how to use the functions.
- **Analysis API Reference.** This Analysis API set can be used for unattended creating of an Analysis session or for custom extraction of data from the results of a test run under the Controller. You can access this reference from the Analysis Help menu.
- **Troubleshooting.** The Output dialog box (**Controller** > **View** > **Show Output**) displays details of any errors received for a scenario. Click the icon in the Help column to open the Troubleshooting guide. This guide provides clear explanations and troubleshooting tips for many Controller connectivity and Web protocol errors. It also provides general troubleshooting tips for Winsock, SAPGUI, and Citrix protocols.

Searching and Navigating the Documentation Library

The following functionality is available from the Documentation Library:

Option	Description
	**Search and Navigate.** Displays the navigation pane. This button is displayed only when the navigation pane is closed. The navigation pane includes the following tabs: ➤ **Contents tab.** Organizes topics in a hierarchical tree, enabling you to directly navigate to a specific guide or topic. ➤ **Index tab.** Displays a detailed alphabetical listing of topics, along with the numbers of the pages on which they are mentioned. Double-click an index entry to display the corresponding topic. If your selection occurs in multiple documents, the right pane displays a list of possible locations, enabling you to select a context. ➤ **Search tab.** Enables you to search for specific topics or keywords. Results are returned in ranked order. You can limit your search to a specific guide by selecting a value from the scope drop-down list. **Note:** The search looks for each individual word in the phrase and not for full phrases, regardless of whether you use quotations ("). ➤ **Favorites tab.** Enables you to bookmark specific topics for quick reference. The **Favorites** tab is available only when using the Java implementation of the Help. If your browser does not support Java, the JavaScript implementation is automatically used and the Favorites tab is not displayed.
	**Show in Contents.** Displays the **Contents** tab in the navigation pane, and highlights the entry corresponding to the currently displayed page. This button is displayed only when the navigation pane is open.
	**Previous and Next.** Navigates to the previous or next page in the currently displayed guide.

Option	Description
	**Send Documentation Feedback to HP.** We welcome your feedback. Use this button in any topic to open an email addressed to us, containing the page reference. Send us your comments, ideas for improvement, and any errors you find.
	**Print.** Prints the currently displayed page. To print a complete guide, access the printer-friendly link from the Documentation Library Home page.
Back	You can use your browser's **Back** function to return to the previously displayed page. In most browsers, you can right-click and select **Back** from the shortcut menu.
Using This Documentation Library	Located on the lower-left corner of each content page. Opens this section.
Glossary	Located on the lower-left corner of each content page. Opens a glossary containing definitions of terms and acronyms.

Topic Types

Note: This section applies to the LoadRunner Controller, VuGen, and Analysis User Guides only.

The content in the above mentioned LoadRunner guides is organized by topics. Three main topic types are in use: **Concepts**, **Tasks**, and **Reference**. The topic types are differentiated visually using icons.

Topic Type	Description	Usage
Concepts	Background, descriptive, or conceptual information.	Learn general information about what a feature does.
Tasks	**Instructional Tasks.** Step-by-step guidance to help you work with the application and accomplish your goals. Task steps can be with or without numbering: ➤ **Numbered steps.** Tasks that are performed by following each step in consecutive order. ➤ **Non-numbered steps.** A list of self-contained operations that you can perform in any order.	➤ Learn about the overall workflow of a task. ➤ Follow the steps listed in a numbered task to complete a task. ➤ Perform independent operations by completing steps in a non-numbered task.
	Use-case Scenario Tasks. Examples of how to perform a task for a specific situation.	Learn how a task could be performed in a realistic scenario.

Topic Type	Description	Usage
Reference	**General Reference**. Detailed lists and explanations of reference-oriented material.	Look up a specific piece of reference information relevant to a particular context.
	User Interface Reference. Specialized reference topics that describe a particular user interface in detail. Selecting **Help on this page** from the Help menu in the product generally open the user interface topics.	Look up specific information about what to enter or how to use one or more specific user interface elements, such as a window, dialog box, or wizard.
Troubleshooting and Limitations	**Troubleshooting and Limitations**. Specialized reference topics that describe commonly encountered problems and their solutions, and list limitations of a feature or product area.	Increase your awareness of important issues before working with a feature, or if you encounter usability problems in the software.

Additional Online Resources

Troubleshooting & Knowledge Base accesses the Troubleshooting page on the HP Software Support Web site where you can search the Self-solve knowledge base. Choose **Help** > **Troubleshooting & Knowledge Base**. The URL for this Web site is http://h20230.www2.hp.com/troubleshooting.jsp.

HP Software Support accesses the HP Software Support Web site. This site enables you to browse the Self-solve knowledge base. You can also post to and search user discussion forums, submit support requests, download patches and updated documentation, and more. Choose **Help** > **HP Software Support**. The URL for this Web site is www.hp.com/go/hpsoftwaresupport.

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

To find more information about access levels, go to:

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

To register for an HP Passport user ID, go to:

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

HP Software Web site accesses the HP Software Web site. This site provides you with the most up-to-date information on HP Software products. This includes new software releases, seminars and trade shows, customer support, and more. Choose **Help** > **HP Software Web site**. The URL for this Web site is www.hp.com/go/software.

Part I

Working with Analysis

1

Introduction to Analysis

This chapter includes:

***Concepts**

***Tasks**

***Reference**

Concepts

Analysis Overview

During load test 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 *HP LoadRunner Controller User Guide*. 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 load test 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 *HP Virtual User Generator User Guide*.
- The **Controller Output window** displays information about the load test scenario run. If your scenario run fails, look for debug information in this window.
- 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 load test 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 of each graph. A report automatically summarizes and displays the test's significant data in graphical and tabular format. You can generate reports based on customizable report templates.

Analysis Basics

This section describes basic concepts that will enhance your understanding of how to work with Analysis.

Creating Analysis Sessions

When you run a load test scenario, data is stored in a result file with an **.lrr** extension. 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). Analysis stores the display information and layout settings for the active graphs in a file with an **.lra** extension.

Starting Analysis

You can open Analysis as an independent application or directly from the Controller. To open Analysis as an independent application, choose one of the following:

- **Start** > **Programs** > **LoadRunner** > **Applications** > **Analysis**
- **Start** > **Programs** > **LoadRunner** > **LoadRunner**, select the **Load Testing** tab, and then click **Analyze Load Tests**.

To open Analysis directly from the Controller, select **Results** > **Analyze Results**. This option is only available after running a load test scenario. 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 Analysis after it completes scenario execution by selecting **Results** > **Auto Load Analysis**.

Collating Execution Results

When you run a load test 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, and clearing the check mark adjacent to the option. To manually collate results, choose **Results** > **Collate Results**. If your results have not been collated, Analysis will automatically collate the results before generating the analysis data. For more information about collating results, refer to the *HP LoadRunner Controller User Guide*.

Analysis Graphs

Analysis graphs are divided into the following categories:

- **Application Component Graphs.** Provide information about resource usage of the Microsoft COM+ server and the Microsoft NET CLR server. For more information, see "Application Component Graphs" on page 409.
- **Application Deployment Solutions Graphs.** Provide information about resource usage of the Citrix MetaFrame server. For more information, see "Application Deployment Solutions Graphs" on page 445.
- **Database Server Resource Graphs.** Provide information about database resources. For more information, see "Database Server Resource Graphs" on page 349.
- **ERP/CRM Server Resource Graphs.** Provide information about ERP/CRM server resource usage. For more information, see "ERP/CRM Server Resource Graphs" on page 389.
- **Error Graphs.** Provide information about the errors that occurred during the load test scenario. For more information, see "Error Graphs" on page 239.
- **Firewall Server Monitor Graphs.** Provide information about firewall server resource usage. For more information, see "Firewall Server Monitor Graphs" on page 319.

- **Infrastructure Resources Graphs.** Provide information about resource usage of FTP, POP3, SMTP, IMAP, and DNS Vusers on the network client. For more information, "Infrastructure Resources Graphs" on page 467.
- **J2EE & .NET Diagnostics Graphs.** Provide information to trace, time, and troubleshoot individual transactions through J2EE & .NET Web, application, and database servers. For more information, see "J2EE & .NET Diagnostics Graphs" on page 595.
- **Middleware Performance Graphs.** Provide information about resource usage of the Tuxedo and IBM WebSphere MQ servers. For more information, see "Middleware Performance Graphs" on page 455.
- **Network Monitor Graphs.** Provide information about the network delays. For more information, see "Network Monitor Graphs" on page 271.
- **Oracle 11i Diagnostics Graphs.** Provide detailed breakdown diagnostics for SQLs generated by transactions on the Oracle NCA system. For more information, see "Oracle 11i Diagnostics Graphs" on page 549.
- **SAP Diagnostics Graphs.** Provide detailed breakdown diagnostics for SAP data generated by transactions on the SAP Server. For more information, see "SAP Diagnostics Graphs" on page 565.
- **Siebel Diagnostics Graphs.** Provide detailed breakdown diagnostics for transactions generated on Siebel Web, Siebel App, and Siebel Database servers. For more information, see "Siebel Diagnostics Graphs" on page 507.
- **Siebel DB Diagnostics Graphs.** Provide detailed breakdown diagnostics for SQLs generated by transactions on the Siebel system. For more information, see "Siebel DB Diagnostics Graphs" on page 531.
- **Streaming Media Graphs.** Provide information about resource usage of streaming media. For more information, see "Streaming Media Graphs" on page 377.
- **System Resource Graphs.** Provide statistics relating to the system resources that were monitored during the load test scenario using the online monitor. This category also includes graphs for SNMP monitoring. For more information, see "System Resource Graphs" on page 305.

- **Transaction Graphs.** Provide information about transaction performance and response time. For more information, see "Transaction Graphs" on page 219.
- **User-Defined Data Point Graphs.** Provide information about the custom data points that were gathered by the online monitor. For more information, see "User-Defined Data Point Graphs" on page 265.
- **Vuser Graphs.** Provide information about Vuser states and other Vuser statistics. For more information, see "Vuser Graphs" on page 233.
- **Web Application Server Resource Graphs.** Provide information about the resource usage for various Web application servers. For more information see "Web Application Server Resource Graphs" on page 331.
- **Web Page Diagnostics Graphs.** Provide information about the size and download time of each Web page component. For more information, see "Web Page Diagnostics Graphs" on page 279.
- **Web Resource Graphs.** Provide information about the throughput, hits per second, HTTP responses per second, number of retries per second, and downloaded pages per second for Web Vusers. For more information, see "Web Resources Graphs" on page 247.
- **Web Server Resource Graphs.** Provide information about the resource usage for the Apache, iPlanet/Netscape, iPlanet(SNMP), and MS IIS Web servers. For more information see "Web Resources Graphs" on page 247.

Analysis API

The LoadRunner Analysis API enables you to write programs to perform some of the functions of the Analysis user interface, and to extract data for use in external applications. Among other capabilities, the API allows you to create an analysis session from test results, analyze raw results of an Analysis session, and extract key session measurements for external use. An application can be launched from the LoadRunner Controller at the completion of a test. For more information, see the *Analysis API Reference*.

WAN Emulation

LoadRunner is integrated with 3rd party software that enables you to accurately test point-to-point performance of WAN-deployed products under real-world network conditions. By installing this 3rd party software on your load generators, you introduce highly probable WAN effects such as latency, packet loss, and link faults over your LAN. As a result of this, your scenario performs the test in an environment that better represents the actual deployment of your application.

You can create more meaningful results by configuring several load generators with the same unique set of WAN effects, and by giving each set a unique location name, for example, London. When viewing scenario results in Analysis, you can group metrics from different load generators according to their location names.

Tasks

How to Customize the Layout of Analysis Windows

This task describes ways to customize the layout of the windows of the Analysis session.

Open Windows

You can open a window or restore a window that was closed by selecting the name of the relevant window from the **Windows** menu.

Lock/Unlock the Layout of the Screen

Select **Windows** > **Layout locked** to lock or unlock the layout of the screen.

Restore the Window Placement to the Default Layout

Select **Windows** > **Restore Default Layout** to restore the placement of the Analysis windows to their default layout.

Note: This option is available only when no Analysis session is open.

Restore the Window Placement to the Classic Layout

Select **Windows** > **Restore Classic Layout** to restore the placement of the Analysis windows to their classic layout. The classic layout resembles the layout of earlier versions of Analysis.

Note: This option is available only when no Analysis session is open.

Reposition and Dock Windows

You can reposition any window by dragging it to the desired position on the screen. You can dock a window by dragging the window and using the arrows of the guide diamond to dock the window in the desired position.

Notes:

- Only document windows (graphs or reports) can be docked in the center portion of the screen.
- Windows > Layout Locked must not be selected when repositioning or docking windows.

Using Auto Hide

You can use the Auto Hide feature to minimize open windows that are not in use. The window is minimized along the edges of the screen.

Click the **Auto Hide** button on the title bar of the window to enable or disable Auto Hide.

Reference

Analysis User Interface

This section includes:

Analysis Toolbars

This section describes the buttons that you access from the main Analysis toolbars.

Common Toolbar

This toolbar is always accessible from the toolbar at top of the page and includes the following buttons:

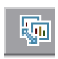
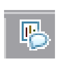
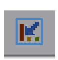
User interface elements are described below:

UI Elements	Description
	Create a new session.
	Open an existing session.
	Generate a Cross Result graph.
	Save a session.
	Print item.
	Create an HTML report.
	View runtime settings.
	Set global filter options.
	Analyze a transaction.
	Undo the most recent action.
	Reapply the last action that was undone.

Graph Toolbar

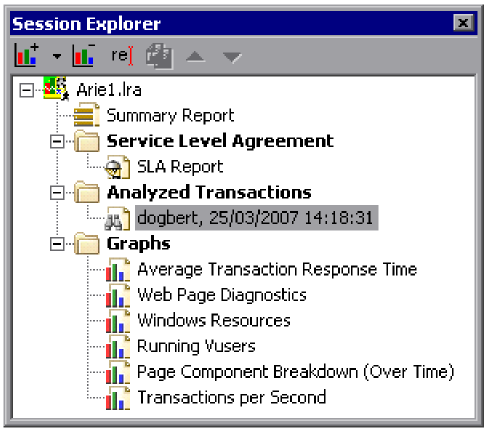
This toolbar is accessible from the top of the page when you have a graph open and includes the following buttons

User interface elements are described below:

UI Elements (A-Z)	Description
	Set filter settings.
	Clear filter settings.
	Set granularity settings.
	Merge graphs.
	Configure auto correlation settings.
	View raw data.
	Add comments to a graph.
	Add arrows to a graph.
	Set display options.

Session Explorer Window

This window displays a tree view of the items (graphs and reports) that are open in the current session.When you click an item in the Session Explorer, it is activated in the main Analysis window.



To access	Use one of the following: ➤ **Session Explorer** ➤ **Session Explorer** > **Reports** > **Summary Report** ➤ **Session Explorer** > **Reports** >**Service Level Agreement Report** ➤ **Session Explorer** >  > **Analyze Transaction** ➤ **Session Explorer** > **Graphs**

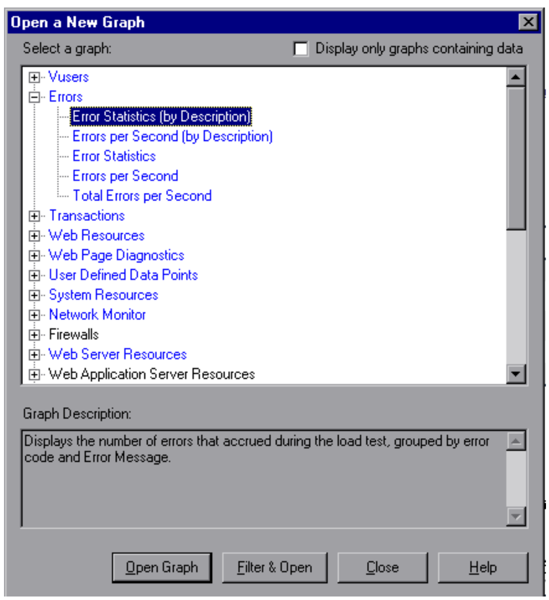
User interface elements are described below:

UI Elements	Description
	Add a new graph or report to the current Analysis session. Opens the Open a New Graph dialog box. For details, see "Open a New Graphs Dialog Box" on page 43
	Delete the selected graph or report.

UI Elements	Description
	Rename the selected graph or report.
	Create a copy of the selected graph.

Open a New Graphs Dialog Box

This dialog box enables you to select the graph type to activate in the main Analysis window.



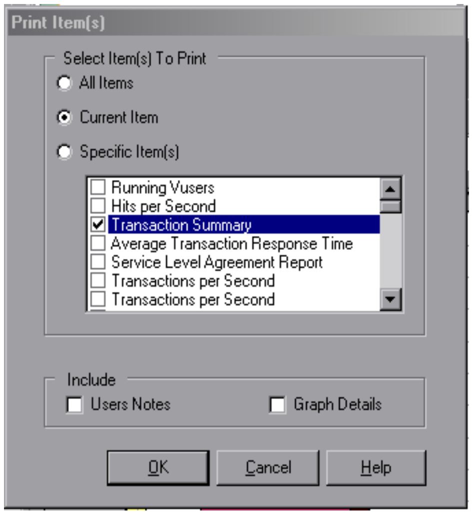
To access	**Session Explorer** > **Graphs** > 

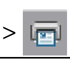
User interface elements are described below:

UI Elements	Description
Select a graph	Displays list of graph types.
Display only graphs containing data	If checked, only graphs that contain data are listed (in blue) in the Select a graph area.
Graph Description	Displays detailed information about the selected graph.
	Analysis generates the selected graph and adds it to the Session Explorer
	Opens the graphs Graph Settings dialog box. For details, see Chapter 4, "Filter Dialog Boxes". This option enables you to apply filter conditions on the selected graph before the graph is displayed.

Print Graphs or Reports

This dialog box enables you to print graphs or reports



To access	Select from one of the two options: ➤ **File** > **Print** ➤ Main toolbar > 

User interface elements are described below:

UI Elements	Description
Select Items to Print	➤ All Items - Prints all graphs and reports in the current session. ➤ Current Item - Prints the graph or report currently selected in the Session Explorer. ➤ Specific Item(s) - Select the graphs or reports to print.
Include	➤ User Notes - Prints the notes in the User Notes window. ➤ Graph Details - Prints details such as graph filters and granularity settings.

2

Configuring Analysis

This chapter includes:

Concepts

Tasks

Reference

Concepts

Summary Data Versus Complete Data

In large load test scenarios, with results exceeding 100 MB, it can take a long time for Analysis to process the data. When you configure how Analysis generates result data from load test scenarios, you can choose to generate complete data or summary data.

Complete data refers to the result data after it has been processed for use within Analysis.

Summary data refers to the raw, unprocessed data. The summary graphs contain general information such as transaction names and times. Some fields are not available for filtering when you work with summary graphs.

The following graphs are not available when viewing summary data only:

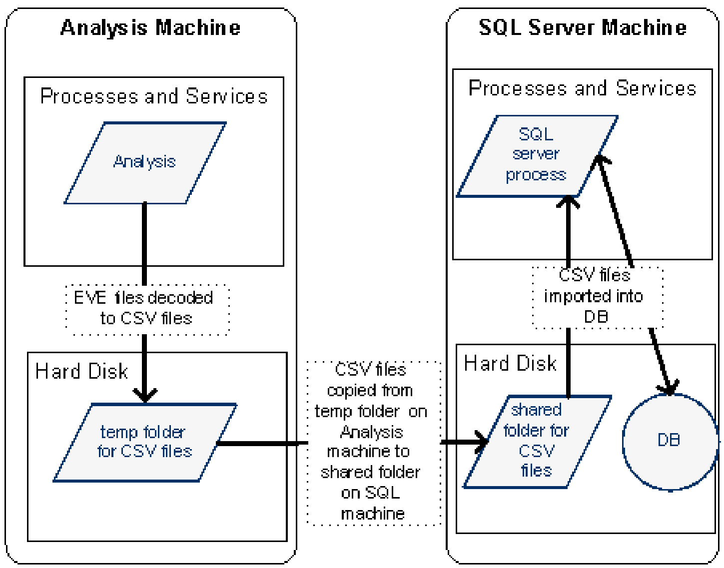
- Data Point (Sum)
- Error
- Network Monitor
- Rendezvous
- Siebel DB Side Transactions
- Siebel DB Side Transactions by SQL Stage
- SQL Average Execution Time
- Web Page Diagnostics

Importing Data Directly from the Analysis Machine

If you are using an SQL server / MSDE machine to store Analysis result data, you can configure Analysis to import data directly from the Analysis machine.

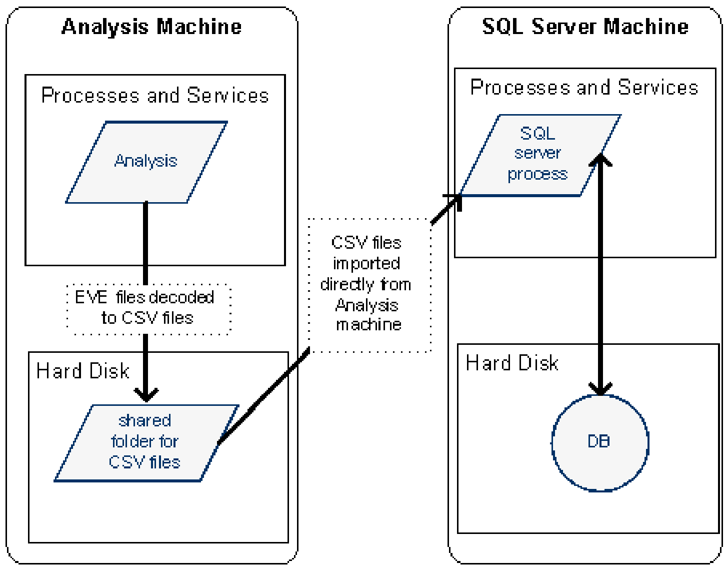
Importing Data from the SQL Server

If you do not select the option to import data directly from the Analysis machine, Analysis creates CSV files in a local temp directory. The CSV files are copied to a shared directory on the SQL Server machine. The SQL server engine then imports the CSV files into the database. The following diagram illustrates the data flow:



Importing Data from the Analysis Machine

If you selected the option to import data directly from the Analysis machine, Analysis creates the CSV files in a shared directory on the Analysis machine and the SQL server imports these CSV files from the Analysis machine directly into the database. The following diagram illustrates the data flow:



Tasks

How to Configure Settings for Analyzing Load Test Results

The following steps describe how to configure certain Analysis settings that significantly impact the way in which Analysis analyzes load test results.

Note: All of these settings are already pre-defined with default options.

Configure how Analysis processes result data

You define how Analysis processes result data from load test scenarios in the **Tools** > **Options** > **Result Collection** tab. For example, you can configure how Analysis aggregates result data, to what extent the data is processed, and whether output messages are copied from the Controller. For details on the user interface, see "Result Collection Tab" on page 64.

Configure template settings

For details on the user interface, see "Template Dialog Box" on page 90.

Configure analysis of transactions

You configure how transactions are analyzed and displayed in the summary report in the **Summary Report** area of the **Tools** > **Options** > **General** tab. For details, see the description of "General Tab (Options Dialog Box)" on page 61.

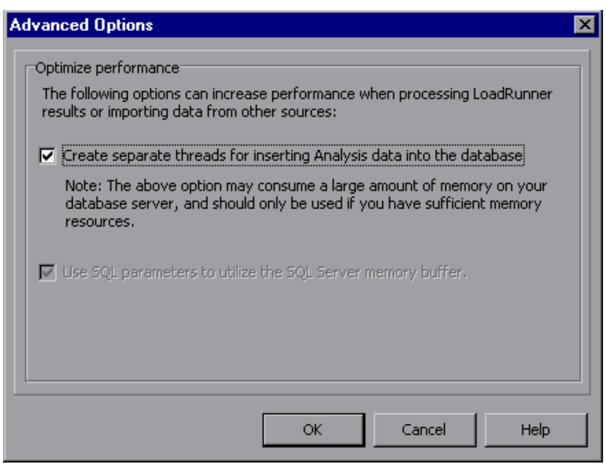
Reference

Configuration Options User Interface

This section includes (in alphabetical order):

Advanced Options Dialog Box (Database Tab)

This dialog box enables you to increase performance when processing LoadRunner results or importing data from other sources.



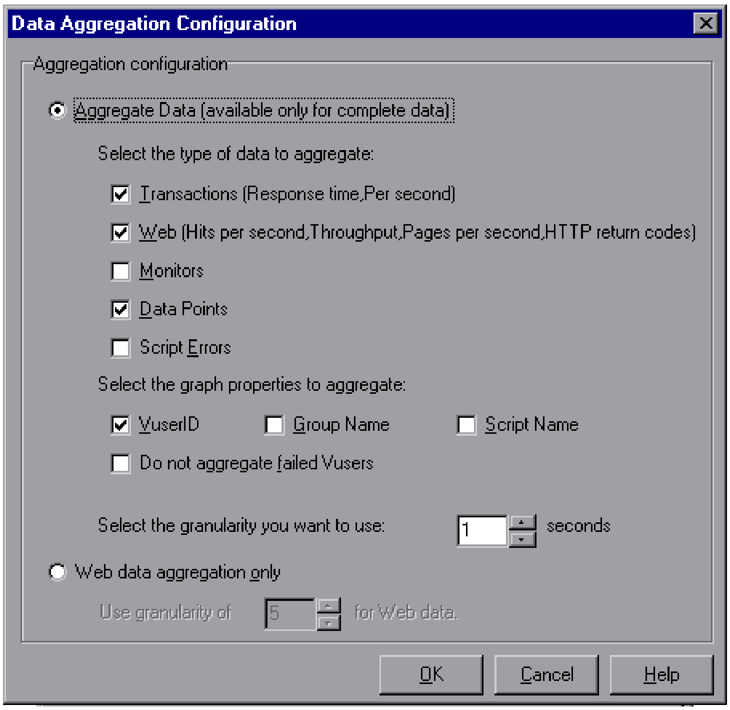
To access	**Tools** > **Options** > **Database** tab > **Advanced** button
See also	"Database Tab (Options Dialog Box)" on page 56

User interface elements are described below:

UI Elements	Description
Create separate threads for inserting Analysis data into the database.	This option may consume a large amount of memory on your database server, and should only be used if you have sufficient memory resources.
Use SQL parameters to utilize the SQL Server memory buffer.	This option is only enabled when you store Analysis result data on an SQL server or MSDE machine.

Data Aggregation Configuration Dialog Box

If you choose to generate the complete data from the load lest scenario results, Analysis aggregates the data using either built-in data aggregation formulas, or aggregation settings that you define. This dialog box enables you to define custom aggregation settings.



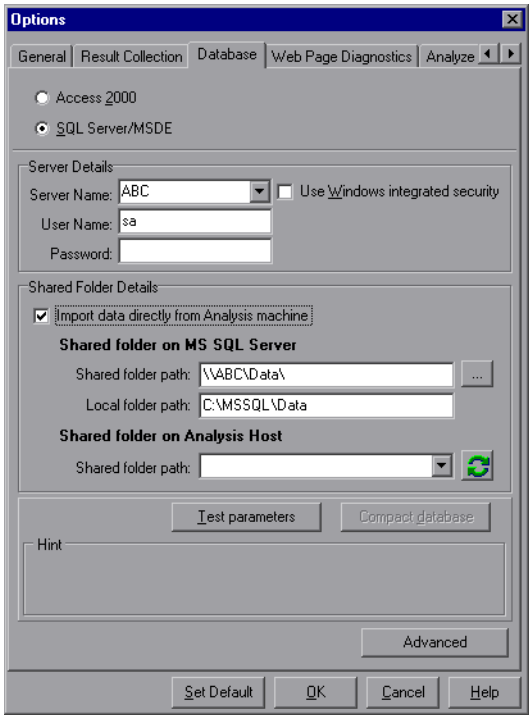
To access	Select **Tools > Options > Result Collection**. Select the **Apply user-defined aggregation** option and click the **Aggregation Configuration** button.
Important information	In this dialog box, you can select granularity settings. To reduce the size of the database, increase the granularity. To focus on more detailed results, decrease the granularity.

User interface elements are described below:

UI Element	Description
Aggregate Data	Select this option to define your custom aggregation settings using the following criteria: ➤ **Select the type of data to aggregate.** Use the check boxes to select the types of graphs for which you want to aggregate data. ➤ **Select graph properties to aggregate**. Use the check boxes to select the graph properties you want to aggregate. To exclude data from failed Vusers, select **Do not aggregate failed Vusers**. **Note:** You will not be able to drill down on the graph properties you select in this list. ➤ **Select the granularity you want to use.** Specify a custom granularity for the data. The minimum granularity is 1 second.
Web data aggregation only	Select this option to aggregate Web data only. In the **Use Granularity of X for Web data** box, specify a custom granularity for Web data. The minimum granularity is 1 second. By default, Analysis summarizes Web measurements every 5 seconds.

Database Tab (Options Dialog Box)

This tab enables you to specify the database in which to store Analysis session result data and to configure the way in which CSV files will be imported into the database.



To access	**Tools** > **Options** > **Database** tab.
Important information	If your Analysis result data exceeds two gigabytes, it is recommended that you store it on an SQL server / MSDE machine.
See also	"Importing Data Directly from the Analysis Machine" on page 49

User interface elements are described below:

UI Elements	Description
Access 2000	Instructs LoadRunner to save Analysis result data in an Access 2000 database format. This setting is the default.
SQL Server/MSDE	Instructs LoadRunner to save Analysis result data on an SQL server / MSDE machine. If you select this option, you have to complete the **Server Details** and **Shared Folder Details**, described below.
Server Details area	SQL server / MSDE machine details. See description below.
Shared Folder Details area	SQL server / MSDE machine shared folder details. See description below.
	Depending on which database you are using, this button performs the following action: ➤ **For Access.** Lets you connect to the Access database and verify that the **list separator** registry options on your machine are the same as those on the database machine. ➤ **For SQL server / MSDE.** Lets you connect to the SQL server / MSDE machine and see that the shared directory you specified exists on the server, and that you have write permissions on the shared server directory. If so, Analysis synchronizes the shared and physical server directories.

UI Elements	Description
Compact database	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. For Access databases, the **Compact database** button enables you to repair and compress your results and optimize your database. **Note:** Long load test scenarios (duration of two hours or more) will require more time for compacting.
Advanced	Opens the Advanced Options dialog box, allowing you to increase performance when processing LoadRunner results or importing data from other sources. For user interface details, see"Advanced Options Dialog Box (Database Tab)" on page 52.

Server Details Area

If you choose to store Analysis result data on an SQL server / MSDE machine, you need to complete the server details. User interface elements are described below:

UI Elements	Description
Server Name	The name of the machine on which the SQL server / MSDE is running.
Use Windows integrated security	Enables you to use your Windows login, instead of specifying a user name and password. By default, the user name "sa" and no password are used for the SQL server.
User Name	The user name for the master database.
Password	The password for the master database.

Shared Folder Details Area

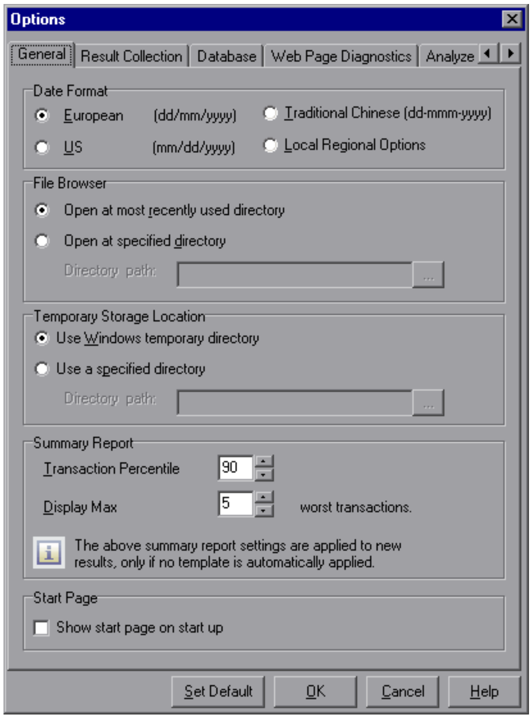
If you store Analysis result data on an SQL server / MSDE machine, you need to provide the shared folder details. User interface elements are described below:

UI Elements	Description
Import Data Directly from Analysis machine	Select this option to import data directly from the Analysis machine. For details on this option, see "Importing Data Directly from the Analysis Machine" on page 49.
Shared Folder on MS SQL Server	➤ **Shared folder path.** Enter a shared directory on the SQL server / MSDE machine. For example, if your SQL server's name is fly, enter \\fly\<Analysis database directory>\. This folder has different functions, depending on how you import the Analysis data: ➤ **If you did not select the option to import data directly from the Analysis machine**, this directory stores permanent and temporary database files. Analysis results stored on an SQL server / MSDE machine can only be viewed on the machine's local LAN. ➤ **If you selected the option to import data directly from the Analysis machine**, this directory is used to store an empty database template copied from the Analysis machine. ➤ **Local folder path.** Enter the real drive and directory path on the SQL server / MSDE machine that correspond to the above shared folder path. 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 directory>. If the SQL server / MSDE and Analysis are on the same machine, the logical storage location and physical storage location are identical.

UI Elements	Description
Shared Folder on Analysis Host	If you selected the option to import data directly from the Analysis machine, the **Shared folder path** box is enabled. Analysis detects all shared folders on your Analysis machine and displays them in a drop-down list. Select a shared directory from the list. **Note:** ➤ Ensure that the user running the SQL server (by default, SYSTEM) has access rights to this shared folder. ➤ If you add a new shared directory on your machine, you can click the refresh button  to display the updated list of shared folders. ➤ Analysis creates the CSV files in this directory and the SQL server imports these CSV files from the Analysis machine directly into the database. This directory stores permanent and temporary database files.

General Tab (Options Dialog Box)

This tab enables you to configure general Analysis options, such as date formats, temporary storage location, and transaction report settings.



To access	**Tools** > **Options** > **General** tab.
See Also	"How to Configure Settings for Analyzing Load Test Results" on page 51

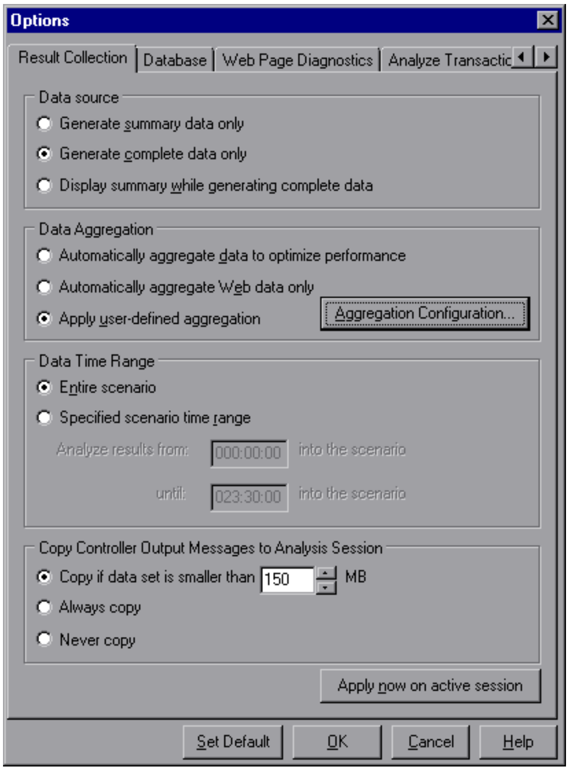
User interface elements are described below:

<table>
<tr><th>UI Elements (A-Z)</th><th>Description</th></tr>
<tr><td>Date Format</td><td>Select a date format for storage and display. (For example, the date displayed in the Summary report)
➤ European. Displays the European date format.
➤ US. Displays the U.S. date format.
➤ Traditional Chinese. Displays the Traditional Chinese date format.
➤ Local Regional Options. Displays the date format as defined in the current user's regional settings.
Note: When you change the date format, it only affects newly created Analysis sessions. The date format of existing sessions is not affected.</td></tr>
<tr><td>File Browser</td><td>Select the directory location at which you want the file browser to open.
➤ Open at most recently used directory. Opens the file browser at the previously used directory location.
➤ Open at specified directory. Opens the file browser at a specified directory.
In the Directory path box, enter the directory location where you want the file browser to open.</td></tr>
<tr><td>Start Page</td><td>Select Show start page on start up to display the Welcome to Analysis tab every time you open the Analysis application.</td></tr>
</table>

UI Elements (A-Z)	Description
Summary Report	Set the following transaction settings in the Summary Report: ➤ **Transaction Percentile.** The Summary Report contains a percentile column showing the response time of 90% of transactions (90% of transactions that fall within this amount of time). To change the value of the default 90 percentile, enter a new figure in the **Transaction Percentile** box. Since this is an application level setting, the new value is only applied the next time you analyze a result file (**File** > **New**). ➤ **Display Max.** Where a Service Level Agreement (SLA) has been defined, the Summary Report contains the Worst Transactions table that displays the transactions that most exceeded the SLA boundary. The setting here defines how many transactions are displayed in that table. To change this number (for example, to **6**), enter a new figure in the **Display Max** box. ➤ Since this is an application level setting, the new value is only applied the next time you analyze a result file (**File** > **New**). **Note:** If a template is automatically applied to new sessions, the transaction settings are defined according to the definitions in the template, and not according to those in the Options dialog box. You define template settings in the Template dialog box (**Tools** > **Templates** > **Apply/Edit Template**).
Temporary Storage Location	Select the directory location in which you want to save temporary files. ➤ **Use Windows temporary directory.** Saves temporary files in your Windows temp directory. ➤ **Use a specified directory.** Saves temporary files in a specified directory. In the **Directory path** box, enter the directory location in which you want to save temporary files.

Result Collection Tab

This tab enables you to configure how Analysis processes result data from load test scenarios.



To access	**Tools** > **Options** > **Result Collection** tab.
Important information	The options in this tab are pre-defined with default settings. It is recommended to use these default settings unless there is a specific need to change them. Changing some of the settings, such as default aggregation, can significantly impact the amount of data stored in the Analysis database.
See Also	"How to Configure Settings for Analyzing Load Test Results" on page 51

User interface elements are described below:

UI Elements	Description
Data Source	In this area, you configure how Analysis generates result data from load test scenarios. **Complete data** refers to the result data after it has been processed for use within Analysis. **Summary data** refers to the raw, unprocessed data. The summary graphs contain general information such as transaction names and times. For more details on summary data versus complete data, see "Summary Data Versus Complete Data" on page 48. Select one of the following options: ➤ **Generate summary data only.** If this option is selected, Analysis will not process the data for advanced use with filtering and grouping. ➤ **Generate complete data only.** If this option is selected, the graphs can then be sorted, filtered, and manipulated. ➤ **Display summary data while generating complete data.** Enables you to view summary data while you wait for the complete data to be processed. **Note:** If you selected one of the options to generate complete data, you can define how Analysis aggregates the complete data in the **Data Aggregation** area.

UI Elements	Description
Data Aggregation	If you chose to generate complete data in the **Data Source** area, you use this area to configure how Analysis aggregates the data. Data aggregation is necessary in order to reduce the size of the database and decrease processing time in large scenarios. Select one of the following options: ➤ **Automatically aggregate data to optimize performance.** Aggregates data using built-in data aggregation formulas. ➤ **Automatically aggregate Web data only.** Aggregates Web data only using built-in data aggregation formulas. ➤ **Apply user-defined aggregation.** Aggregates data using settings you define. Click the **Aggregation Configuration** button to open the Data Aggregation Configuration Dialog Box and define your custom aggregation settings. For details on the user interface, see "Data Aggregation Configuration Dialog Box" on page 54.

UI Elements	Description

Data Time Range	In this area you specify whether to display data for the complete duration of the scenario, or for a specified time range only. Select one of the following options:
	➤ **Entire scenario.** Displays data for the complete duration of the load test scenario
	➤ **Specified scenario time range.** Specify the time range using the following boxes:
	    ➤ **Analyze results from.** Enter the amount of scenario time you want to elapse (in hh:mm:ss format) before Analysis begins displaying data.
	    ➤ **until.** Enter the point in the scenario (in hh:mm:ss format) at which you want Analysis to stop displaying data.
	Notes:
	➤ It is not recommended to use the **Specified scenario time range** option when analyzing the Oracle 11i and Siebel DB Diagnostics graphs, since the data may be incomplete.
	➤ The **Specified scenario time range** settings are not applied to the Connections and Running Vusers graphs.

Copy Controller Output Messages to Analysis Session	Controller output messages are displayed in Analysis in the Controller Output Messages window. Select one of the following options for copying output messages generated by the Controller to the Analysis session. ➤ **Copy if data set is smaller than X MB.** Copies the Controller output data to the Analysis session if the data set is smaller than the amount you specify. ➤ **Always Copy.** Always copies the Controller output data to the Analysis session. ➤ **Never Copy.** Never copies the Controller output data to the Analysis session.
	Click this button to apply the settings in the Result Collection tab to the current session. The Controller output data is copied when the Analysis session is saved.

Session Information Dialog Box

This dialog box enables you to view a summary of the configuration properties of the current Analysis session.

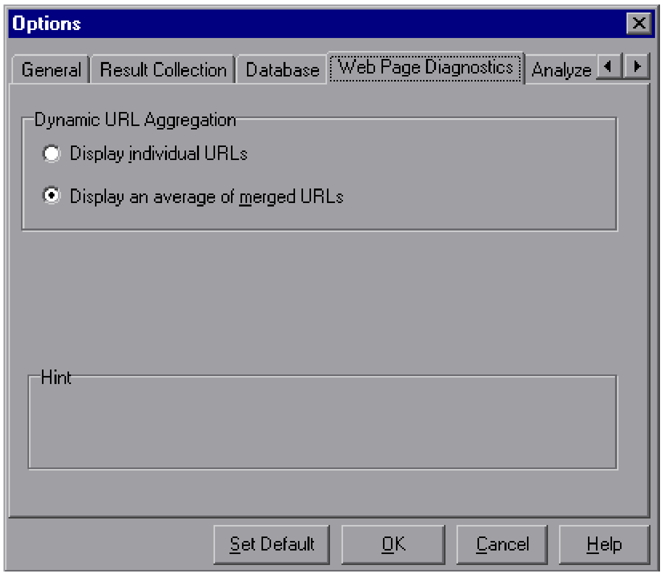
To access	**File > Session Information**

User interface elements are described below:

UI Elements (A-Z)	Description
	Displays the type of data aggregated, the criteria according to which it is aggregated, and the time granularity of the aggregated data.
	Displays the properties of the SQL server and MSDE databases.
Aggregation	Indicates whether the session data has been aggregated.
Data Collection Mode	Indicates whether the session displays complete data or summary data.
Data Time Filter	Indicates whether a time filter has been applied to the session.
Database Name	Displays the name and directory path of the database.
Database Type	Displays the type of database used to store the load test scenario data.
Results	Displays the name of the LoadRunner result file.
Session Name	Displays the name of the current session.
Web Granularity	Displays the Web granularity used in the session.

Web Page Diagnostics Tab

This tab enables you to set Web page breakdown options. You can choose how to aggregate the display of URLs that include dynamic information, such as a session ID. You can display these URLs individually, or you can unify them and display them as one line with merged data points.



To access	**Tools** > **Options** > **Web Page Diagnostics** tab

User interface elements are described below:

UI Elements	Description
Display individual URLs	Displays each URL individually
Display an average of merged URLs.	Merges URLs from the same script step into one URL, and displays it with merged (average) data points.

3

Configuring Graph Display

Analysis allows you to customize the display of the graphs and measurements in your session so that you can view the data displayed in the most effective way possible.

This chapter includes:

Concepts

Tasks

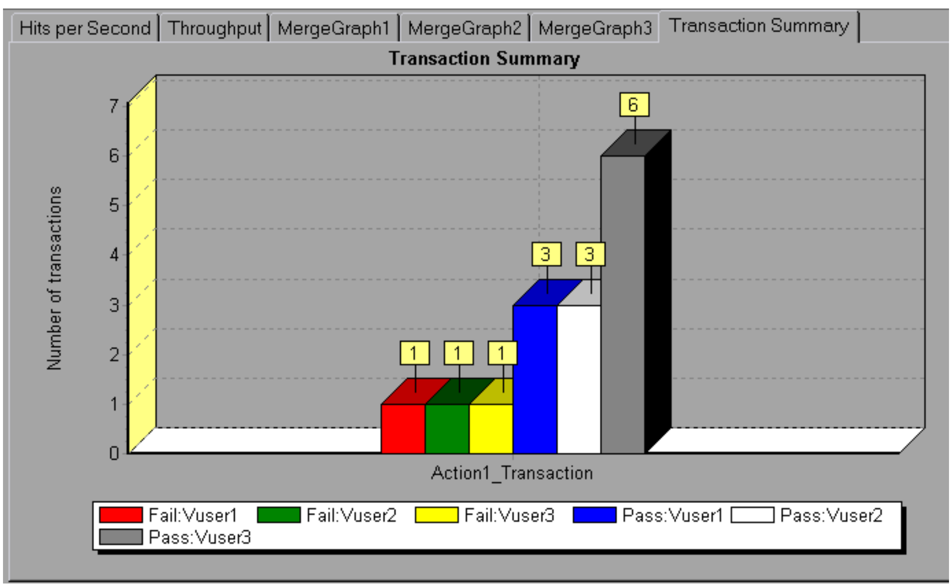
Reference

Concepts

Sorting Graph Data Overview

You can sort graph data to show the data in more relevant ways. For example, Transaction graphs can be grouped by the Transaction End Status, and Vuser graphs can be grouped by Scenario Elapsed Time, Vuser End Status, Vuser Status, and VuserID.

You can sort by one or several groups—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 grouping order by rearranging the list. The following graph shows the Transaction Summary grouped according to Vusers.



Tasks

How to Customize the Analysis Display

The following steps describes how to customize the display of analysis. You can customize the display of the graphs and measurements in your session so that you can view the data displayed in the most effective way possible.

- "Enlarging a section of the graph" on page 75
- "Using comments in a graph" on page 75
- "Using arrows in a graph" on page 75
- "Using the User Notes Window" on page 76

Enlarging a section of the graph

To zoom in or enlarge a section of the graph, move and hold down the left mouse button over the section of the graph you want to enlarge.

Using comments in a graph

To add a comment to a graph, click  and then click the mouse over the section of the graph where you would like to add a comment. Type your comment in the Add Comment dialog box.

To edit, format or delete a comment from the graph, click the comment and apply your change in the Edit Comments dialog box. In the left pane, verify the relevant comment is selected before you edit, format or delete.

Using arrows in a graph

To add an arrow to a graph, click  and then click the mouse button within the graph to position the base of the arrow.

To delete an arrow from a graph, select the arrow and press Delete.

Using the User Notes Window

In the User Notes window (**Windows** > **User Notes**), you can enter text about the graph or report that is currently open. The text in the User Notes window is saved with the session.

To view the text that you entered for a specific graph or report, select the relevant graph or report and open the User Notes window (**Windows** > **User Notes**).

Reference

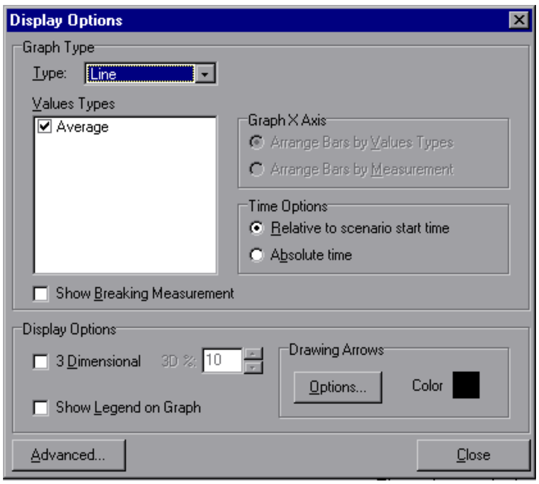
Configuring Graph Display User Interface

This section includes:

Display Options Dialog Box

This dialog box enables you to select the graph type and configure the display of the graph.

Note: This option is not available for all graph types.



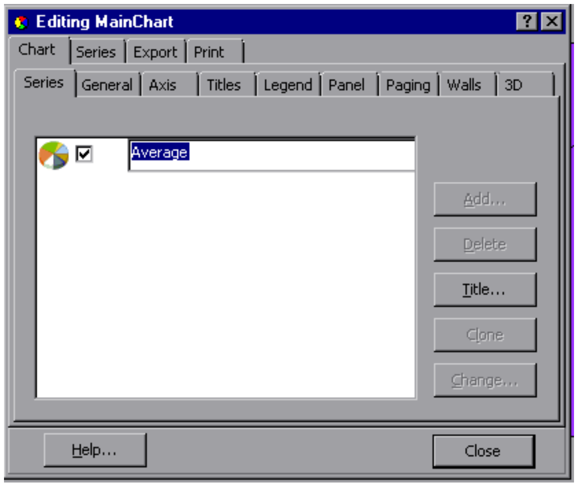
To access	**View > Display Options**
See also	➤ "Editing MainChart Dialog Box" on page 80 ➤ "Chart Tab" on page 81 ➤ "Series Tab" on page 82

User interface elements are described below:

UI Elements	Description
Type	Select the type of graph to display from the drop-down list.
Values Types	Select the type of display information from the list of available values. For example, a bar graph displaying Average Transaction Response Time can be configured to display minimum, maximum, average, STD, count, and sum averages.
Graph X Axis (Bar graphs only)	Select the bar arrangement along the x-axis. You can arrange the bars by value types or measurement.
Time Options	Select the way in which the graph shows the Elapsed Scenario Time on the x-axis. You can choose an elapsed time relative to the beginning of the scenario or an elapsed time from the absolute time of the machine's system clock.
Show Breaking Measurement	Select this check box to display the name and properties of the breaking measurement at the top of the graph (disabled by default).
3 Dimensional	Select this check box to enable a 3-dimensional display of the graph.
3D %	Specify a percentage for the 3-dimensional aspect of lines in the graph. This percentage indicates the thickness if the bar, grid, or pie chart.
Show Legend on Graph	Select this check box to display a legend at the bottom of the graph.
Drawing Arrows	Allows you to configure the style, color, and width of arrows you draw to highlight graph information.
Advanced...	Opens the **Editing MainChart** dialog box. For more information, see "Editing MainChart Dialog Box" on page 80.

Editing MainChart Dialog Box

This dialog box enables you to configure the look and feel of your graph as well as its title and the format of the data.



To access	**View** > **Display Options** > **Advanced** button
See also	"Display Options Dialog Box" on page 78 "Chart Tab" on page 81 "Series Tab" on page 82

User interface elements are described below:

UI Elements	Description
Chart tab	Enables you to configure the look and feel of your entire graph. You set Chart preferences using the following tabs: For details, see "Chart Tab" on page 81,
Series tab	Enables you to control the appearance of the individual points plotted in the graph. You set Series preferences using the following tabs. For details, see "Series Tab" on page 82.

UI Elements	Description
Export tab	Enables you to store the current graph to an image file in the format of your choice—BMP, JPG, or EMF. You can also export the graph's data to HTML, Excel, or XML
Print tab	Enables you to print only the graph itself without the legend and other data such as the User Notes.

Chart Tab

This tab enables you to configure the look and feel of your entire graph.

To access	**View** > **Display Options** > **Advanced button** > **Chart** tab
See also	"Display Options Dialog Box" on page 78 "Editing MainChart Dialog Box" on page 80 "Series Tab" on page 82

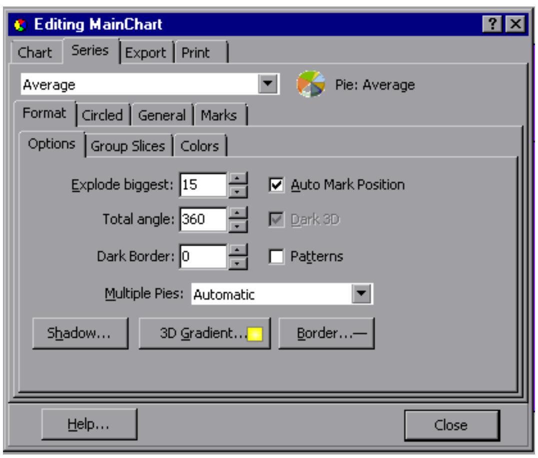
User interface elements are described below:

UI Elements	Description
Series tab	Select the graph style (bar, line, etc.), the hide/show settings, line and fill color, and the title of the series.
General tab	Select options for print preview, export, margins, scrolling, and magnification.
Axis tab	Select which axes to show, as well as their scales, titles, ticks, and position.
Titles tab	Set the title of the graph, its font, background color, border, and alignment.
Legend tab	Set all legend related settings, such as position, fonts, and divider lines.
Panel tab	Show the background panel layout of the graph. You can modify its color, set a gradient option, or specify a background image.

UI Elements	Description
Paging tab	Set all page related settings, such as amount of data per page, scale, and page numbering. These settings are relevant when the graph data exceeds a single page.
Walls tab	Set colors for the walls of 3-dimensional graphs.
3D	Select the 3-dimensional settings, offset, magnification, and rotation angle for the active graph.

Series Tab

This page enables you to control the appearance of the individual points plotted in the graph.



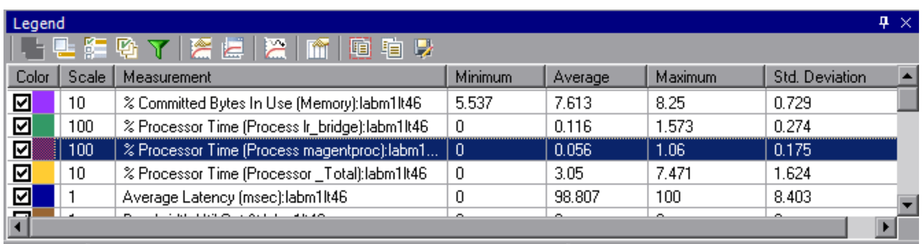
To access	**View** > **Display Options** > **Advanced button** > **Series** tab
See also	"Display Options Dialog Box" on page 78 "Editing MainChart Dialog Box" on page 80 "Chart Tab" on page 81

User interface elements are described below:

UI Elements	Description
Format tab	Set the border color, line color, pattern, and invert property for the lines or bars in your graph.
Point tab	Set the size, color, and shape of the points that appear within your line graph.
General tab	Select the type of cursor, the format of the axis values, and show/hide settings for the horizontal and vertical axes.
Marks tab	Configure the format for each point in the graph.

Legend Window

This window enables you to configure the color, scale, minimum, maximum, average, median, and standard deviation of each measurement appearing in the graph.



To access	**Analysis Window** > **Legend** window
See also	"Measurement Description Dialog Box" on page 86 "Measurement Options Dialog Box" on page 87

Legend Toolbar

User interface elements are described below:

UI Elements	Description
	Displays a measurement in the graph.
	Hides a measurement in the graph.
	Displays the highlighted measurement only.
	Displays all the available measurements in the graph.
	Filters the graph by the measurements selected in the Legend window. You can select multiple measurements. To clear the filter, select **View** > **Clear Filter/Group by**.
	Opens the Measurement Options dialog box that enables you to configure measurement options (for example, set color and measurement scale). For more information, see "Measurement Options Dialog Box" on page 87.
	Opens the Measurement Description dialog box that displays the name, monitor type, and description of the selected measurement. For more information, see "Measurement Description Dialog Box" on page 86.
	Displays the selected measurement as a flashing line.
	Opens the Legend Columns Options dialog box that enables you to configure the columns displayed in the Legend window. For more information, see "Legend Window" on page 83.
	Copies the data that you have selected to the Clipboard. You can paste the data in a text file or a spreadsheet.

UI Elements	Description
	Copies all the legend data to the Clipboard, regardless of what is selected. You can paste the data in a text file or a spreadsheet.
	Saves the legend data to a CSV file.

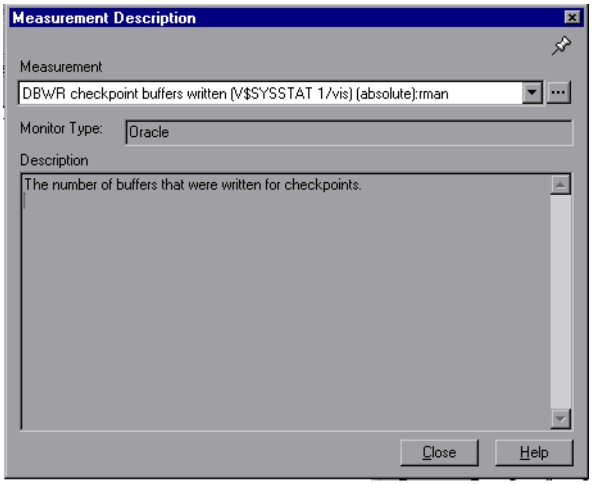
Legend grid shortcut menu

User interface elements are described below:

UI Elements(A-Z)	Description
Auto Correlate.	Opens the Auto Correlate dialog box that enables you to correlate the selected measurement with other monitor measurements in the load test scenario. For more information on auto correlation, see "Auto Correlating Measurements" on page 123.
Web Page Diagnostics for <selected measurement	Appears for measurements in the Average Transaction Response Time and Transaction Performance Summary graphs): Displays a Web Page Diagnostics graph for the selected transaction measurement.
Sort by measurement column	Sorts the measurements according to the selected column, in ascending or descending order.
Break down	(appears for measurements in the Web Page Diagnostics graphs). Displays a graph with a breakdown of the selected page.

Measurement Description Dialog Box

This dialog box shows you additional information about the selected measurement.



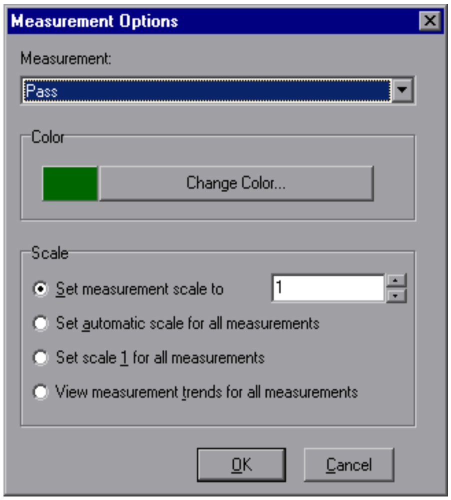
To access	Legend Toolbar > 
See also	"Legend Window" on page 83 "Measurement Options Dialog Box" on page 87

User interface elements are described below:

UI Elements	Description
Measurement	Displays the name of the selected measurement. Click the drop-down arrow to select a different measurement.
Monitor Type	Displays the type of monitor used to obtain the selected measurement.
Description	Displays a description of the selected monitored measurement.
SQL	If an SQL logical name is in use, displays the full SQL statement.

Measurement Options Dialog Box

This dialog box enables you to set the color and the scale for any measurement of the graph you selected.



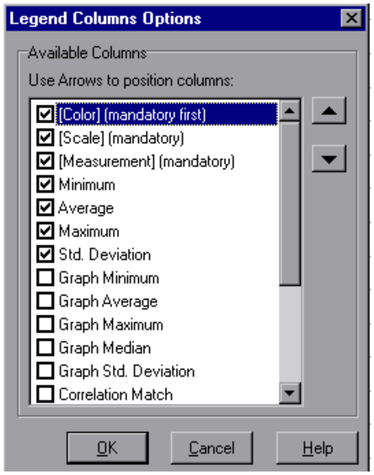
To access	Legend Toolbar >
See also	"Legend Window" on page 83 "Measurement Description Dialog Box" on page 86

User interface elements are described below:

UI Elements	Description
Measurement	Select a measurement to configure.
Change Color	Select a new color for the selected measurement.
Scale	Select the desired scale option: ➤ Set measurement scale to x. Select the scale with which you want to view the selected measurement. ➤ Set automatic scale for all measurements. Uses an automatic scale optimized to display each measurement in the graph. ➤ Set scale 1 for all measurements. Sets the scale to one for all measurements in the graph. ➤ View measurement trends for all measurements. Standardizes the y-axis values in the graph, according to the following formula: New Y value = (Previous Y Value - Average of previous values) / STD of previous values.

Legend Columns Options Dialog Box

This dialog box enables you to select the columns to be displayed.



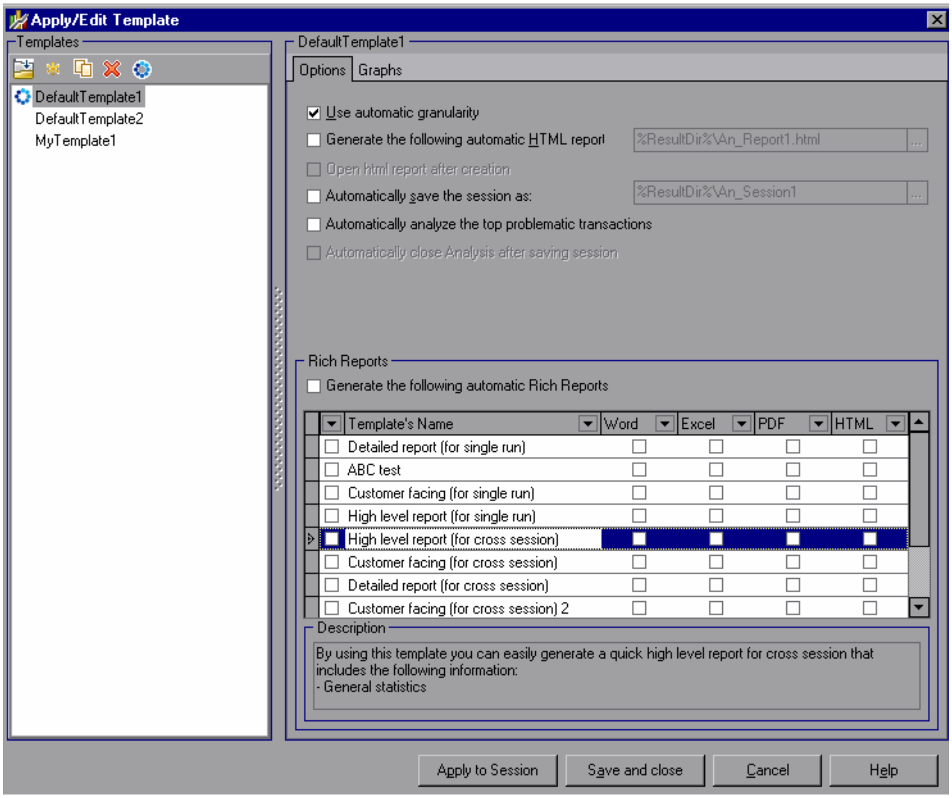
To access	**View > Legend Columns**
See also	"Legend Window" on page 83

User interface elements are described below:

UI Elements	Description
Available Columns	Select or deselect the check boxes to the left of the column names to show or hide the columns respectively. **Notes:** ➤ The Color, Scale, and Measurement columns are mandatory and cannot be deselected. ➤ To rearrange the order in which the columns appear (from left to right), you use the vertical arrows to the right of the Available Columns list to place the columns in the desired order.

Template Dialog Box

This dialog box enables you to set template settings, automatic activity options, and select report template options.



To access	**Tools > Templates**

User interface elements are described below:

UI Elements	Description
Templates .	Select one of the following buttons: ➤ Select to browse for a template. ➤ - Select to add a template. Enter the title of the new template in the **Add new template** dialog box. ➤ - Select a template to duplicate. ➤ - Select to delete the selected template. ➤ - Select to set the selected template as the default.
Use automatic granularity.	Applies the default Analysis granularity (one second) to the template. For information about setting Analysis granularity, see "Changing the Granularity of the Data" on page 121.
Generate the following automatic HTML report.	Generates an HTML report using the template. Specify or select a report name. For information about generating HTML reports, see "HTML Report" on page 482.
Open html report after creation.	If you selected the option of generating an automatic HTML report, select this option to automatically open the HTML report after it is created.
Automatically save the session as	Automatically saves the session using the template you specify. Specify or select a file name.
Automatically analyze the top problematic transactions	Automatically generates Transaction Analysis reports for the transactions with the worst SLA violations. Reports are generated for a maximum of five transactions. For more information about Transaction Analysis reports, see "Analyze Transactions Dialog Box" on page 479.
Automatically close Analysis after saving session.	Automatically closes Analysis after a session is automatically saved (using the previous option). This prevents the running of multiple instances of Analysis.

UI Elements	Description
Generate the following automatic Rich Reports	The selected reports are added to the template.
<Check-box on left of Template's Name>	Select to add report template to selected template. The reports are added to the session.
Word	Generates a report using the selected report template to MS Word. **Note:** Take into account that the content load may affect the table format within the MS Word document.
Excel	Generates a report using the selected report template to Excel.
PDF	Generates a report using the selected report template to PDF.
HTML	Generates a report using the selected report template to HTML.
Graphs <tab>	Displays the list of graphs that are included in the template. When the template is applied to a session, the graphs are displayed under Graphs in Session Explorer. If there is no data in the session, the graphs are not created.

4

Filtering and Sorting Graph Data

This chapter includes:

Concepts

Reference

Concepts

Filtering Graph Data Overview

You can filter graph data to show fewer transactions for a specific segment of the load test scenario. More specifically, you can display four transactions beginning from five minutes into the scenario and ending three minutes before the end of the scenario.

You can filter for a single graph, in all graphs in a load test scenario, or in the summary graph.

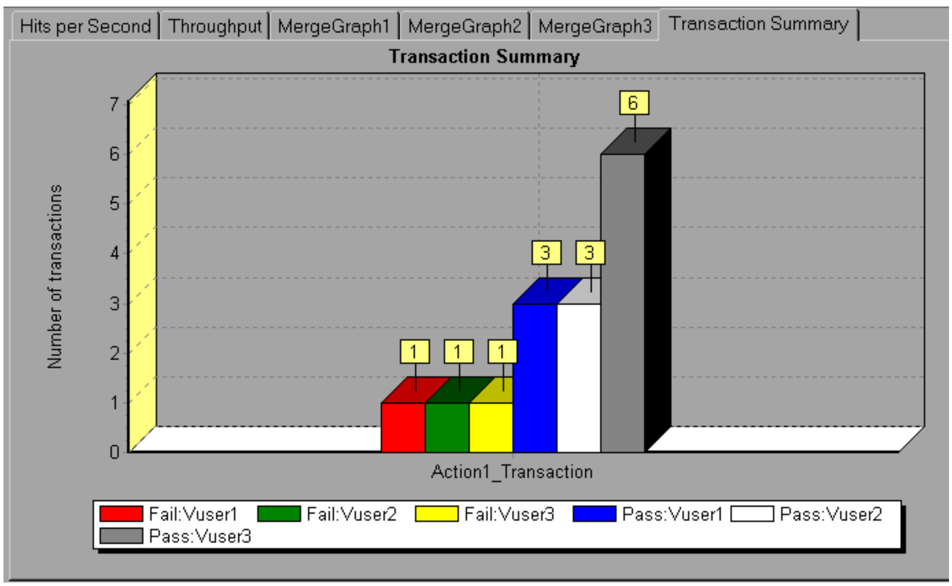
The available filter conditions differ for each type of graph. The filter conditions also depend on your scenario. For example, if you only had one group or one load generator machine in your scenario, the Group Name and Load Generator Name filter conditions do not apply.

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

Sorting Graph Data Overview

You can sort graph data to show the data in more relevant ways. For example, Transaction graphs can be grouped by the Transaction End Status, and Vuser graphs can be grouped by Scenario Elapsed Time, Vuser End Status, Vuser Status, and VuserID.

You can sort by one or several groups—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 grouping order by rearranging the list. The following graph shows the Transaction Summary grouped according to Vusers.



Reference

Filter Conditions

Common Filter Condition Options

The following filter conditions are common to many graphs:

Filter Condition	Filters the graph according to...
Host Name	The name of the Host machine. Select one or more hosts from the drop down list.
Transaction End Status	The end status of a transaction: *pass, fail, stop*.
Scenario Elapsed Time	The time that elapsed from the beginning to the end of the load test scenario. For more information about setting the time range, see "Scenario Elapsed Time Dialog Box" on page 113.
Vuser ID	The Vuser ID. For more information, see "Vuser ID Dialog Box" on page 115.
Script Name	The name of the script.
Group Name	The name of the group to filter by.
Think Time	The Think Time option in the graph filter for complete mode is turned off by default. The transaction time displayed shows pure time.

Vuser Graphs

You can apply the following filter conditions to Vuser graphs:

Filter Condition	Filters the graph according to...
Vuser Status	The Vuser status: *load, pause, quit, ready, run*
Vuser End Status	The status of the Vuser at the end of the transaction: *error, failed, passed, stopped.*
Number of Released Vusers	The number of Vusers that were released.
Rendezvous Name	The name of the rendezvous point.

Error Graphs

You can apply the following filter conditions to Error graphs:

Filter Condition	Filters the graph according to...
Error Type	The type of error (per error number).
Parent Transaction	The parent transaction.
Line Number in Script	The line number in the script.

Transaction Graphs

You can apply the following filter conditions to Transaction graphs:

Filter Condition	Filters the graph according to...
Transaction Name	The name of the transaction.
Transaction Response Time	The response time of the transaction.
Transaction Hierarchical Path	The hierarchical path of the transaction. For more information on setting this condition, see "Hierarchical Path Dialog Box" on page 112.

Web Resource Graphs

You can apply the following filter conditions to Web Resources graphs:

Filter Condition	Filters the graph according to...
Web Resource Name	The name of the Web resource.
Web Resource Value	The value of the Web resource.
Web Server Resource Name	The name of the Web Server resource.
Web Server Resource Value	The value of the Web Server resource.

Web Page Diagnostics Graphs

You can apply the following filter conditions to Web Page Diagnostics graphs:

Filter Condition	Filters the graph according to...
Component Name	The name of the component.
Component Response Time	The response time of the component.
Component DNS Resolution Time	The amount of time the component needs to resolve the DNS name to an IP address, using the closest DNS server.
Component Connection Time	The time taken for the component to establish an initial connection with the Web server hosting the specified URL.
Component First Buffer Time	The time that passes from the component's initial HTTP request (usually GET) until the first buffer is successfully received back from the Web server.
Component Receive Time	The time that passes until the component's last byte arrives from the server and the downloading is complete.

Filter Condition	Filters the graph according to...
Component SSL Handshaking Time	The time take for the component to establish an SSL connection. (Applicable to HTTPS communication only.)
Component FTP Authentication Time	The time taken for the component to authenticate the client. (Applicable to FTP protocol communication only).
Component Error Time	The average amount of time that passes from the moment a component's HTTP request is sent until the moment an error message (HTTP errors only) is returned.
Component Size (KB)	The size of the component (in kilobytes).
Component Type	The type of component: *Application; Image; Page; Text*
Component Hierarchical Path	The hierarchical path of the component. For more information on setting this condition, see "Hierarchical Path Dialog Box" on page 112.
Component Network Time	The amount of time from the component's first HTTP request, until receipt of ACK.
Component Server Time	The amount of time from when the component receives of ACK, until the first buffer is successfully received back from the Web server.
Component Client Time	The average amount of time that passes while a component request is delayed on the client machine due to browser think time or other client-related delays.

User Defined Data Point Graphs

You can apply the following filter conditions to User-Defined Data Point graphs:

Filter Condition	Filters the graph according to...
Datapoint Name	The name of the data point.
Datapoint Value	The value of the data point.

System Resources Graphs

You can apply the following filter conditions to System Resource graphs:

Filter Condition	Filters the graph according to...
System Resource Name	The name of the system resource.
System Resource Value	The value of the system resource. See "Set Dimension Information Dialog Box" on page 114.

Network Monitor Graphs

You can apply the following filter conditions to Network Monitor graphs:

Filter Condition	Filters the graph according to...
Network Path Name	The name of the network path.
Network Path Delay	The delay of the network path.
Network Path Father	The father of the network path.
Network SubPath Name	The name of the network subpath.
Network SubPath Delay	The delay of the network subpath.
Network Full Path	The full network path.
Network Segment Name	The name of the network segment.

Filter Condition	Filters the graph according to...
Network Segment Delay	The delay of the network segment.
Network Segment Full Path	The full network segment path.

Firewall Graphs

You can apply the following filter conditions to Firewall graphs:

Filter Condition	Filters the graph according to...
Firewall Resource Name	The name of the Firewall resource.
Firewall Resource Value	The value of the firewall resource. See "Set Dimension Information Dialog Box" on page 114.

Web Server Resource Graphs

You can apply the following filter conditions to Web Server Resource graphs:

Filter Condition	Filters the graph according to...
Measurement Name	The name of the measurement.
Measurement Value	The measurement value. See "Set Dimension Information Dialog Box" on page 114.

Web Application Server Resource Graphs

You can apply the following filter conditions to Web Application Server Resource graphs:

Filter Condition	Filters the graph according to...
Resource Name	The name of the resource.
Resource Value	The value of the resource. See "Set Dimension Information Dialog Box" on page 114.

Database Server Resource Graphs

You can apply the following filter conditions to Database Server Resource graphs:

Filter Condition	Filters the graph according to...
Database Resource Name	The name of the database resource.
Database Resource Value	The value of the database resource. See "Set Dimension Information Dialog Box" on page 114.

Streaming Media Graphs

You can apply the following filter conditions to Streaming Media graphs:

Filter Condition	Filters the graph according to...
Streaming Media Name	The name of the streaming media.
Streaming Media Value	The value of the streaming media. See "Set Dimension Information Dialog Box" on page 114.

ERP/CRM Server Resource Graphs

You can apply the following filter conditions to ERP/CRM Server Resource graphs:

Filter Condition	Filters the graph according to...
ERP/CRM Server Resource Name	The name of the ERP/CRM server resource.
ERP/CRM Server Resource Value	The value of the ERP/CRM Server resource. See "Set Dimension Information Dialog Box" on page 114.
ERP Server Resource Name	The name of the ERP server resource.
ERP Server Resource Value	The value of the ERP server resource. See "Set Dimension Information Dialog Box" on page 114.

Siebel Diagnostics Graphs

You can apply the following filter conditions to Siebel Diagnostics graphs:

Filter Condition	Filters the graph according to...
Siebel Transaction Name	The name of the Siebel transaction.
Siebel Request Name	The name of the Siebel request.
Siebel Layer Name	The name of the Siebel layer.
Siebel Area Name	The name of the Siebel area.
Siebel Sub-Area Name	The name of the Siebel sub-area.
Siebel Server Name	The name of the Siebel server.
Siebel Script Name	The name of the Siebel script.
Response Time	The response time of the Siebel transaction.
Siebel Chain of Calls	The chain of calls for the Siebel transaction.

Siebel DB Diagnostics Graphs

You can apply the following filter conditions to Siebel DB Diagnostics graphs:

Filter Condition	Filters the graph according to...
Transaction Name - SIEBEL	The name of the Siebel DB transaction.
SQL Chain of Calls	The SQL chain of calls for the Siebel DB transaction.
SQL Alias Name	The SQL alias name for the Siebel DB transaction.
SQL Response Time	The SQL response time of the Siebel DB transaction.

Oracle 11i Diagnostics Graphs

You can apply the following filter conditions to Oracle 11i Diagnostics graphs:

Filter Condition	Filters the graph according to...
Transaction Name - ORACLE	The name of the Oracle transaction.
SQL Chain of Calls	The SQL chain of calls for the Oracle transaction.
SQL Alias Name - Oracle	The SQL alias name for the Oracle transaction.
SQL Response Time	The SQL response time of the Oracle transaction.
Oracle SQL Parse Time	The SQL parse time of the Oracle transaction.
Oracle SQL Execute Time	The SQL execute time of the Oracle transaction.
Oracle SQL Fetch Time	The SQL fetch time of the Oracle transaction.
Oracle SQL Other Time	Other SQL time for the Oracle transaction.

Java Performance Graphs

You can apply the following filter conditions to Java Performance graphs:

Filter Condition	Filters the graph according to...
Java Performance Resource Name	The name of the Java performance resource.
Java Performance Resource Value	The value of the Java performance resource.

J2EE & .NET Diagnostics Graphs

You can apply the following filter conditions to J2EE & .NET Diagnostics graphs:

Filter Condition	Filters the graph according to...
Transaction Name	The name of the Java transaction.
Method Chain of Calls	The chain of calls for the Java method.
Layer Name	The name of the layer.
Class Name	The name of the class.
Method Name	The name of the method.
SQL Logical Name	The SQL logical name for the Java transaction.
Response Time	The response time of the Java transaction.
Host Name - J2EE/.NET	The name of the host for the J2EE & .NET transaction.
Application Host Name - (VM)	The name of the application host for the VM.
Transaction Request	The request for the transaction.
Transaction Hierarchical Path	The hierarchical path of the transaction. For more information on setting this condition, see "Hierarchical Path Dialog Box" on page 112.

Application Component Graphs

You can apply the following filter conditions to Application Component graphs:

Filter Condition	Filters the graph according to...
Component Resource Name	The resource name of the component.
Component Resource Value	The value of the component resource. See "Set Dimension Information Dialog Box" on page 114.

Filter Condition	Filters the graph according to...
COM+ Interface	The interface of the COM+ component.
COM+ Response Time	The response time of the COM+ component.
COM+ Call Count	The call count of the COM+ component.
COM+ Method	The method of the COM+ component.
.Net Resource Name	The resource name of the .NET component.
.Net Value	The .NET resource value. See "Set Dimension Information Dialog Box" on page 114.
.Net Class	The class of the .NET component.
.Net Response Time	The response time of the .NET component.
.Net Call Count	The call count of the .NET component.
.Net Method	The method of the .NET component.

Application Deployment Graphs

You can apply the following filter conditions to Application Deployment graphs:

Filter Condition	Filters the graph according to...
Citrix Resource Name	The name of the Citrix resource.
Citrix Resource Value	The value of the Citrix resource. See "Set Dimension Information Dialog Box" on page 114.

Middleware Performance Graphs

You can apply the following filter conditions to Middleware Performance graphs:

Filter Condition	Filters the graph according to...
Message Queue Resource Name	The name of the message queue resource.
Message Queue Resource Value	The value of the Message Queue resource. See "Set Dimension Information Dialog Box" on page 114.

Infrastructure Resource Graphs

You can apply the following filter conditions to Infrastructure Resource graphs:

Filter Condition	Filters the graph according to...
Network Client	The name of the network client.
Network Client Value	The value of the network client. See "Set Dimension Information Dialog Box" on page 114.

External Monitor Graphs

You can apply the following filter conditions to External Monitor graphs:

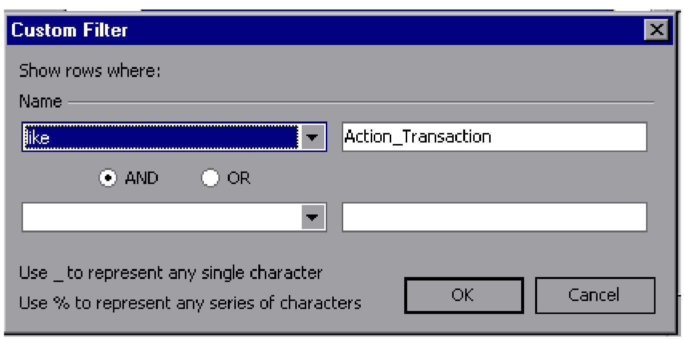
Filter Condition	Filters the graph according to...
External Monitor Resource Name	The name of the external monitor resource.
External Monitor Resource Value	The value of the external monitor resource. See "Set Dimension Information Dialog Box" on page 114.

Filter Conditions User Interface

This section includes (in alphabetical order):

Custom Filter Dialog Box

This dialog box enables you to customize your filter criteria.



To access	**View** menu > **Set Filter/Group by** > **Values for (discrete value)** > **Drop-down arrow from top of Select or Name column** > **(Custom...)**
Note	If you are specifying the start and end time for a transaction (in minutes:seconds format), the time is relative to the beginning of the load test scenario.

User interface elements are described below:

UI Elements	Description
Minimum	Specify a minimum value for the measurement.
Maximum	Specify a maximum value for the measurement.

Filter Dialog Boxes

The filter dialog boxes (Graph Settings, Global Filter, and Analysis Summary Filter) enable you to filter the data that is displayed in the graph or report.

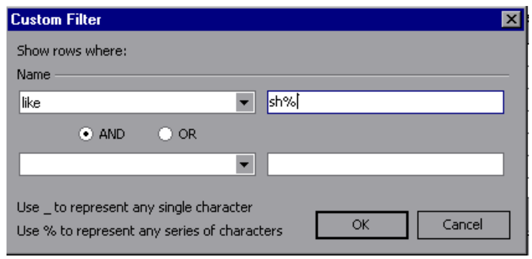
When adding a graph, the filter and sort button is displayed which enables you to filter and sort data before the graph is displayed.

To access	**Use one of the following:** ➤ **View** > **Set Filter/Group By** or click  ➤ **File** > **Set Global Filter** or click  ➤ **View** > **Summary Filter** or click 
Note	Some of the following fields are not displayed in all of the filter boxes.

User interface elements are described below:

UI Elements	Description
Filter Condition	Select criteria and values for each filter condition that you want to employ. The applicable filter conditions are displayed for each graph. For details on each graphs filter conditions, see the chapter on the relevant graph.
Criteria	Select "=" (equals) or "<>" (does not equal).

<table>
<tr><th>UI Elements</th><th>Description</th></tr>
<tr><td>Values</td><td>
The filter conditions are grouped into three value types (discrete, continuous, and time-based).
A discrete value is a distinct integer (whole number) or string value such as Transaction Name or Vuser ID. Select the check box(es) of the value(s) that you want to include in your filter. You can also customize your filter by entering wild cards to depict any single character or any series of characters.



A continuous value is a variable dimension that can take any value within the minimum and maximum range limits, such as Transaction Response Time. You set the dimension information for each measurement in the "Set Dimension Information Dialog Box" on page 114.
A time-based value is a value that is based on time relative to the start of the load test scenario. Scenario Elapsed Time is the only condition that uses time-based values. You specify time-based values in the "Scenario Elapsed Time Dialog Box" on page 113.

For some filter conditions, one of the following dialog boxes opens to enable you to specify additional filtering details:

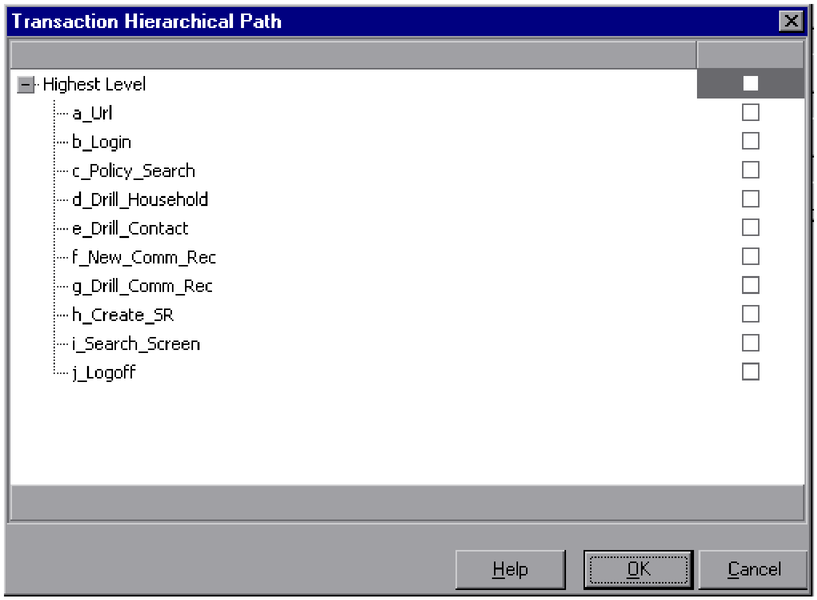
"Set Dimension Information Dialog Box" on page 114
"Vuser ID Dialog Box" on page 115
"Scenario Elapsed Time Dialog Box" on page 113
"Hierarchical Path Dialog Box" on page 112: Enables you to display the hierarchial path of a transaction or component, or a method chain of calls.

</td></tr>
</table>

UI Elements	Description
Transaction Percentile	The Summary Report contains a percentile column showing the response time of 90% of transactions (90% of transactions that fall within this amount of time). To change the value of the default 90 percentile, enter a new figure in the **Transaction Percentile** box.
Set Default	Displays the default criteria and values for each filter condition.
Clear All	Deletes all of the information you entered in the dialog box.
Group By settings	Use these settings to sort the graph display by grouping the data. You can group the data by: ➤ **Available groups**. Select the group by which you want to the sort the results, and click the right arrow. ➤ **Selected groups**. Displays a list of all the selected groups by which the results will be sorted. To remove a value, select it and click the left arrow.
Reset all graphs to their defaults prior to applying the Global Filter	All graphs filter settings are reverted to their default.

Hierarchical Path Dialog Box

This dialog box enables you to display the hierarchical path of a transaction or component, or a method chain of calls.



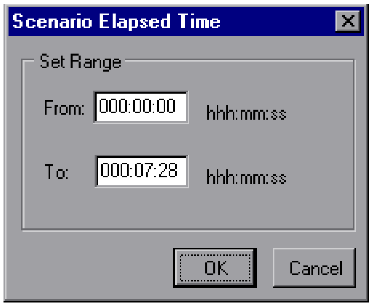
To access	**View** menu > **Set Filter/Group by** > **Filter condition** pane > **Transaction, Component Hierarchical Path or a method chain of calls.**

User interface elements are described below:

UI Elements	Description
Transaction, Component Hierarchical Path or a method chain of calls	Select the box for the path where you want to start to see results. Only the selected path and its immediate sub-nodes will be displayed.

Scenario Elapsed Time Dialog Box

This dialog box enables you to specify the start and end time range for the graph's x-axis.



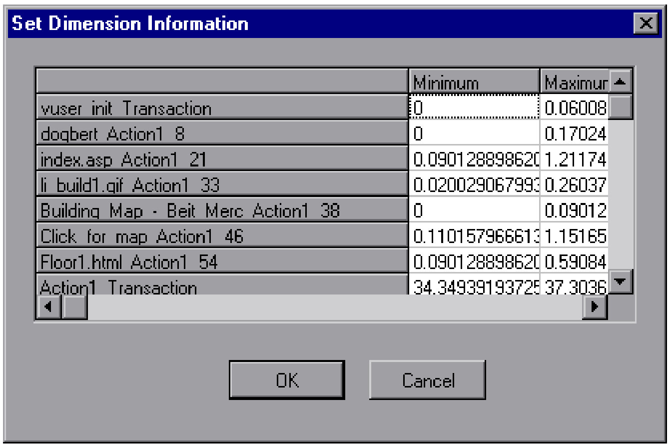
To access	**View** menu > **Set Filter/Group by** > **Filter condition** pane > **Scenario Elapsed Time**
Note	The time is relative to the start of the scenario.

User interface elements are described below:

UI Elements	Description
From	Specify a start value for the desired range.
To	Specify an end value for the desired range.

Set Dimension Information Dialog Box

This dialog box enables you to set the dimension information for each measurement (transaction, number of released Vusers, resource) in the result set. You specify the minimum and maximum values for each measurement you want in the analysis. By default, the full range of values for each measurement is displayed.



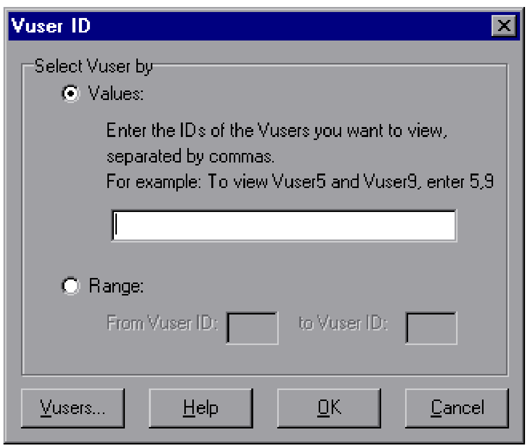
To access	You can open this dialog box from the following locations: ➤ **Transaction** graphs > **View** menu > **Set Filter/Group by** > **Filter condition** pane > **Transaction Response Time** ➤ **Vusers** graph > **Rendezvous** graph > **View** menu > **Set Filter/Group by** > **Filter condition** pane > **Number of Released Vusers** ➤ All graphs that measure resources (**Web Server**, **Database Server**, etc.) > **View** menu > **Set Filter/Group by** > **Filter condition** pane > **Resource Value**
Note	If you are specifying the start and end time for a transaction (in minutes:seconds format), the time is relative to the beginning of the load test scenario.

User interface elements are described below:

UI Elements	Description
Minimum	Specify a minimum value for the measurement.
Maximum	Specify a maximum value for the measurement.

Vuser ID Dialog Box

This dialog box opens to enable the entering of additional filter information for the Vuser ID filter condition.



To access	**View menu** > **Set Filter/Group by** > **Filter condition pane** > **Vuser ID**

User interface elements are described below:

UI Elements	Description
Value	Enter the Vuser IDs of the Vusers you want the graph(s) to display, separated by commas.
Range	Specify the beginning and end of the desired range of Vusers you want the graph(s) to display
Vusers	Displays the existing Vuser IDs from which you can choose.

5

Working with Analysis Graph Data

Analysis contains several utilities that enable you to manage graph data to most effectively view the displayed data.

This chapter includes:

Concepts

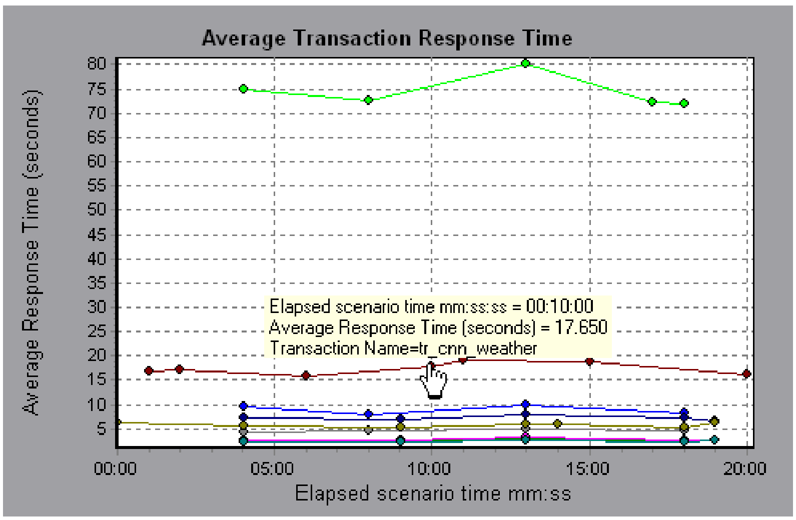
Tasks

Reference

Concepts

Determining a Point's Coordinates

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

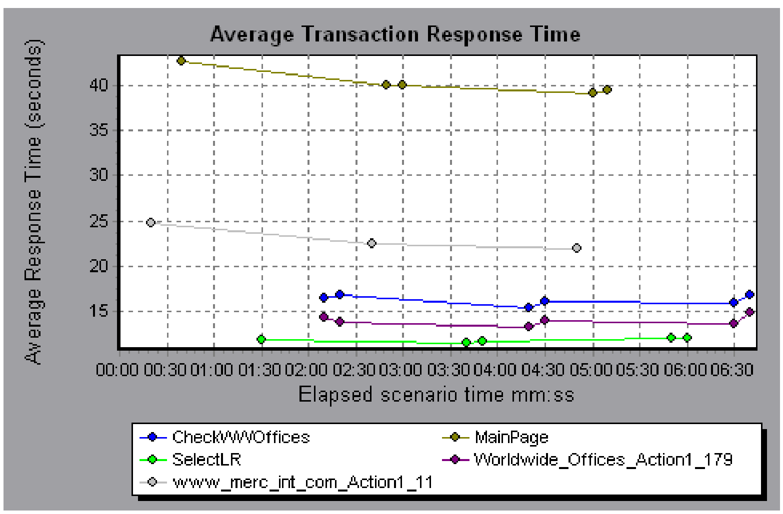


Drilling Down in a Graph

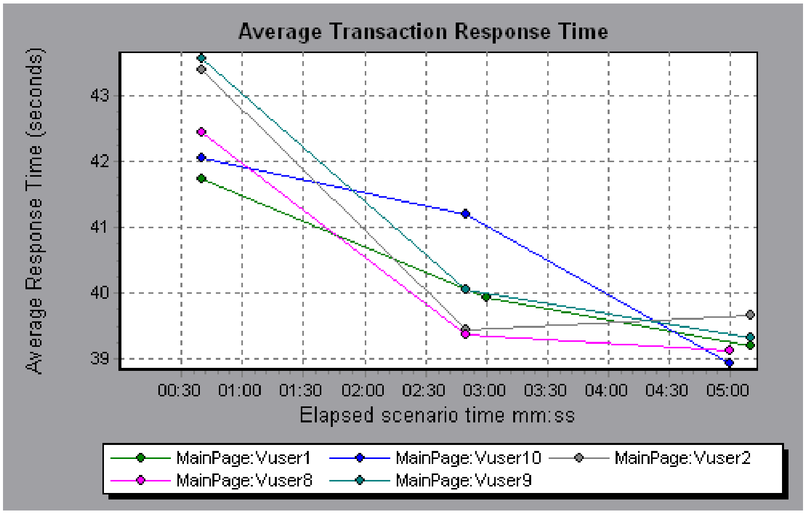
Drill down enables you to focus on a specific measurement within your graph and display it according to a desired grouping. The available groupings depend on the graph. For example, the Average Transaction Response Time graph shows one line per transaction. To determine the response time for each Vuser, 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 Diagnostics graph.

The following graph shows a line for each of five transactions.



When you drill down on the MainPage transaction, grouped by 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.

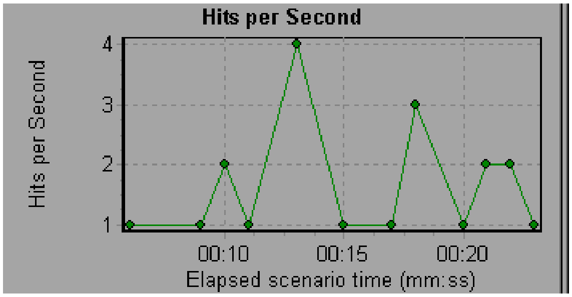
To determine the response time for each host, you drill down on one transaction and sort it according to host. The graph displays a separate line for the transaction response time on each host.

Changing the Granularity of the Data

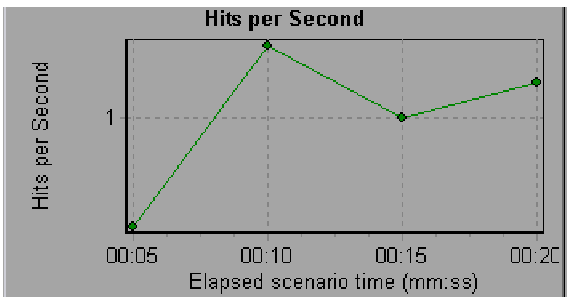
You can make the graphs easier to read and analyze by changing the granularity (scale) of the *x-axis*. The maximum granularity is half of the graph's time range. To ensure readability and clarity, Analysis automatically adjusts the minimum granularity of graphs with ranges of 500 seconds or more.

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 load test scenario.

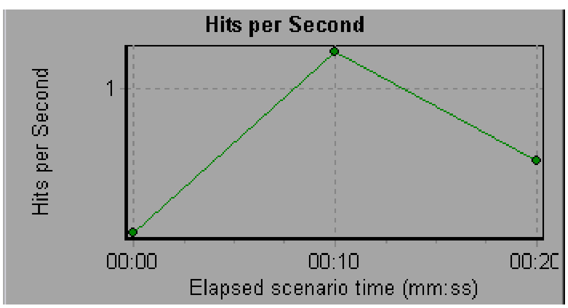
For a granularity of 5, the y-axis shows the number of hits per second for every five-second period of the scenario.



GRANULARITY=1



GRANULARITY=5



GRANULARITY=10

In the above graphs, the same load test 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 Vuser 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.

Viewing Measurement Trends

You can view the pattern of a line graph more effectively by standardizing the graph's y-axis values. Standardizing a graph causes the graph's y-axis values to converge around zero. This cancels the measurements' actual values and allows you to focus on the behavior pattern of the graph during the course of the load test scenario.

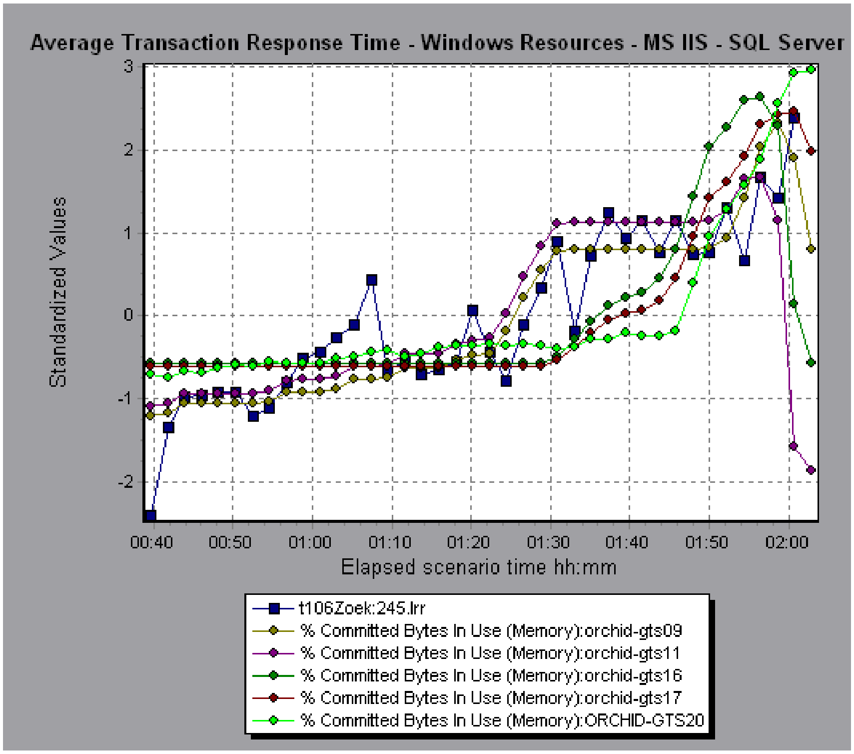
Analysis standardizes the y-axis values in a graph according to the following formula:

New Y value = (Previous Y Value - Average of previous values) / STD of previous values

Auto Correlating Measurements

You can detect similar trends among measurements by correlating a measurement in one graph with measurements in other graphs. Correlation cancels the measurements' actual values and allows you to focus on the behavior pattern of the measurements during a specified time range of the load test scenario.

In the following example, the **t106Zoek:245.lrr** measurement in the Average Transaction Response Time graph is correlated with the measurements in the Windows Resources, Microsoft IIS, and SQL Server graphs. The five measurements most closely correlated with **t106Zoek:245.lrr** are displayed in the graph below.



Note: This feature can be applied to all line graphs except the Web Page Diagnostics graph.

Viewing Raw Data

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 in the following cases:

➤ To determine specific details about a peak—for example, which Vuser was running the transaction that caused the peak value(s).

➤ To perform a complete export of unprocessed data for your own spreadsheet application.

For user interface details, click "Graph Data View Table" on page 136.

Tasks

How to Manage Graph Data

The following list includes the utilities you can use in Analysis to enable you to manage graph data to most effectively view the displayed data.

Determine a point's coordinates

To determine the coordinates and values at any point in a graph, place the cursor over the point you want to evaluate. Analysis displays the axis values and other grouping information.

Drill down in a graph

Drill down enables you to focus on a specific measurement within your graph and display it according to the desired grouping.

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**. Analysis drills down and displays the new graph.

   To undo the last drill down settings, choose **Undo Set Filter/Group By** from the right-click menu.

   ➤ To perform additional drill-downs, repeat steps 1 to 4.

   ➤ To clear all filter and drill down settings, choose **Clear Filter/Group By** from the right-click menu.

Change the granularity of the data

The task describes how to change the granularity of a graph.

1 Click inside a graph.

2 Select **View** > **Set Granularity**, or click **Set Granularity**. The Granularity dialog box opens.

3 Enter the granularity of the x-axis and select a time measurement. The maximum granularity is half of the graph's time range.

4 To ensure readability and clarity, LoadRunner automatically adjusts the minimum granularity of graphs with ranges of 500 seconds or more.

5 Click **OK**.

View measurement trends

This task describes how to activate the View Measurements Trends option from a line graph.

1 Select **View** > **View Measurement Trends**, or right-click the graph and choose **View Measurement Trends**. Alternatively, you can select **View** > **Configure Measurements** and check the View measurement trends for all measurements box.

Note: The standardization feature can be applied to all line graphs except the Web Page Diagnostics graph.

---

2 View the standardized values for the line graph you selected. The values in the Minimum, Average, Maximum, and Std. Deviation legend columns are real values.

To undo the standardization of a graph, repeat step 1.

Note: If you standardize two line graphs, the two y-axes merge into one y-axis.

Auto correlate measurements

You can detect similar trends among measurements by correlating a measurement in one graph with measurements in other graphs. Correlation cancels the measurements' actual values and allows you to focus on the behavior pattern of the measurements during a specified time range of the load test scenario.

1 From a graph or legend, right-click the measurement you want to correlate and choose **Auto Correlate**. The Auto Correlate dialog box opens with the selected measurement displayed in the graph.

2 Select a suggested time range method and time range.

3 If you applied a time filter to your graph, you can correlate values for the complete scenario time range by clicking the **Display** button in the upper right-hand corner of the dialog box.

4 To specify the graphs you want to correlate with a selected measurement and the type of graph output to be displayed, perform the following:

- ➤ Select the **Correlation Options** tab.
- ➤ Select the graphs to correlate, the data interval, and output options, as described in "Drill Down Options Dialog Box" on page 131.
- ➤ On the **Time Range** tab, click **OK**. Analysis generates the correlated graph you specified. Note the two new columns—**Correlation Match** and **Correlation**—that appear in the Legend window below the graph.

To specify another measurement to correlate, select the measurement from the Measurement to Correlate box at the top of the Auto Correlate dialog box.

The minimum time range should be more than 5% of the total time range of the measurement. Trends which are smaller than 5% of the whole measurement will be contained in other larger segments.

Sometimes, very strong changes in a measurement can hide smaller changes. In cases like these, only the strong change is suggested, and the **Next** button will be disabled.

Note: This feature can be applied to all line graphs except the Web Page Diagnostics graph.

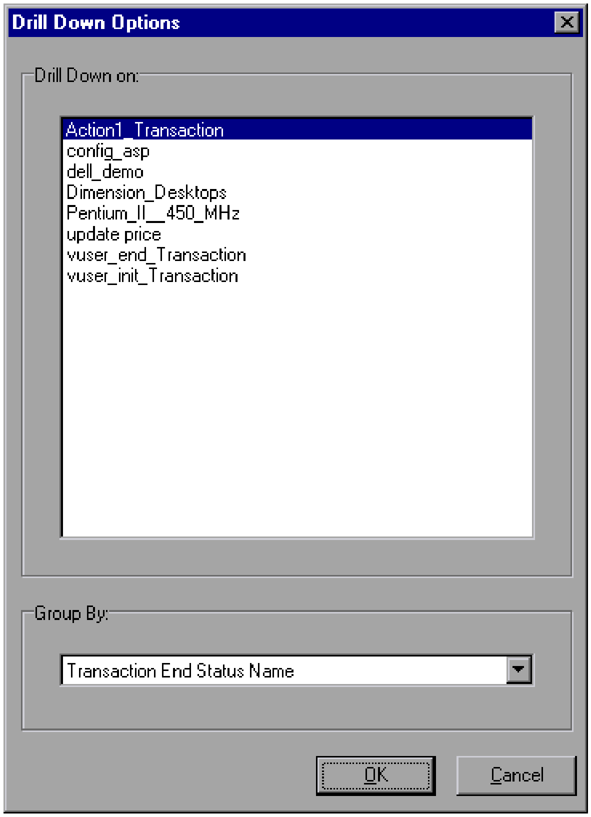
Reference

Analysis Graph Data User Interface

This section includes:

Drill Down Options Dialog Box

This dialog box lists all the measurements in the graph.



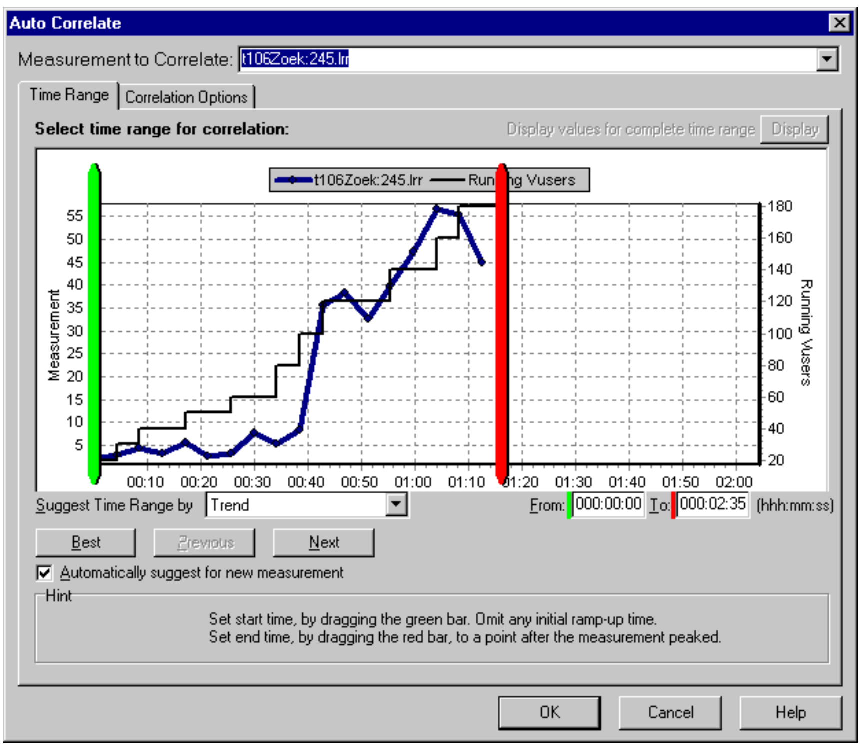
To access	<Right-click> graph line/bar/segment > **Drill Down**
See also	"Drilling Down in a Graph" on page 119

User interface elements are described below:

UI Elements (A-Z)	Description
Drill Down on	Filter graph by selected transaction.
Group By	The selected transaction is sorted by selected criteria.

Auto Correlate Dialog Box

This dialog box enables you to configure settings used to correlate measurements from the selected graph with measurements in other graphs.



To access	Use one of the following; Right click on graph > **Auto Correlate** Right click on graph > **Auto Correlate** > **Time Range** tab Right click on graph > **Auto Correlate** > **Correlation Options** tab
Important information	You can also use the green and red vertical drag bars to specify the start and end values for the scenario time range.
Note	The granularity of the correlated measurements graph may differ from that of the original graph, depending on the scenario time range defined.
See also	"Auto Correlating Measurements" on page 123

Time Range Tab

The Time Range tab of the Auto Correlate dialog box enables you to specify a load test scenario time range for the correlated measurement graph.

User interface elements are described below:

UI Elements	Description
Measurement to Correlate.	Select the measurement you want to correlate.
Display values for complete time range.	Click **Display** to correlate values for the complete scenario time range. This option is available only if you applied a time filter to your graph.
Suggest Time Range By	Analysis automatically demarcates the most significant time period for the measurement in the scenario. ➤ **Trend.** Demarcated an extended time segment which contains the most significant changes. ➤ **Feature.** Demarcates a smaller dimension segment which forms the trend.

UI Elements	Description
Best	Choose the time segment most dissimilar to its adjacent segments.
Next	Suggest the next time segment for auto correlation. Each suggestion is successively less dissimilar.
Previous	Return to the previous suggestion of a time segment.
Automatically suggest for new measurement	Generates new suggestions each time that the Measurement to Correlate item changes.
From	Specify a start value (in hh:mm:ss format) for the desired scenario time range.
To	Specify an end value (in hh:mm:ss format) for the desired scenario time range.

Correlation Options tab

You use the Correlation Options tab of the Auto Correlate dialog box to set the graphs to correlate, the data interval, and the output options.

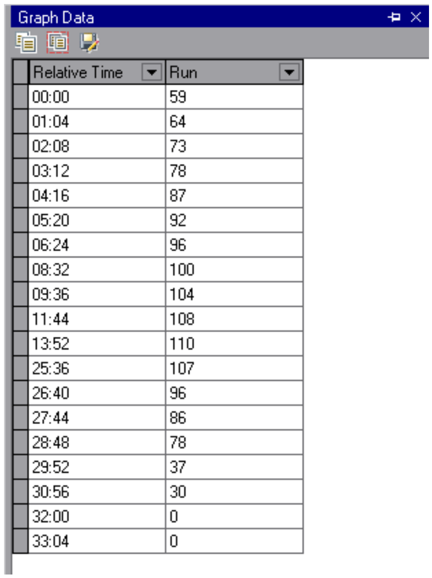
User interface elements are described below:

UI Elements	Description
Select Graphs for Correlation	Select the graphs whose measurements you want to correlate with your selected measurement.

UI Elements	Description
Data Interval	Calculate the interval between correlation measurement polls. ➤ **Automatic**. Uses an automatic value, determined by the time range. ➤ **Correlate data based on X second intervals**. Enter a fixed value.
Output	Choose the level of output displayed. ➤ **Show the X most closely correlated measurements.** Displays only the specified number of measurements most closely related to the selected measurement. The default setting is 5. ➤ **Show measurements with an influence factor of at least X%.** Displays only those measurements that converge to the specified percent with the selected measurement. The default setting is 50%.

Graph Data View Table

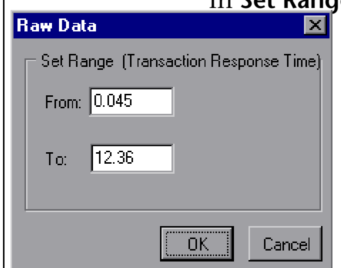
You can view graph data in spreadsheet view or raw data view. The data is instantly displayed on request.


Graph Data

Relative Time	Run
00:00	59
01:04	64
02:08	73
03:12	78
04:16	87
05:20	92
06:24	96
08:32	100
09:36	104
11:44	108
13:52	110
25:36	107
26:40	96
27:44	86
28:48	78
29:52	37
30:56	30
32:00	0
33:04	0

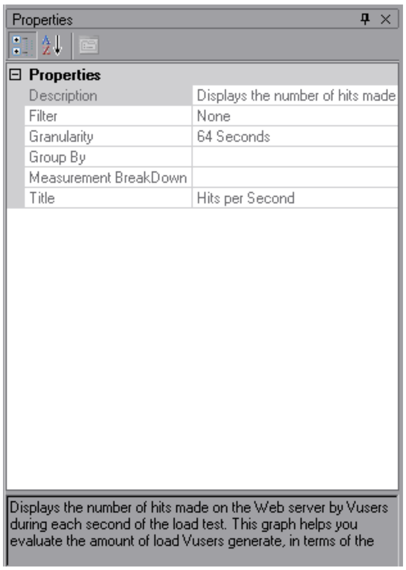
To access	Use one of the following: **Windows** > **Graph Data** **Windows** > **Raw Data**
Note	Raw Data is not available for all graphs.

User interface elements are described below:

UI Elements	Description
	Copies the data that you have selected.
	Copies the spreadsheet to the clipboard. You can paste to a spreadsheet.
	Saves the spreadsheet data to an Excel file. Once you have the data in Excel, you can generate your own customized graphs.
	Use the buttons on the toolbar to navigate through the table, and mark any records for future reference.
Relative Time	The first column in the Graph Data window. displays the elapsed scenario time (the x-axis values). The following columns displays the relative y-axis values for each measurement represented on the graph.
Raw Data dialog box	In **Set Range**, set a time range. 

Graph Properties Window

This window displays the details of the graph or report selected in the Session Explorer. Fields that appear in black are editable. When you select an editable field, an edit button is displayed next to the selected field value.



To access	**Windows** > **Properties**

User interface elements are described below:

UI Elements	Description
	Enables you to edit the value for the selected field.
Graph fields	➤ **Filter**. Shows configured filter. ➤ **Granularity.** Shows configured granularity. ➤ **Group By**. Shows the filter for selected group. ➤ **Measurement Breakdown**. Shows the measurements of the graph. ➤ **Title**. Shows the name of the graph in the graph display window.
Summary Report fields	➤ **Description**. An short summary of what is included in the summary report. ➤ **Filter**. Shows configured filter for the summary report. ➤ **Percentile.** The Summary Report contains a percentile column showing the response time of 90% of transactions (90% of transactions that fall within this amount of time). To change the value of the default 90 percentile, enter a new figure in the **Transaction Percentile** box. ➤ **Title**. The name of the summary report.
Transaction Analysis Report fields	When clicking the edit button for some of the fields, the Analyze Transaction Settings dialog box opens, enabling you to edit some of the Analyze Transaction settings.

6

Viewing Load Test Scenario Information

This chapter includes:

Concepts

Tasks

Reference

Concepts

Viewing Load Test Scenario Information

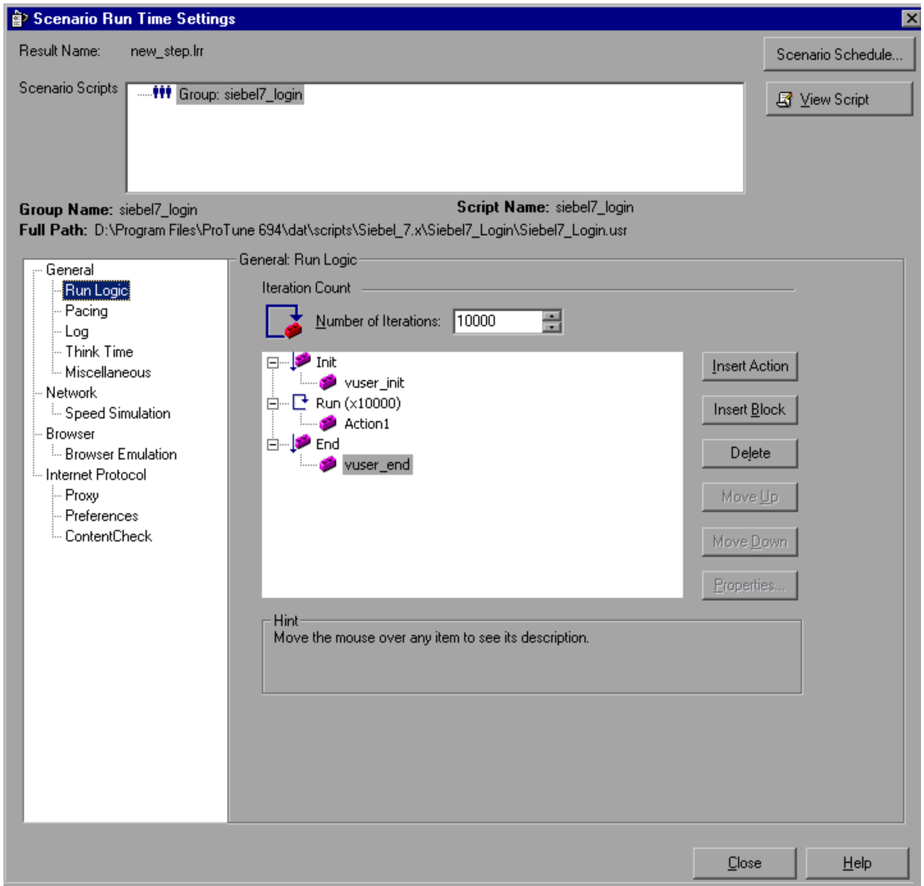
In Analysis, you can view information about the load test scenario which you are analyzing. You can view the scenario run time settings and output messages that were generated by the Controller during the scenario.

You can view information about the Vuser groups and scripts that were run in each scenario, as well as the run time settings for each script in a scenario, in the Scenario Run Time Settings dialog box.

Note: The run time settings allow you to customize the way a Vuser script is executed. You configure the run time settings from the Controller or Virtual User Generator (*VuGen*) before running a scenario. For information on configuring the run time settings, refer to the *HP Virtual User Generator User Guide*.

Choose **File** > **View Scenario Run Time Settings**, or click **View Run Time Settings** on the toolbar.

The Scenario Run Time Settings dialog box opens, displaying the Vuser groups, scripts, and scheduling information for each scenario. For each script in a scenario, you can view the run time settings that were configured in the Controller or VuGen before scenario execution.



Tasks

How to Configure Controller Output Messages Settings

This task describes how to configure settings for output messages.

1 Choose **Tools** > **Options** and select the **Result Collection** tab.

2 In the **Copy Controller Output Messages to Analysis Session** area, choose one of the following options:

- **Copy if data set is smaller than X MB.** Copies the Controller output data to the Analysis session if the data set is smaller than the amount you specify.
- **Always Copy.** Always copies the Controller output data to the Analysis session.
- **Never Copy.** Never copies the Controller output data to the Analysis session.

3 Apply your settings.

- To apply these settings to the current session, click **Apply now to active session**.
- To apply these settings after the current session is saved, click **OK**.

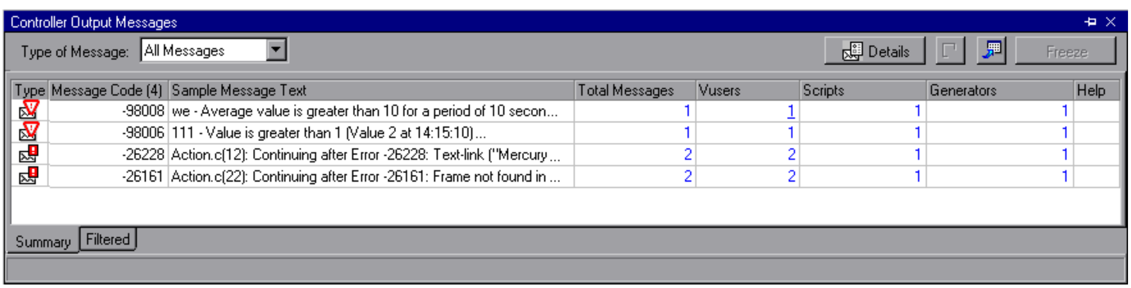
Reference

Load Test Scenario User Interface

This section includes (in alphabetical order):

Controller Output Messages Window

This window displays error, notification, warning, debug, and batch messages that are sent to the Controller by the Vusers and load generators during a scenario run.



To access	**Window** > **Controller Output Messages**
Important information	➤ The Summary tab is displayed by default when you open this window. ➤ Analysis searches for the output data in the current Analysis session. If the data is not found, it searches in the scenario results directory. If Analysis cannot locate the results directory, no messages are displayed.

User interface elements are described below:

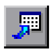
UI Elements	Description
Summary Tab	See "Summary Tab" on page 146
Filtered Tab	See "Filtered Tab" on page 149

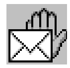
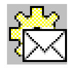
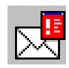
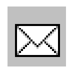
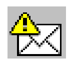
Summary Tab

This tab displays summary information about the messages sent during a scenario run.

To access	**Controller Output Messages** window > **Summary** tab
Important Information	You can drill down further on any information displayed in blue.
Parent topic	"Controller Output Messages Window" on page 145
See also	"Filtered Tab" on page 149

User interface elements are described below:

UI Elements	Description
	Displays the full text of the selected output message in the Detailed Message Text area at the bottom of the Output window.
	**Remove all messages.** Clears all log information from the Output window.
	**Export the view.** Saves the output to a specified file.
 	➤ **Freeze.** Stops updating the Output window with messages. ➤ **Resume.** Resumes updating the Output window with messages. The newly updated log information is displayed in a red frame.
Detailed Message Text	Displays the full text of the selected output message when you click the **Details** button.
Generators	Displays the number of load generators that generated messages with the specified message code.
Help	Displays an icon if there is a link to troubleshooting for the message.
Message Code	Displays the code assigned to all similar messages. The number in parentheses indicates the number of different codes displayed in the Output window.
Sample Message Text	Displays an example of the text of a message with the specified code.
Scripts	Displays the number of scripts whose execution generated messages with the specified code.
Total Messages	Displays the total number of sent messages with the specified code.

UI Elements	Description
Type	The type of message being displayed. The following icons indicate the various message types. For more information about each type, see **Type of Message** below: ➤  Batch ➤  Debug ➤  Errors ➤  Notifications ➤  Warnings ➤  Alerts
Type of Message	Filters the output messages to display only certain message types. Select one of the following filters: ➤ **All messages.** Displays all message types. ➤ **Batch.** Sent instead of message boxes appearing in the Controller, if you are using automation. ➤ **Debug.** Sent only if the debugging feature is enabled in the Controller. (Expert mode: **Tools** > **Options** > **Debug Information**). For more information, see "Options > Debug Information Tab" on page 267. ➤ **Errors.** Usually indicate that the script failed. ➤ **Notifications.** Provides run-time information, such as message sent using **lr_output_message**. ➤ **Warnings.** Indicates that the Vuser encountered a problem, but the scenario continued to run. ➤ **Alerts**. Indicates a warning.
Vusers	Displays the number of Vusers that generated messages with the specified code.

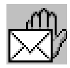
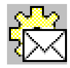
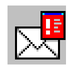
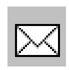
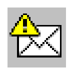
Filtered Tab

This tab displays a drilled down view by message, Vuser, script, or load generator. For example, if you drill down on the Vuser column, the Filtered tab displays all the messages with the code you selected, grouped by the Vusers that sent the messages.

To access	**Controller Output Messages** window > **Summary** tab. Click the blue link on the column that you wish to view more information about.
Important information	The tab is appears when you click on a blue link in the Summary tab.
Parent topic	"Controller Output Messages Window" on page 145
See also	"Summary Tab" on page 146

User interface elements are described below:

UI Elements	Description
	**Previous/Next View.** Enables you to move between the various drill down levels.
	Displays the full text of the selected output message in the Detailed Message Text area at the bottom of the Output window.
	**Export the view.** Saves the output to a specified file.
	Refreshes the Filtered tab with new log information that arrived in the Output window updated in the Summary tab.
<Message icon>	Displays an icon indicating the type of message by which the current Output view is filtered.
Active Filter	Displays the category or categories by which the current Output view is filtered.

UI Elements	Description
Viewed By	Displays the name of the column on which you selected to drill down. The following icons indicate the various message types: ➤  Batch ➤  Debug ➤  Errors ➤  Notifications ➤  Warnings ➤  Alerts
Detailed Message Text	Displays the full text of the selected output message when the **Details** button is selected.
Message	Displays all instances of the sample message text.
Script	The script on which the message was generated. If you click the blue link, VuGen opens displaying the script.
Action	The action in the script where the message was generated. If you click the blue link, VuGen opens the script to the relevant action.
Line #	The line in the script where the message was generated. If you click the blue link, VuGen opens the script and highlights the relevant line.
# Lines	The total number of lines in the script where the Vuser failed.
Time	The time the message was generated.
Iteration	The iteration during which the message was generated.
Vuser	The Vuser that generated the message.

UI Elements	Description
Generator	The load generator on which the message was generated. If you click the blue link, the Load Generator dialog box opens.
# Messages	The number of messages generated by a specific Vuser.

Scenario Run Time Settings Dialog Box

This dialog box enables you to view information about executed load test scenarios, as well as the run time settings for each script in a scenario.

To access	Toolbar >
See also	"Viewing Load Test Scenario Information" on page 142

User interface elements are described below

UI Elements	Description
Result Name	The name of the result file.
Scenario Scripts	Displays the result set for each executed scenario, as well as the Vuser groups and scripts that were run in the scenario.
Group Name	Displays the name of the group to which the selected script belongs.
Full Path	Displays the script's full directory path.
Script Name	Displays the name of the selected script.
Scenario Schedule	Displays goal-oriented or manual scenario scheduling information for the selected scenario.
View Script	Opens the Virtual User Generator, so that you can edit the script. For more information on editing scripts, refer to the *HP Virtual User Generator User Guide*.

7

Cross Result and Merged Graphs

This chapter includes:

Concepts

Tasks

Reference

Concepts

Cross Result and Merged Graphs Overview

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

Cross Result Graphs Overview

Cross Result graphs are useful for:

- Benchmarking hardware
- Testing software versions
- Determining system capacity

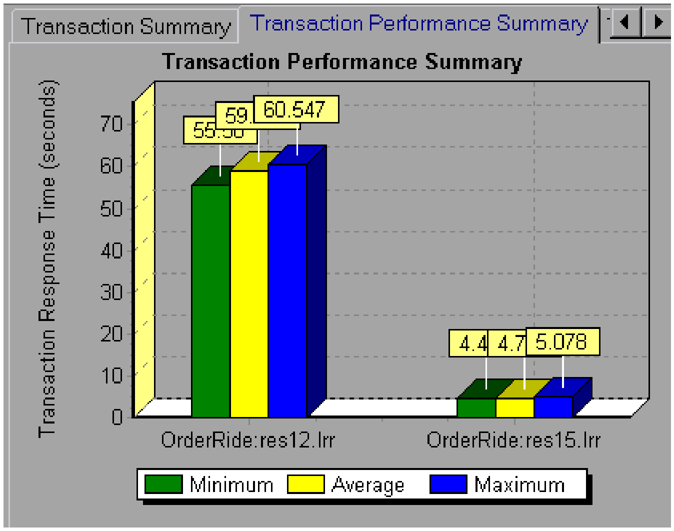
If you want to benchmark two hardware configurations, you run the same load test scenario with both configurations and compare the transaction response times using a single Cross Result graph.

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**.

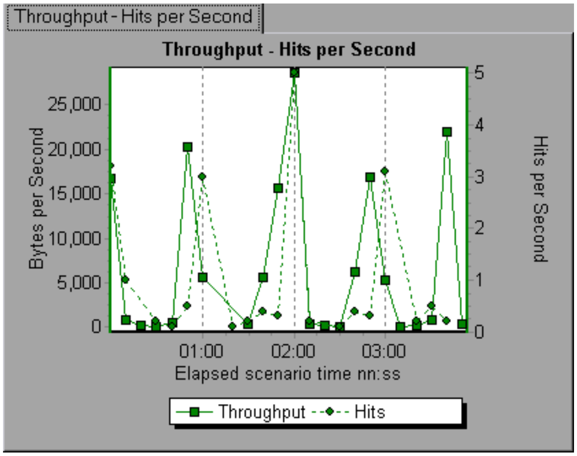
Merging Types Overview

Analysis provides three types of merging:

Overlay

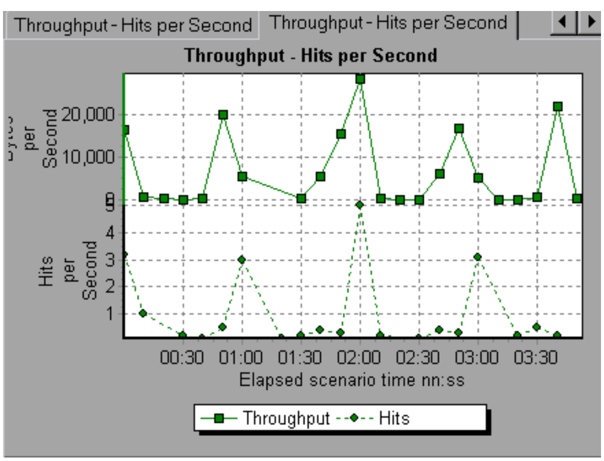
Superimpose the contents of two graphs that share a common x- axis. The left y-axis on the merged graph shows the current graph's values. The right y-axis shows the values of the graph that was merged. There is no limit to the number of graphs that you can overlay. When you overlay two graphs, the y-axis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

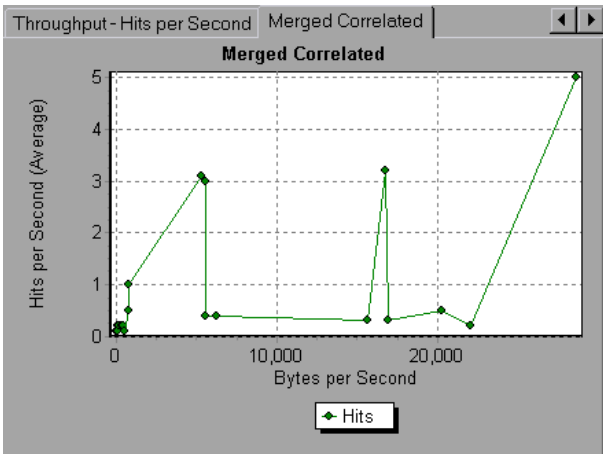
View contents of two graphs that share a common x-axis in a tiled layout, one above the other. In the following example the Throughput and Hits per Second graph are tiled one above the other.



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 average hits per second.



Tasks

How to Generate Cross Results Graphs

This task describes how to create a Cross Result graph for two or more result sets. The Cross Result dialog box enables you to compare the results of multiple load test scenario runs.

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** box.
6. Click **OK**. Analysis processes the result data and asks for a confirmation to open the default graphs.

Note: When generating a Cross Results Session, verify that the transaction names do not contain a <_> or <@> symbol. This will cause errors to occur when attempting to open the Cross Results 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.

You can view a summary report for the Cross Result graph.

How to Generate Merged Graphs

This task describes how to merge the results of two graphs from the same load test 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.

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

1. Select a graph in the Session Explorer or select its tab to make it active.
2. Choose **View** > **Merge Graphs** or click **Merge Graphs**. 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 merge type and a title for the merged graph. By default, Analysis combines the titles of the two graphs being merged. For more information, see "Merge Graphs Dialog Box" on page 160.
5. Click **OK**.
6. Filter the graph just as you would filter any ordinary graph.

Reference

Merge Graphs User Interface

This section includes:

Merge Graphs Dialog Box

This dialog box enables you to merge two graphs into a single graph.

To access	**View** > **Merge Graphs**
Important Information	In order to merge graphs, the graphs' x-axes must be the same measurement. For example, you can merge Web Throughput and Hits per Second graphs, because their x-axes are Scenario Elapsed Time.
See Also	"Merging Types Overview" on page 155

User interface elements are described below:

UI Elements (A-Z)	Description
Select Graph to merge with	The drop-down list shows all of the open graphs that share a common x-axis measurement with the current graph. Select one of the graphs in the list.
Select type of merge	➤ **Overlay**. View 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 with the current graph. ➤ **Tile**. View contents of two graphs that share a common x-axis in a tiled layout, one above the other. ➤ **Correlate**. Plot the y-axes 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.
Title of Merged Graph	Enter a title for the merged graph. This title will appear in the Session Explorer (**Windows** > **Session Explorer**).

8

Defining Service Level Agreements

This chapter includes:

Concepts

Tasks

Reference

Concepts

Service Level Agreements Overview

Service level agreements (SLAs) are specific goals that you define for your load test scenario. After a scenario run, HP LoadRunner Analysis compares these goals against performance related data that was gathered and stored during the course of the run, and determines whether the SLA passed or failed.

Depending on the measurements that you are evaluating for your goal, LoadRunner determines the SLA status in one of the following ways:

SLA Type	Description
SLA status determined at time intervals over a timeline	Analysis displays SLA statuses at set time intervals over a timeline within the run. At each time interval in the timeline—for example, every 10 seconds—Analysis checks to see if the measurement's performance deviated from the threshold defined in the SLA. Measurements that can be evaluated in this way: ➤ **Transaction Response Time (Average) per time interval** ➤ **Errors per Second per time interval**
SLA status determined over the whole run	Analysis displays a single SLA status for the whole scenario run. Measurements that can be evaluated in this way: ➤ **Transaction Response Time (Percentile) per run** ➤ **Total Hits per run** ➤ **Average Hits (hits/second) per run** ➤ **Total Throughput (bytes) per run** ➤ **Average Throughput (bytes/second) per run**

You can define and edit SLAs in the Controller or in Analysis.

Tracking Period

When you define an SLA for measurements that are evaluated over a timeline, Analysis determines SLA statuses at specified time intervals within that timeline. The frequency of the time intervals is called the **tracking period**.

An internally-calculated tracking period is defined by default. You can change the tracking period by entering a value in the Advanced Options dialog box which Analysis plugs into a built-in algorithm to calculate the tracking period. For details, see "Advanced Options Dialog Box" on page 173.

Tasks

How to Define Service Level Agreements

This task describes how to define service level agreements (SLAs).

You can define service level agreements (SLAs) which measure scenario goals over time intervals, or over a whole scenario run. For details, see "Service Level Agreements Overview" on page 164.

Tip: For a use-case scenario related to this task, see "How to Define Service Level Agreements - Use-Case Scenario" on page 168.

This task includes the following steps:

- "Prerequisites" on page 167
- "Run through the SLA wizard" on page 167
- "Define a tracking period - optional" on page 167
- "Results" on page 168

1 Prerequisites

If you are defining an SLA for Average Transaction Response Time, your scenario must include a script that contains at least one transaction.

2 Run through the SLA wizard

In the Service Level Agreement pane, click **New** to open the Service Level Agreement wizard. For user interface details, see "Service Level Agreement Wizard" on page 176.

a Select a measurement for the SLA.

b If you are defining an SLA for Average Transaction Response Time or Transaction Response Time (Percentile), select the transactions to include in your goal.

c (Optional) When evaluating SLA statuses over a timeline, select load criteria to take into account and define appropriate load value ranges for the load criteria. For an example, see "How to Define Service Level Agreements - Use-Case Scenario" on page 168.

d Set thresholds for the measurements.

- If the **Average Transaction Response Time** or **Errors per Second** exceed the defined thresholds, Analysis will produce a **Failed** SLA status.
- If **Transaction Response Time(Percentile)**, **Total Hits per run**, **Average Hits (hits/second) per run**, **Total Throughput (bytes) per run**, or **Average Throughput (bytes/second) per run** are lower than the defined threshold, Analysis will produce a **Failed** SLA status.

3 Define a tracking period - optional

For measurements whose SLA statuses are determined over time intervals, you need to define the frequency of the time intervals, that is, the **tracking period**. For details, see "Tracking Period" on page 165.

For user interface details, see "Advanced Options Dialog Box" on page 173.

4 Results

When analyzing your scenario run, HP LoadRunner Analysis compares the data collected from the scenario run against the SLA settings, and determines SLA statuses which are included in the default Summary Report.

How to Define Service Level Agreements - Use-Case Scenario

This use-case scenario describes how to define a service level agreement (SLA) for Average Transaction Response Time.

This scenario includes the following steps:

➤ "Background" on page 168

➤ "Start the SLA wizard" on page 168

➤ "Select the measurement for the SLA" on page 169

➤ "Select the transactions to evaluate in your goal" on page 169

➤ "Select a load criterion and define appropriate ranges of load - optional" on page 169

➤ "Set thresholds" on page 171

➤ "Results" on page 172

1 Background

The administrator of HP Web Tours would like to know when the average transaction response time for booking a flight and searching for a flight exceeds a certain value. Assume that your scenario includes a script that includes the following transactions: **book_flight** and **search_flight**.

2 Start the SLA wizard

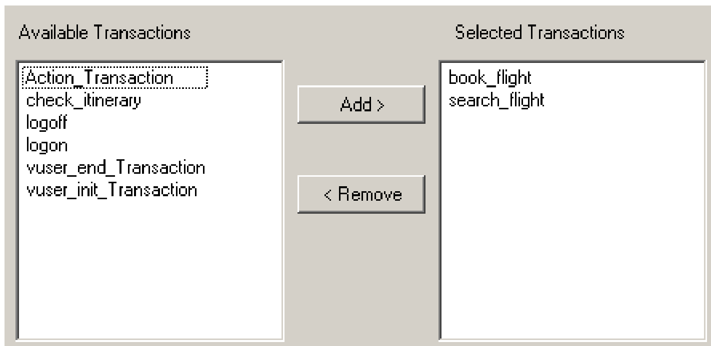
In the Service Level Agreement pane, click **New** to open the Service Level Agreement wizard.

3 Select the measurement for the SLA

On the Select a Measurement page, under **Select a Measurement for Your Goal,** in the **Transaction Response Time** box, select **Average**.

4 Select the transactions to evaluate in your goal

On the Select a Transaction page, select the transactions to be evaluated: **book_flight** and **search_flight**.



5 Select a load criterion and define appropriate ranges of load - optional

On the Select Load Criteria page, select the load criterion to take into account when evaluating the average transaction response time.

In this case, to see the effect that various quantities of Vusers running on the system has on the average transaction response time of each transaction, in the **Load Criteria** box, select **Running Vusers**.

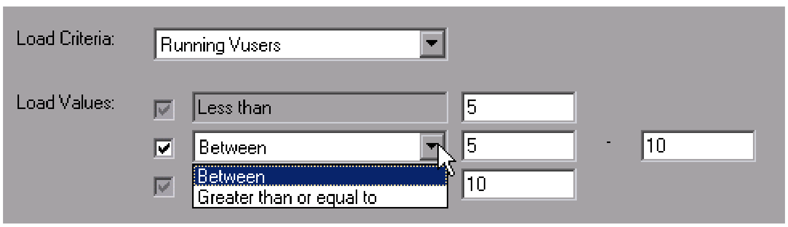
Then set the value ranges for the running Vusers:

Consider less than 20 Vusers to be a light load, 20 – 50 Vusers an average load, and 50 Vusers or more a heavy load. Enter these values in the Load Values boxes.

Note:

➤ You can set up to three in-between ranges.

➤ Valid load value ranges are consecutive—there are no gaps in the range—and span all values from zero to infinity.



6 Set thresholds

On the Set Threshold Values page, you define the acceptable average transaction response times for the transactions, taking into account the defined load criteria.

In this case, define the same threshold values for both transactions as follows: for a light load, a reasonable average response time can be up to 5 seconds, for an average load, up to 10 seconds, and for a heavy load, up to 15 seconds.

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

Tip: To define the same thresholds for all the transactions, you can type the values in the table nearer the bottom of the Set Threshold Values page, and click **Apply to all transactions**.

---

7 Define a tracking period - optional

When SLA statuses for a measurement are determined at time intervals over a timeline, the frequency of the time intervals is determined by the **tracking period**.

This step is optional because an internally-calculated tracking period of at least 5 seconds is defined by default. You can change the tracking period in the Advanced Options dialog box:

a In the Service Level Agreement pane, click the **Advanced** button.

b Select **Tracking period of at least X seconds**, and select a tracking period. The time intervals are calculated by Analysis according to a built-in algorithm and as a function of the value you enter here.

Example:

If you select a tracking period of 10, and the aggregation granularity for the scenario (defined in Analysis) is 6, then the tracking period is set to the nearest multiple of 6 that is greater than or equal to 10, that is, Tracking Period = 12.

For details, see "Tracking Period" on page 165.

For user interface details, see "Advanced Options Dialog Box" on page 173.

8 Results

When analyzing your scenario run, Analysis applies your SLA settings to the default Summary Report and the report is updated to include all the relevant SLA information.

For example, it displays the worst performing transactions in terms of defined SLAs, how specific transactions performed over set time intervals, and overall SLA statuses.

Reference

Service Level Agreements User Interface

This section includes (in alphabetical order):

Advanced Options Dialog Box

This dialog box enables you to define a tracking period for load test scenario.

To access	**Tools** menu > **Configure SLA Rules** > **Service Level Agreement** pane > Advanced
Important Information	The tracking period is calculated by Analysis according to a built-in algorithm and as a function of the value entered here.
Relevant tasks	➤ "How to Define Service Level Agreements" on page 166 ➤ "How to Define Service Level Agreements - Use-Case Scenario" on page 168
See also	"Service Level Agreements Overview" on page 164

User interface elements are described below:

UI Elements	**Description**
Internally calculated tracking period	Analysis sets the tracking period to the minimum value possible, taking into account the aggregation granularity defined for the scenario. This value is at least 5 seconds. It uses the following formula: Tracking Period = Max (5 seconds, aggregation granularity)
Tracking period of at least X seconds	Determines the minimum amount of time for the tracking period. This value can never be less than 5 seconds. Analysis sets the tracking period to the nearest multiple of the scenario's aggregation granularity that is greater than or equal to the value (X) that you selected. For this option, Analysis uses the following formula: Tracking Period = Max(5 seconds, m(Aggregation Granularity)) where **m** is a multiple of the scenario's aggregation granularity such that **m(Aggregation Granularity)** is greater than or equal to **X.** **Example:** If you select a tracking period of X=10, and the aggregation granularity for the scenario is 6, then the tracking period is set to the nearest multiple of 6 that is greater than or equal to 10, that is, Tracking Period = 12.

Goal Details Dialog Box

This dialog box displays the thresholds that were set for the selected SLA.

To access	**Tools** menu > **Configure SLA Rules** > **Service Level Agreement** pane > Details
Important information	If you defined load criteria as part of your SLA, the threshold values are displayed per the defined load value ranges.
See also	"Service Level Agreements Overview" on page 164

Service Level Agreement Pane

This pane lists all the service level agreements (SLAs) defined for the scenario.

To access	**Tools** menu > **Configure SLA Rules** > **Service Level Agreement** pane
Relevant Tasks	➤ "How to Design a Goal-Oriented Scenario" on page 78 ➤ "How to Design a Manual Scenario" on page 80 ➤ "How to Define Service Level Agreements" on page 166 ➤ "How to Define Service Level Agreements - Use-Case Scenario" on page 168
See also	"Service Level Agreements Overview" on page 164

User interface elements are described below:

UI Elements	Description
New	Starts the Service Level Agreement wizard where you can define new goals for the load test scenario.
Details	Opens the Goal Details dialog box which displays a summary of the details of the selected SLA.
Edit	Opens the Service Level Agreement wizard where you can modify the goals defined in the SLA.
Delete	Deletes the selected SLA.
Advanced	Opens the Advanced Options dialog box where you can adjust the tracking period for measurements that are evaluated per time interval over a timeline. For more information, "Tracking Period" on page 165. For user interface details, see "Advanced Options Dialog Box" on page 173.
Service Level Agreement list	Lists the SLAs defined for the scenario.

Service Level Agreement Wizard

This wizard enables you to define goals or **service level agreements** (SLAs) for your load test scenario.

To access	**Tools** menu > **Configure SLA Rules** > **Service Level Agreement** pane > New
Important information	There are two modes for the Service Level Agreement wizard. The pages included in the wizard depend on the measurement that is selected. See the wizard maps below.
Relevant tasks	➤ "How to Design a Goal-Oriented Scenario" on page 78 ➤ "How to Design a Manual Scenario" on page 80 ➤ "How to Define Service Level Agreements" on page 166 ➤ "How to Define Service Level Agreements - Use-Case Scenario" on page 168
Wizard map - Goal measured per time interval	The Service Level Agreement Wizard contains: **Welcome** > Select a Measurement Page > (Select Transactions Page) > Set Load Criteria Page > Set Threshold Values Page (Goal Per Time Interval)
Wizard map - Goal measured over whole scenario run	The Service Level Agreement Wizard contains: **Welcome** > Select a Measurement Page > (Select Transactions Page) > Set Threshold Values Page (Goal Per Whole Run)
See also	"Service Level Agreements Overview" on page 164

Select a Measurement Page

This wizard page enables you to select a measurement for your goal.

Important information	➤ General information about this wizard is available here: "Service Level Agreement Wizard" on page 176. ➤ There are two modes for the Service Level Agreement wizard. The wizard pages that follow depend on the measurement that you select on this page. See the wizard maps below.
Wizard map - Goal measured per time interval	The Service Level Agreement Wizard contains: **Welcome** > **Select a Measurement Page** > (Select Transactions Page) > Set Load Criteria Page > Set Threshold Values Page (Goal Per Time Interval)
Wizard map - Goal measured over whole scenario run	The Service Level Agreement Wizard contains: **Welcome** > **Select a Measurement Page** > (Select Transactions Page) > Set Threshold Values Page (Goal Per Whole Run)
See also	"Service Level Agreements Overview" on page 164

User interface elements are described below:

UI Elements	Description
SLA status determined over the whole run	Evaluates a single SLA status for the whole scenario run. Select one of the following measurements: ➤ **Transaction Response Time (Percentile)** ➤ **Total Hits per run** ➤ **Average Hits (hits/second) per run** ➤ **Total Throughput (bytes) per run** ➤ **Average Throughput (bytes/second) per run**
SLA status determined per time intervals over a timeline	Evaluates SLA statuses at set time intervals within the run. Select one of the following measurements: ➤ **Average Transaction Response Time** ➤ **Errors per Second** The time intervals at which the SLA statuses are evaluated are known as the **tracking period**. For details, see"Tracking Period" on page 165.

Select Transactions Page

This wizard page enables you to select transactions to evaluate as part of your goal.

Important information	➤ General information about this wizard is available here: "Service Level Agreement Wizard" on page 176. ➤ This page is displayed when creating an SLA for Transaction Response Time by Average or by Percentile. ➤ In order to define an SLA for Transaction Response Time by Average or by Percentile, at least one of the Vuser scripts participating in the scenario must include a transaction. ➤ You can select multiple transactions using the CTRL key.
Wizard map - Goal measured per time interval	The Service Level Agreement Wizard contains: **Welcome** > Select a Measurement Page > (**Select Transactions Page**) > Set Load Criteria Page > Set Threshold Values Page (Goal Per Time Interval)
See also	"Service Level Agreements Overview" on page 164

User interface elements are described below:

UI Elements (A-Z)	Description
Available Transactions	Lists the transactions in the Vuser scripts participating in the scenario. To move a script to the **Selected Transaction** list, select it and click **Add**.
Selected Transactions	Lists the transactions in the Vuser scripts participating in the scenario that have been selected for the SLA. To remove a script from this list, select it and click **Remove**.

Set Load Criteria Page

This wizard page enables you to select load criteria to take into account when testing your goal.

Important information	➤ General information about this wizard is available here: "Service Level Agreement Wizard" on page 176. ➤ This page is displayed only when defining an SLA that determines SLA statuses per time interval over a timeline. ➤ In the next wizard step (Set Threshold Values page), you will set different thresholds per each of the load ranges that you select here.
Wizard map - Goal measured per time interval	The Service Level Agreement Wizard contains: **Welcome** > Select a Measurement Page > (Select Transactions Page) > **Set Load Criteria Page** > Set Threshold Values Page (Goal Per Time Interval)
See also	"Service Level Agreements Overview" on page 164

User interface elements are described below:

UI Elements	Description
Load Criteria	The relevant load criteria that you want to use **Example:** If you want to see the impact of running Vusers on the measurement, select **Running Vusers**. To define an SLA without load criteria, select **None**.
Load Values	Valid load value ranges are consecutive—there are no gaps in the range—and span all values from zero to infinity. ➤ **Less than.** Enter the upper value for the lower range of values for the load criteria. The lower range is between 0 and the value you entered. It does not include the upper value. **Example:** If you enter 5, the lower range of values for the load criteria is between 0 and 5, but does not include 5. ➤ **Between.** The in-between range of values for the load criteria. Enter lower and upper values for this range. The lower range is included in this range; it does not include the upper value. **Example:** If you enter 5 and 10, the in-between range of values for the load criteria is from 5 and up to, but not including, 10. **Note:** You can set up to three in-between ranges. ➤ **Greater than.** Enter the lower value for the upper range of values for the load criteria. The upper range includes values from the value you entered and on. **Example:** If you enter 10, the upper range of values for the load criteria is from 10 and on.
Selected Measurement	The measurement selected for the goal.

Set Percentile Threshold Values Page

This wizard page enables you to select load criteria to take into account when testing your goal.

Important information	➤ General information about this wizard is available here: "Service Level Agreement Wizard" on page 176. ➤ The Percentile SLA enables you to measure whether the percentage of transaction samples meets the defined threshold criteria. ➤ You can enter a threshold value to 3 decimal places.
Wizard map - Goal measured over whole scenario run	The Service Level Agreement Wizard contains: **Welcome** > Select a Measurement Page > (Select Transactions Page) > **Set Percentile Threshold Values Page**
See also	"Service Level Agreements Overview" on page 164

User interface elements are described below:

UI Elements	Description
Selected Measurement	The measurement selected for the goal.
Percentile	Percentage of transactions to measure against the configured threshold.
Provide threshold value for all transactions	To apply one set of threshold values to all transactions selected for the goal, enter the threshold value and click **Apply to all**. These values are applied to all the transactions in the Thresholds table at the bottom of the page.
Transaction name	The transaction from the scenario run.
Threshold	The threshold value for the selected transaction.

Set Threshold Values Page (Goal Per Time Interval)

This wizard page enables you to set thresholds for the measurements you are evaluating in your goal.

Important information	➤ General information about this wizard is available here: "Service Level Agreement Wizard" on page 176. ➤ If you defined load criteria in the Set Load Criteria Page, you must set thresholds per each of the defined load ranges. If you did not define load criteria, you set one threshold value. For Average Transaction response time, you set threshold values for each transaction. ➤ You can enter a threshold value to 3 decimal places.
Wizard map - Goal measured per time interval	The Service Level Agreement Wizard contains: **Welcome** > Select a Measurement Page > (Select Transactions Page) > Set Load Criteria Page > **Set Threshold Values Page (Goal Per Time Interval)**
See also	"Service Level Agreements Overview" on page 164

User interface elements are described below (unlabeled elements are shown in angle brackets):

UI Elements	Description
<Thresholds table>	The thresholds for your goal. If you defined load criteria, enter thresholds for each range of values. **Note:** If the maximum threshold value is exceeded during a particular time interval during the run, Analysis displays an SLA status of **Failed** for that time interval.

UI Elements	Description
Apply to all (Average Transaction Response Time goal only)	To apply one set of threshold values to all transactions selected for the goal, enter the threshold values in this table and click **Apply to all transactions**. These values are applied to all the transactions in the Thresholds table at the top of the page. **Note:** Threshold values for selected transactions do not have to be the same. You can assign different values for each transaction.
Selected Measurement	The measurement selected for the goal.

Set Threshold Values Page (Goal Per Whole Run)

This wizard page enables you to set minimum thresholds for the measurements you are evaluating in your goal.

Important information	General information about this wizard is available here: "Service Level Agreement Wizard" on page 176.
Wizard map - Goal measured over whole scenario run	The Service Level Agreement Wizard contains: **Welcome** > Select a Measurement Page > **Set Threshold Values Page (Goal Per Whole Run)**
See also	"Service Level Agreements Overview" on page 164

User interface elements are described below:

UI Elements	Description
Selected measurement	The measurement selected for the goal.
Threshold	The minimum threshold value for the selected measurement. **Note:** If the value of the measurement is lower than this threshold during the run, Analysis displays an SLA status of **Failed** for the entire run.

9

Working with Application Lifecycle Management

This chapter includes:

Concepts

Tasks

Reference

Concepts

Managing Results Using ALM Overview

Analysis works together with HP Application Lifecycle Management (ALM). ALM provides an efficient method for storing and retrieving scenario and analysis results. You can store results in an ALM project and organize them into unique groups.

In order for the Analysis to access an ALM project, you must connect it to the Web server on which ALM is installed. You can connect to either a local or remote Web server.

When working against an ALM server with Performance Center, the ALM integration has several additional capabilities, such as the ability to save the Analysis session to a new location, and upload a report from the file system to ALM. For details, see "How to Work with Results in ALM - With Performance Center" on page 191.

For more information on working with ALM, see the *Application Lifecycle Management User Guide*.

Tasks

How to Connect to ALM

To store and retrieve results from ALM, you need to connect to an ALM project. You can connect or disconnect from an ALM project at any time during the testing process.

You can connect to one version of HP ALM from Analysis and a different version from your browser. For more information, see the **Important Information** section in "HP ALM Connection Dialog Box" on page 198.

To connect to ALM:

1. Select **Tools** > **HP ALM Connection**. The HP ALM Connection dialog box opens.
2. Enter the required information in the HP ALM Connection dialog box, as described in "HP ALM Connection Dialog Box" on page 198.
3. To disconnect from ALM, click **Disconnect**.

How to Work with Results in ALM - Without Performance Center

The following steps describe the workflow for working with results saved in an ALM project, whose server does not have a Performance Center installation.

When working against an ALM server with HP Performance Center, there are several differences. For more information, see "How to Work with Results in ALM - With Performance Center" on page 191.

- "Connect to ALM" on page 190
- "Open an existing Analysis session file - optional" on page 190

➤ "Create a new Analysis session file from the raw data - optional" on page 191

➤ "Save the LoadRunner results file" on page 191

Connect to ALM

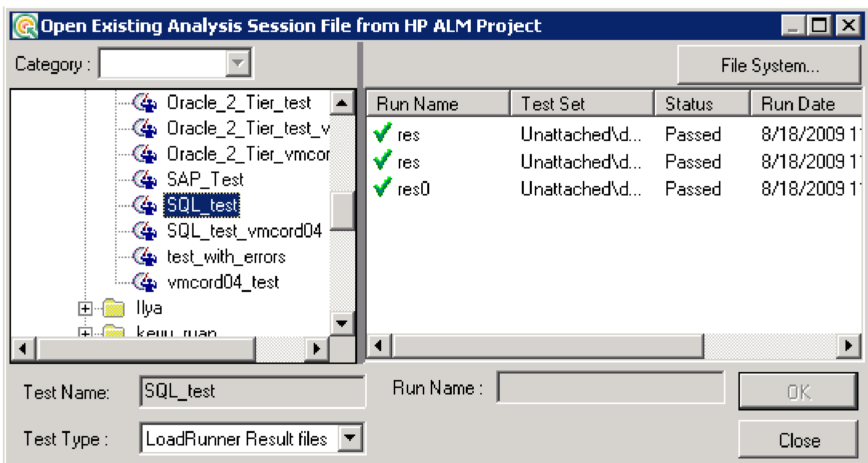
Open a connection to the ALM server and project that contains the LoadRunner result or Analysis session files. For task details, see "How to Connect to ALM" on page 189.

Open an existing Analysis session file - optional

a Select **File** > **Open**.

b In the left pane select a script.

c In the right pane, select the results for which the Analysis session file was created.



d Click **OK**.

Create a new Analysis session file from the raw data - optional

This procedure describes how to create a new Analysis session file on the ALM server, from the raw results file. If an Analysis session file already exists for the raw data, you can choose to overwrite the existing file.

a Select **File** > **New**.

b In the left pane select a script.

c In the right pane, select the results you want to analyze.

d Click **OK**.

Save the LoadRunner results file

When you are finished analyzing your results and creating reports or graphs, save the changes. Select **File** > **Save**. The Analysis session file is in the ALM project.

Note: When working with ALM without Performance Center, **Save As** is not supported—you cannot save the Analysis session file to another location.

How to Work with Results in ALM - With Performance Center

ALM servers with Performance Center, allow you to perform the following operations:

- "Open an existing Analysis Session file" on page 192
- "Open raw data and create a new Analysis session" on page 192
- "Save the changes to the Analysis session file" on page 193
- "Save the Analysis session file to a new ALM location" on page 194

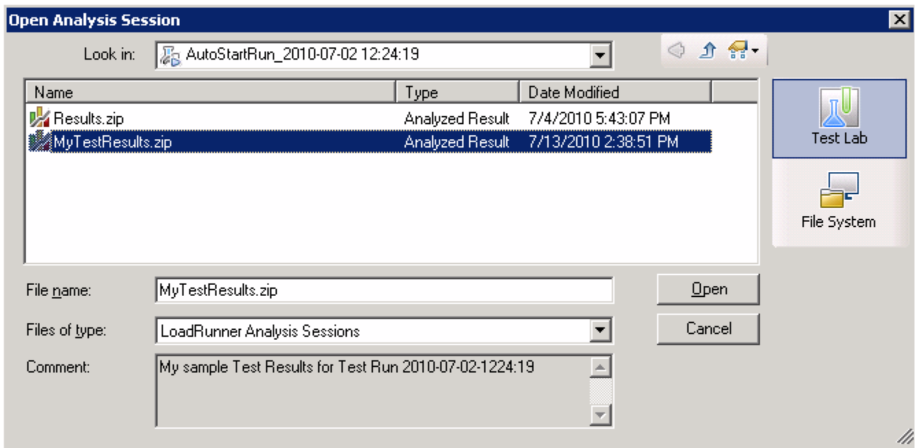
Open an existing Analysis Session file

a Select **Tools** > **HP ALM Connection** and make sure your connection to ALM is open.

b Select **File** > **Open**.

c Drill down to the Run level within the Test Plan module, and select an individual run.

d Select a zip file containing the Analysis session file.



e Click **Open**.

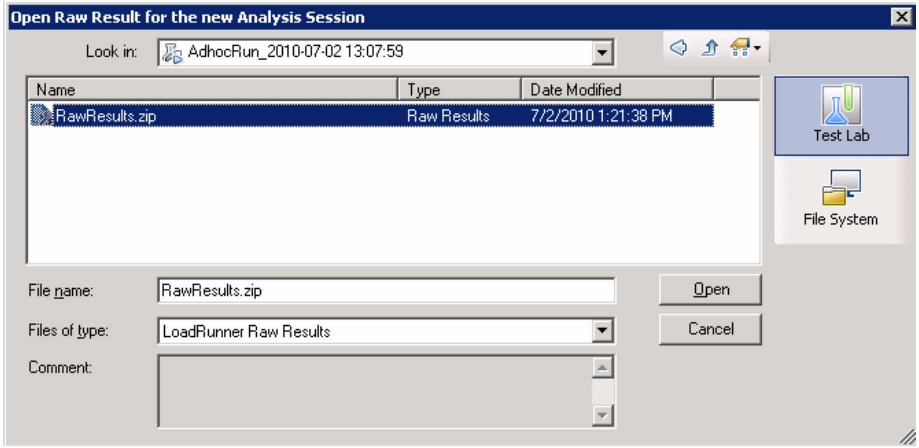
Open raw data and create a new Analysis session

a Select **Tools** > **HP ALM Connection** and make sure your connection to ALM is open.

b To create a new Analysis session file from the raw data, select **File** > **New**.

c Drill down to the Run level within the Test Plan module, and select an individual run.

d Select a zip file containing the run's raw data.



e Click **Open**.

Save the changes to the Analysis session file

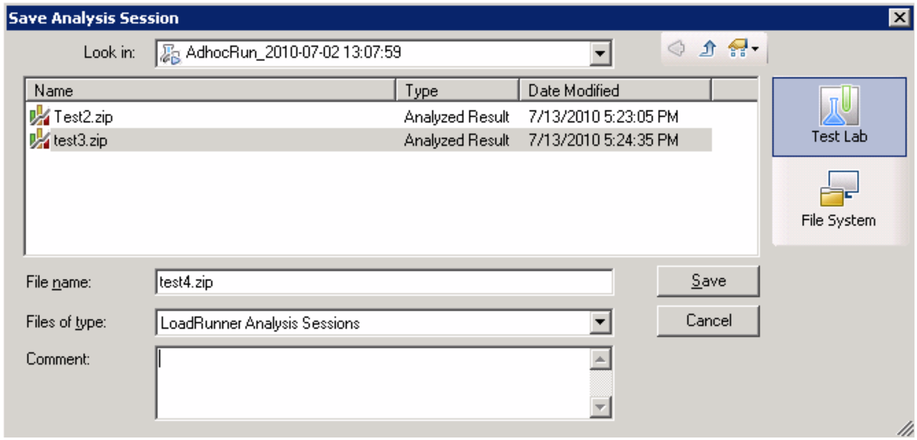
a Complete your changes to the Analysis results.

b Select **Tools** > **HP ALM Connection** and make sure your connection to ALM is open.

c Select **File** > **Save**.

d To save an Analysis session that was opened from the file system, click the **Test Lab** module button.

e Drill down to the Run level within the Test Plan module, and specify a name for the zip file.



f Provide a comment about the Analysis session (optional).

g Click **Save**.

Save the Analysis session file to a new ALM location

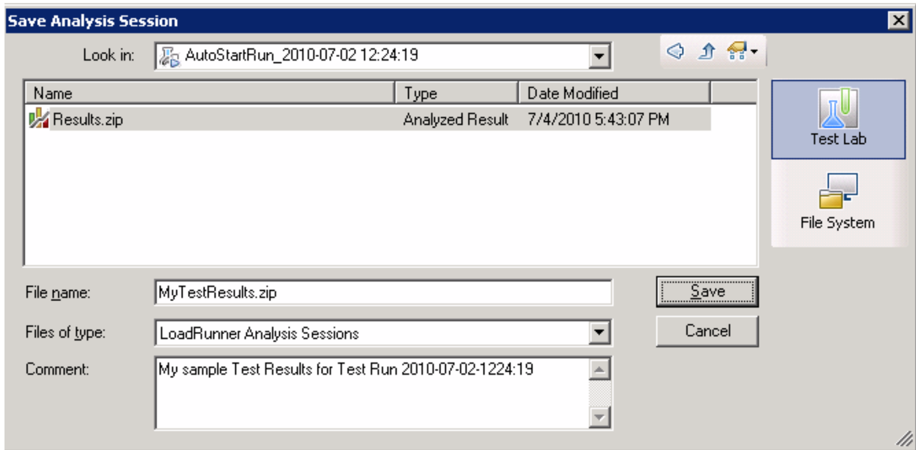
a Select **Tools** > **HP ALM Connection** and make sure your connection to ALM is open.

b Open an Analysis session file from the file system, or from ALM as described above.

c Select **File** > **Save as**.

d Drill down to the Run level within the Test Plan module, and select an individual run.

e Specify a name for the Analysis session zip file. The name *Results* is reserved.



f Provide a comment about the Analysis session (optional).

g Click **Save**.

How to Upload a Report to ALM

The following steps describe how to upload a report from the file system to an ALM's Test Lab module. This capability is only available for ALM installation with Performance Center.

When working against an ALM server with HP Performance Center, there are several differences. For more information, see "How to Work with Results in ALM - With Performance Center" on page 191.

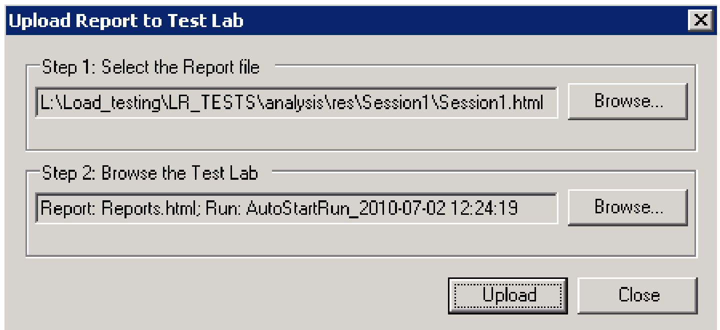
- "Connect to ALM" on page 196
- "Open the Upload dialog box" on page 196
- "Select a report" on page 196
- "Select a location on ALM" on page 196
- "Begin the upload" on page 196

Connect to ALM

Open a connection to the ALM server and project that contains the LoadRunner result or Analysis session files. For task details, see "How to Connect to ALM" on page 189.

Open the Upload dialog box

Select **Tools** > **Upload Report to Test Lab**.



Select a report

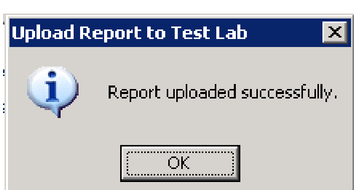
Click **Browse** in the **Step 1** section. The **Select the Report file** dialog box opens. Select an HTML or XML file from the file system. Click **Open**.

Select a location on ALM

Click **Browse** in the **Step 2** section. The **Select Location for the Report** dialog box opens. Navigate to a Run level in the Test Lab module. Specify a name for the report and include any relevant comments. Click **OK**.

Begin the upload

Click **Upload**. If it succeeds, the Analysis issues a message.



Reference

ALM User Interface

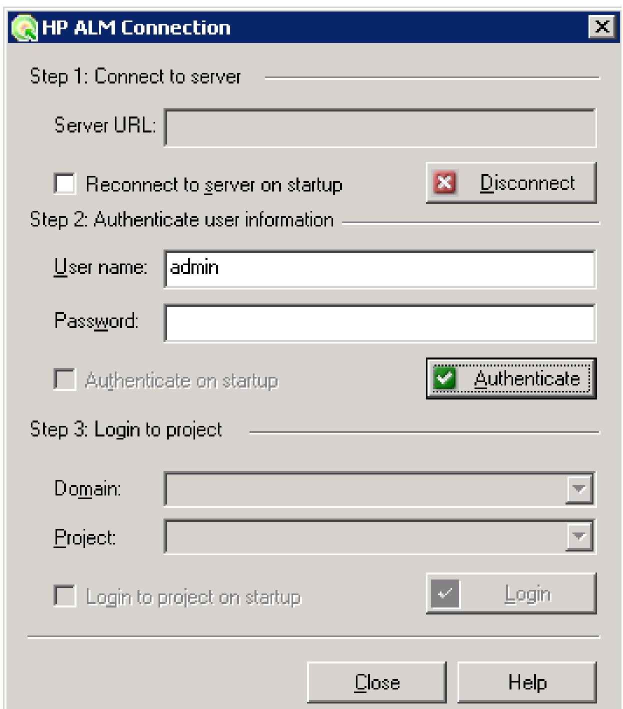
This section includes (in alphabetical order):

➤ HP ALM Connection Dialog Box on page 198

➤ Upload Report to Test Lab Dialog Box on page 202

HP ALM Connection Dialog Box

This dialog box enables you to connect to an ALM project from within the Analysis.



To access	**Tools > HP ALM Connection ...**
Important information	You can connect to one version of HP ALM from Analysis and a different version of HP ALM from your browser. You can only connect to different versions of HP ALM if one of the versions is HP ALM 11.00 or higher. If you are connecting from Analysis to a different HP ALM version than the one in your browser, you must first download the client files. **1** From the browser, navigate to the HP ALM server that you will be connecting to from Analysis. **2** Once the login screen is displayed, the client files have been downloaded. There is no need to log in.
Relevant tasks	"How to Connect to ALM" on page 189

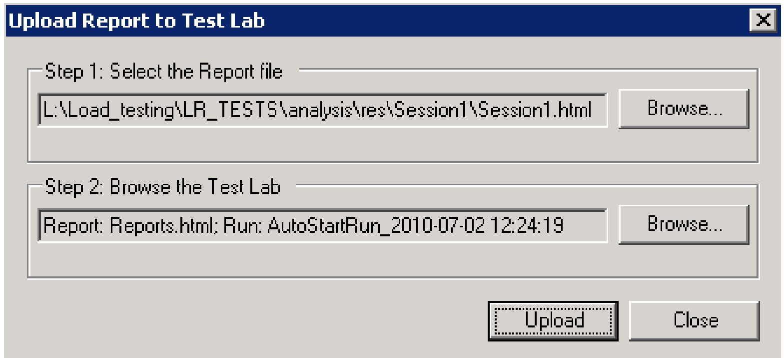
User interface elements are described below:

UI Elements (A-Z)	Description
Step 1: Connect to Server	➤ **Server URL.** The URL of the server that contains ALM in the following form **http://<server_name:port>/qcbin**. ➤ **Reconnect to server on startup.** Automatically reconnect to the server every time you start the application. ➤  /  . Connects to the server specified in the **Server URL** box. Only one button is visible at a time, depending on your connection status.

UI Elements (A-Z)	Description
Step 2: Authenticate User Information	➤ **User Name.** Your ALM project user name. ➤ **Password.** Your ALM project password. ➤ **Authenticate on startup.** Authenticates your user information automatically, the next time you open the application. This option is only available if you selected **Reconnect to server on startup** above. ➤ Authenticate. Authenticates your user information against the ALM server. After your user information has been authenticated, the fields in the Authenticate user information area are displayed in read-only format. The Authenticate button changes to Change User. You can log in to the same ALM server using a different user name by clicking **Change User**, entering a new user name and password, and then clicking **Authenticate** again.
Step 3: Login to Project	➤ **Domain.** The domain that contains the ALM project. Only those domains containing projects to which you have permission to connect to are displayed. (If you are working with a project in versions of TestDirector earlier than version 7.5, the **Domain** box is not relevant.) ➤ **Project.** Enter the ALM project name or select a project from the list. Only those projects that you have permission to connect to are displayed. ➤ **Login to project on startup.** This option is only enabled when the **Authenticate on startup** check box is selected. ➤ Login / Logout.Logs into and out of the ALM project.

Upload Report to Test Lab Dialog Box

This dialog box enables you to upload an Analysis report to an ALM project's Test Lab module.



To access	**Reports > Upload Report to Test Lab**

User interface elements are described below:

UI Elements (A-Z)	Description
Step 1: Select the report file	Allows you to select an Analysis report from the file system. You can select an HTML report, or Rich report in XML format.
Step 2: Browse the test lab	Allows you to select an location within the Test Lab module, for the report. **Note:** You must drill down to the level of a Run within the Test Lab module.
Upload	Begins the uploading of the report. If the uploading succeeds, the Analysis issues a message.

10

Importing External Data

This chapter includes:

***Concepts**

***Tasks**

***Reference**

Concepts

Import Data Tool Overview

The LoadRunner Analysis Import Data tool enables you to import and integrate non-HP data into a LoadRunner Analysis session. After the import procedure, you can view the data files as graphs within the session, using all the capabilities of the Analysis tool.

Suppose an NT Performance Monitor runs on a server and measures its behavior. Following a LoadRunner scenario on the server, you can retrieve the results of the Performance Monitor, and integrate the data into LoadRunner's results. This enables you to correlate trends and relationships between the two sets of data: LoadRunner's and the Performance Monitor's.

In this case, the results of the NT Performance Monitor are saved as a **.csv** file. You launch the Import Data tool, direct it to the **.csv** file, and specify its format. LoadRunner reads the file and integrates the results into its own Analysis session.

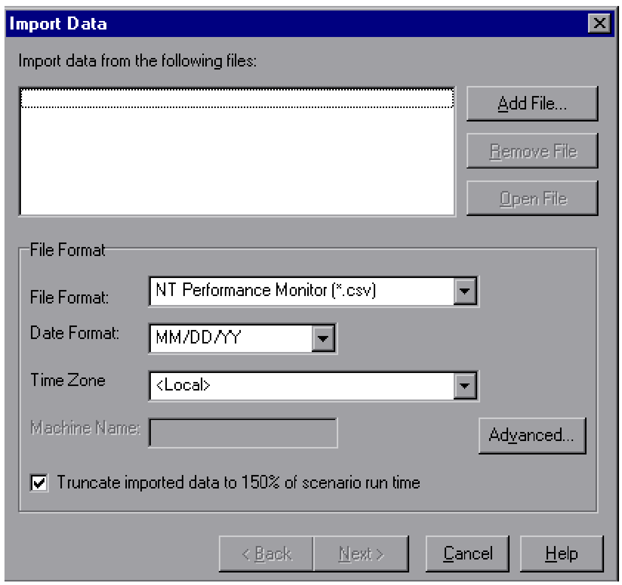
For a list of data formats that are supported, see "Supported File Types" on page 208. To define your own custom data files, see "How to Define Custom File Formats" on page 206.

Tasks

How to Use the Import Data Tool

This task describes how to import data files to integrate into your analysis session.

1 Choose **Tools** > **External Monitors** > **Import Data**. The Import Data dialog box opens.



2 Select the format of the external data file from the **File format** list box.

3 Click **Add File**. In the **Select File to Import** dialog box that opens, the **Files of type** list box shows the type chosen in step 2.

4 Set other file format options, as described in "Import Data Dialog Box" on page 215. You must enter a machine name.

5 To specify character separators and symbols, click **Advanced**. For more information, see "Advanced Settings Dialog Box" on page 211.

6 Click **Next**. The Import Data dialog box opens.

7 Select the type of monitor that generated the external data file. If your monitor type does not exist, you can add it, as described in "How to Customize Monitor Types for Import" on page 207.

   When opening a new graph, you will see your monitor added to the list of available graphs under this particular category. (See "Open a New Graphs Dialog Box" on page 43.)

8 Click **Finish**. LoadRunner Analysis imports the data file or files, and refreshes all graphs currently displayed in the session.

Note: When importing data into a scenario with two or more cross results, the imported data will be integrated into the last set of results listed in the **File** > **Cross with Result** dialog box. For more information, see "How to Generate Merged Graphs" on page 159.

How to Define Custom File Formats

This task describes how to define a custom format, if the file format of your import file is not supported.

If the file format of your import file is not supported, you can define a custom format.

1 Choose **Tools** > **External Monitors** > **Import Data**. The Import Data dialog box opens.

2 From the **File Format** list, select <**Custom File Format**>. The Enter New Format Name dialog box opens.

3 Enter a name for the new format (in this case, my_monitor_format).

4 Click **OK**. The Define External Format dialog box opens.

5 Specify the mandatory and optional data, as described in "Define External Format Dialog Box" on page 212.

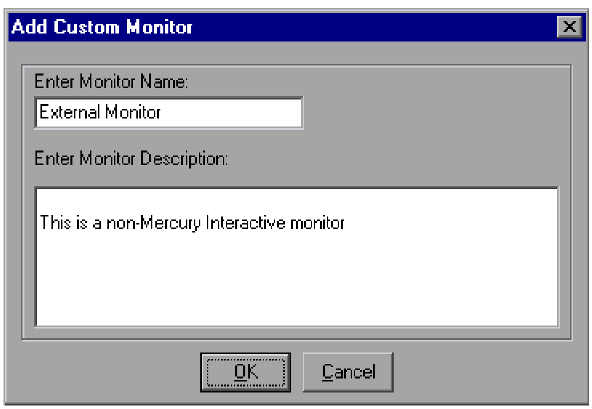
6 Click **Save**.

How to Customize Monitor Types for Import

This task describes how to define a new monitor type, if your monitor is not included in any of the categories fount in the **Monitor Type** list.

1 Choose **Tools** > **External Monitors** > **Import Data**. The Import Data dialog box opens. For more information, see "Import Data Dialog Box" on page 215.

2 In the **Import Data** dialog box, select **External Monitors** > **Add Custom Monitor**.



3 Enter a monitor name and description and click **OK**.

The new monitor can now be selected in the list of available monitors.

Reference

Supported File Types

The following file types are supported:

NT Performance Monitor (.csv)

Default file type of NT Performance monitor, in comma separated value (CSV) format.

For example:

```
Reported on \\WINTER
Date: 10/23/01
Time: 10:08:39 AM
Data: Current Activity
Interval:   1.000 seconds

,,% Privileged Time,% Processor Time,% User Time,
,,0,0,0,
,,,,,
,,Processor,Processor,Processor,
Date,Time,\\WINTER,\\WINTER,\\WINTER,
10/23/01,10:07:00 AM ,     0.998,     1.174,     0.000,
10/23/01,10:07:01 AM ,     0.000,     0.275,     0.000,
```

Windows 2000 Performance Monitor (.csv)

Default file type of Windows 2000 Performance monitor, but incompatible with NT Performance monitor. In comma separated value (CSV) format.

For example:

```
"(PDH-CSV 4.0)","\\MACRON\Processor(_Total)\% Processor Time","\\MACRON\Processor(_Total)\% User
Time","\\MACRON\Processor(_Total)\Interrupts/sec","\\MACRON\System\File Control Bytes/sec"
"10/29/2001
13:09:33.746","99.999148401465547","0.0021716772078191897","997.21487008127474","488.53479318892
"10/29/2001
13:09:48.747","18.157543391188248","8.4112149532710276","1116.5859176246415","9843.293330312279
"10/29/2001
13:10:03.749","5.941255006675572","1.5353805073431241","1100.9651204860379","623.18277489319848"
```

Standard Comma Separated File (.csv)

This file type has the following format:

Date,Time,Measurement_1,Measurement_2, ...

where fields are comma separated and first row contains column titles.

The following example from a standard CSV file shows 3 measurements: an interrupt rate, a file IO rate and a CPU usage. The first row shows an interrupt rate of 1122.19 and an IO rate of 4.18:

```
date, time, interrupt rate, File IO rate, CPU bust percent
25/05/01,10:09:01,1122.19,4.18,1.59
25/05/01,10:10:01,1123.7,6.43,1.42
```

Master-Detail Comma Separated File (.csv)

This file type is identical to Standard Comma Separated Files except for an additional **Master** column which specifies that row's particular breakdown of a more general measurement. For example, a Standard CSV file may contain data points of a machine's total CPU usage at a given moment:

Date,Time,CPU_Usage

However, if the total CPU usage can be further broken up into CPU time per-process, then a Master-Detail CSV file can be created with an extra column **ProcessName**, containing the name of a process.

Each row contains the measurement of a specific process's CPU usage only. The format will be the following:

Date,Time,ProcessName,CPU_Usage

as in the following example:

```
date, time,process name, CPU used, elapsed time used
25/05/01,10:06:01,edaSend,0.1,47981.36
25/05/01,10:06:01,PDS,0,47981.17
```

Microsoft Excel File (.xls)

Created by the Microsoft Excel application. The first row contains column titles.

	A	B	C	D	E
1	date	time	interrupt rate	File IO rate	CPU bust percent
2	25/05/01	10:09:01	1122.19	4.18	1.59
3	25/05/01	10:10:01	1123.7	6.43	1.42
4	25/05/01	10:11:01	1103.62	5.33	1.17
5	25/05/01	10:12:01	1118.89	12.18	2.37
6	25/05/01	10:13:01	1116.89	19.85	3.87
7	25/05/01	10:14:01	1128.12	19.9	4.15
8	25/05/01	10:15:01	1151.98	20.82	4.25
9	25/05/01	10:16:01	1110.1	4.83	1.34

Master-Detail Microsoft Excel file (.xls)

Created by Microsoft's Excel application. The first row contains column titles. It contains an extra **Master** column. For an explanation of this column, see "Master-Detail Comma Separated File (.csv)" on page 209.

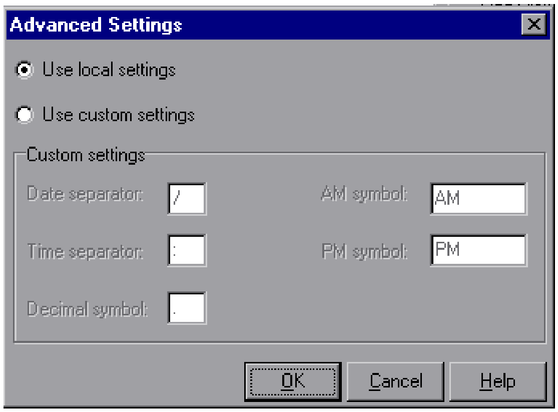
	A	B	C	D	E
1	date	time	process name	CPU used	elapsed time used
2	25/05/01	10:06:01	edaSend	0.1	47981.36
3	25/05/01	10:06:01	PDS	0	47981.17
4					

Import Data User Interface

This section includes:

Advanced Settings Dialog Box

This dialog box enables you to define the data format of the imported file to settings other than of the regional configuration.



To access	**Tools** > **External Monitors** > **Import Data** > **Advanced**

User interface elements are described below:

UI Elements	Description
Use local settings	Keep default settings of the regional configuration. Disables the **Custom Settings** area of the dialog box.
Use custom settings	Define your own settings. Enables the **Custom Settings** area of the dialog box. ➤ **Date Separator.** Enter a custom symbol, for example, the slash ('/') character in 11/10/02 ➤ **Time Separator.** Enter a custom symbol, for example, the colon ':' character in 9:54:19 ➤ **Decimal symbol.** Enter a custom symbol, for example, the '.' character in the number 2.5 ➤ **AM symbol.** Enter a custom symbol for the hours between midnight and noon. ➤ **PM symbol.** Enter a custom symbol for the hours between noon and midnight.

Define External Format Dialog Box

This dialog box enables you to define a new file format for external data files not supported by Analysis.

The Define External Format dialog box is divided into mandatory and optional information.

To access	**Tools** > **External Monitors** > **Import data** > **File Format** > **<Custom File Format>**
Relevant tasks	"How to Define Custom File Formats" on page 206

Mandatory tab

User interface elements are described below:

UI Elements	Description
Date Column Number	Enter the column that contains the date. If there is a master column (see "Master-Detail Comma Separated File (.csv)" on page 209), specify its number.
Time Column Number	Enter the column that contains the time.
Use Master Column	Select this if the data file contains a master column. A master column specifies the row's particular breakdown of a more general measurement.
File Extension	Enter the file suffix.
Field Separator	Enter the character that separates a field in a row from its neighbor. To select a field separator character, click **Browse** and select a character from the define Field Separator dialog box.

Optional tab

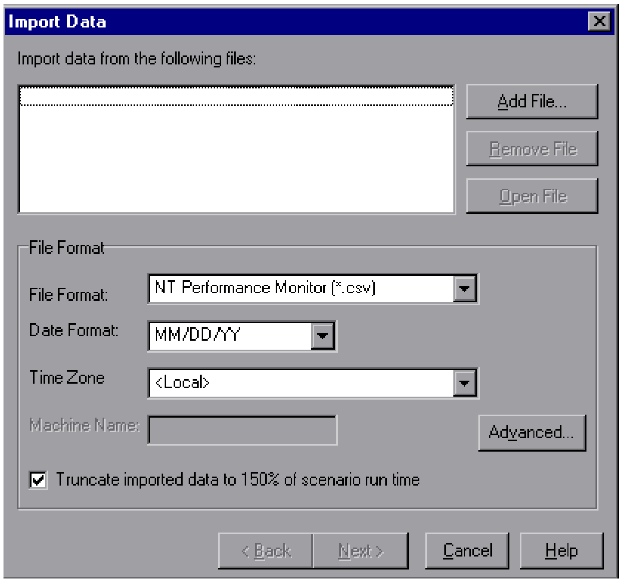
User interface elements are described below:

UI Elements	Description
Date Format	Specify the format of the date in the imported data file. For example, for European dates with a 4 digit year, choose **DD/MM/YYYY**.
Time Zone	Select the time zone where the external data file was recorded. LoadRunner Analysis aligns the times in the file with local time zone settings to match LoadRunner results. (LoadRunner does not alter the file itself).
Machine Name	Specify the machine name the monitor runs on. This associates the machine name with the measurement.

UI Elements	Description
Exclude Columns	Indicate which columns are to be excluded from the data import, such as columns containing descriptive comments. When there is more than one column to be excluded, specify the columns in a comma-separated list. For example, 1,3,8.
Convert file from UNIX to DOS format	Monitors often run on UNIX machines. Check this option to convert data files to Windows format. A carriage return (Ascii character 13) is appended to all line feed characters (Ascii character 10) in the UNIX file.
Skip the first [] lines	Specify the number of lines at the start of the file to ignore before reading in data. Typically, the first few lines in a file contain headings and sub-headings.

Import Data Dialog Box

This dialog box enables you to import and integrate non-HP data files into Analysis session.



To access	**Tools** > **External Monitors** > **Import Data**

User interface elements are described below:

UI Elements	Description
Import data from the following files	Displays the files that you selected for import.
Add file	Select an external data file to import. A dialog box opens to enable you to select files.
Remove file	Delete an external data file from the list.

UI Elements	Description
Open File	Open an external data file using the associated application.
File Format	Set the file format options. ➤ **File Format**. Choose the format of the external data file. For an explanation of available formats, see "Supported File Types" on page 208. ➤ **Date Format.** Specify the format of the date in the imported data file. For example, for European dates with a 4 digit year, choose **DD/MM/YYYY**.
Time Zone	Select the time zone where the external data file was recorded. LoadRunner Analysis compensates for the various international time zones and aligns the times in the file with local time zone settings in order to match LoadRunner results. If the times in the imported file are erroneous by a constant offset, you can synchronize the time.
<Synchronize with scenario start time>	Time Zone also contains the option **<Synchronize with scenario start time>**. Choose this to align the earliest measurement found in the data file to the start time of the LoadRunner scenario.
Machine Name	Specify the machine name the monitor runs on. This associates the machine name with the measurement. For example, a file IO rate on the machine fender will be named File IO Rate:fender. This enables you to apply Graph settings by the machine name. For more information, see "Filtering and Sorting Graph Data" on page 93.
Advanced	For more information, see "Advanced Settings Dialog Box" on page 211.
Truncate Imported data to 150% of scenario run time	In certain cases, the external monitor may have collected data over a time period that was larger than the actual load test. This option deletes data that was collected while the load test was not running, limiting the data collection period to 150% of the load testing period.

Part II

Analysis Graphs

11

Transaction Graphs

This chapter includes:

Concepts

Reference

Concepts

Transaction Graphs Overview

During load test scenario execution, Vusers generate data as they perform transactions. 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 Analysis, see section Working with Analysis on page 25.

Reference

Transaction Graphs User Interface

This section includes:

Average Transaction Response Time Graph

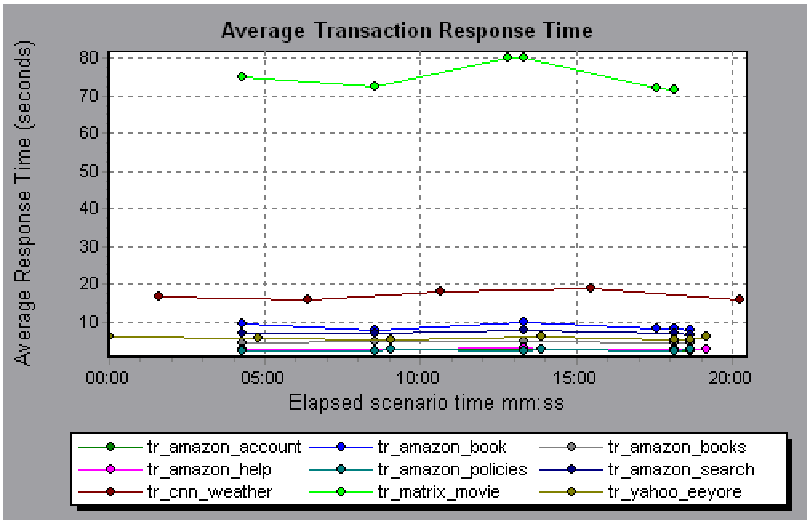
This graph displays the average time taken to perform transactions during each second of the load test scenario run.

Purpose	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.
X-axis	Elapsed time since the start of the run.
Y-axis	Average response time (in seconds) of each transaction

Breakdown options	Transaction Breakdown You can view a breakdown of a transaction by right-clicking the transaction in the graph and selecting **Show Transaction Breakdown Tree**. In the Transaction Breakdown Tree, right-click the transaction you want to break down, and select **Break Down <transaction name>**. The Average Transaction Response Time graph displays data for the sub-transactions. For more details, see "Transaction Breakdown Tree" on page 224. Web Page Breakdown To view a breakdown of the Web page(s) included in a transaction or sub-transaction, right-click it and select **Web Page Diagnostics for <transaction name>**. For more information on the Web Page Diagnostics graphs, see "Web Page Diagnostics Graphs" on page 279.”
Tips	Granularity 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 "Change the granularity of the data" on page 127. Compare with Running Vusers You can compare the Average Transaction Response Time graph to the Running Vusers graph to see how the number of running Vusers affects the transaction performance time. For example, if the Average Transaction Response Time 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.

Note	By default, only transactions that passed are displayed.
See also	"Transaction Graphs Overview" on page 220

Example

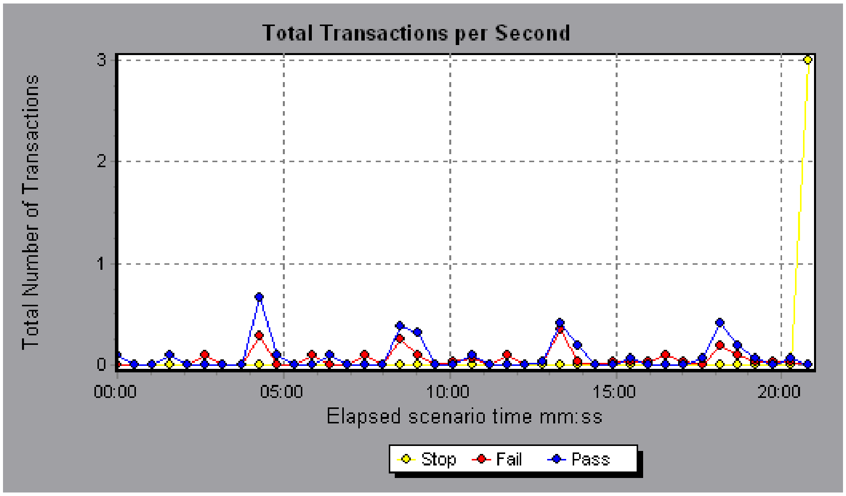


Total Transactions per Second Graph

This graph displays the total number of transactions that passed, the total number of transactions that failed, and the total number of transactions that were stopped, during each second of a load test scenario run.

Purpose	Helps you determine the actual transaction load on your system at any given moment.
X-axis	Elapsed time since the start of the run.
Y-axis	Total number of transactions performed during the scenario run.
See also	"Transaction Graphs Overview" on page 220

Example



Transaction Breakdown Tree

The Transaction Breakdown Tree displays a tree view of the transactions and sub-transactions in the current session. From the tree, you can breakdown transactions and view the results of the breakdown in either the Average Transaction Response Time or Transaction Performance Summary graph.

To access	In either the Average Transaction Response Time or Transaction Performance Summary graph, right-click in the graph and select **Show Transaction Breakdown Tree.**
Important information	After you breakdown a transaction, you can return to the original transaction graph by reapplying the global filter (**File** > **Set Global Filter**) or by undoing your breakdown actions using **Edit** > **Undo Last Action**.

User interface elements are described below:

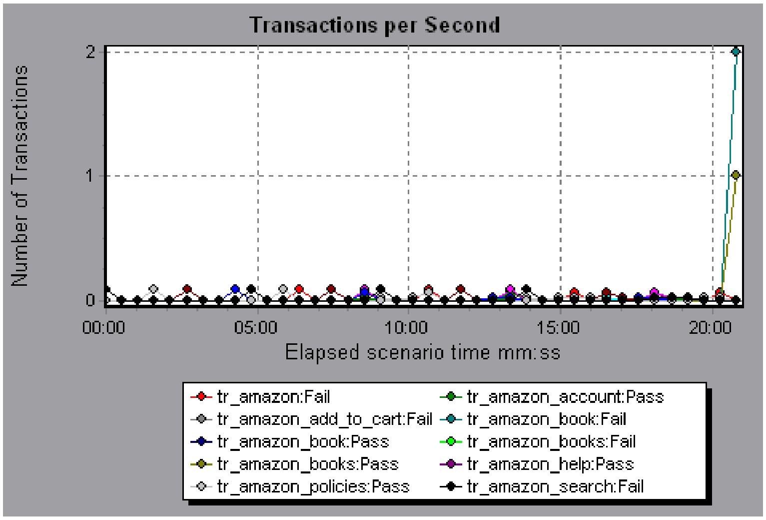
UI Elements	Description
<Right-click menu>	➤ **Break Down From Highest Level.** Displays data for the highest level hierarchical path of a transaction. ➤ **Break Down <transaction name>.** Displays data for the sub-transactions in the Average Transaction Response Time or Transaction Performance Summary graph. ➤ **Show Only <transaction name>.** Displays data only for the selected transaction/sub-transaction. ➤ **Web Page Diagnostics for <page name>.** Displays a breakdown of the Web page(s) included in a transaction or sub-transaction in the Web Page Diagnostics graphs. For details, see "Web Page Diagnostics Graphs" on page 279.

Transactions per Second Graph

This graph displays, for each transaction, the number of times it passed, failed, and stopped during each second of a load test scenario run.

Purpose	Helps you determine the actual transaction load on your system at any given moment.
X-axis	Elapsed time since the start of the run.
Y-axis	Number of transactions performed during the scenario run.
Tips	Compare with Average Transaction Response Time Graph You can compare this graph to the Average Transaction Response Time graph in order to analyze the effect of the number of transactions on the performance time.
See also	"Transaction Graphs Overview" on page 220

Example



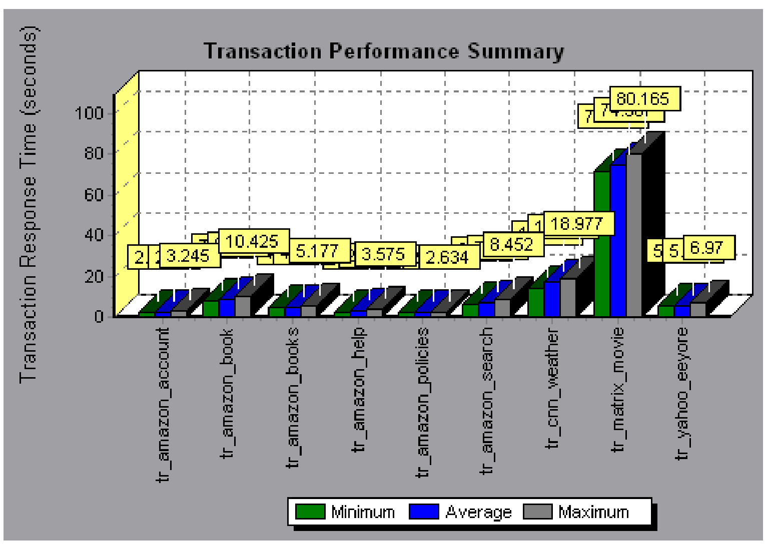
Transaction Performance Summary Graph

This graph displays the minimum, maximum and average performance time for all the transactions in the load test scenario.

X-axis	Name of the transaction.
Y-axis	Response time—rounded off to the nearest second—of each transaction.

Breakdown options	Transaction Breakdown You can view breakdown of a transaction in the Transaction Performance Summary graph by right-clicking the transaction in the graph and selecting **Show Transaction Breakdown Tree**. In the Transaction Breakdown Tree, right-click the transaction you want to break down, and select **Break Down <transaction name>**. The Transaction Performance Summary graph displays data for the sub-transactions. For more details, see "Transaction Breakdown Tree" on page 224. Web Page Breakdown To view a breakdown of the Web page(s) included in a transaction or sub-transaction, right-click it and select **Web Page Diagnostics for <transaction name>**.
See also	"Transaction Graphs Overview" on page 220

Example



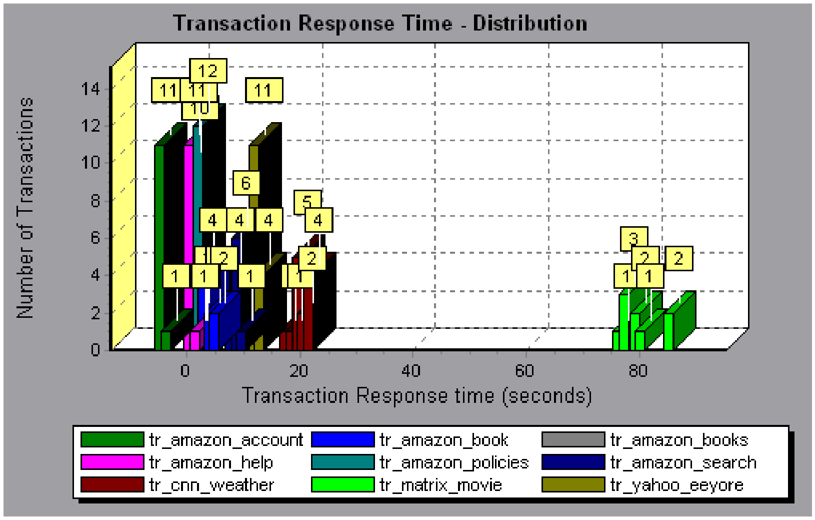
Transaction Response Time (Distribution) Graph

This graph displays the distribution of the time taken to perform transactions in a load test scenario.

Purpose	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.
X-axis	Transaction response time (rounded down to the nearest second).
Y-axis	Number of transactions executed during the scenario.
Tips	Compare with Transaction Performance Summary Graph If you compare it with the Transaction Performance Summary graph, you can see how the average performance was calculated.
Note	This graph can only be displayed as a bar graph.
See also	"Transaction Graphs Overview" on page 220

Example

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



Transaction Response Time (Percentile) Graph

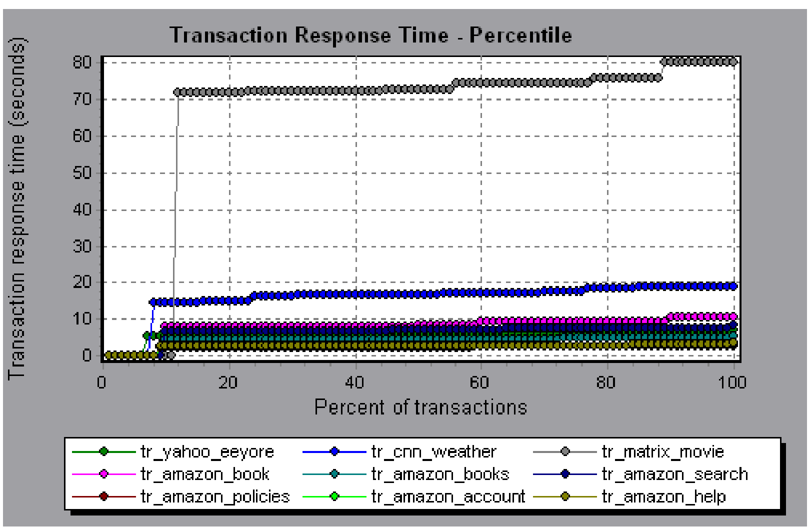
This graph analyzes the percentage of transactions that were performed within a given time range.

Purpose	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.
X-axis	Percentage of the total number of transactions measured during the load test scenario run.

Y-axis	Maximum transaction response time (in seconds). **Note:** Analysis approximates the transaction response time for each available percentage of transactions. The y-axis values, therefore, may not be exact.
Tips	Compare with Average Response Time Graph It is recommended to compare the Percentile graph with a graph indicating average response time such as the Average Transaction Response Time 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.
See also	"Transaction Graphs Overview" on page 220

Example

In the following example, fewer than 20 percent of the tr_matrix_movie transactions had a response time less than 70 seconds.

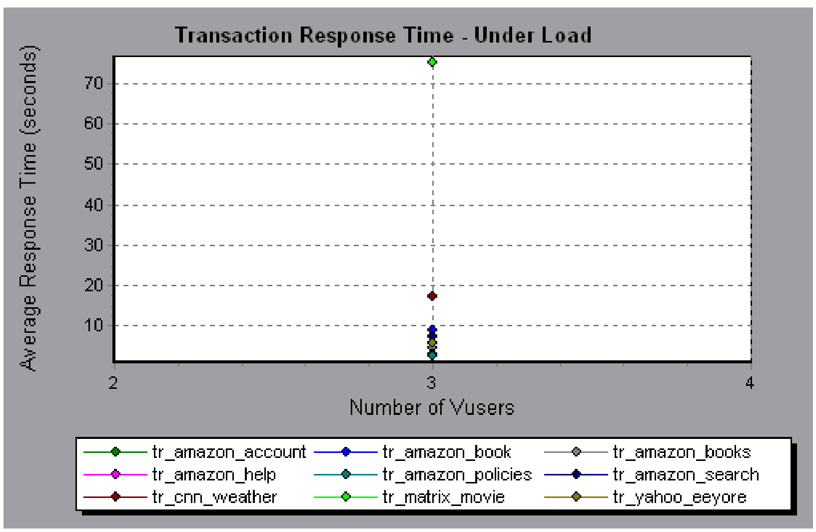


Transaction Response Time (Under Load) Graph

This graph is a combination of the Running Vusers and Average Transaction Response Time graphs and indicates transaction times relative to the number of Vusers running at any given point during the load test scenario.

Purpose	Helps you view the general impact of Vuser load on performance time and is most useful when analyzing a scenario with a gradual load.
X-axis	Number of running Vusers
Y-axis	Average response time (in seconds) of each transaction.
See also	"Transaction Graphs Overview" on page 220

Example

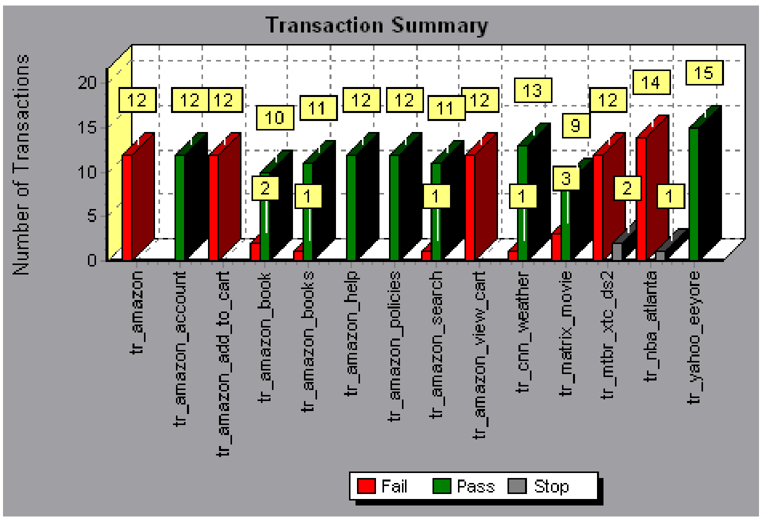


Transaction Summary Graph

This graph summarizes the number of transactions in the load test scenario that failed, passed, stopped, and ended in error.

X-axis	Name of the transaction
Y-axis	Number of transactions performed during the scenario run.
See also	"Transaction Graphs Overview" on page 220

Example



12

Vuser Graphs

This chapter includes:

Concepts

Reference

Concepts

Vuser Graphs Overview

During load test 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. For more information about Transaction graphs, see "Transaction Graphs" on page 219.

Reference

Vuser Graphs User Interface

This section includes (in alphabetical order):

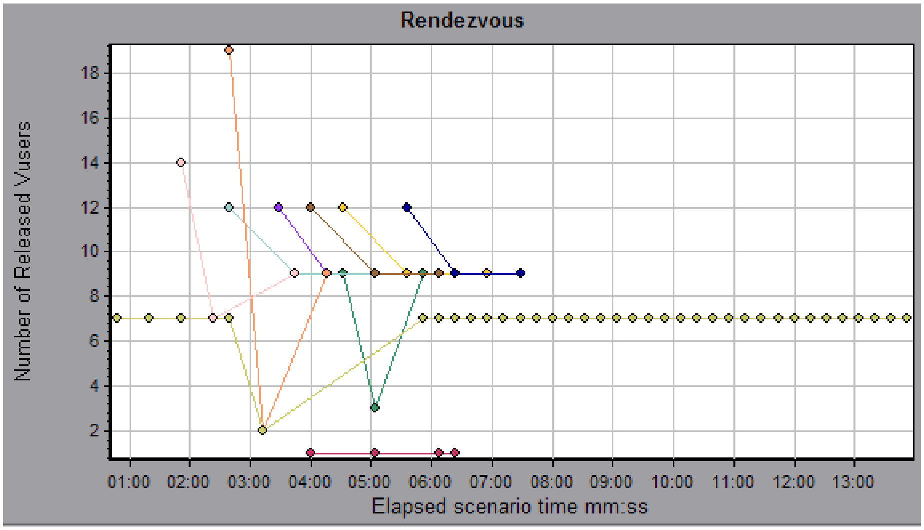
Rendezvous Graph

During a scenario run, you can instruct multiple Vusers to perform tasks simultaneously by using **rendezvous points**. A rendezvous point creates intense user load on the server and enables LoadRunner to measure server performance under load. For more information about using rendezvous points, see the *HP Virtual User Generator User Guide*.

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

Purpose	Helps you understand transaction performance times.
X-axis	Elapsed time since the start of the run.
Y-axis	Number of Vusers that were released from the rendezvous.
Tips	Compare to Average Transaction Response Time graph If you compare the Rendezvous graph to the Average Transaction Response Time graph, you can see how the load peak created by a rendezvous influences transaction times.
See also	"Vuser Graphs Overview" on page 234

Example

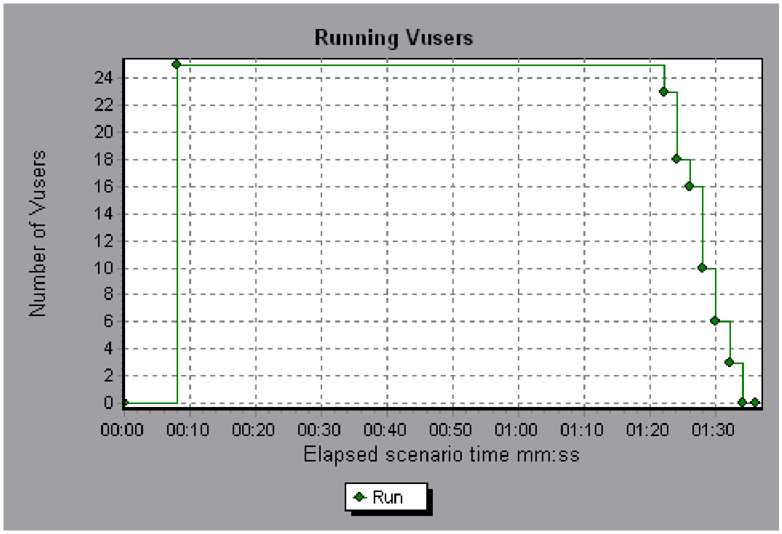


Running Vusers Graph

This graph displays the number of Vusers that executed Vuser scripts and their status during each second of the test.

Purpose	Helps you determine the Vuser load on your server at any given moment.
X-axis	Elapsed time since the start of the run.
Y-axis	Number of Vusers in the scenario.
Note	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.
See also	"Vuser Graphs Overview" on page 234

Example

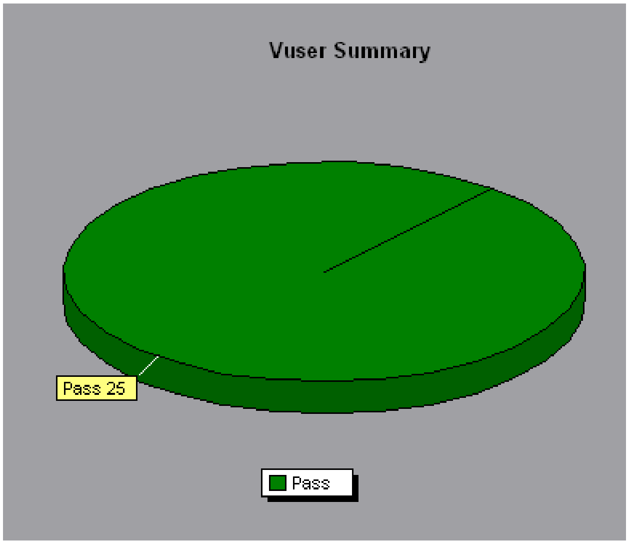


Vuser Summary Graph

This graph displays a summary of Vuser performance.

Purpose	Lets you view the number of Vusers that successfully completed the load test scenario run relative to those that did not.
Note	This graph may only be viewed as a pie.
See also	"Vuser Graphs Overview" on page 234

Example



13

Error Graphs

This chapter includes:

Concepts

Reference

Concepts

Error Graphs Overview

During load test scenario execution, Vusers may not complete all transactions successfully. The Error graphs let you view information about the transactions that failed, stopped, 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.

Reference

Error Graphs User Interface

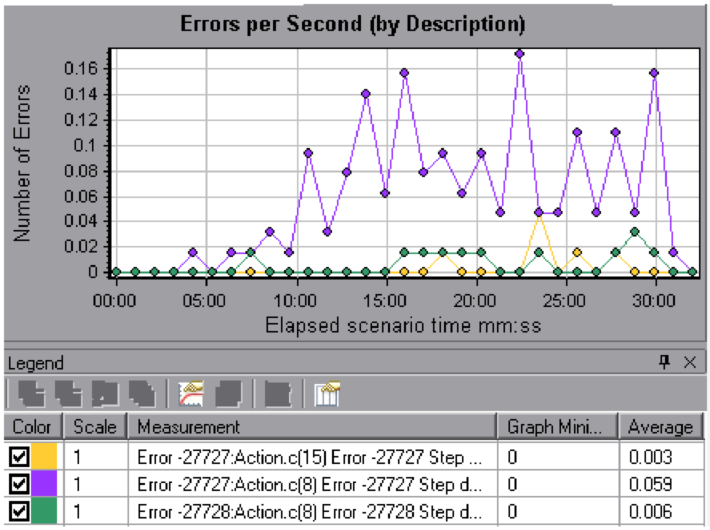
This section includes (in alphabetical order):

Errors per Second (by Description) Graph

This graph displays the average number of errors that occurred during each second of the load test scenario run, grouped by error description. The error description is displayed in the legend.

X-axis	Elapsed time since the start of the run.
Y-axis	Number of errors.
See also	"Error Graphs Overview" on page 240

Example

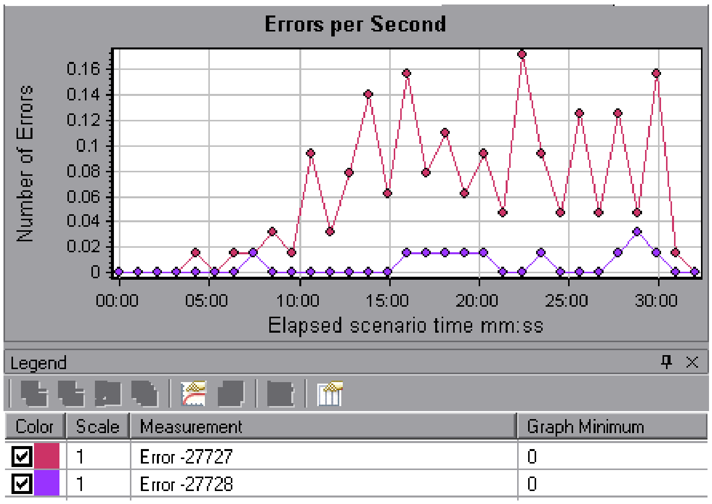


Errors per Second Graph

This graph displays the average number of errors that occurred during each second of the load test scenario run, grouped by error code.

X-axis	Elapsed time since the start of the run.
Y-axis	Number of errors.
See also	"Error Graphs Overview" on page 240

Example

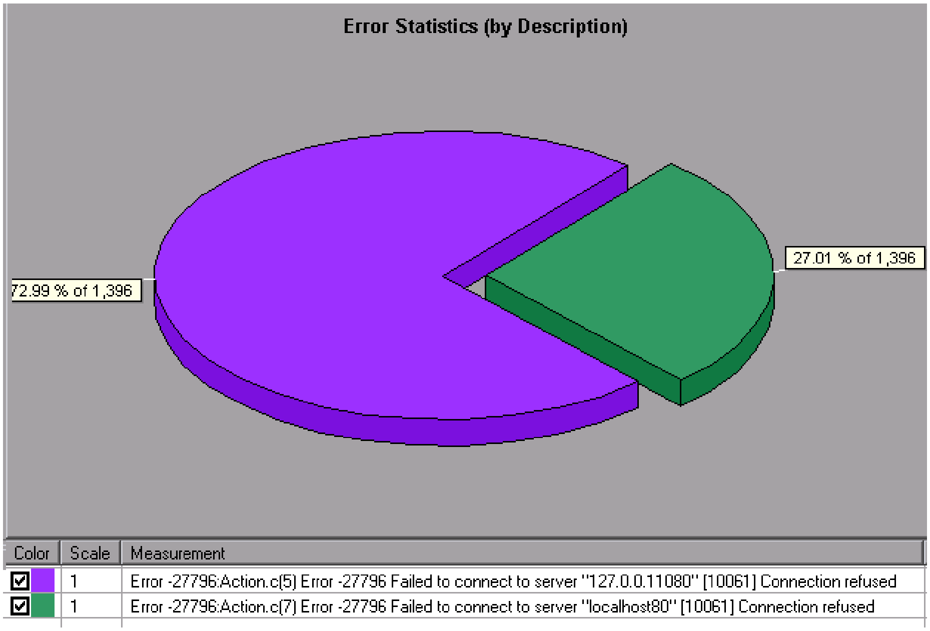


Error Statistics (by Description) Graph

This graph displays the number of errors that accrued during load test scenario execution, grouped by error description. The error description is displayed in the legend.

Note	This graph may only be viewed as a pie.
See also	"Error Graphs Overview" on page 240

Example



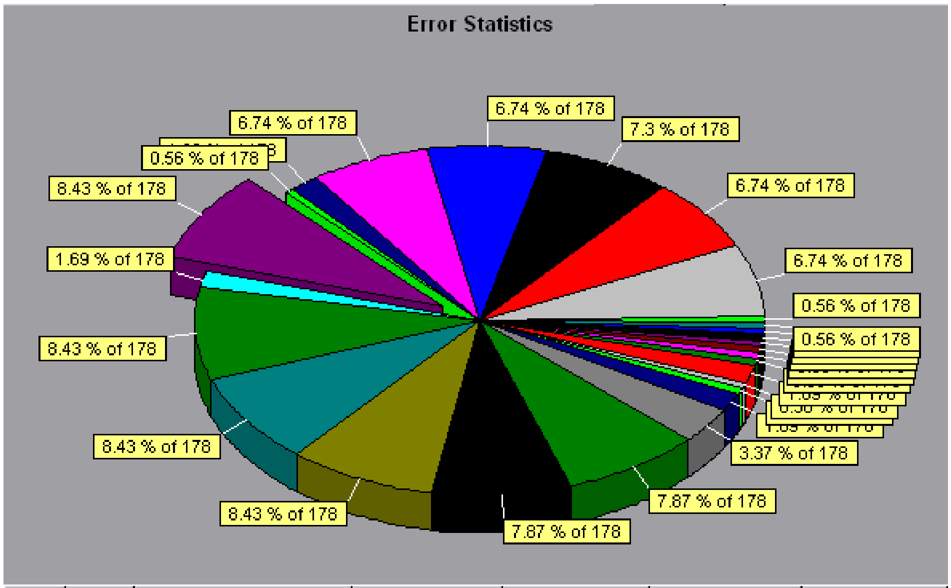
Error Statistics Graph

This graph displays the number of errors that accrued during load test scenario execution, grouped by error code.

Note	This graph may only be viewed as a pie.
See also	"Error Graphs Overview" on page 240

Example

In the following example, 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% of the errors.



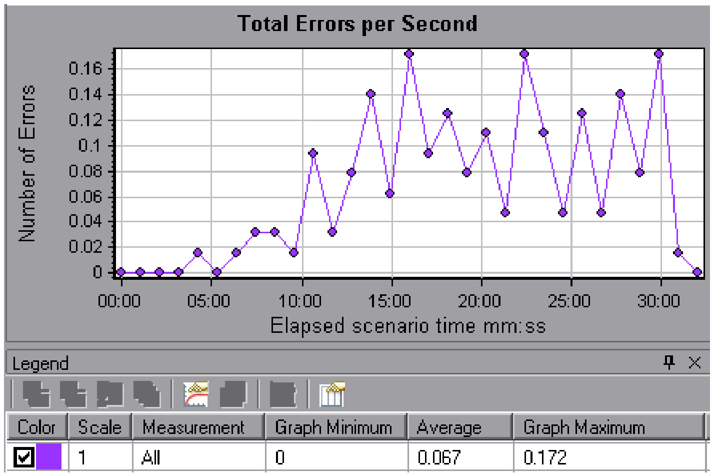
Color	Scale	Measurement	Graph's Minimum	Graph's Average	Graph's Maximum	Graph's Median
	1	Error -26627:-26627 Ac...	13	13	13	13
	1	Error -26627:-26627 Ac...	12	12	12	12
	1	Error -26627:-26627 Ac...	12	12	12	12
	1	Error -26627:-26627 Ac...	3	3	3	3
	1	Error -26627:-26627 Ac...	1	1	1	1

Total Errors per Second Graph

This graph displays the average number of errors that occurred during each second of the load test scenario run.

X-axis	Elapsed time since the start of the run.
Y-axis	Number of errors.
See also	"Error Graphs Overview" on page 240

Example


Total Errors per Second
Number of Errors
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
00:00
05:00
10:00
15:00
20:00
25:00
30:00
Elapsed scenario time mm:ss
Legend
Color
Scale
Measurement
Graph Minimum
Average
Graph Maximum
1
All
0
0.067
0.172

14

Web Resources Graphs

This chapter includes:

Concepts

Reference

Concepts

Web Resources Graphs Overview

Web Resource graphs provide you with information about the performance of your Web server. You use the Web Resource graphs to analyze the following data:

- Throughput on the Web server
- The number of hits per second
- The number of HTTP responses per second
- The HTTP status codes returned from the Web server
- The number of downloaded pages per second
- The number of server retries per second
- A summary of the server retries during the load test scenario
- The number of open TCP/IP connections
- The number of new TCP/IP connections opened
- The number of connections that are shut down
- The number of new and reused SSL connections opened

Note: Due to the asynchronous nature of the Ajax TruClient protocol, data collected and displayed in Web Resources graphs may not represent actual performance.

Reference

HTTP Status Codes

The following table displays a list of HTTP status codes:

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

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

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

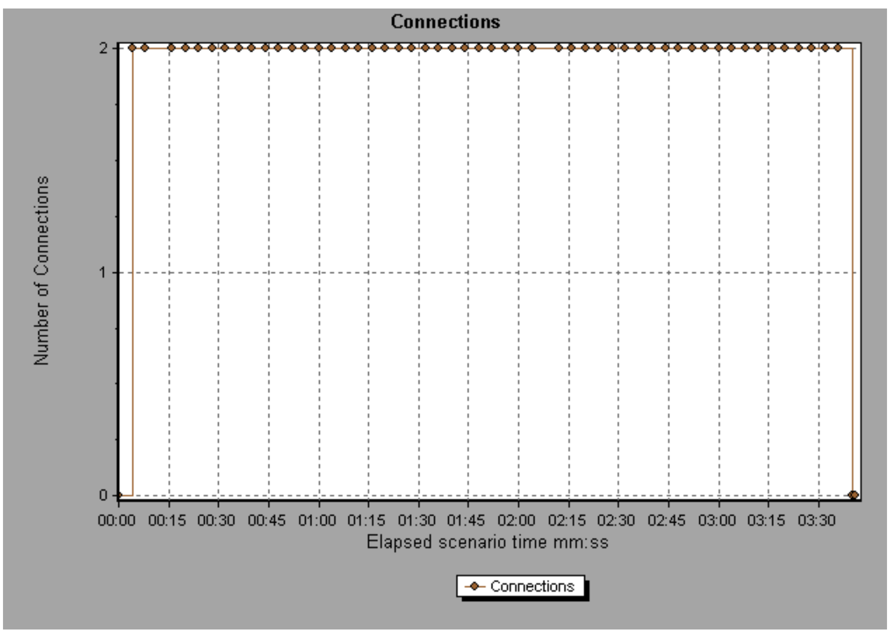
Web Resources Graphs User Interface

This section includes:

Connections Graph

This graph shows the number of open TCP/IP connections (y-axis) at each point in time of the load test scenario (x-axis). One HTML page may cause the browser to open several connections, when links on the page go to different Web addresses. Two connections are opened for each Web server.

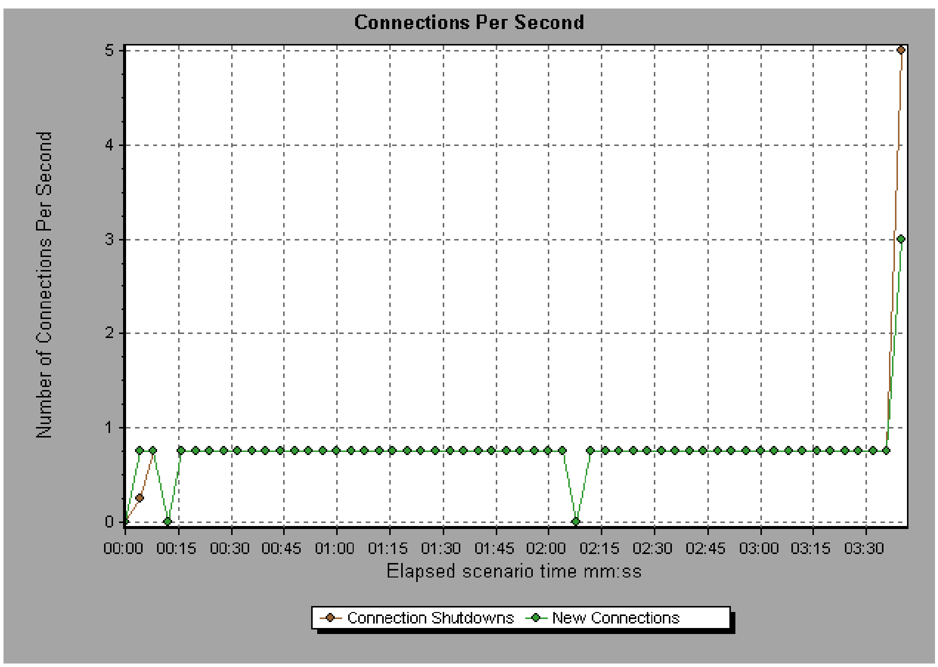
Purpose	This graph is useful in indicating when additional connections are needed. For example, if the number of connections reaches a plateau, and the transaction response time increases sharply, adding connections would probably cause a dramatic improvement in performance (reduction in the transaction response time).
X-axis	Elapsed time since the start of the run.
Y-axis	Open TCP/IP connections.
See also	"Web Resources Graphs Overview" on page 248



Connections per Second Graph

This graph shows the number of new TCP/IP connections (y-axis) opened and the number of connections that are shut down for each second of the load test scenario (x-axis).

X-axis	Elapsed time since the start of the run.
Y-axis	TCP/IP connections per second.
Tips	New connections versus hits per second: The number of new connections should be a small fraction of the number of hits per second, because new TCP/IP connections are very expensive in terms of server, router and network resource consumption. Ideally, many HTTP requests should use the same connection, instead of opening a new connection for each request.
See also	"Web Resources Graphs Overview" on page 248



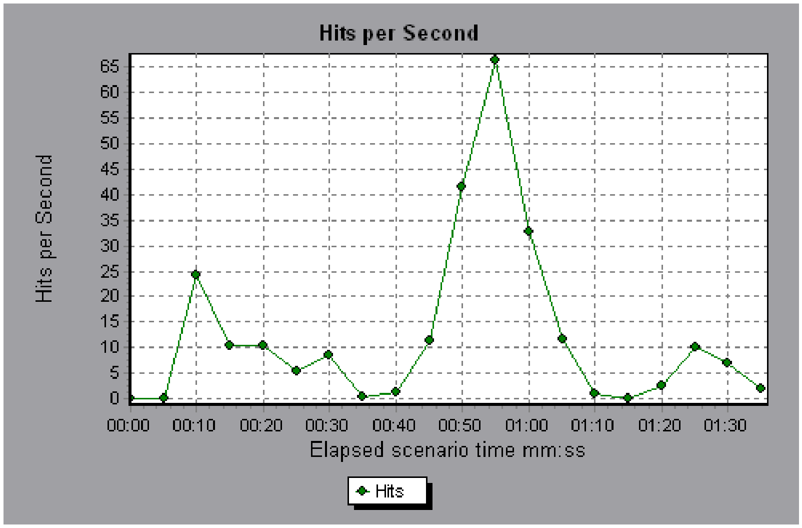
Hits per Second Graph

This graph shows the number of HTTP requests made by Vusers to the Web server during each second of the load test scenario run.

Purpose	Helps you evaluate the amount of load Vusers generate, in terms of the number of hits.
X-axis	Elapsed time since the start of the run.
Y-axis	Number of hits on the server.
Tips	Compare to Average Transaction Response Time graph: You can compare this graph to the Average Transaction Response Time graph to see how the number of hits affects transaction performance.
Note	You cannot change the granularity of the x-axis to a value that is less than the Web granularity you defined in the General tab of the Options dialog box.
See also	"Web Resources Graphs Overview" on page 248

Example

In the following example, the most hits per second took place during the fifty-fifth second of the scenario.



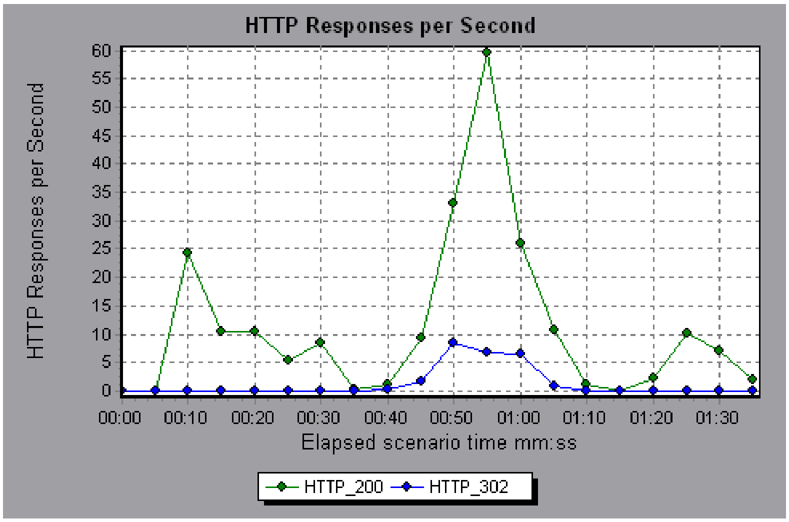
HTTP Responses per Second Graph

This graph shows the number of HTTP status codes returned from the Web server during each second of the load test scenario run, grouped by status code. HTTP status codes indicate the status of HTTP requests, for example, "the request was successful", "the page was not found”.

X-axis	Elapsed time since the start of the run.
Y-axis	Number of HTTP responses per second.
Tips	Locate scripts which generated error codes: You can group the results shown in this graph by script (using the "Group By" function) to locate scripts which generated error codes.
See also	➤ "Web Resources Graphs Overview" on page 248 ➤ "HTTP Status Codes" on page 249

Example

In the following example, 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.



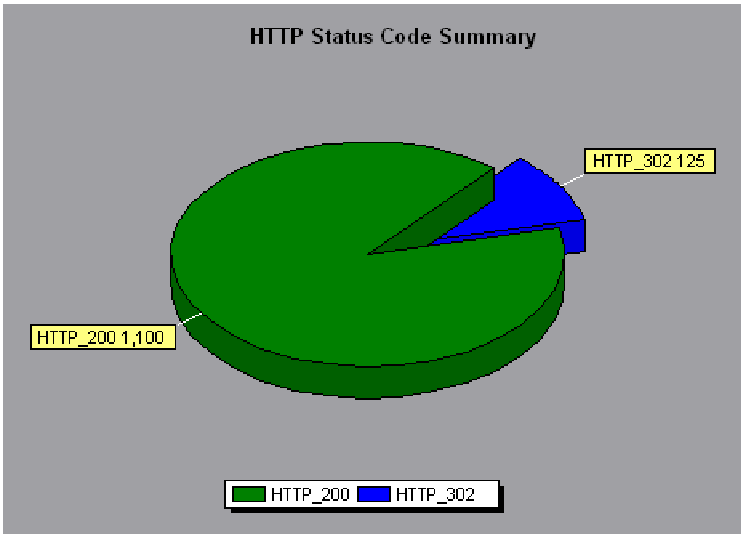
HTTP Status Code Summary Graph

This graph shows the number of HTTP status codes returned from the Web server during the load test scenario run, grouped by status code. HTTP status codes indicate the status of HTTP requests, for example, "the request was successful","the page was not found”.

Tips	Locate scripts which generated error codes: Use this graph together with the HTTP Responses per Second graph to locate those scripts which generated error codes.
Note	This graph can only be viewed as a pie.
See also	➤ "Web Resources Graphs Overview" on page 248 ➤ "HTTP Status Codes" on page 249

Example

In the following example, the 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.



Pages Downloaded per Second Graph

This graph shows the number of Web pages downloaded from the server during each second of the load test scenario run.

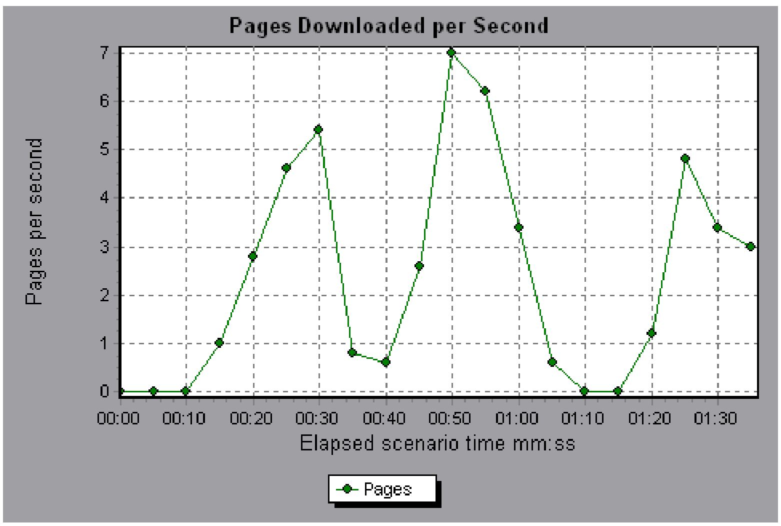
Like the Throughput graph, the Pages Downloaded per Second graph represents the amount of data that the Vusers received from the server at any given second. However, the Throughput graph takes into account each resource and its size (for example, the size of each **.gif** file, the size of each Web page). The Pages Downloaded per Second graph takes into account only the number of pages.

Purpose	Helps you evaluate the amount of load Vusers generate, in terms of the number of pages downloaded.
X-axis	Elapsed time since the start of the run.

Y-axis	Number of Web pages downloaded from the server.
Note	To view the Pages Downloaded per Second graph, you must select **Pages per second (HTML Mode only)** from the run time settings Preferences tab before running your scenario.
See also	"Web Resources Graphs Overview" on page 248

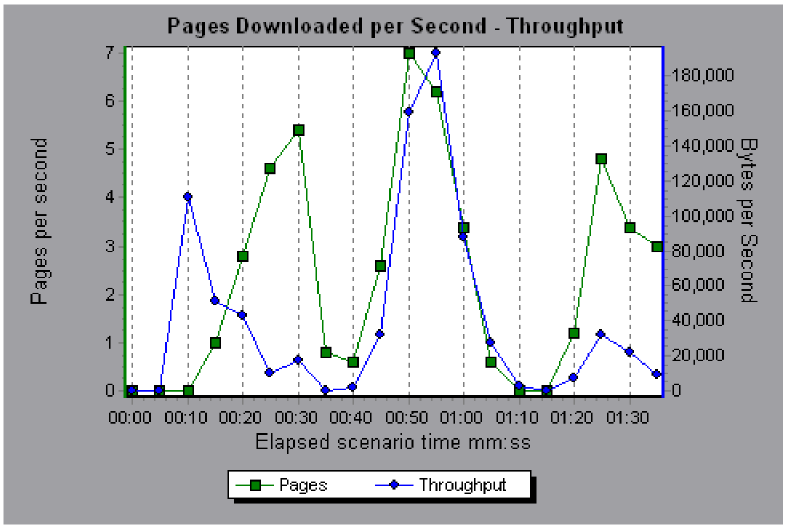
Example1

In the following example, the greatest number of pages downloaded per second, about 7, occurred in the fiftieth second of the scenario run.



Example 2

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.



Retries per Second Graph

This graph displays the number of attempted server connections during each second of the load test scenario run. A server connection is retried when:

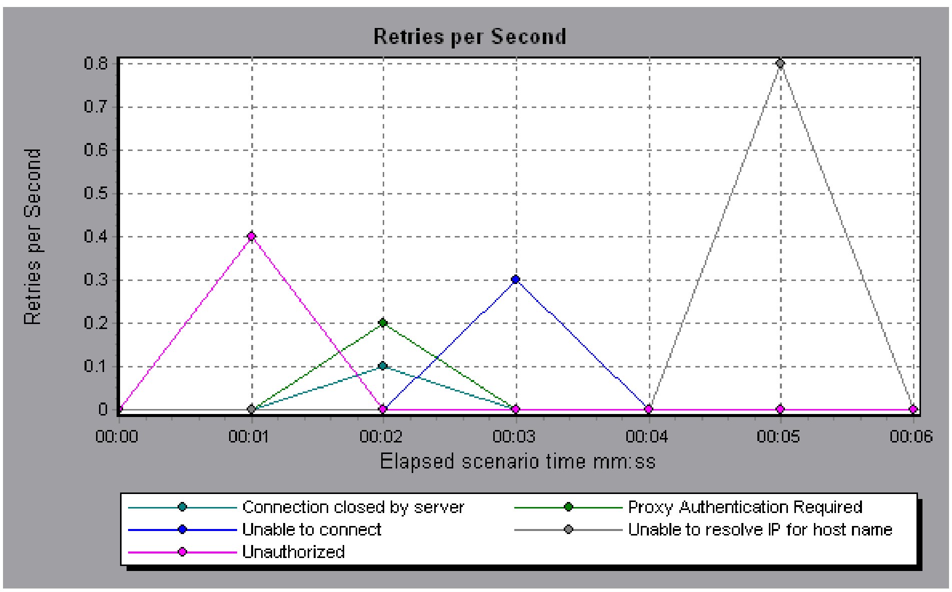
- the initial connection was unauthorized
- proxy authentication is required
- the initial connection was closed by the server

- the initial connection to the server could not be made
- when the server was initially unable to resolve the load generator's IP address

X-axis	Elapsed time since the start of the run.
Y-axis	Number of server retries per second.
See also	"Web Resources Graphs Overview" on page 248

Example

In the following example, the graph shows that during the first second of the scenario, the number of retries was 0.4, whereas in the fifth second of the scenario, the number of retries per second rose to 0.8.



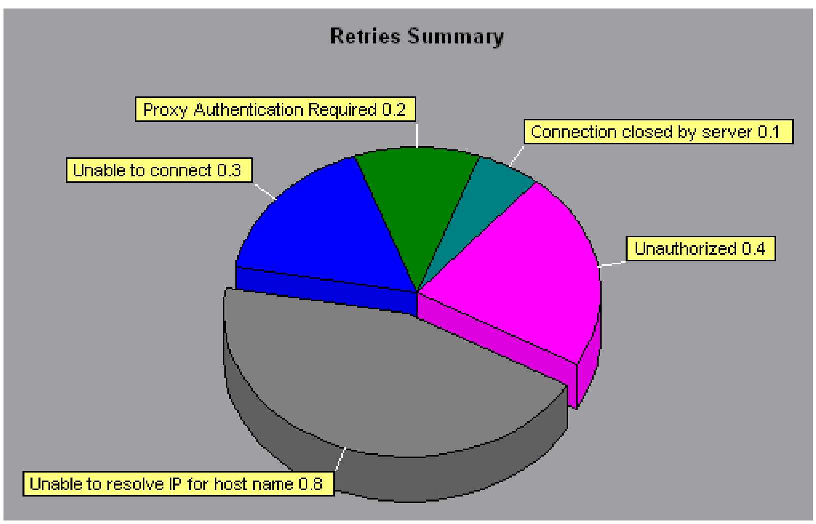
Retries Summary Graph

This graph shows the number of attempted server connections during the load test scenario run, grouped by the cause of the retry.

Tips	Determine when server retries were attempted: Use this graph together with the Retries per Second graph to determine at what point during the scenario the server retries were attempted.
Note	This graph may only be viewed as a pie.
See also	"Web Resources Graphs Overview" on page 248

Example

In the following example, the graph shows that the server's inability to resolve the load generator's IP address was the leading cause of server retries during the scenario run.

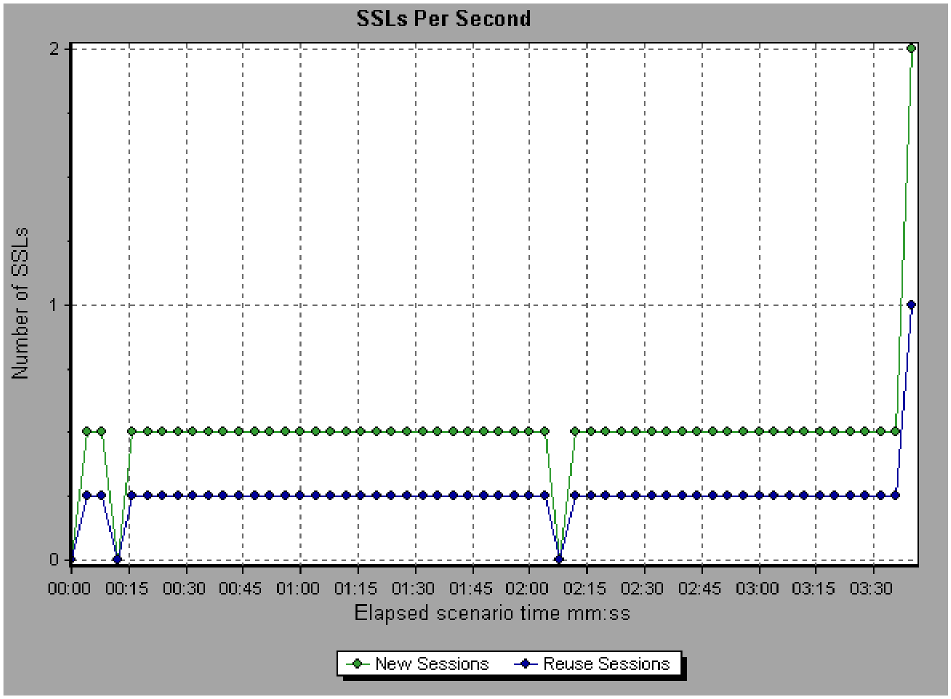


SSLs per Second Graph

This graph shows the number of new and reused SSL Connections (y-axis) opened in each second of the load test scenario (x-axis). An SSL connection is opened by the browser after a TCP/IP connection has been opened to a secure server.

X-axis	Elapsed time since the start of the run.
Y-axis	Number of SSL Connections
Tips	Reduce SSL connections: Creating a new SSL connection entails heavy resource consumption. Therefore, you should try to open as few new SSL connections as possible. Once you've established an SSL connection, you should reuse it. There should be no more than one new SSL connection per Vuser. In cases where you reset TCP connections between iterations (**VuGen Run-Time Settings** > **Browser Emulation node** > **Simulate a new user on each iteration**), you should have no more than one new SSL connection per iteration.
See also	"Web Resources Graphs Overview" on page 248

Example



Throughput Graph

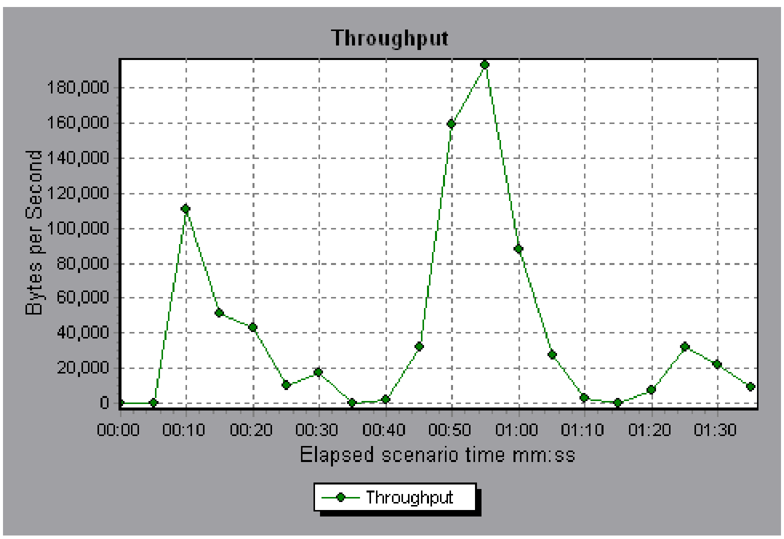
This graph shows the amount of throughput on the server during each second of the load test scenario run. Throughput is measured in bytes or megabytes and represents the amount of data that the Vusers received from the server at any given second. To view throughput in megabytes, use the **Throughput (MB)** graph.

Purpose	Helps you evaluate the amount of load that Vusers generate, in terms of server throughput.
X-axis	Elapsed time since the start of the scenario run.
Y-axis	Throughput of the server, in bytes or megabytes.

Tips	Compare to Average Transaction Response Time graph: You can compare this graph to the Average Transaction Response Time graph to see how the throughput affects transaction performance.
Note	You cannot change the granularity of the x-axis to a value that is less than the Web granularity you defined in the General tab of the Options dialog box.
See also	"Web Resources Graphs Overview" on page 248

Example

In the following example, the highest throughput is 193,242 bytes during the fifty-fifth second of the scenario.



15

User-Defined Data Point Graphs

This chapter includes:

Concepts

Reference

Concepts

User-Defined Data Point Graphs Overview

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, refer to the online *HP LoadRunner Online Function Reference*.

Data points, like other Analysis data, are aggregated every few seconds, resulting in less data points shown on the graph than actually recorded.

Reference

User-Defined Data Point Graphs User Interface

This section includes (in alphabetical order):

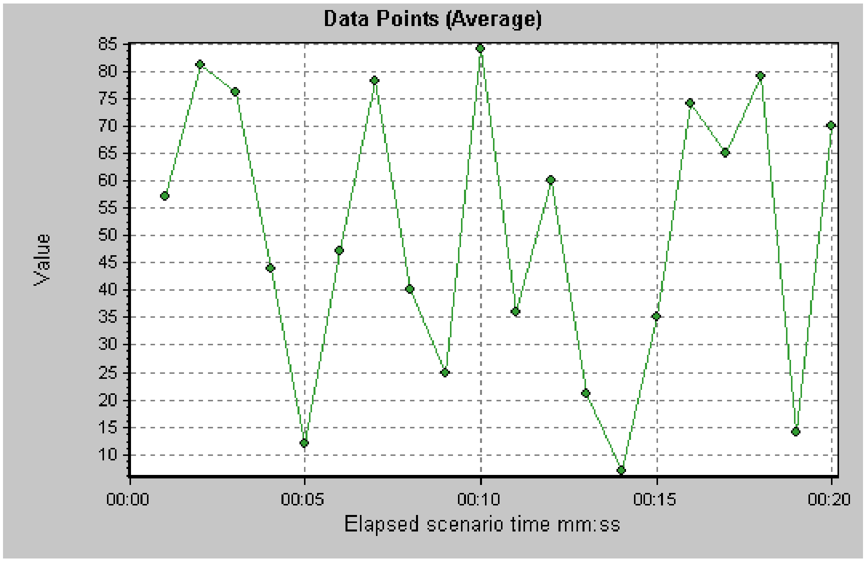
Data Points (Average) Graph

This graph shows the average values that were recorded for user-defined data points during the load test scenario run.

Purpose	This graph is typically used in cases where the actual value of the measurement is required. Suppose that each Vuser monitors CPU utilization on its machine and records it as a data point. In this case, the actual recorded value of CPU utilization is required. The Average graph displays the average value recorded throughout the scenario.
X-axis	Elapsed time since the start of the run.
Y-axis	The average values of the recorded data point statements.
See also	"User-Defined Data Point Graphs Overview" on page 266

Example

In the following example, the CPU utilization is recorded as the data point **user_data_point_val_1**. It is shown as a function of the elapsed scenario time.



Data Points (Sum) Graph

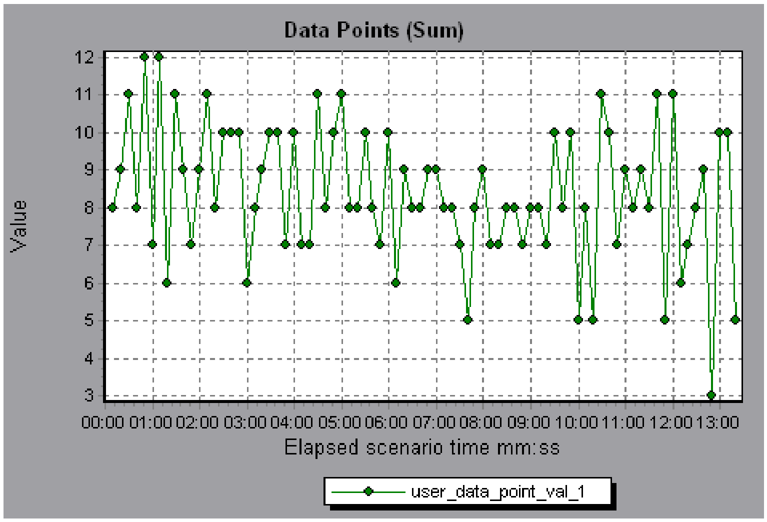
This graph shows the sum of the values for user-defined data points throughout the load test scenario run.

This graph typically indicates the total amount of measurements which all Vusers are able to generate. For example, suppose only a certain set of circumstances allow a Vuser to call a server. Each time it does, a data point is recorded. In this case, the Sum graph displays the total number of times that Vusers call the function.

X-axis	Elapsed time since the start of the run.
Y-axis	The sum of the recorded data point values.
See also	"User-Defined Data Point Graphs Overview" on page 266

Example

In the following example, the call to the server is recorded as the data point user_data_point_val_1. It is shown as a function of the elapsed scenario time.



16

Network Monitor Graphs

This chapter includes:

Concepts

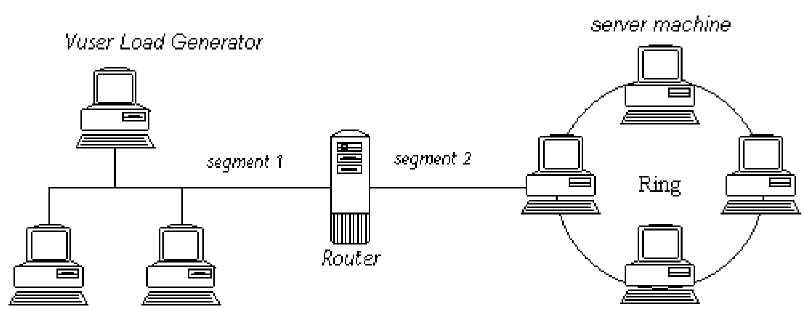
Reference

Concepts

Network Monitor Graphs Overview

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.

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



To measure network performance, the Network monitor sends packets of data across the network. When a packet returns, the monitor calculates the time it takes for the packet to go to the requested node and return.

The Network Sub-Path Time graph displays the delay from the source machine to each node along the path. The Network Segment Delay 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 Analysis to generate Network monitor graphs, you must activate the Network monitor before executing the load test scenario. In the Network monitor settings, you specify the path you want to monitor. For information about setting up the Network monitor, refer to the *HP LoadRunner Controller User Guide*.

Reference

Network Monitor Graphs User Interface

This section includes (in alphabetical order):

Network Delay Time Graph

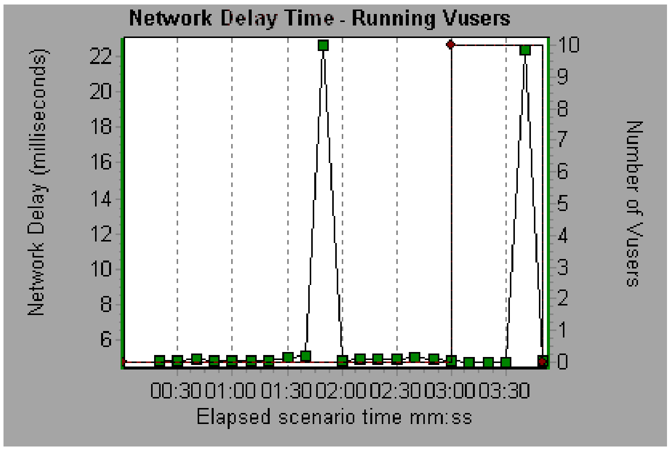
This graph shows the delays for the complete path between the source and destination machines (for example, the database server and Vuser load generator). The graph maps the delay as a function of the elapsed load test scenario time.

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

X-axis	Elapsed time since the start of the run.
Y-axis	Network delay time.
Tips	**Merge graphs to determine network 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.
See also	"Network Monitor Graphs Overview" on page 272

Example

In the following example of a 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.



Network Segment Delay Graph

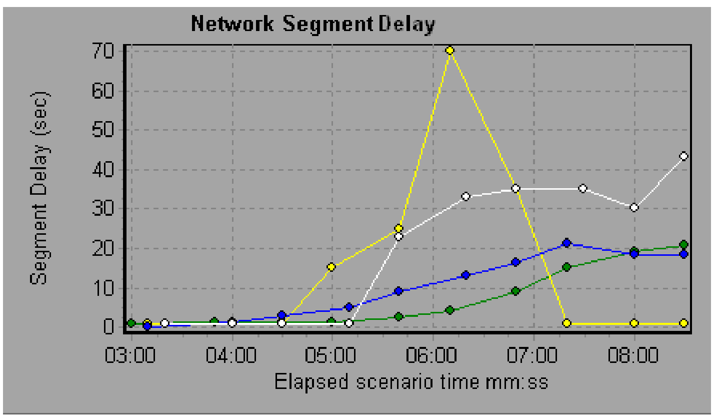
This graph shows the delay for each segment of the path according to the elapsed load test scenario time. Each segment is displayed as a separate line with a different color.

X-axis	Elapsed time since the start of the run.
Y-axis	Network delay time.

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.
See also	"Network Monitor Graphs Overview" on page 272

Example

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



Network Sub-Path Time Graph

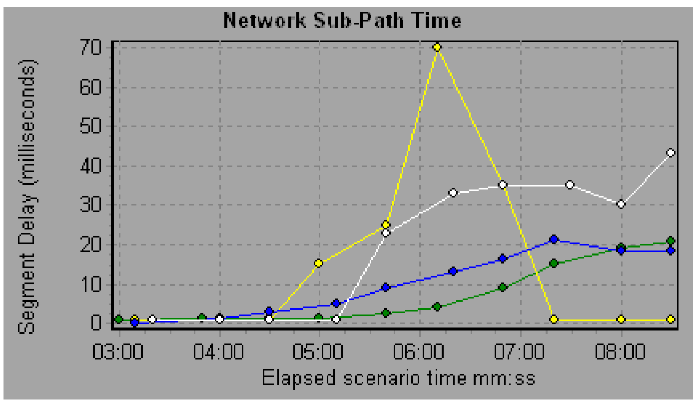
This graph displays the delay from the source machine to each node along the path according to the elapsed load test scenario time. Each segment is displayed as a separate line with a different color.

X-axis	Elapsed time since the start of the run.
Y-axis	Network delay time.

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.
See also	"Network Monitor Graphs Overview" on page 272

Example

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



17

Web Page Diagnostics Graphs

This chapter includes:

Concepts

Tasks

Reference

Concepts

Web Page Diagnostics Tree View Overview

The Web Page Diagnostics tree view displays a tree view of the transactions, sub-transactions, and Web pages for which you can view Web Page Diagnostics graphs. For more information about Web Page Diagnostics graphs, see "Web Page Diagnostics Graphs Overview" on page 280

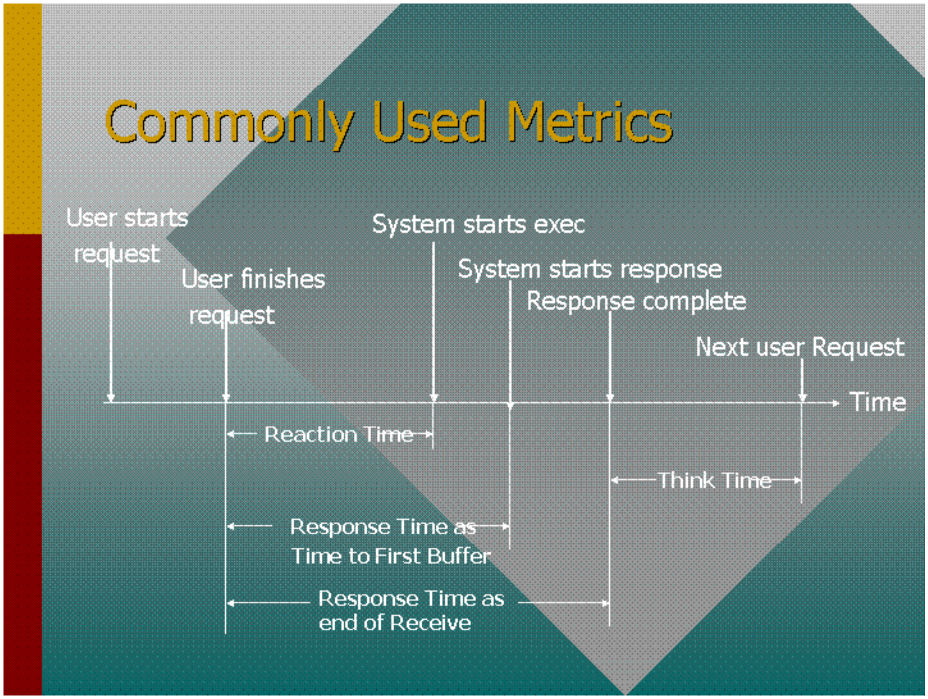
The Web Page Diagnostics graphs enable you to assess whether transaction response times were affected by page content. Using the Web Page Diagnostics graphs, you can analyze problematic elements—for example, images that download slowly, or broken links—of a Web site.

Web Page Diagnostics Graphs Overview

Web Page Diagnostics 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. Analysis displays both average download time and download time over time data.

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

The following diagram illustrates the sequence of events from the time an HTTP request is sent:



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 begin analyzing the Transaction Performance Summary and Average Transaction Response Time graphs with the Web Page Diagnostics graph, which displays the average download time (in seconds) for each monitored Web page during each second of the load test scenario run. The x-axis represents the elapsed time from the beginning of the scenario run. The y-axis represents the average download time (in seconds) for each Web page.

In order for Analysis to generate Web Page Diagnostics graphs, you must enable the Web Page Diagnostics feature in the Controller before recording your script. From the Controller menu, choose **Diagnostics** > **Configuration** and select the **Enable the following diagnostics** check box. Also next to **Web Page Diagnostics (Max. Vuser Sampling: 10%)** click **Enable.**

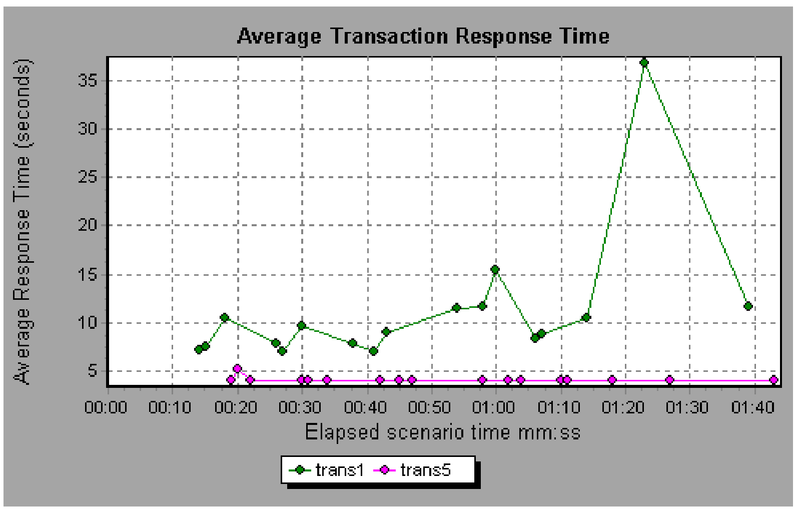
Note: It is recommended that, in VuGen, you select **HTML-based script** in the Recording tab of the Recording Options dialog box.

For more information on recording Web Vuser scripts, refer to the *HP Virtual User Generator User Guide*.

Tasks

How to View the Breakdown of a Transaction

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



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

This task describes how to breakdown from a transaction.

1. Right-click **trans1** and select **Web Page Diagnostics for trans1**. The Web Page Diagnostics graph opens and the Web Page Diagnostics tree appear. An icon appears next to the page name indicating the page content. See "Web Page Diagnostics Content Icons" on page 285.
2. In the Web Page Diagnostics tree, right-click the problematic page you want to break down, and select **Break Down <component name>**. Alternatively, select a page in the **Select Page to Break Down** box that appears under the Web Page Diagnostics graph. The Web Page Diagnostics graph for that page appears.

Note: You can open a browser displaying the problematic page by right-clicking the page in the Web Page Diagnostics tree and selecting **View page in browser**.

---

3. Select one of the following available breakdown options:
   - **Download Time.** Displays a table with a breakdown of the selected page's download time. The size of each page component (including the component's header) is displayed. See the "Page Download Time Breakdown Graph" on page 292 for more information about this display.
   - **Component (Over Time).** Displays the Page Component Breakdown (Over Time) Graph for the selected Web page.
   - **Download Time (Over Time).** Displays the Page Download Time Breakdown (Over Time) Graph for the selected Web page.
   - **Time to First Buffer (Over Time).** Displays the Time to First Buffer Breakdown (Over Time) Graph for the selected Web page.

   To display the graphs in full view, click the  button. You can also access these graphs, as well as additional Web Page Diagnostics graphs, from the Open a New Graph dialog box.

Reference

Web Page Diagnostics Content Icons

The following icons appear in the Web Page Diagnostics tree. They indicate the HTTP content of the page.

Name	Description
	**Transaction**. Specifies that the ensuing content is part of the transaction.
	**Page Content**. Specifies that the ensuing content, which may include text, images, and so on, is all part of one logical page.
	**Text content.** Textual information. Plain text is intended to be displayed as-is. Includes HTML text and style sheets.
	**Multipart content.** Data consisting of multiple entities of independent data types.
	**Message content.** An encapsulated message. Common subtypes are news, or external-body which specifies large bodies by reference to an external data source.
	**Application content.** Some other kind of data, typically either uninterpreted binary data or information to be processed by an application. An example subtype is Postscript data.
	**Image content.** Image data. Two common subtypes are the jpeg and gif format.
	**Resource content.** Other resources not listed above. Also, content that is defined as "not available" are likewise included.

Web Page Diagnostics Graphs User Interface

This section includes (in alphabetical order):

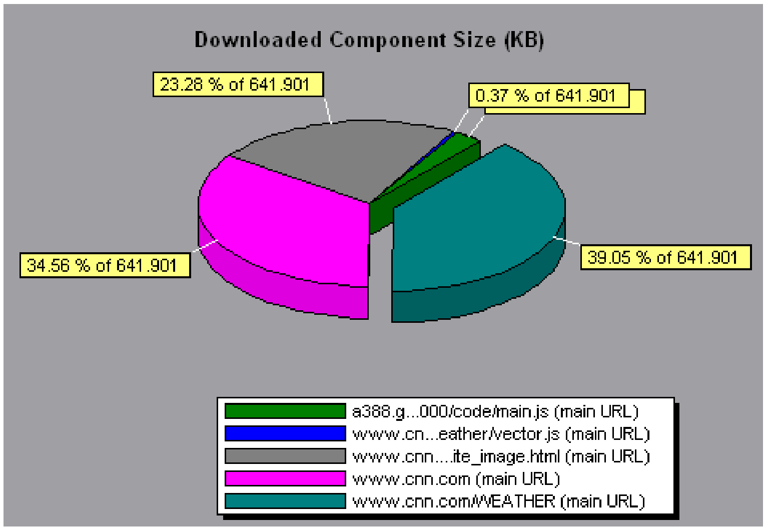
Downloaded Component Size Graph

This graph displays the size of each Web page component.

Note	➤ The Web page size is a sum of the sizes of each of its components. ➤ The Downloaded Component Size graph can only be viewed as a pie graph.
See also	"Web Page Diagnostics Graphs Overview" on page 280

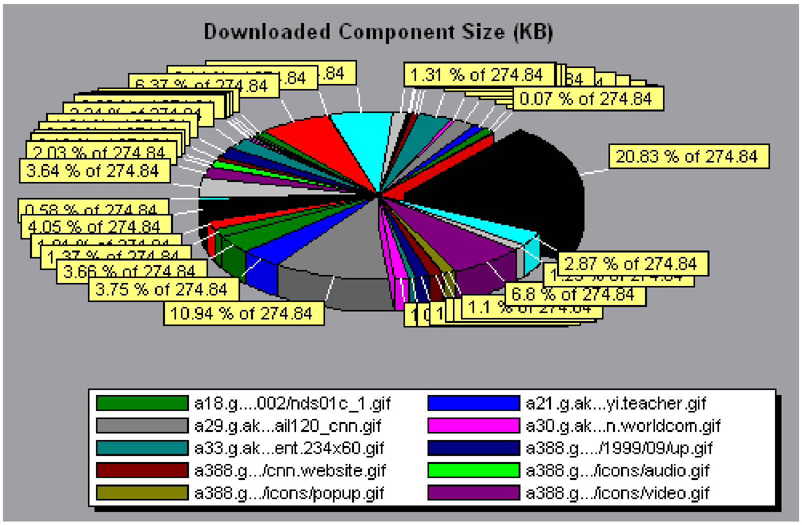
Example

In the following example 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.



Example

In the following 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.



Page Component Breakdown Graph

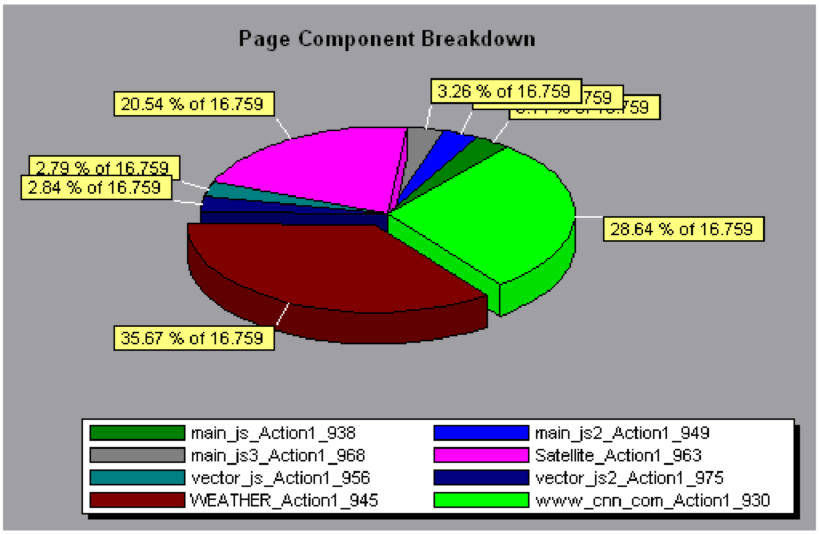
This graph displays the average download time (in seconds) for each Web page and its components.

Breakdown options	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 Diagnostics tree.
Tips	To isolate problematic components, 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 **Graph's Average** column.

Note	The graph can only be viewed as a pie.
See also	"Web Page Diagnostics Graphs Overview" on page 280

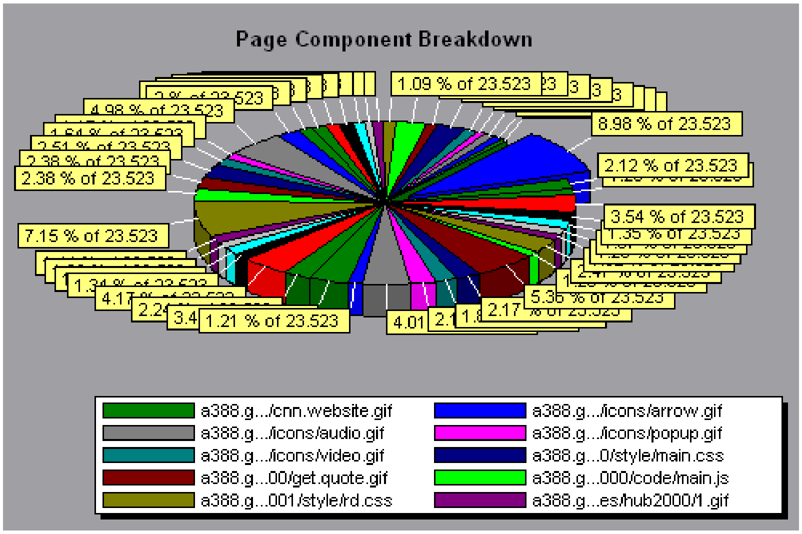
Example

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.



Example

The graph shows that the main cnn.com/WEATHER component took the longest time to download (8.98% of the total download time).



Page Component Breakdown (Over Time) Graph

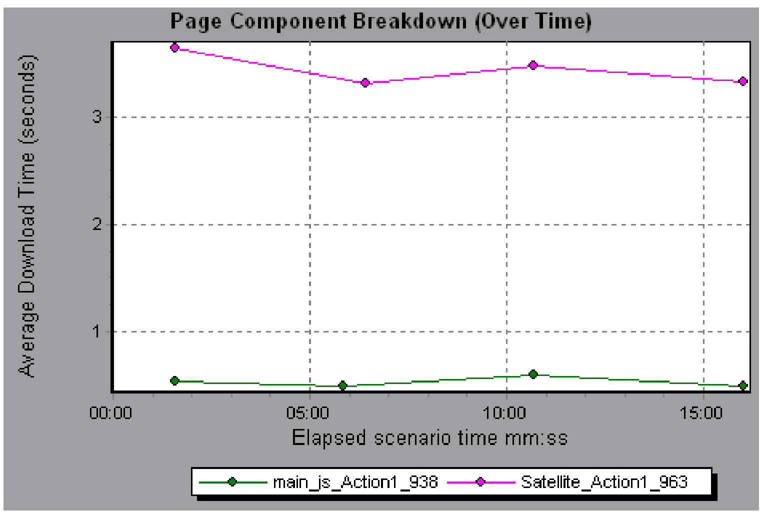
This graph displays the average response time (in seconds) for each Web page and its components during each second of the load test scenario run.

X-axis	The elapsed time from the beginning of the scenario run.
Y-axis	The average response time (in seconds) for each component.

Tips	➤ To isolate the most problematic components, it may be helpful to sort the legend window according to the average number of seconds taken to download a component. To sort the legend by average, double-click the **Average** column heading. ➤ To identify a component in the graph, you can select it. The corresponding line in the legend window is selected.
See also	"Web Page Diagnostics Graphs Overview" on page 280

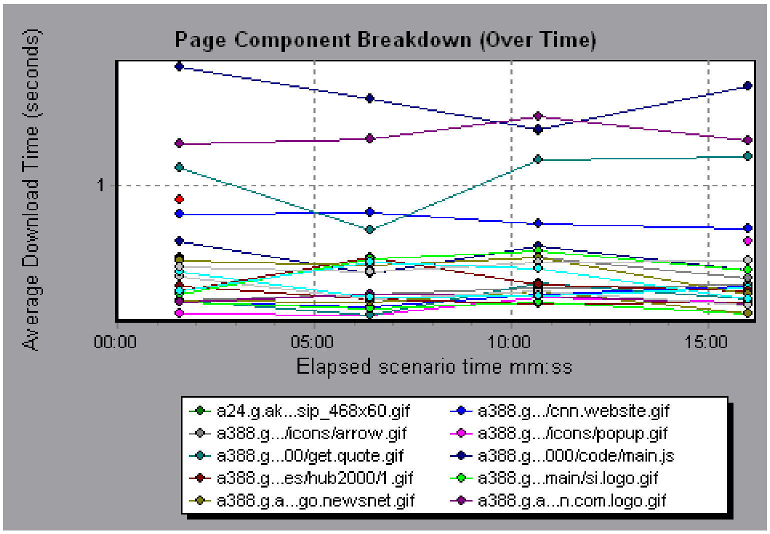
Example

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.



Example

Using the 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.



Page Download Time Breakdown Graph

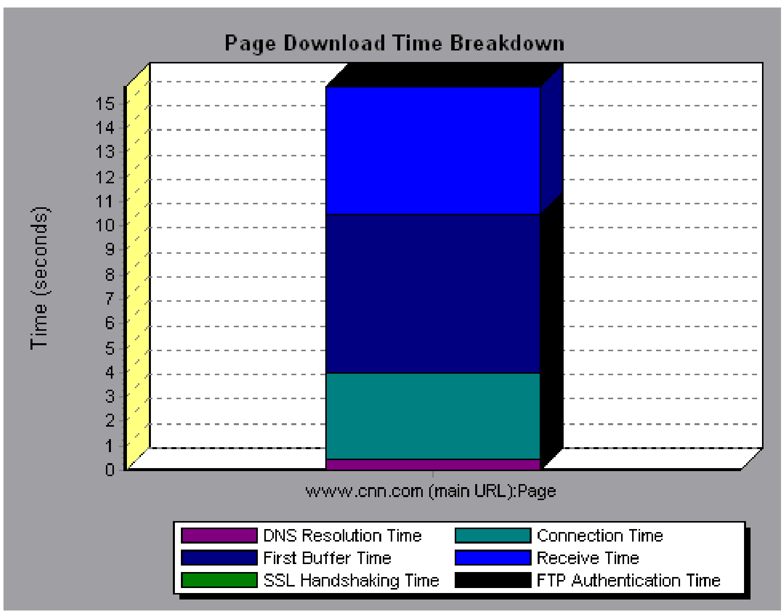
This graph displays a breakdown of each page component's download time.

Purpose	Enables you to determine whether slow response times are being caused by network or server errors during Web page download.

Breakdown options	For breakdown options, see "Page Download Time Breakdown Graph Breakdown Options" on page 294 **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 is the sum of the Connection Time for each of the page's components.
See also	"Web Page Diagnostics Graphs Overview" on page 280

Example

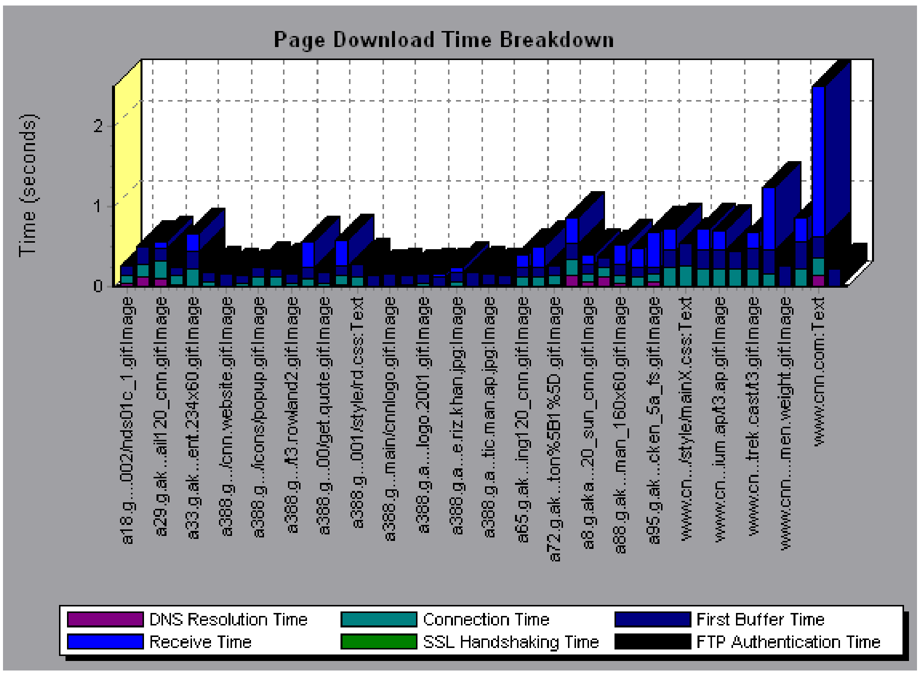
The Page Download Time Breakdown graph demonstrates that receive time, connection time, and first buffer time accounted for a large portion of the time taken to download the main cnn.com URL.



Example

If you break the 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.



Page Download Time Breakdown Graph Breakdown Options

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, FTP authentication time, client time, and error 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. **Note:** 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.
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).

Name	Description
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.
Client Time	Displays the average amount of time that passes while a request is delayed on the client machine due to browser think time or other client-related delays.
Error Time	Displays the average amount of time that passes from the moment an HTTP request is sent until the moment an error message (HTTP errors only) is returned.

Page Download Time Breakdown (Over Time) Graph

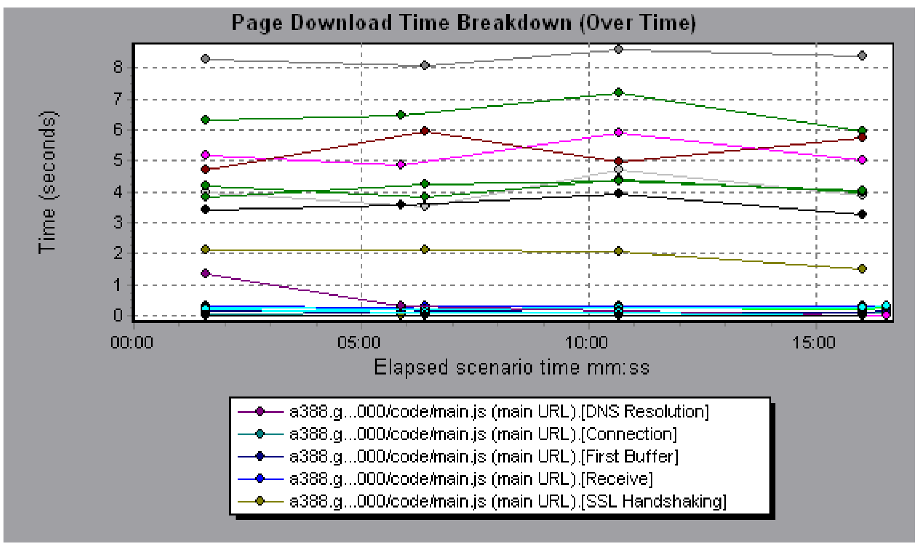
The graph displays a breakdown of each page component's download time during each second of the load test scenario run.

Purpose	This graph enables you to determine at what point during scenario execution network or server problems occurred.
X-axis	Elapsed time from the beginning of the scenario run.
Y-axis	Time (in seconds) taken for each step in the download process.
Tips	To isolate the most problematic components, you can sort the legend window according to the average number of seconds taken to download a component. To sort the legend by average, double-click the **Average** column heading.

Notes	➤ 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 is the sum of the Connection Time for each of the page's components. ➤ When the Page Download Time Breakdown (Over Time) graph is selected from the Web Page Diagnostics graph, it appears as an area graph.
See also	"Web Page Diagnostics Graphs Overview" on page 280

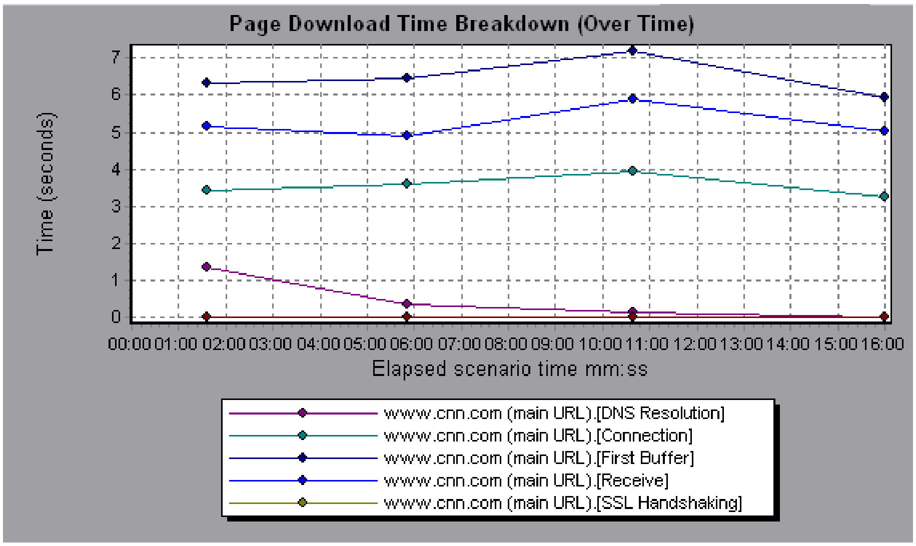
Example

This graph enables you to determine at what point during scenario execution network or server problems occurred.



Example

In the example in the previous section, it is apparent that cnn.com was the most problematic component. If you examine the 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 during the scenario.



Time to First Buffer Breakdown Graph

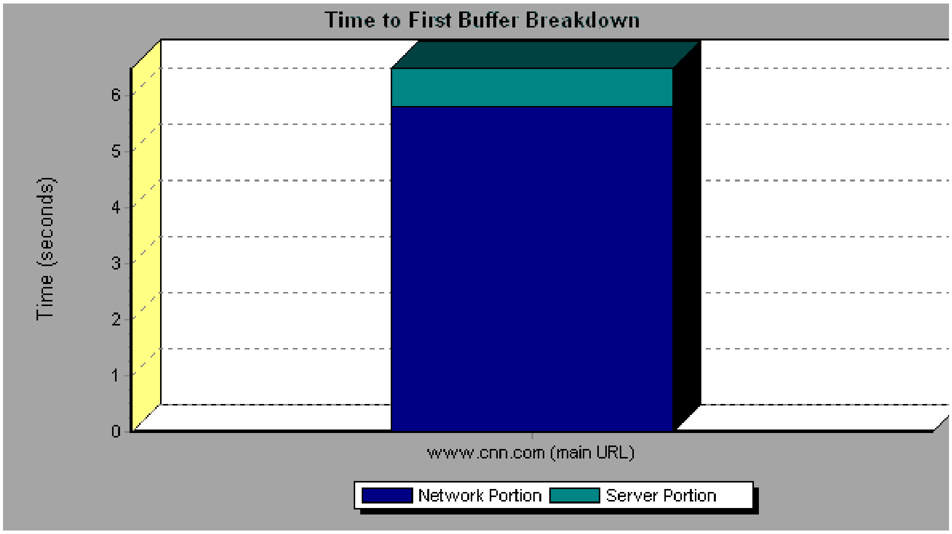
This graph displays each Web page component's relative server/network time (in seconds) for the period of time until the first buffer is successfully received back from the Web server.

Purpose	If the download time for a component is high, you can use this graph to determine whether the problem is server- or network-related.
X-axis	Specifies the name of the component.

Y-axis	Shows the average network/server time (in seconds) for each component.
Measurements	➤ 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 is the sum of the network time for each of the page's components. ➤ 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.
See also	"Web Page Diagnostics Graphs Overview" on page 280

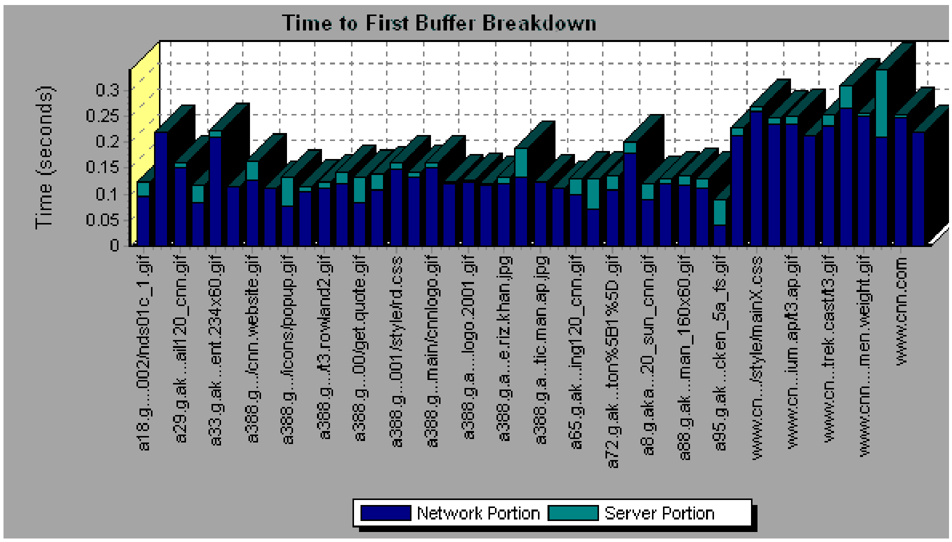
Example

In the following example it is apparent that network time is greater than server time.



Example

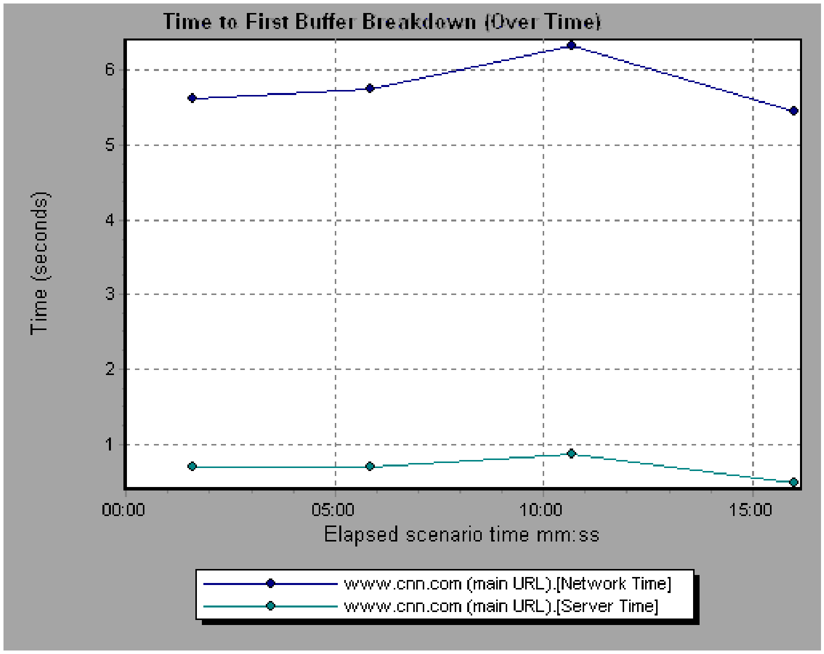
In the following example shows that you can break the main cnn.com URL down further to view the time to first buffer breakdown for each of its components. It is apparent that for the main cnn.com component (the first component on the right), the time to first buffer breakdown is almost all network time.



Time to First Buffer Breakdown (Over Time) Graph

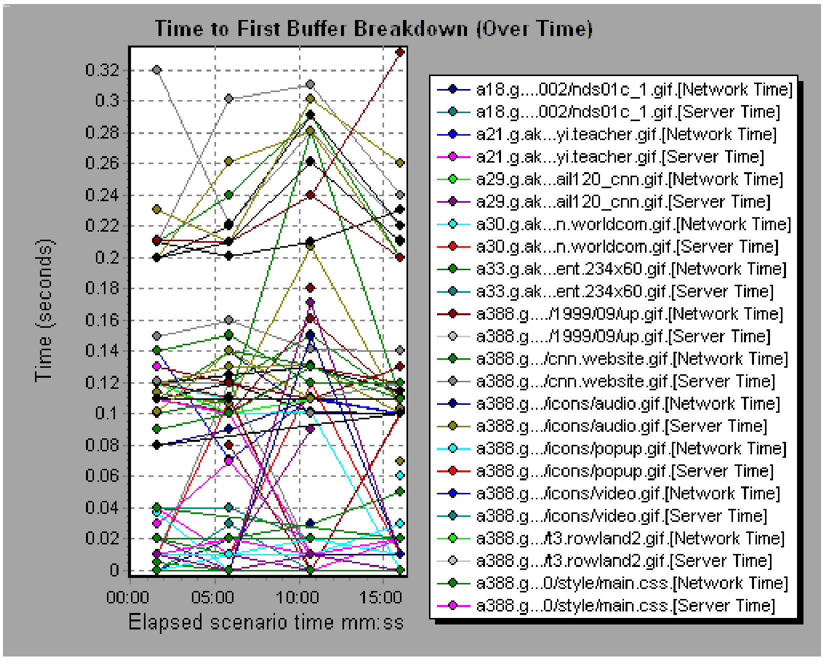
This graph displays each Web page component's server and network time (in seconds) during each second of the load test scenario run, for the period of time until the first buffer is successfully received back from the Web server.

Purpose	You can use this graph to determine when during the scenario run a server- or network-related problem occurred.
X-axis	Elapsed time from the beginning of the scenario run.
Y-axis	Average network or server time (in seconds) for each component.
Measurements	➤ 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:** 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.
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 is the sum of the network time for each of the page's components. ➤ When the Time to First Buffer Breakdown (Over Time) graph is selected from the Web Page Diagnostics graph, it appears as an area graph.
See also	"Web Page Diagnostics Graphs Overview" on page 280


Time to First Buffer Breakdown (Over Time)
Time (seconds)
6
5
4
3
2
1
00:00
05:00
10:00
15:00
Elapsed scenario time mm:ss
www.cnn.com (main URL).[Network Time]
www.cnn.com (main URL).[Server Time]

Example

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



18

System Resource Graphs

This chapter includes:

Concepts

Reference

Concepts

System Resource Graphs Overview

System Resource graphs display the system resource usage measured by the online monitors during the load test scenario run. These graphs require that you specify the resources you want to measure *before* running the scenario. For more information, refer to the section on online monitors in the *HP LoadRunner Controller User Guide*.

Reference

Server Resources Performance Counters

The following table describes the available counters:

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

Unix Resources Default Measurements

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.

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

Windows Resources Default Measurements

The following default measurements are available for Windows Resources:

Object	**Measurement**	**Description**
System	**% Total Processor Time**	The average percentage of time that all the processors on the system are busy executing non-idle threads. On a multi-processor system, if all processors are always busy, this is 100%, if all processors are 50% busy this is 50% and if 1/4 of the processors are 100% busy this is 25%. It can be viewed as the fraction of the time spent doing useful work. Each processor is assigned an Idle thread in the Idle process which consumes those unproductive processor cycles not used by any other threads.
Processor	**% Processor Time**	The percentage of time that the processor is executing a non-idle thread. This counter was designed as a primary indicator of processor activity. It is calculated by measuring the time that the processor spends executing the thread of the idle process in each sample interval, and subtracting that value from 100%. (Each processor has an idle thread which consumes cycles when no other threads are ready to run.) It can be viewed as the percentage of the sample interval spent doing useful work. This counter displays the average percentage of busy time observed during the sample interval. It is calculated by monitoring the time the service was inactive, and then subtracting that value from 100%.

Object	Measurement	Description
System	**File Data Operations/sec**	The rate at which the computer issues read and write operations to file system devices. This does not include File Control Operations.
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.

System Resource Graphs User Interface

This section includes (in alphabetical order):

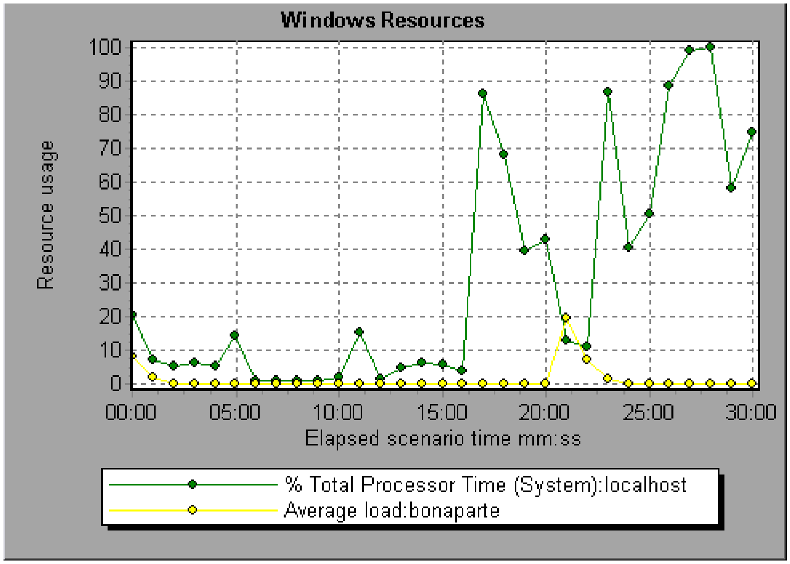
Server Resources Graph

This graph shows the resources (CPU, disk space, memory, or services) used on remote Unix servers measured during the load test scenario.

Purpose	This graph helps you determine the impact of Vuser load on the various system resources.
X-axis	Elapsed time since the start of the run.
Y-axis	The usage of resources on the Unix server.
See also	"System Resource Graphs Overview" on page 306 "Server Resources Performance Counters" on page 307

Example

In the following example, Windows resource utilization is measured during the load test scenario. It is shown as a function of the elapsed scenario time.



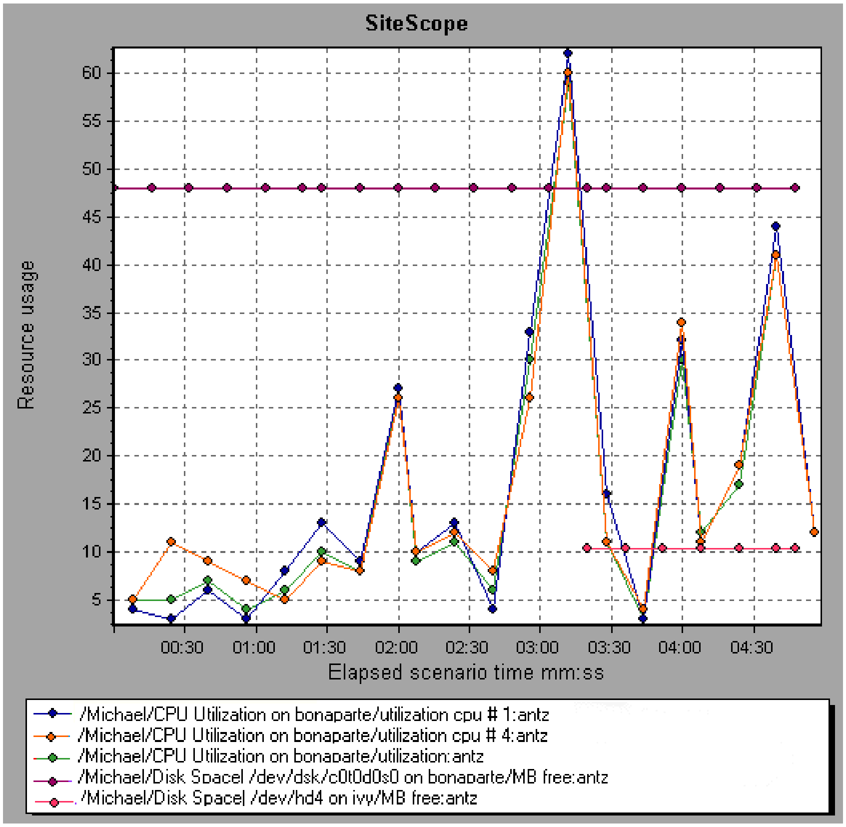
SiteScope Graph

This graph displays statistics about the resource usage on the SiteScope machine during the load test scenario run.

X-axis	Elapsed time since the start of the run.
Y-axis	The resource usage on the Sitescope machine.
See also	"System Resource Graphs Overview" on page 306

Example

In the following example, the percentage of memory resources used, the number of pages read per second, and the CPU utilization.



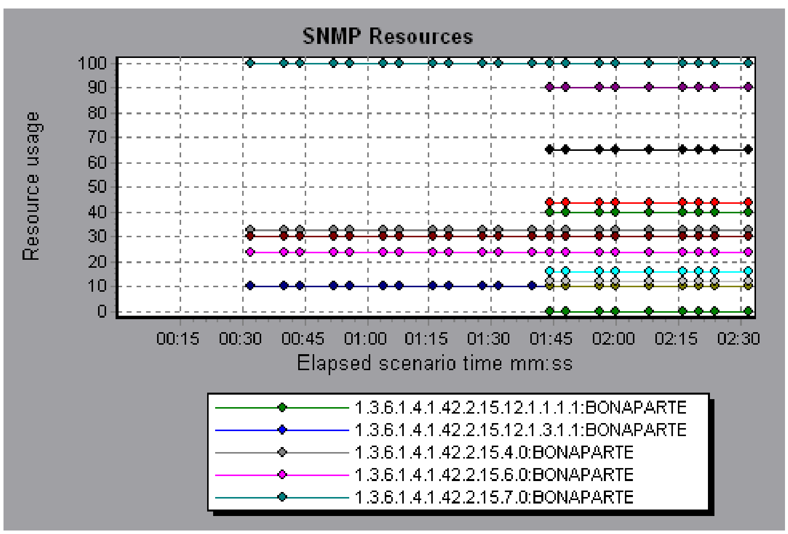
SNMP Resources Graph

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

X-axis	Elapsed time since the start of the run.
Y-axis	The usage of resources on a machine running the SNMP agent.
Note	To obtain data for this graph, you need to enable the SNMP monitor (from the Controller) and select the default measurements you want to display, before running the scenario.
See also	"System Resource Graphs Overview" on page 306

Example

In the following example SNMP measurements are displayed for a machine called bonaporte.



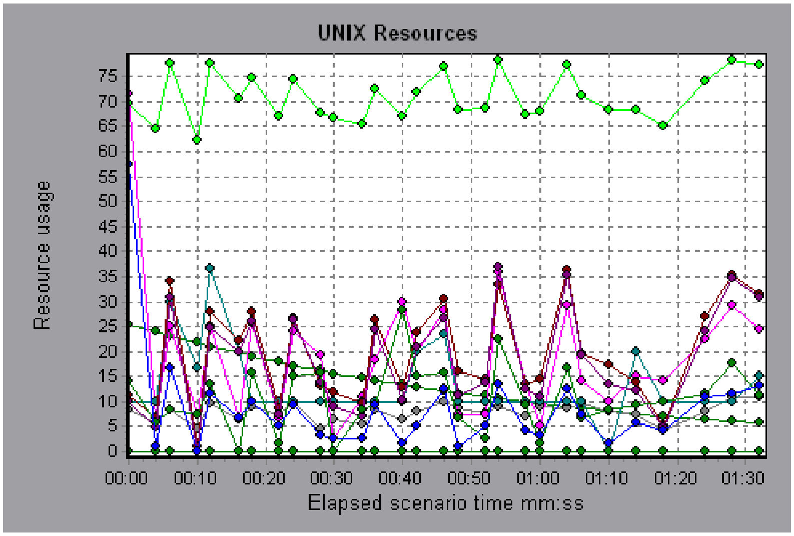
UNIX Resources Graph

This graph shows the UNIX resources measured during the load test 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.

Purpose	This graph helps you determine the impact of Vuser load on the various system resources.
X-axis	Elapsed time since the start of the run.
Y-axis	The usage of resources on the UNIX machine.
Note	To obtain data for this graph, you need to select the desired measurements for the online monitor (from the Controller) before running the scenario.
See also	"Unix Resources Default Measurements" on page 307

Example

In the following example UNIX resources are measured during the load test scenario.



Windows Resources Graph

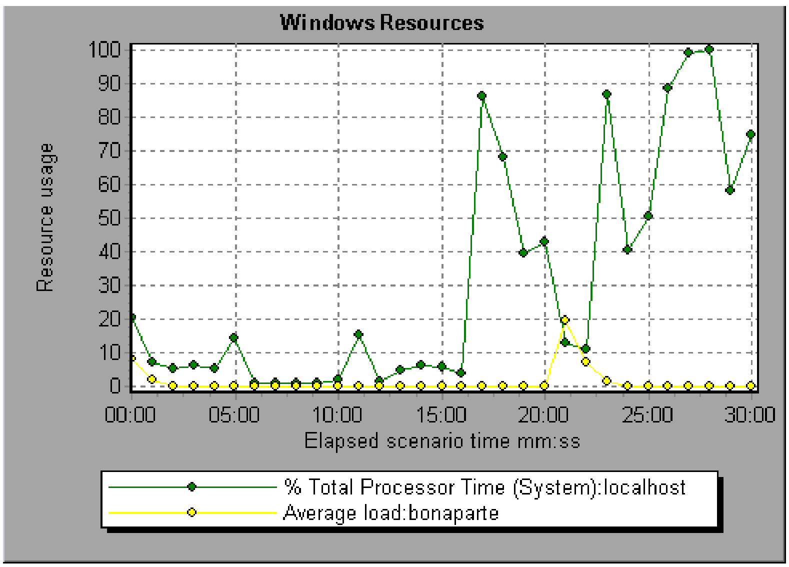
This graph shows the Windows resources measured during the load test scenario. The Windows measurements correspond to the built-in counters available from the Windows Performance Monitor.

Purpose	This graph helps you determine the impact of Vuser load on the various system resources.
X-axis	Elapsed time since the start of the run.
Y-axis	The usage of resources on the Windows machine running the load test scenario.

Note	To obtain data for this graph, you need to select the desired measurements for the online monitor (from the Controller) before running the scenario.
See also	"System Resource Graphs Overview" on page 306 "Windows Resources Default Measurements" on page 309

Example

In the following example Windows resources are measured on the server running the load test scenario.



19

Firewall Server Monitor Graphs

This chapter includes:

Concepts

Reference

Concepts

Firewall Server Monitor Graphs Overview

Firewall server monitor graphs provide you with performance information for firewall servers. In order to obtain data for this graph, you need to activate the Firewall server online monitor before running the load test 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, refer to the *HP LoadRunner Controller User Guide*.

Reference

Check Point FireWall-1 Server Measurements

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

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

Firewall Server Monitor Graphs User Interface

This section includes:

➤ Check Point FireWall-1 Server Graph on page 321

Check Point FireWall-1 Server Graph

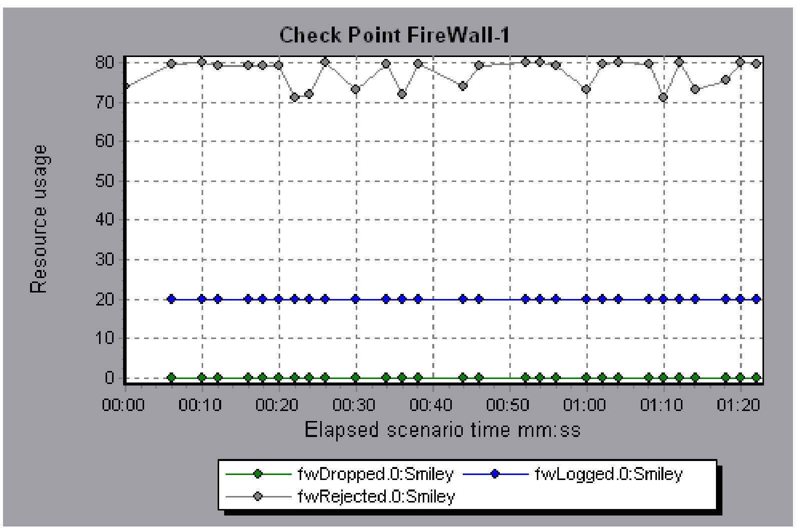
This graph shows statistics on Check Point's Firewall server as a function of the elapsed load test scenario time.

X-axis	Elapsed time since the start of the run.
Y-axis	The resource usage on the Check Point Firewall-1 server.

Note	To obtain data for this graph, you need to enable the Check Point FireWall-1 monitor (from the Controller) and select the default measurements you want to display, before running the scenario.
See also	"Firewall Server Monitor Graphs Overview" on page 320 "Check Point FireWall-1 Server Measurements" on page 321

Example

In the following example the graph displays the **fwDropped**, **fwLogged**, and **fwRejected** measurements during the first minute and twenty seconds of the scenario. 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.



20

Web Server Resource Graphs

This chapter includes:

Concepts

Reference

Concepts

Web Server Resource Graphs Overview

Web Server Resource graphs provide you with information about the resource usage of the Apache, Microsoft IIS, iPlanet/Netscape, and iPlanet (SNMP) 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 load test scenario. For information on activating and configuring the Web Server Resource monitors, refer to the *HP LoadRunner Controller User Guide*.

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

Reference

Apache Server Measurements

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

IIS Server Measurements

The following default measurements are available for the IIS server:

Object	Measurement	Description
Web Service	**Bytes Sent/sec**	The rate at which the data bytes are sent by the Web service.
Web Service	**Bytes Received/sec**	The rate at which the data bytes are received by the Web service.
Web Service	**Get Requests/sec**	The rate at which HTTP requests using the GET method are made. Get requests are generally used for basic file retrievals or image maps, though they can be used with forms.

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

Web Server Resource Graphs User Interface

This section includes (in alphabetical order):

➤ Apache Server Graph on page 327

➤ Microsoft Information Internet Server (IIS) Graph on page 328

Apache Server Graph

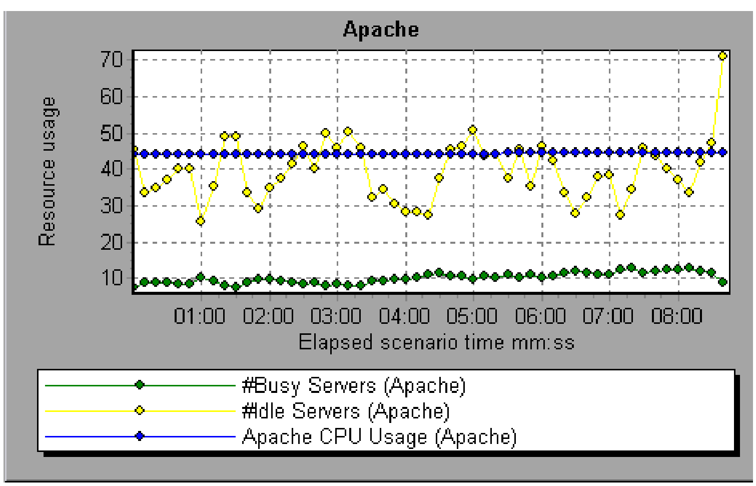
This graph shows server statistics as a function of the elapsed load test scenario time.

X-axis	Elapsed time since the start of the run.
Y-axis	The resource usage on the Apache server during the scenario run.
Note	To obtain data for this graph, you need to enable the Apache online monitor (from the Controller) and select the default measurements you want to display, before running the scenario.
See also	"Web Server Resource Graphs Overview" on page 324 "Apache Server Measurements" on page 325

Example

In the following example, 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.

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



Microsoft Information Internet Server (IIS) Graph

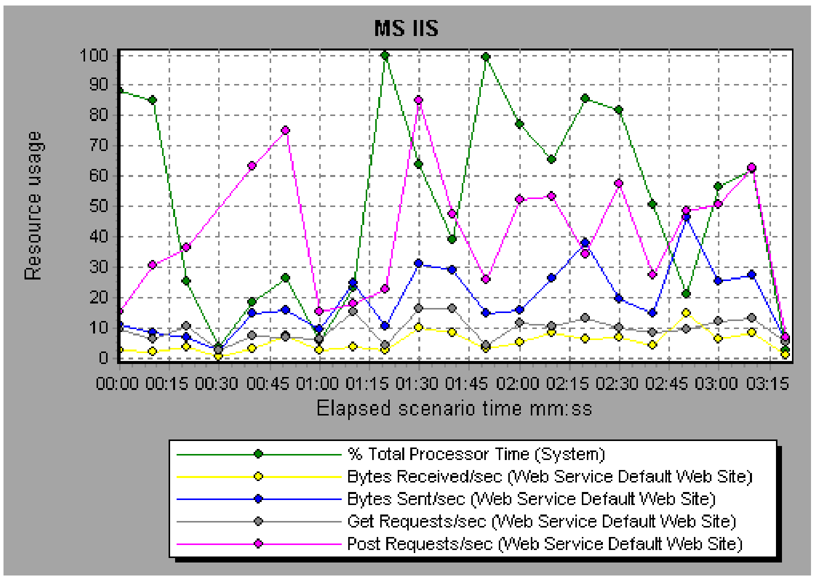
This graph shows server statistics as a function of the elapsed load test scenario time.

X-axis	Elapsed time since the start of the run.
Y-axis	The resource usage on the MS IIS.
Note	To obtain data for this graph, you need to enable the MS IIS online monitor (from the Controller) and select the default measurements you want to display, before running the scenario.
See also	"Web Server Resource Graphs Overview" on page 324 "IIS Server Measurements" on page 325

Example

In the following example 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.

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.



21

Web Application Server Resource Graphs

This chapter includes:

Concepts

Reference

Concepts

Web Application Server Resource Graphs Overview

Web Application Server Resource graphs provide you with resource usage information about the Ariba, ATG Dynamo, BroadVision, ColdFusion, Fujitsu INTERSTAGE, iPlanet (NAS), Microsoft ASP, Oracle9iAS HTTP, SilverStream, WebLogic (SNMP), WebLogic (JMX), and WebSphere application servers.

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 load test scenario. For information on activating and configuring the Web Application Server Resource monitors, refer to the *HP LoadRunner Controller User Guide*.

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, Analysis may scale them. The Legend window indicates the scale factor for each resource. To obtain the true value, multiply the scale factor by the displayed value. For more information on scaled measurements, see the example in "Web Server Resource Graphs Overview" on page 324.

Reference

Web Application Server Resource Graphs Measurements

Microsoft Active Server Pages (ASP) Measurements

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

Measurement	Description
Errors per Second	The number of errors per second.
Requests Wait Time	The number of milliseconds the most recent request was waiting in the queue.
Requests Executing	The number of requests currently executing.
Requests Queued	The number of requests waiting in the queue for service.
Requests Rejected	The total number of requests not executed because there were insufficient resources to process them.
Requests Not Found	The number of requests for files that were not found.
Requests/sec	The number of requests executed per second.
Memory Allocated	The total amount of memory (in bytes) currently allocated by Active Server Pages.
Errors During Script Run-Time	The number of failed requests due to run time errors.
Sessions Current	The current number of sessions being serviced.
Transactions/sec	The number of transactions started per second.

Oracle9iAS HTTP Server Modules

The following table describes some of the modules that are available for the Oracle9iAS HTTP server:

Measurement	Description
mod_mime.c	Determines document types using file extensions.
mod_mime_magic.c	Determines document types using "magic numbers".
mod_auth_anon.c	Provides anonymous user access to authenticated areas.
mod_auth_dbm.c	Provides user authentication using DBM files.
mod_auth_digest.c	Provides MD5 authentication.
mod_cern_meta.c	Supports HTTP header metafiles.
mod_digest.c	Provides MD5 authentication (deprecated by mod_auth_digest).
mod_expires.c	Applies Expires: headers to resources.
mod_headers.c	Adds arbitrary HTTP headers to resources.
mod_proxy.c	Provides caching proxy abilities.
mod_rewrite.c	Provides powerful URI-to-filename mapping using regular expressions.
mod_speling.c	Automatically corrects minor typos in URLs.
mod_info.c	Provides server configuration information.
mod_status.c	Displays server status.
mod_usertrack.c	Provides user tracking using cookies.
mod_dms.c	Provides access to DMS Apache statistics.
mod_perl.c	Allows execution of Perl scripts.
mod_fastcgi.c	Supports CGI access to long-lived programs.
mod_ssl.c	Provides SSL support.
mod_plsql.c	Handles requests for Oracle stored procedures.
mod_isapi.c	Provides Windows ISAPI extension support.

Measurement	Description
mod_setenvif.c	Sets environment variables based on client information.
mod_actions.c	Executes CGI scripts based on media type or request method.
mod_imap.c	Handles imagemap files.
mod_asis.c	Sends files that contain their own HTTP headers.
mod_log_config.c	Provides user-configurable logging replacement for mod_log_common.
mod_env.c	Passes environments to CGI scripts.
mod_alias.c	Maps different parts of the host file system in the document tree, and redirects URLs.
mod_userdir.c	Handles user home directories.
mod_cgi.c	Invokes CGI scripts.
mod_dir.c	Handles the basic directory.
mod_autoindex.c	Provides automatic directory listings.
mod_include.c	Provides server-parsed documents.
mod_negotiation.c	Handles content negotiation.
mod_auth.c	Provides user authentication using text files.
mod_access.c	Provides access control based on the client host name or IP address.
mod_so.c	Supports loading modules (.so on UNIX, .dll on Win32) at run time.
mod_oprocmgr.c	Monitors JServ processes and restarts them if they fail.
mod_jserv.c	Routes HTTP requests to JServ server processes. Balances load across multiple JServs by distributing new requests in round-robin order.

Measurement	Description
mod_ose.c	Routes requests to the JVM embedded in Oracle's database server.
http_core.c	Handles requests for static Web pages.

Oracle9iAS HTTP Server Counters

The following table describes the counters that are available for the Oracle9iAS HTTP server:

Measurement	Description
handle.minTime	The minimum time spent in the module handler.
handle.avg	The average time spent in the module handler.
handle.active	The number of threads currently in the handle processing phase.
handle.time	The total amount of time spent in the module handler.
handle.completed	The number of times the handle processing phase was completed.
request.maxTime	The maximum amount of time required to service an HTTP request.
request.minTime	The minimum amount of time required to service an HTTP request.
request.avg	The average amount of time required to service an HTTP request.
request.active	The number of threads currently in the request processing phase.
request.time	The total amount of time required to service an HTTP request.
request.completed	The number of times the request processing phase was completed.

Measurement	Description
connection.maxTime	The maximum amount of time spent servicing any HTTP connection.
connection.minTime	The minimum amount of time spent servicing any HTTP connection.
connection.avg	The average amount of time spent servicing HTTP connections.
connection.active	The number of connections with currently open threads.
connection.time	The total amount of time spent servicing HTTP connections.
connection.completed	The number of times the connection processing phase was completed.
numMods.value	The number of loaded modules.
childFinish.count	The number of times the Apache parent server started a child server, for any reason.
childStart.count	The number of times "children"finished "gracefully."There are some ungraceful error/crash cases that are not counted in childFinish.count.
Decline.count	The number of times each module declined HTTP requests.
internalRedirect.count	The number of times that any module passed control to another module using an "internal redirect”.
cpuTime.value	The total CPU time utilized by all processes on the Apache server (measured in CPU milliseconds).
heapSize.value	The total heap memory utilized by all processes on the Apache server (measured in kilobytes).
pid.value	The process identifier of the parent Apache process.
upTime.value	The amount of time the server has been running (measured in milliseconds).

WebLogic (SNMP) Server Table Measurements

The Server Table lists all WebLogic (SNMP) servers that are being monitored by the agent. A server must be contacted or be reported as a member of a cluster at least once before it will appear in this table. Servers are only reported as a member of a cluster when they are actively participating in the cluster, or shortly thereafter.

Measurement	Description
ServerState	The state of the WebLogic server, as inferred by the SNMP agent. **Up** implies that the agent can contact the server. **Down** implies that the agent cannot contact the server.
ServerLoginEnable	**True** if client logins are enabled on the server.
ServerMaxHeapSpace	The maximum heap size for this server (in KB).
ServerHeapUsedPct	The percentage of heap space currently in use on the server.
ServerQueueLength	The current length of the server execute queue.
ServerQueueThroughput	The current throughput of execute queue, expressed as the number of requests processed per second.
ServerNumEJBDeployment	The total number of EJB deployment units known to the server.
ServerNumEJBBeansDeployed	The total number of EJB beans actively deployed on the server.

WebLogic (SNMP) Listen Table Measurements

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

Measurement	Description
ListenPort	Port number.
ListenAdminOK	**True** if admin requests are allowed on this (protocol, ipAddr, port) combination; otherwise false.
ListenState	**Listening** if the (protocol, ipAddr, port) combination 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.

WebLogic (SNMP) ClassPath Table Measurements

The ClassPath Table is the table of classpath elements for Java, WebLogic (SNMP) server, and servlets. There are multiple entries in this table for each server. There may also be multiple entries for each path on a server. If clustering is used, the clustering-related MIB objects will assume a higher priority.

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

Websphere Application Server Monitor Run-Time Resource Measurements

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 on the Java Virtual Machine.

Websphere Application Server Monitor BeanData Measurements

Every home on the server provides performance data, depending on the type of bean deployed in the home. The top level bean data holds an aggregate of all the containers.

Measurement	Description
BeanDestroys	The number of times an individual bean object was destroyed. This applies to any bean, regardless of its type.
StatelessBeanDestroys	The number of times a stateless session bean object was destroyed.
StatefulBeanDestroys	The number of times a stateful session bean object was destroyed.

Websphere Application Server Monitor BeanObjectPool Measurements

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
NumGetFound	The number of calls to the pool that resulted in finding an available bean.
NumPutsDiscarded	The number of times releasing a bean to the pool resulted in the bean being discarded because the pool was full.

Websphere Application Server Monitor OrbThreadPool Measurements

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.

Websphere Application Server Monitor DBConnectionMgr Measurements

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
ConnectionWaitTime	The average time (in seconds) of a connection grant.
ConnectionTime	The average time (in seconds) that a connection is in use.
ConnectionPercentUsed	The average percentage of the pool that is in use.

Websphere Application Server Monitor TransactionData Measurements

These are resources that pertain to transactions.

Measurement	Description
NumTransactions	The number of transactions processed.
ActiveTransactions	The average number of active transactions.

Measurement	Description
TransactionRT	The average duration of each transaction.
RolledBack	The number of transactions rolled back.
Timeouts	The number of transactions that timed out due to inactivity timeouts.
TransactionSuspended	The average number of times that a transaction was suspended.

Websphere Application Server Monitor ServletEngine Measurements

These are resources that are related to servlets and JSPs.

Measurement	Description
ServletErrors	The number of requests that resulted in an error or an exception.

Websphere Application Server Monitor Session Measurements

These are general metrics regarding the HTTP session pool.

Measurement	Description
SessionsInvalidated	The number of invalidated sessions. May not be valid when using sessions in the database mode.

Web Application Server Resource Graphs User Interface

This section includes (in alphabetical order):

- Microsoft Active Server Pages (ASP) Graph on page 344
- Oracle9iAS HTTP Server Graph on page 345

Microsoft Active Server Pages (ASP) Graph

This graph displays statistics about the resource usage on the ASP server during the load test scenario run.

X-axis	Elapsed time since the start of the run.
Y-axis	The resource usage on the ASP server.
Note	To obtain data for this graph, you need to enable the Microsoft ASP online monitor (from the Controller) and select the default measurements you want to display, before running the scenario.
See also	"Web Application Server Resource Graphs Overview" on page 332 "Microsoft Active Server Pages (ASP) Measurements" on page 333

Oracle9iAS HTTP Server Graph

This graph displays statistics about the resource usage on the Oracle9iAS HTTP server during the load test scenario run.

X-axis	Elapsed time since the start of the run.
Y-axis	The resource usage on the Oracle9iAS HTTP server.
Note	To obtain data for this graph, you need to enable the Oracle9iAS HTTP online monitor (from the Controller), and select the default measurements you want to display, before running the scenario.
See also	"Web Application Server Resource Graphs Overview" on page 332 "Oracle9iAS HTTP Server Modules" on page 334 "Oracle9iAS HTTP Server Counters" on page 336

WebLogic (SNMP) Graph

This graph displays statistics about the resource usage on the WebLogic (SNMP) server (version 6.0 and earlier) during the load test scenario run.

X-axis	The elapsed time since the start of the run.
Y-axis	The resource usage on the WebLogic (SNMP) server.
Note	To obtain data for this graph, you need to enable the WebLogic (SNMP) online monitor (from the Controller) and select the default measurements you want to display, before running the scenario.
See also	"Web Application Server Resource Graphs Overview" on page 332 "WebLogic (SNMP) Server Table Measurements" on page 338 "WebLogic (SNMP) Listen Table Measurements" on page 339 "WebLogic (SNMP) ClassPath Table Measurements" on page 340

WebSphere Application Server Graph

This graph displays statistics about the resource usage on the WebSphere application server during the load test scenario run.

X-axis	Elapsed time since the start of the run.
Y-axis	The resource usage on the WebSphere Application server.
Note	To obtain data for this graph, you need to configure the WebSphere Application Server online monitor (from the Controller) and select the default measurements you want to display, before running the scenario.
See also	"Web Application Server Resource Graphs Overview" on page 332 "Websphere Application Server Monitor Run-Time Resource Measurements" on page 340 "Websphere Application Server Monitor BeanData Measurements" on page 341 "Websphere Application Server Monitor BeanObjectPool Measurements" on page 341 "Websphere Application Server Monitor OrbThreadPool Measurements" on page 342 "Websphere Application Server Monitor DBConnectionMgr Measurements" on page 342 "Websphere Application Server Monitor TransactionData Measurements" on page 342 "Websphere Application Server Monitor ServletEngine Measurements" on page 343 "Websphere Application Server Monitor Session Measurements" on page 343

22

Database Server Resource Graphs

This chapter includes:

Concepts

Reference

Concepts

Database Server Resource Graphs Overview

The Database Server Resource graphs show statistics for several database servers. Currently DB2, Oracle, SQL server, and Sybase databases are supported. These graphs require that you specify the resources you want to measure *before* running the load test scenario. For more information, refer to the section on online monitors in the *HP LoadRunner Controller User Guide*.

Reference

DB2 Database Manager Counters

Measurement	Description
rem_cons_in	The current number of connections initiated from remote clients to the instance of the database manager that is being monitored.
rem_cons_in_exec	The number of remote applications that are currently connected to a database and are currently processing a unit of work within the database manager instance being monitored.
local_cons	The number of local applications that are currently connected to a database within the database manager instance being monitored.
local_cons_in_exec	The number of local applications that are currently connected to a database within the database manager instance being monitored and are currently processing a unit of work.
con_local_dbases	The number of local databases that have applications connected.
agents_registered	The number of agents registered in the database manager instance that is being monitored (coordinator agents and subagents).
agents_waiting_on_token	The number of agents waiting for a token so they can execute a transaction in the database manager.
idle_agents	The number of agents in the agent pool that are currently unassigned to an application and are therefore "idle".
agents_from_pool	The number of agents assigned from the agent pool.

Measurement	Description
agents_created_empty_pool	The number of agents created because the agent pool was empty.
agents_stolen	The number of times that agents are stolen from an application. Agents are stolen when an idle agent associated with an application is reassigned to work on a different application.
comm_private_mem	The amount of private memory that the instance of the database manager has currently committed at the time of the snapshot.
inactive_gw_agents	The number of DRDA agents in the DRDA connections pool that are primed with a connection to a DRDA database, but are inactive.
num_gw_conn_switches	The number of times that an agent from the agents pool was primed with a connection and was stolen for use with a different DRDA database.
sort_heap_allocated	The total number of allocated pages of sort heap space for all sorts at the level chosen and at the time the snapshot was taken.
post_threshold_sorts	The number of sorts that have requested heaps after the sort heap threshold has been reached.
piped_sorts_requested	The number of piped sorts that have been requested.
piped_sorts_accepted	The number of piped sorts that have been accepted.

DB2 Database Counters

Measurement	Description
appls_cur_cons	Indicates the number of applications that are currently connected to the database.
appls_in_db2	Indicates the number of applications that are currently connected to the database, and for which the database manager is currently processing a request.
total_sec_cons	The number of connections made by a sub-agent to the database at the node.
num_assoc_agents	At the application level, this is the number of sub-agents associated with an application. At the database level, it is the number of sub-agents for all applications.
sort_heap_allocated	The total number of allocated pages of sort heap space for all sorts at the level chosen and at the time the snapshot was taken.
total_sorts	The total number of sorts that have been executed.
total_sort_time	The total elapsed time (in milliseconds) for all sorts that have been executed.
sort_overflows	The total number of sorts that ran out of sort heap and may have required disk space for temporary storage.
active_sorts	The number of sorts in the database that currently have a sort heap allocated.
total_hash_joins	The total number of hash joins executed.
total_hash_loops	The total number of times that a single partition of a hash join was larger than the available sort heap space.
hash_join_overflows	The number of times that hash join data exceeded the available sort heap space.

Measurement	Description
hash_join_small_overflows	The number of times that hash join data exceeded the available sort heap space by less than 10%.
pool_data_l_reads	The number of logical read requests for data pages that have gone through the buffer pool.
pool_data_p_reads	The number of read requests that required I/O to get data pages into the buffer pool.
pool_data_writes	Indicates the number of times a buffer pool data page was physically written to disk.
pool_index_l_reads	The number of logical read requests for index pages that have gone through the buffer pool.
pool_index_p_reads	The number of physical read requests to get index pages into the buffer pool.
pool_index_writes	The number of times a buffer pool index page was physically written to disk.
pool_read_time	The total amount of elapsed time spent processing read requests that caused data or index pages to be physically read from disk to buffer pool.
pool_write_time	The total amount of time spent physically writing data or index pages from the buffer pool to disk.
files_closed	The total number of database files closed.
pool_async_data_reads	The number of pages read asynchronously into the buffer pool.
pool_async_data_writes	The number of times a buffer pool data page was physically written to disk by either an asynchronous page cleaner, or a pre-fetcher. A pre-fetcher may have written dirty pages to disk to make space for the pages being pre-fetched.
pool_async_index_writes	The number of times a buffer pool index page was physically written to disk by either an asynchronous page cleaner, or a pre-fetcher. A pre-fetcher may have written dirty pages to disk to make space for the pages being pre-fetched.

Measurement	Description
pool_async_index_reads	The number of index pages read asynchronously into the buffer pool by a pre-fetcher.
pool_async_read_time	The total elapsed time spent reading by database manager pre-fetchers.
pool_async_write_time	The total elapsed time spent writing data or index pages from the buffer pool to disk by database manager page cleaners.
pool_async_data_read_reqs	The number of asynchronous read requests.
pool_lsn_gap_clns	The number of times a page cleaner was invoked because the logging space used had reached a pre-defined criterion for the database.
pool_drty_pg_steal_clns	The number of times a page cleaner was invoked because a synchronous write was needed during the victim buffer replacement for the database.
pool_drty_pg_thrsh_clns	The number of times a page cleaner was invoked because a buffer pool had reached the dirty page threshold criterion for the database.
prefetch_wait_time	The time an application spent waiting for an I/O server (pre-fetcher) to finish loading pages into the buffer pool.
pool_data_to_estore	The number of buffer pool data pages copied to extended storage.
pool_index_to_estore	The number of buffer pool index pages copied to extended storage.
pool_data_from_estore	The number of buffer pool data pages copied from extended storage.
pool_index_from_estore	The number of buffer pool index pages copied from extended storage.
direct_reads	The number of read operations that do not use the buffer pool.
direct_writes	The number of write operations that do not use the buffer pool.

Measurement	Description
direct_read_reqs	The number of requests to perform a direct read of one or more sectors of data.
direct_write_reqs	The number of requests to perform a direct write of one or more sectors of data.
direct_read_time	The elapsed time (in milliseconds) required to perform the direct reads.
direct_write_time	The elapsed time (in milliseconds) required to perform the direct writes.
cat_cache_lookups	The number of times that the catalog cache was referenced to obtain table descriptor information.
cat_cache_inserts	The number of times that the system tried to insert table descriptor information into the catalog cache.
cat_cache_overflows	The number of times that an insert into the catalog cache failed due the catalog cache being full.
cat_cache_heap_full	The number of times that an insert into the catalog cache failed due to a heap-full condition in the database heap.
pkg_cache_lookups	The number of times that an application looked for a section or package in the package cache. At a database level, it indicates the overall number of references since the database was started, or monitor data was reset.
pkg_cache_inserts	The total number of times that a requested section was not available for use and had to be loaded into the package cache. This count includes any implicit prepares performed by the system.
pkg_cache_num_overflows	The number of times that the package cache overflowed the bounds of its allocated memory.
appl_section_lookups	Lookups of SQL sections by an application from its SQL work area.

Measurement	Description
appl_section_inserts	Inserts of SQL sections by an application from its SQL work area.
sec_logs_allocated	The total number of secondary log files that are currently being used for the database.
log_reads	The number of log pages read from disk by the logger.
log_writes	The number of log pages written to disk by the logger.
total_log_used	The total amount of active log space currently used (in bytes) in the database.
locks_held	The number of locks currently held.
lock_list_in_use	The total amount of lock list memory (in bytes) that is in use.
deadlocks	The total number of deadlocks that have occurred.
lock_escals	The number of times that locks have been escalated from several row locks to a table lock.
x_lock_escals	The number of times that locks have been escalated from several row locks to one exclusive table lock, or the number of times an exclusive lock on a row caused the table lock to become an exclusive lock.
lock_timeouts	The number of times that a request to lock an object timed-out instead of being granted.
lock_waits	The total number of times that applications or connections waited for locks.
lock_wait_time	The total elapsed time waited for a lock.
locks_waiting	The number of agents waiting on a lock.
rows_deleted	The number of row deletions attempted.
rows_inserted	The number of row insertions attempted.

Measurement	Description
rows_updated	The number of row updates attempted.
rows_selected	The number of rows that have been selected and returned to the application.
int_rows_deleted	The number of rows deleted from the database as a result of internal activity.
int_rows_updated	The number of rows updated from the database as a result of internal activity.
int_rows_inserted	The number of rows inserted into the database as a result of internal activity caused by triggers.
static_sql_stmts	The number of static SQL statements that were attempted.
dynamic_sql_stmts	The number of dynamic SQL statements that were attempted.
failed_sql_stmts	The number of SQL statements that were attempted, but failed.
commit_sql_stmts	The total number of SQL COMMIT statements that have been attempted.
rollback_sql_stmts	The total number of SQL ROLLBACK statements that have been attempted.
select_sql_stmts	The number of SQL SELECT statements that were executed.
uid_sql_stmts	The number of SQL UPDATE, INSERT, and DELETE statements that were executed.
ddl_sql_stmts	The number of SQL Data Definition Language (DDL) statements that were executed.
int_auto_rebinds	The number of automatic rebinds (or recompiles) that have been attempted.
int_commits	The total number of commits initiated internally by the database manager.
int_rollbacks	The total number of rollbacks initiated internally by the database manager.

Measurement	Description
int_deadlock_rollbacks	The total number of forced rollbacks initiated by the database manager due to a deadlock. A rollback is performed on the current unit of work in an application selected by the database manager to resolve the deadlock.
binds_precompiles	The number of binds and pre-compiles attempted.

DB2 Application Counters

Measurement	Description
agents_stolen	The number of times that agents are stolen from an application. Agents are stolen when an idle agent associated with an application is reassigned to work on a different application.
num_assoc_agents	At the application level, this is the number of sub-agents associated with an application. At the database level, it is the number of sub-agents for all applications.
total_sorts	The total number of sorts that have been executed.
total_sort_time	The total elapsed time (in milliseconds) for all sorts that have been executed.
sort_overflows	The total number of sorts that ran out of sort heap and may have required disk space for temporary storage.
total_hash_joins	The total number of hash joins executed.
total_hash_loops	The total number of times that a single partition of a hash join was larger than the available sort heap space.
hash_join_overflows	The number of times that hash join data exceeded the available sort heap space

Measurement	Description
hash_join_small_overflows	The number of times that hash join data exceeded the available sort heap space by less than 10%.
pool_data_l_reads	The number of logical read requests for data pages that have gone through the buffer pool.
pool_data_p_reads	The number of read requests that required I/O to get data pages into the buffer pool.
pool_data_writes	The number of times a buffer pool data page was physically written to disk.
pool_index_l_reads	The number of logical read requests for index pages that have gone through the buffer pool.
pool_index_p_reads	The number of physical read requests to get index pages into the buffer pool.
pool_index_writes	The number of times a buffer pool index page was physically written to disk.
pool_read_time	The total amount of elapsed time spent processing read requests that caused data or index pages to be physically read from disk to buffer pool.
prefetch_wait_time	The time an application spent waiting for an I/O server (pre-fetcher) to finish loading pages into the buffer pool.
pool_data_to_estore	The number of buffer pool data pages copied to extended storage.
pool_index_to_estore	The number of buffer pool index pages copied to extended storage.
pool_data_from_estore	The number of buffer pool data pages copied from extended storage.
pool_index_from_estore	The number of buffer pool index pages copied from extended storage.
direct_reads	The number of read operations that do not use the buffer pool.

Measurement	Description
direct_writes	The number of write operations that do not use the buffer pool.
direct_read_reqs	The number of requests to perform a direct read of one or more sectors of data.
direct_write_reqs	The number of requests to perform a direct write of one or more sectors of data.
direct_read_time	The elapsed time (in milliseconds) required to perform the direct reads.
direct_write_time	The elapsed time (in milliseconds) required to perform the direct writes.
cat_cache_lookups	The number of times that the catalog cache was referenced to obtain table descriptor information.
cat_cache_inserts	The number of times that the system tried to insert table descriptor information into the catalog cache.
cat_cache_overflows	The number of times that an insert into the catalog cache failed due to the catalog cache being full.
cat_cache_heap_full	The number of times that an insert into the catalog cache failed due to a heap-full condition in the database heap.
pkg_cache_lookups	The number of times that an application looked for a section or package in the package cache. At a database level, it indicates the overall number of references since the database was started, or monitor data was reset.
pkg_cache_inserts	The total number of times that a requested section was not available for use and had to be loaded into the package cache. This count includes any implicit prepares performed by the system.
appl_section_lookups	Lookups of SQL sections by an application from its SQL work area.

Measurement	Description
appl_section_inserts	Inserts of SQL sections by an application from its SQL work area.
uow_log_space_used	The amount of log space (in bytes) used in the current unit of work of the monitored application.
locks_held	The number of locks currently held.
deadlocks	The total number of deadlocks that have occurred.
lock_escals	The number of times that locks have been escalated from several row locks to a table lock.
x_lock_escals	The number of times that locks have been escalated from several row locks to one exclusive table lock, or the number of times an exclusive lock on a row caused the table lock to become an exclusive lock.
lock_timeouts	The number of times that a request to lock an object timed-out instead of being granted.
lock_waits	The total number of times that applications or connections waited for locks.
lock_wait_time	The total elapsed time waited for a lock.
locks_waiting	The number of agents waiting on a lock.
uow_lock_wait_time	The total amount of elapsed time this unit of work has spent waiting for locks.
rows_deleted	The number of row deletions attempted.
rows_inserted	The number of row insertions attempted.
rows_updated	The number of row updates attempted.
rows_selected	The number of rows that have been selected and returned to the application.
rows_written	The number of rows changed (inserted, deleted or updated) in the table.

Measurement	Description
rows_read	The number of rows read from the table.
int_rows_deleted	The number of rows deleted from the database as a result of internal activity.
int_rows_updated	The number of rows updated from the database as a result of internal activity.
int_rows_inserted	The number of rows inserted into the database as a result of internal activity caused by triggers.
open_rem_curs	The number of remote cursors currently open for this application, including those cursors counted by 'open_rem_curs_blk'.
open_rem_curs_blk	The number of remote blocking cursors currently open for this application.
rej_curs_blk	The number of times that a request for an I/O block at server was rejected and the request was converted to non-blocked I/O.
acc_curs_blk	The number of times that a request for an I/O block was accepted.
open_loc_curs	The number of local cursors currently open for this application, including those cursors counted by 'open_loc_curs_blk'.
open_loc_curs_blk	The number of local blocking cursors currently open for this application.
static_sql_stmts	The number of static SQL statements that were attempted.
dynamic_sql_stmts	The number of dynamic SQL statements that were attempted.
failed_sql_stmts	The number of SQL statements that were attempted, but failed.
commit_sql_stmts	The total number of SQL COMMIT statements that have been attempted.

Measurement	Description
rollback_sql_stmts	The total number of SQL ROLLBACK statements that have been attempted.
select_sql_stmts	The number of SQL SELECT statements that were executed.
uid_sql_stmts	The number of SQL UPDATE, INSERT, and DELETE statements that were executed.
ddl_sql_stmts	This element indicates the number of SQL Data Definition Language (DDL) statements that were executed.
int_auto_rebinds	The number of automatic rebinds (or recompiles) that have been attempted.
int_commits	The total number of commits initiated internally by the database manager.
int_rollbacks	The total number of rollbacks initiated internally by the database manager.
int_deadlock_rollbacks	The total number of forced rollbacks initiated by the database manager due to a deadlock. A rollback is performed on the current unit of work in an application selected by the database manager to resolve the deadlock.
binds_precompiles	The number of binds and pre-compiles attempted.

Oracle Server Monitoring Measurements

The following measurements are most commonly used when monitoring the Oracle server (from the V$SYSSTAT table):

Measurement	Description
CPU used by this session	The amount of CPU time (in tens of milliseconds) used by a session between the time a user call started and ended. Some user calls can be completed within 10 milliseconds and, as a result, the start- and end-user call time can be the same. In this case, 0 milliseconds are added to the statistic. A similar problem can exist in the operating system reporting, especially on systems that suffer from many context switches.
Bytes received via SQL*Net from client	The total number of bytes received from the client over Net8.
Logons current	The total number of current logons.
Opens of replaced files	The total number of files that needed to be reopened because they were no longer in the process file cache.
User calls	Oracle allocates resources (Call State Objects) to keep track of relevant user call data structures every time you log in, parse, or execute. When determining activity, the ratio of user calls to RPI calls gives you an indication of how much internal work is generated as a result of the type of requests the user is sending to Oracle.
SQL*Net roundtrips to/ from client	The total number of Net8 messages sent to, and received from, the client.
Bytes sent via SQL*Net to client	The total number of bytes sent to the client from the foreground process(es).
Opened cursors current	The total number of current open cursors.

Measurement	Description
DB block changes	Closely related to consistent changes, this statistic counts the total number of changes that were made to all blocks in the SGA that were part of an update or delete operation. These are changes that generate redo log entries and hence cause permanent changes to the database if the transaction is committed. This statistic is a rough indication of total database work and indicates (possibly on a per-transaction level) the rate at which buffers are being dirtied.
Total file opens	The total number of file opens being performed by the instance. Each process needs a number of files (control file, log file, database file) in order to work against the database.

SQL Server Default Counters

Measurement	Description
% Total Processor Time	The average percentage of time that all the processors on the system are busy executing non-idle threads. On a multi-processor system, if all processors are always busy, this is 100%, if all processors are 50% busy this is 50% and if 1/4 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.
Cache Hit Ratio	The percentage of time that a requested data page was found in the data cache (instead of being read from disk).
I/O - Batch Writes/ sec	The number of pages written to disk per second, using Batch I/O. The checkpoint thread is the primary user of Batch I/O.
I/O - Lazy Writes/sec	The number of pages flushed to disk per second by the Lazy Writer.

Measurement	Description
I/O - Outstanding Reads	The number of physical reads pending.
I/O - Outstanding Writes	The number of physical writes pending.
I/O - Page Reads/sec	The number of physical page reads per second.
I/O - Transactions/ sec	The number of Transact-SQL command batches executed per second.
User Connections	The number of open user connections.
% Processor Time	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%.

Sybase Server Monitoring Measurements

The following tables describe the measurements that can be monitored on a Sybase server:

Object	Measurement	Description
Network	**Average packet size (Read)**	Reports the number of network packets received.
	Average packet size (Send)	Reports the number of network packets sent.
	Network bytes (Read)	Reports the number of bytes received, over the sampling interval.
	Network bytes (Read)/sec	Reports the number of bytes received, per second.
	Network bytes (Send)	Reports the number of bytes sent, over the sampling interval.
	Network bytes (Send)/sec	Reports the number of bytes sent, per second.
	Network packets (Read)	Reports the number of network packets received, over the sampling interval.
	Network packets (Read)/ sec	Reports the number of network packets received, per second.
	Network packets (Send)	Reports the number of network packets sent, over the sampling interval.
	Network packets (Send)/ sec	Reports the number of network packets sent, per second.
Memory	**Memory**	Reports the amount of memory (in bytes) allocated for the page cache.

Object	Measurement	Description
Disk	**Reads**	Reports the number of reads made from a database device.
	Writes	Reports the number of writes made to a database device.
	Waits	Reports the number of times that access to a device had to wait.
	Grants	Reports the number of times access to a device was granted.
Engine	**Server is busy (%)**	Reports the percentage of time during which the Adaptive Server is in a "busy" state.
	CPU time	Reports how much "busy" time was used by the engine.
	Logical pages (Read)	Reports the number of data page reads, whether satisfied from cache or from a database device.
	Pages from disk (Read)	Reports the number of data page reads that could not be satisfied from the data cache.
	Pages stored	Reports the number of data pages written to a database device.
Stored Procedures	**Executed (sampling period)**	Reports the number of times a stored procedure was executed, over the sampling interval.
	Executed (session)	Reports the number of times a stored procedure was executed, during the session.
	Average duration (sampling period)	Reports the time (in seconds) spent executing a stored procedure, over the sampling interval.
	Average duration (session)	Reports the time (in seconds) spent executing a stored procedure, during the session.

Object	Measurement	Description
Locks	**% Requests**	Reports the percentage of successful requests for locks.
	Locks count	Reports the number of locks. This is an accumulated value.
	Granted immediately	Reports the number of locks that were granted immediately, without having to wait for another lock to be released.
	Granted after wait	Reports the number of locks that were granted after waiting for another lock to be released.
	Not granted	Reports the number of locks that were requested but not granted.
	Wait time (avg.)	Reports the average wait time for a lock.
SqlSrvr	**Locks/sec**	Reports the number of locks. This is an accumulated value.
	% Processor time (server)	Reports the percentage of time that the Adaptive Server is in a "busy" state.
	Transactions	Reports the number of committed Transact-SQL statement blocks (transactions).
	Deadlocks	Reports the number of deadlocks.
Cache	**% Hits**	Reports the percentage of times that a data page read could be satisfied from cache without requiring a physical page read.
	Pages (Read)	Reports the number of data page reads, whether satisfied from cache or from a database device.

Object	Measurement	Description
Cache	**Pages (Read)/sec**	Reports the number of data page reads, whether satisfied from cache or from a database device, per second.
	Pages from disk (Read)	Reports the number of data page reads that could not be satisfied from the data cache.
	Pages from disk (Read)/sec	Reports the number of data page reads, per second, that could not be satisfied from the data cache.
	Pages (Write)	Reports the number of data pages written to a database device.
	Pages (Write)/sec	Reports the number of data pages written to a database device, per second.
Process	**% Processor time (process)**	Reports the percentage of time that a process running a given application was in the "Running" state (out of the time that all processes were in the "Running" state).
	Locks/sec	Reports the number of locks, by process. This is an accumulated value.
	% Cache hit	Reports the percentage of times that a data page read could be satisfied from cache without requiring a physical page read, by process.
	Pages (Write)	Reports the number of data pages written to a database device, by process.

Object	Measurement	Description
Transaction	**Transactions**	Reports the number of committed Transact-SQL statement blocks (transactions), during the session.
Transaction	**Rows (Deleted)**	Reports the number of rows deleted from database tables during the session.
	Inserts	Reports the number of insertions into a database table during the session.
	Updates	Reports the updates to database tables during the session.
	Updates in place	Reports the sum of expensive, in-place and not-in-place updates (everything except updates deferred) during the session.
	Transactions/sec	Reports the number of committed Transact-SQL statement blocks (transactions) per second.
	Rows (Deleted)/sec	Reports the number of rows deleted from database tables, per second.
	Inserts/sec	Reports the number of insertions into a database table, per second.
	Updates/sec	Reports the updates to database tables, per second.
	Updates in place/sec	Reports the sum of expensive, in-place and not-in-place updates (everything except updates deferred), per second.

Database Server Resource Graphs User Interface

This section includes (in alphabetical order):

DB2 Graph

This graph shows the resource usage on the DB2 database server machine as a function of the elapsed load test scenario time.

X-axis	Elapsed time since the start of the run.
Y-axis	The resource usage on the DB2 database server.
Note	In order to monitor the DB2 database server machine, you must first set up the DB2 monitor environment. You then enable the DB2 monitor (from the Controller) by selecting the counters you want the monitor to measure.
See also	"Database Server Resource Graphs Overview" on page 350 "DB2 Database Manager Counters" on page 351 "DB2 Database Counters" on page 353 "DB2 Application Counters" on page 359

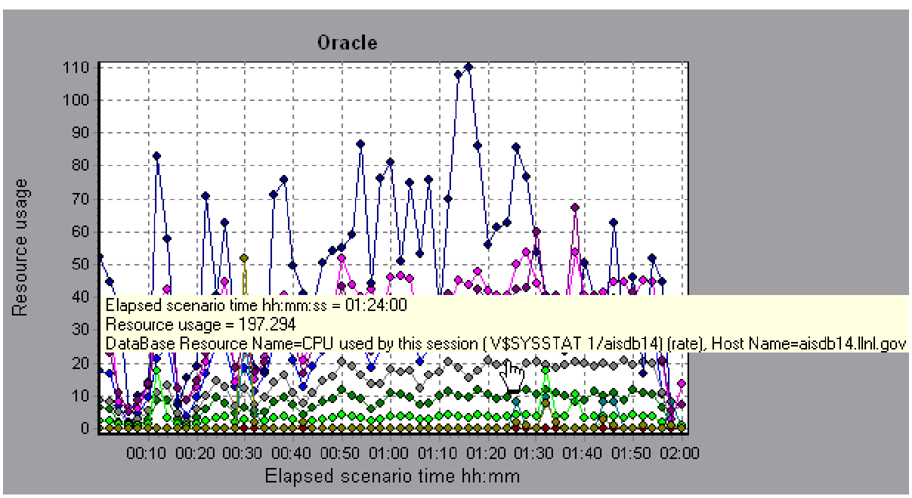
Oracle Graph

This graph displays information from Oracle V$ tables: Session statistics, V$SESSTAT, system statistics, V$SYSSTAT, and other table counters defined by the user in the custom query.

X-axis	Elapsed time since the start of the run.
Y-axis	The resource usage on the Oracle server.
Note	To obtain data for this graph, you need to enable the Oracle online monitor (from the Controller) and select the default measurements you want to display, before running the scenario.
See also	"Database Server Resource Graphs Overview" on page 350 "Oracle Server Monitoring Measurements" on page 365

Example

In the following example, the V$SYSSTAT resource values are shown as a function of the elapsed load test scenario time:

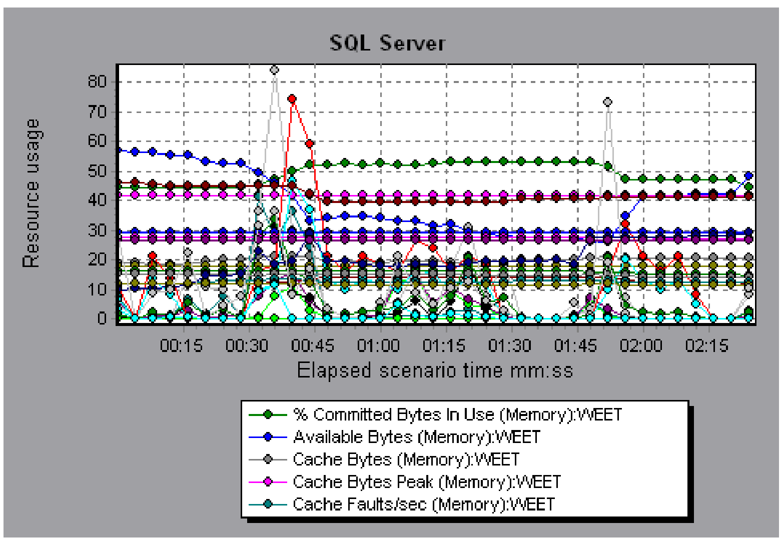


SQL Server Graph

This graph shows the standard Windows resources on the SQL server machine.

X-axis	Elapsed time since the start of the load test scenario run.
Y-axis	Resource usage
Note	To obtain data for this graph, you need to enable the SQL Server online monitor (from the Controller) and select the default measurements you want to display, before running the scenario.
See also	"Database Server Resource Graphs Overview" on page 350 "SQL Server Default Counters" on page 366

Example



Sybase Graph

This graph shows the resource usage on the Sybase database server machine as a function of the elapsed load test scenario time.

X-axis	Elapsed time since the start of the run.
Y-axis	The resource usage on the Sybase database server.
Note	In order to monitor the Sybase database server machine, you must first set up the Sybase monitor environment. You then enable the Sybase monitor (from the Controller) by selecting the counters you want the monitor to measure.
See also	"Database Server Resource Graphs Overview" on page 350 "SQL Server Default Counters" on page 366

23

Streaming Media Graphs

This chapter includes:

Concepts

Reference

Concepts

Streaming Media Graphs Overview

Streaming Media Resource graphs provide you with performance information for the RealPlayer Client, RealPlayer Server, Windows Media Server, and Media Player Client machines.

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 load test 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, refer to the *HP LoadRunner Controller User Guide*.

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

Reference

Media Player Client Monitoring Measurements

The following table describes the Media Player Client measurements that are monitored:

Measurement	Description
Average Buffering Events	The number of times Media Player Client had to buffer incoming media data due to insufficient media content.
Average Buffering Time (sec)	The time spent by Media Player Client waiting for sufficient amount of media data in order to continue playing media clip.
Current bandwidth (Kbits/ sec)	The number of kbits per second received.
Number of Packets	The number of packets sent by server for a particular media clip.
Stream Interruptions	The number of interruptions encountered by Media Player Client while playing a media clip. This measurement includes the number of times Media Player Client had to buffer incoming media data, and any errors that occurred during playback.
Stream Quality (Packet-level)	The percentage ratio of packets received to total packets.
Stream Quality (Sampling-level)	The percentage of stream samples received on time (no delays in reception).
Total number of recovered packets	The number of lost packets that were recovered. This value is only relevant during network playback.
Total number of lost packets	The number of lost packets that were not recovered. This value is only relevant during network playback.

RealPlayer Client Monitoring Measurements

The following table describes the RealPlayer Client measurements that are monitored:

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

RealPlayer Server Monitoring Measurements

The following table describes the RealPlayer Client measurements that are monitored:

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

Windows Media Server Default Measurements

Measurement	Description
Active Live Unicast Streams (Windows)	The number of live unicast streams that are being streamed.
Active Streams	The number of streams that are being streamed.
Active TCP Streams	The number of TCP streams that are being streamed.
Active UDP Streams	The number of UDP streams that are being streamed.
Aggregate Read Rate	The total, aggregate rate (bytes/sec) of file reads.
Aggregate Send Rate	The total, aggregate rate (bytes/sec) of stream transmission.
Connected Clients	The number of clients connected to the server.
Connection Rate	The rate at which clients are connecting to the server.
Controllers	The number of controllers currently connected to the server.
HTTP Streams	The number of HTTP streams being streamed.
Late Reads	The number of late read completions per second.
Pending Connections	The number of clients that are attempting to connect to the server, but are not yet connected. This number may be high if the server is running near maximum capacity and cannot process a large number of connection requests in a timely manner.
Stations	The number of station objects that currently exist on the server.
Streams	The number of stream objects that currently exist on the server.
Stream Errors	The cumulative number of errors occurring per second.

Streaming Media Graphs User Interface

This section includes (in alphabetical order):

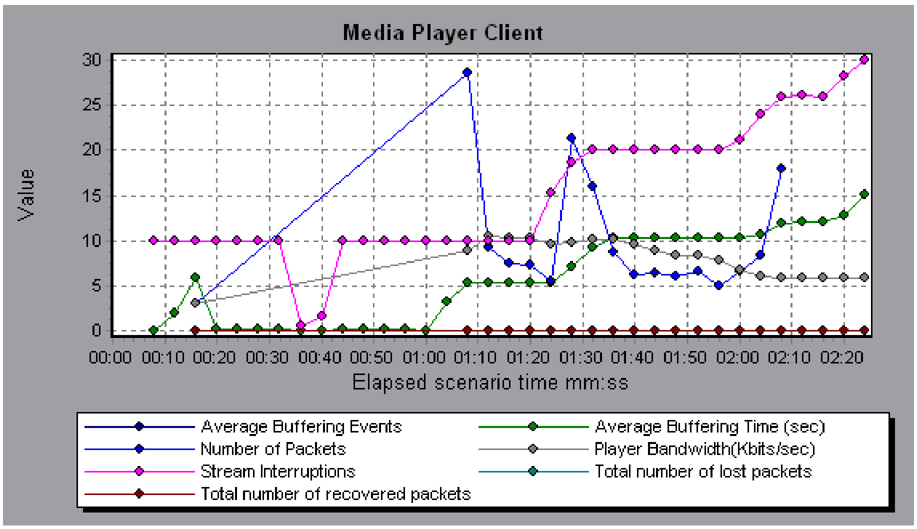
Media Player Client Graph

This graph shows statistics on the Windows Media Player client machine as a function of the elapsed load test scenario time.

X-axis	Elapsed time since the start of the run.
Y-axis	The resource usage on the Windows Media Player client machine.
See also	"Streaming Media Graphs Overview" on page 378 "Media Player Client Monitoring Measurements" on page 379

Example

In the following example the **Total number of recovered packets** remained steady during the first two and a half minutes of the scenario. The **Number of Packets** and **Stream Interruptions** fluctuated significantly. The **Average Buffering Time** increased moderately, and the **Player Bandwidth** increased and then decreased moderately.The scale factor for the **Stream Interruptions** and **Average Buffering Events** measurements is 10, and the scale factor for **Player Bandwidth** is 1/10.



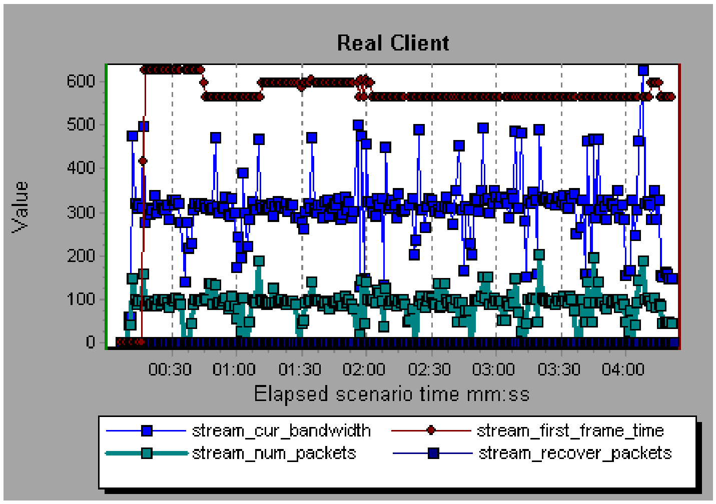
Real Client Graph

This graph shows statistics on the RealPlayer client machine as a function of the elapsed load test scenario time.

X-axis	Elapsed time since the start of the run.
Y-axis	The resource usage on the RealPlayer client machine.
See also	"Streaming Media Graphs Overview" on page 378 "RealPlayer Client Monitoring Measurements" on page 380

Example

In the following example 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. The scale factor is the same for all of the measurements.



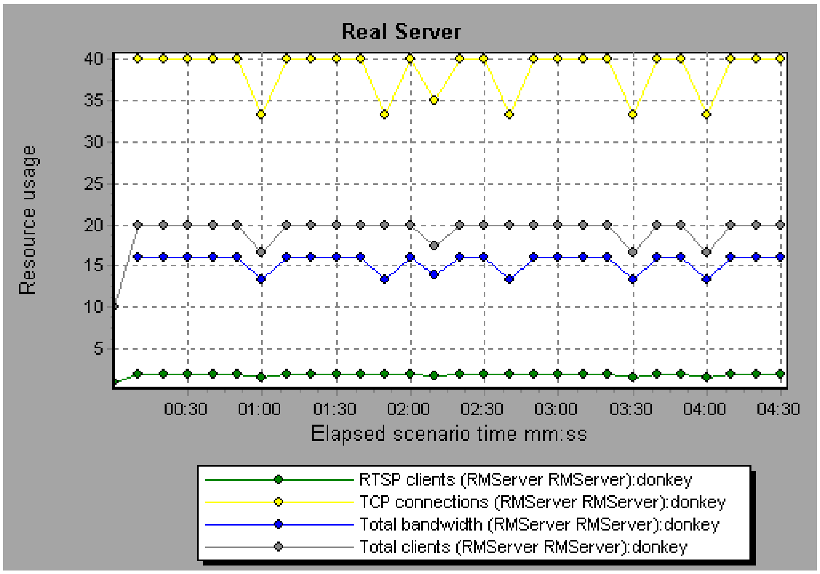
Real Server Graph

This graph shows RealPlayer server statistics as a function of the elapsed load test scenario time.

X-axis	Elapsed time since the start of the run.
Y-axis	The resource usage of the RealPlayer server machine.
Note	To obtain data for this graph, you need to enable the RealPlayer Server online monitor (from the Controller) and select the default measurements you want to display, before running the scenario.
See also	"Streaming Media Graphs Overview" on page 378 "RealPlayer Server Monitoring Measurements" on page 381

Example

In the following example 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. The scale factor is the same for all of the measurements.



Windows Media Server Graph

This graph shows the Windows Media server statistics as a function of the elapsed load test scenario time.

X-axis	Elapsed time since the start of the run.
Y-axis	Resource usage.

Note	To obtain data for this graph, you need to enable the Windows Media Server online monitor (from the Controller) and select the default measurements you want to display, before running the scenario.
See also	"Streaming Media Graphs Overview" on page 378 "Windows Media Server Default Measurements" on page 382

24

ERP/CRM Server Resource Graphs

This chapter includes:

Concepts

Reference

Concepts

ERP/CRM Server Resource Graphs Overview

ERP/CRM server resource monitor graphs provide you with performance information for ERP/CRM servers. To obtain data for these graphs, you must activate the ERP/CRM server resource online monitor before running the scenario. When you set up the online monitor for ERP/CRM server resources, you indicate which statistics and measurements to monitor. For more information on activating and configuring ERP/CRM server resource monitors, refer to the *HP LoadRunner Controller User Guide*.

Reference

ERP/CRM Server Resources Graphs Measurements

PeopleSoft(Tuxedo) Graph Counters

The following table describes the default counters that can be measured. It is recommended to pay particular attention to the following measurements: % Busy Clients, Active Clients, Busy Clients, Idle Clients, and all the queue counters for the APPQ/PSAPPSRV queue.

Monitor	Measurements
Machine	**% Busy Clients**. The percentage of active clients currently logged in to the Tuxedo application server that are waiting for a response from the application server.
	Active Clients. The total number of active clients currently logged in to the Tuxedo application server.
	Busy Clients. The total number of active clients currently logged in to the Tuxedo application server that are waiting for a response from the application server.
	Current Accessers. The number of clients and servers currently accessing the application either directly on this machine or through a workstation handler on this machine.
	Current Transactions. The number of in use transaction table entries on this machine.
	Idle Clients. The total number of active clients currently logged in to the Tuxedo application server that are not waiting for a response from the application server.
	Workload Completed/second. The total workload on all the servers for the machine that was completed, per unit time.
	Workload Initiated/second. The total workload on all the servers for the machine that was initiated, per unit time.

Monitor	Measurements
Queue	**% Busy Servers**. The percentage of active servers currently handling Tuxedo requests.
	Active Servers. The total number of active servers either handling or waiting to handle Tuxedo requests.
	Busy Servers. The total number of active servers currently busy handling Tuxedo requests.
	Idle Servers. The total number of active servers currently waiting to handle Tuxedo requests.
	Number Queued. The total number of messages which have been placed on the queue.
Server	**Requests/second.** The number of server requests handled per second
	Workload/second. The workload is a weighted measure of the server requests. Some requests could have a different weight than others. By default, the workload is always 50 times the number of requests.
Workstation Handler (WSH)	**Bytes Received/sec**. The total number of bytes received by the workstation handler, per second.
	Bytes Sent/sec. The total number of bytes sent back to the clients by the workstation handler, per second.
	Messages Received/sec. The number of messages received by the workstation handler, per second.
	Messages Sent/sec. The number of messages sent back to the clients by the workstation handler, per second.
	Number of Queue Blocks/sec. The number of times the queue for the workstation handler blocked, per second. This gives an idea of how often the workstation handler was overloaded.

SAP Server Graph Counters

The following are the most commonly monitored counters for a SAP server:

Measurement	Description
Average CPU time	The average CPU time used in the work process.
Average response time	The average response time, measured from the time a dialog sends a request to the dispatcher work process, through the processing of the dialog, until the dialog is completed and the data is passed to the presentation layer. The response time between the SAP GUI and the dispatcher is not included in this value.
Average wait time	The average amount of time that an unprocessed dialog step waits in the dispatcher queue for a free work process. Under normal conditions, the dispatcher work process should pass a dialog step to the application process immediately after receiving the request from the dialog step. Under these conditions, the average wait time would be a few milliseconds. A heavy load on the application server or on the entire system causes queues at the dispatcher queue.
Average load time	The time needed to load and generate objects, such as ABAP source code and screen information, from the database.
Database calls	The number of parsed requests sent to the database.
Database requests	The number of logical ABAP requests for data in the database. These requests are passed through the R/3 database interface and parsed into individual database calls. The proportion of database calls to database requests is important. If access to information in a table is buffered in the SAP buffers, database calls to the database server are not required. Therefore, the ratio of calls/requests gives an overall indication of the efficiency of table buffering. A good ratio would be 1:10.

Measurement	Description
Roll ins	The number of rolled-in user contexts.
Roll outs	The number of rolled-out user contexts.
Roll in time	The processing time for roll ins.
Roll out time	The processing time for roll outs.
Roll wait time	The queue time in the roll area. When synchronous RFCs are called, the work process executes a roll out and may have to wait for the end of the RFC in the roll area, even if the dialog step is not yet completed. In the roll area, RFC server programs can also wait for other RFCs sent to them.
Average time per logical DB call	The average response time for all commands sent to the database system (in milliseconds). The time depends on the CPU capacity of the database server, the network, the buffering, and on the input/output capabilities of the database server. Access times for buffered tables are many magnitudes faster and are not considered in the measurement.

SAPGUI Graph Counters

The following are the most commonly monitored counters:

Measurement	Description
Average CPU time	The average CPU time used in the work process.
Average response time	The average response time, measured from the time a dialog sends a request to the dispatcher work process, through the processing of the dialog, until the dialog is completed and the data is passed to the presentation layer. The response time between the SAPGUI and the dispatcher is not included in this value.

Measurement	Description
Average wait time	The average amount of time that an unprocessed dialog step waits in the dispatcher queue for a free work process. Under normal conditions, the dispatcher work process should pass a dialog step to the application process immediately after receiving the request from the dialog step. Under these conditions, the average wait time would be a few milliseconds. A heavy load on the application server or on the entire system causes queues at the dispatcher queue.
Average load time	The time needed to load and generate objects, such as ABAP source code and screen information, from the database.
Database calls	The number of parsed requests sent to the database.
Database requests	The number of logical ABAP requests for data in the database. These requests are passed through the R/3 database interface and parsed into individual database calls. The proportion of database calls to database requests is important. If access to information in a table is buffered in the SAP buffers, database calls to the database server are not required. Therefore, the ratio of calls/requests gives an overall indication of the efficiency of table buffering. A good ratio would be 1:10.
Roll ins	The number of rolled-in user contexts.
Roll outs	The number of rolled-out user contexts.
Roll in time	The processing time for roll ins.
Roll out time	The processing time for roll outs.

Measurement	Description
Roll wait time	The queue time in the roll area. When synchronous RFCs are called, the work process executes a roll out and may have to wait for the end of the RFC in the roll area, even if the dialog step is not yet completed. In the roll area, RFC server programs can also wait for other RFCs sent to them.
Average time per logical DB call	The average response time for all commands sent to the database system (in milliseconds). The time depends on the CPU capacity of the database server, the network, the buffering, and on the input/output capabilities of the database server. Access times for buffered tables are many magnitudes faster and are not considered in the measurement.

SAP Portal Graph Counters

The following are the monitored counters for a SAP Portal system server:

Measurement	Description
Accumulated Amount of Outbound Data (bytes)	The accumulated amount of outbound data, measured in bytes.
Time for all Requests (ms)	The total time (in milliseconds) taken for processing all requests.
Average Amount of Outbound Data per Request (bytes)	The average amount of outbound data per request, measured in bytes.
Average Number of Component Calls per Request (bytes)	The average number of component calls per request, measured in bytes.
Average Time of a Request (ms)	The average amount of time (in milliseconds) taken to process a request.
Number of Calls with Outbound Data	The total number of calls with outbound data.

Measurement	Description
Number of Component Calls for all Requests	The total number of component calls for all requests.
Number of Requests since First Request	The total number of requests since the first request was made.
Requests per Second	The number of requests made per second.
Time Stamp of First Request	The time stamp of the first request.

Siebel Server Manager Graph Counters

The following are the monitored counters for a Siebel Server Manager server.

Measurement	Description
Average Connect Time	The average connection time.
Average Reply Size	The average size of a user reply.
Average Request Size	The average size of a user request.
Average Requests Per Session	The average number of user requests per session.
Average Response Time	The average amount of time that it takes the server to respond to a request.
Average Think Time	The average amount of think time taken to respond to a request.
Avg SQL Execute Time	The average SQL execute time.
Avg SQL Fetch Time	The average SQL fetch time.
Avg SQL Parse Time	The average SQL parse time.
CPU Time	The CPU time used in the work process.
Elapsed Time	The total amount of elapsed time.
Num of DBConn Retries	The number of database connection retries.
Num of DLRbk Retries	The number of DLRbk retries.

Measurement	Description
Num of Exhausted Retries	The total number of retries that expired.
Number of SQL Executes	The total number of SQL executes.
Number of SQL Fetches	The total number of SQL fetches.
Number of SQL Parses	The total number of SQL parses.
Number of Sleeps	The number of sleeps.
Object Manager Errors	The total number of object manager errors.
Reply Messages	The total number of reply messages.
Request Messages	The total number of request messages.
SQL Execute Time	The total SQL execute time.
SQL Fetch Time	The total SQL fetch time.
SQL Parse Time	The total SQL parse time.
Sleep Time	The total sleep time.
Tests Attempted	The number of tests attempted.
Tests Failed	The number of tests that failed.
Tests Successful	The number of tests that were successful.
Total Reply Size	The total reply size, measured in bytes.
Total Request Size	The total request size, measured in bytes.
Total Response Time	The total response time.
Total Tasks	The total number of tasks.
Total Think Time	The total think time.

Siebel Web Server Graph Counters

The following are the monitored counters for a Siebel Web Server:

Measurement	Description
Anonymous sessions requested from the pool	The number of anonymous sessions requested from the pool.
Open Session Time	The time taken for users to log on to the system.
Anon Session Removed	The number of anonymous sessions removed from the pool.
Anon Session Available	The number of anonymous sessions available in the pool.
Anonymous sessions returns to the pool	The number of anonymous sessions returned to the pool.
Response Time	The time taken to respond to a user request.
Close Session Time	The time taken for users to log off the system.
Request Time	The time taken to process the user request.

ERP/CRM Server Resource Graphs User Interface

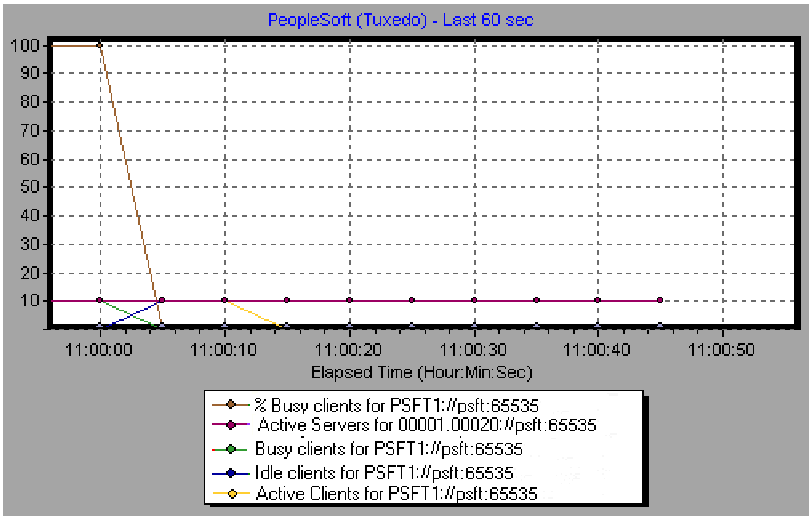
This section includes (in alphabetical order):

PeopleSoft (Tuxedo) Graph

This graph shows the resource usage of your Tuxedo server as a function of the elapsed load test scenario time

X-axis	Elapsed time from the start of the run.
Y-axis	The resource usage on the Tuxedo server.
Note	To obtain data for this graph, you need to enable the PeopleSoft (Tuxedo) monitor (from the Controller) and select the default measurements you want to display, before running the scenario.
See also	"ERP/CRM Server Resource Graphs Overview" on page 390 "PeopleSoft(Tuxedo) Graph Counters" on page 391

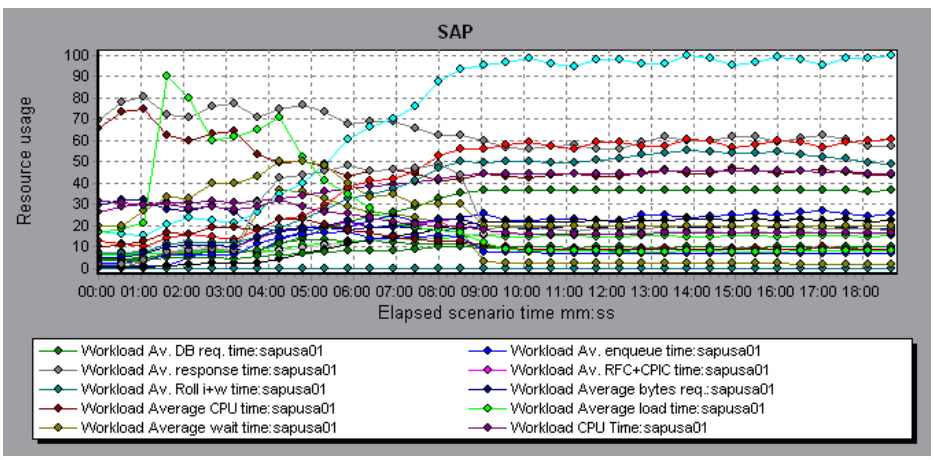
Example


PeopleSoft (Tuxedo) - Last 60 sec
100
90
80
70
60
50
40
30
20
10
11:00:00
11:00:10
11:00:20
11:00:30
11:00:40
11:00:50
Elapsed Time (Hour:Min:Sec)
% Busy clients for PSFT1://psft:65535
Active Servers for 00001.00020://psft:65535
Busy clients for PSFT1://psft:65535
Idle clients for PSFT1://psft:65535
Active Clients for PSFT1://psft:65535

SAP Graph

This graph shows the resource usage of a SAP server as a function of the elapsed load test scenario time.

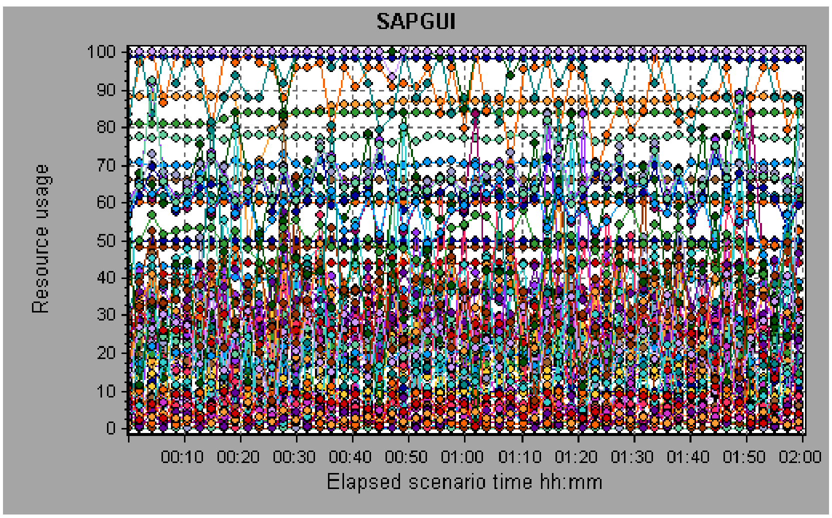
X-axis	Elapsed time since the start of the run
Y-axis	The resource usage on the SAP server.
Note	To obtain data for this graph, you need to enable the SAP online monitor (from the Controller) and select the default measurements you want to display, before running the scenario. (There are differences in the scale factor for some of the measurements).
See also	"ERP/CRM Server Resource Graphs Overview" on page 390 "SAP Server Graph Counters" on page 394



SAPGUI Graph

This graph shows the resource usage of a SAP server as a function of the elapsed load test scenario time.

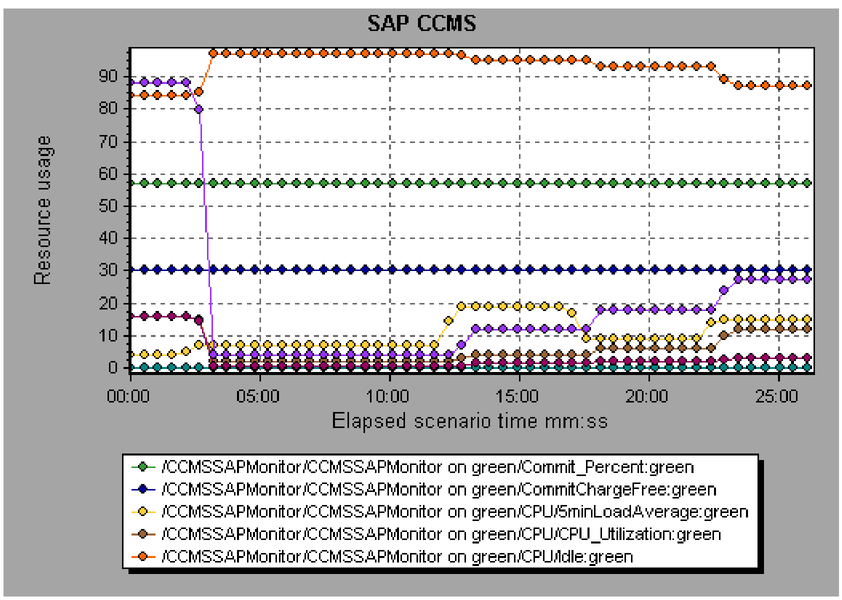
X-axis	Elapsed time since the start of the run.
Y-axis	The resource usage on the SAP server.
Note	To obtain data for this graph, you need to enable the SAP online monitor (from the Controller) and select the default measurements you want to display, before running the scenario. (There are differences in the scale factor for some of the measurements).
See also	"ERP/CRM Server Resource Graphs Overview" on page 390 "SAPGUI Graph Counters" on page 395



SAP CCMS Graph

The SAP CCMS (Computer Center Management System) graph shows statistics about the resource usage of all the servers in a SAP R/3 landscape server during the load test scenario run. The x-axis represents the time that has elapsed since the start of the scenario run. The y-axis represents the resource usage.

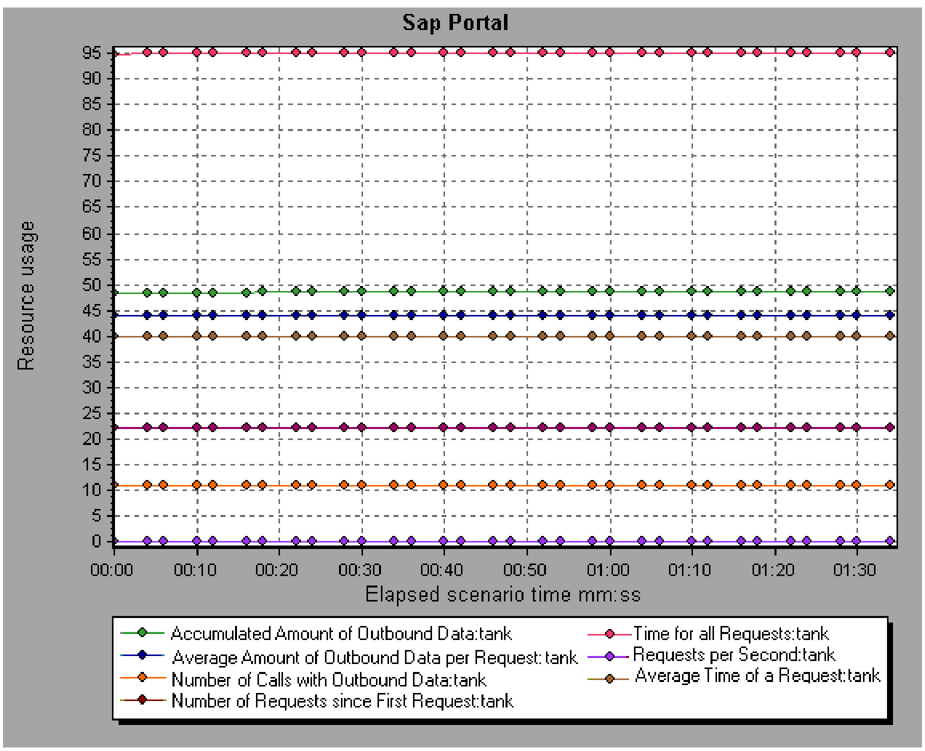
X-axis	Elapsed time since the start of the run.
Y-axis	The resource usage on all servers in a SAP R/3 landscape server.
Note	To obtain data for this graph, you need to enable the SAP CCMS monitor (from the Controller) and select the default measurements you want to display, before running the scenario.
See also	"ERP/CRM Server Resource Graphs Overview" on page 390



SAP Portal Graph

This graph shows the resource usage of your SAP Portal server as a function of the elapsed load test scenario time.

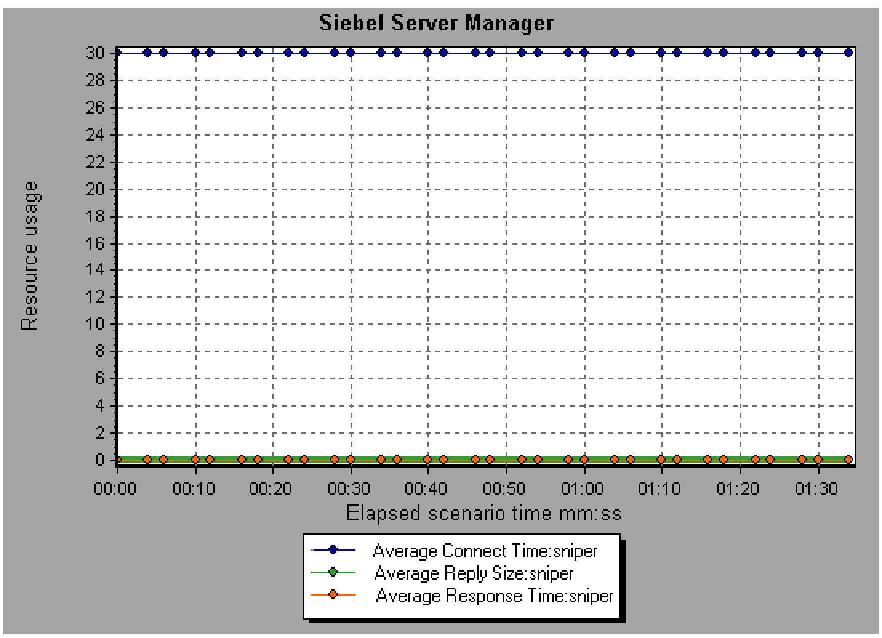
X-axis	Elapsed time since the start of the run.
Y-axis	The resource usage on the SAP Portal server.
Note	To obtain data for this graph, you need to enable the SAP Portal monitor (from the Controller) and select the default measurements you want to display, before running the scenario.
See also	"ERP/CRM Server Resource Graphs Overview" on page 390 "SAP Portal Graph Counters" on page 397



Siebel Server Manager Graph

This graph shows the resource usage of your Siebel Server Manager server as a function of the elapsed load test scenario time.

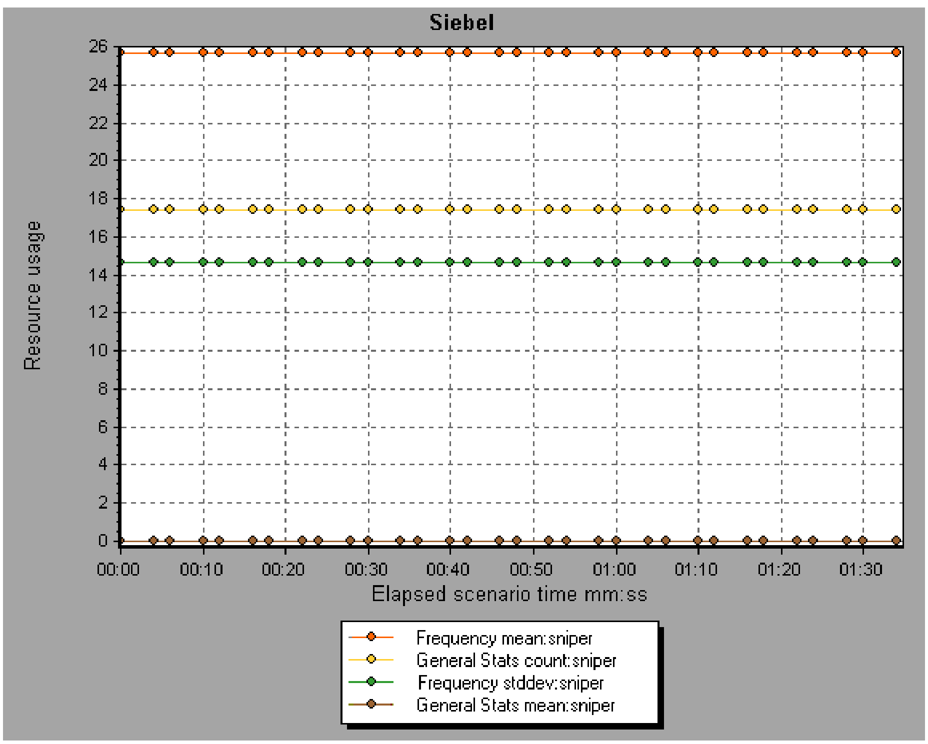
X-axis	Elapsed time since the start of the run.
Y-axis	The resource usage on the Siebel Server Manager server.
Note	To obtain data for this graph, you need to enable the Siebel Server Manager monitor (from the Controller) and select the default measurements you want to display, before running the scenario.
See also	"ERP/CRM Server Resource Graphs Overview" on page 390 "Siebel Server Manager Graph Counters" on page 398



Siebel Web Server Graph

This graph shows the resource usage of your Siebel Web Server as a function of the elapsed load test scenario time.

X-axis	Elapsed time since the start of the run.
Y-axis	The resource usage on the Siebel Web Server.
Note	To obtain data for this graph, you need to enable the Siebel Web Server monitor (from the Controller) and select the default measurements you want to display, before running the scenario.
See also	"ERP/CRM Server Resource Graphs Overview" on page 390 "Siebel Web Server Graph Counters" on page 400



25

Application Component Graphs

This chapter includes:

Concepts

Reference

Concepts

Microsoft COM+ Performance Graphs Overview

Microsoft COM+ performance graphs provide you with performance information for COM+ interfaces and methods.

In order to obtain data for these graphs, you need to activate the various Microsoft COM+ performance monitors before running the load test scenario.

When you set up the Microsoft COM+ performance online monitors, you indicate which statistics and measurements to monitor. For more information on activating and configuring the Microsoft COM+ performance monitors, refer to the *HP LoadRunner Controller User Guide*.

Microsoft .NET CLR Performance Graphs Overview

The .NET CLR performance graphs provide you with performance information for .NET classes and methods. To obtain data for these graphs, you must activate the .NET CLR performance monitor before running the load test scenario run.

Displayed measurements are specified using the .NET monitor. For more information on activating and configuring the .NET CLR performance monitor, refer to the *HP LoadRunner Controller User Guide*.

Reference

Application Component Graphs User Interface

This section includes (in alphabetical order):

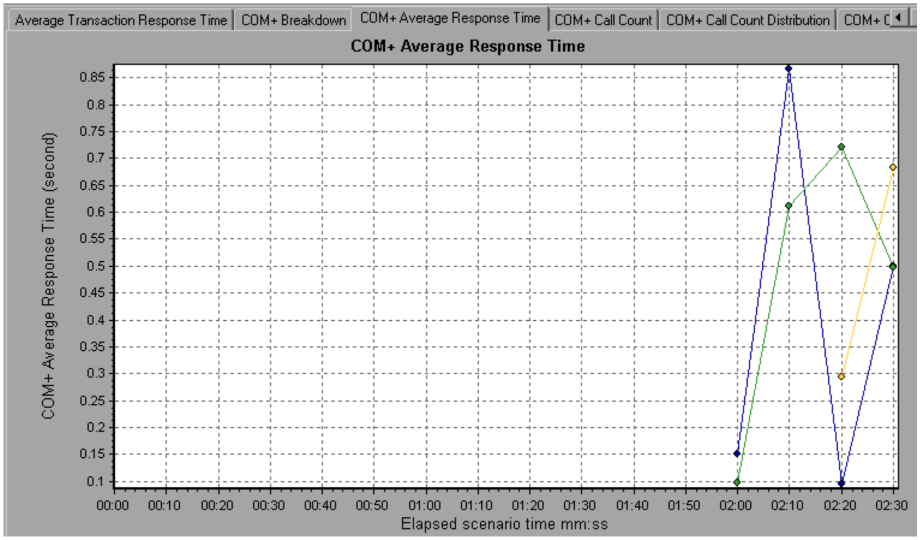
COM+ Average Response Time Graph

This graph specifies the average time COM+ interfaces or methods take to perform during the load test scenario.

<table>
<tr><td>X-axis</td><td>Elapsed time from the beginning of the scenario run.</td></tr>
<tr><td>Y-axis</td><td>Average response time of a COM+ interface or method.</td></tr>
<tr><td>Breakdown options</td><td>Each interface or method is represented by a different colored line on the graph. The legend frame (which is found below the graph) identifies the interfaces by color:
<table>
<tr><th>Color</th><th>Scale</th><th>Measurement</th><th>Minimum</th><th>Average</th><th>Maximum</th></tr>
<tr><td>☑</td><td>1</td><td>ContrRnd\COMPlusServer.ConstTime\IDispatch</td><td>0.096</td><td>0.499</td><td>1.501</td></tr>
<tr><td>☑</td><td>1</td><td>ContrRnd\COMPlusServer.ConstTime_ConstTime</td><td>0</td><td>0.5</td><td>1.502</td></tr>
<tr><td>☑</td><td>1</td><td>ContrRnd\COMPlusServer.RandomTime_RandomTime</td><td>0.058</td><td>0.391</td><td>0.747</td></tr>
</table>
This legend shows that the blue colored line belongs to the COM+ interface _ConstTime. Looking at the graph above, we see that this interface has higher response times than all other COM+ interfaces. At 2:10 minutes into the scenario, it records an average response time of 0.87 seconds.

Note: The 0.87 second data point is an average, taken from all data points recorded within a 10 second interval (the default granularity). You can change the length of this sample interval.

Viewing CON+ Methods
The table initially displays COM+ interfaces, but you can also view the list of COM+ methods by using drill-down or filtering techniques. For more information, see "Filtering and Sorting Graph Data" on page 93 and "Drilling Down in a Graph" on page 119.</td></tr>
<tr><td>Tips</td><td>To highlight a specific interface line in the graph, select the interface row in the legend.</td></tr>
<tr><td>See also</td><td>"Microsoft COM+ Performance Graphs Overview" on page 410</td></tr>
</table>

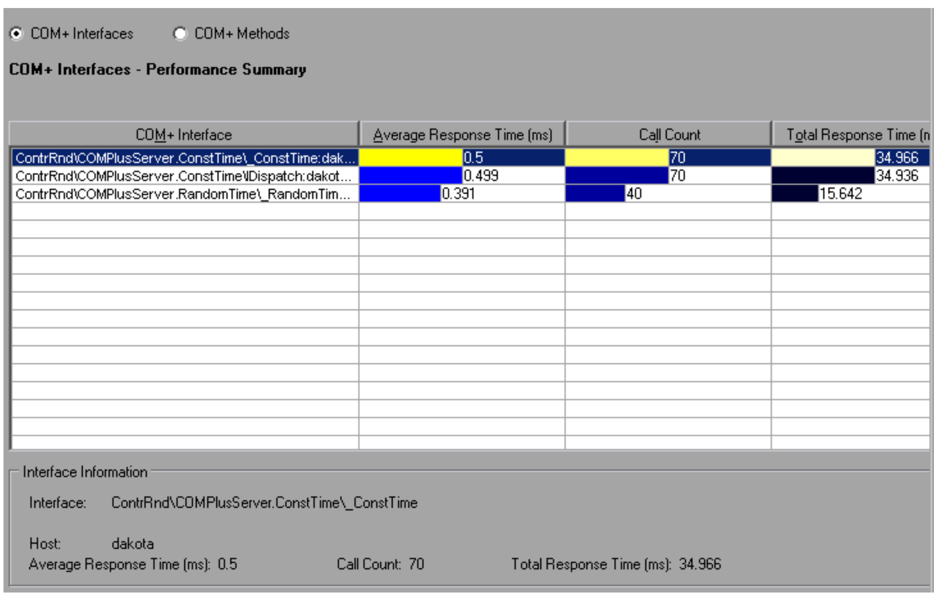

Average Transaction Response Time
COM+ Breakdown
COM+ Average Response Time
COM+ Call Count
COM+ Call Count Distribution
COM+ Average Response Time
COM+ Average Response Time (second)
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
00:00 00:10 00:20 00:30 00:40 00:50 01:00 01:10 01:20 01:30 01:40 01:50 02:00 02:10 02:20 02:30
Elapsed scenario time mm:ss

COM+ Breakdown Graph

This graph summarizes fundamental result data about COM+ interfaces or methods and presents it in table format.

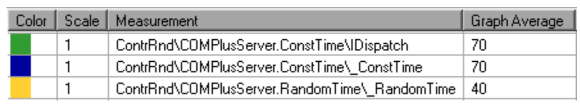
Purpose	Using the COM+ Breakdown table, you can quickly identify the COM+ interfaces or methods which consume the most time during the test. The table can be sorted by column, and the data can be viewed either by COM+ interface or COM+ method.
Breakdown options	Average Response Time The **Average Response Time** column shows how long, on average, an interface or method takes to perform. The graphical representation of this column is the COM+ Average Response Time Graph. Call Count The next column, **Call Count**, specifies the number of times the interface or method was invoked. The graphical representation of this column is the COM+ Average Response Time Graph. Total Response Time The final column, **Total Response Time**, specifies how much time was spent overall on the interface or method. It is calculated by multiplying the first two data columns together. The graphical representation of this column is the COM+ Average Response Time Graph. The graphical representations of each of these columns are the COM+ Average Response Time Graph, the COM+ Call Count Distribution Graph and the COM+ Total Operation Time Distribution Graph Interfaces are listed in the **COM+ Interface** column in the form **Interface:Host**. In the table above, the **_ConstTime** interface took an average of .5 seconds to execute and was called 70 times. Overall, this interface took 34.966 seconds to execute.

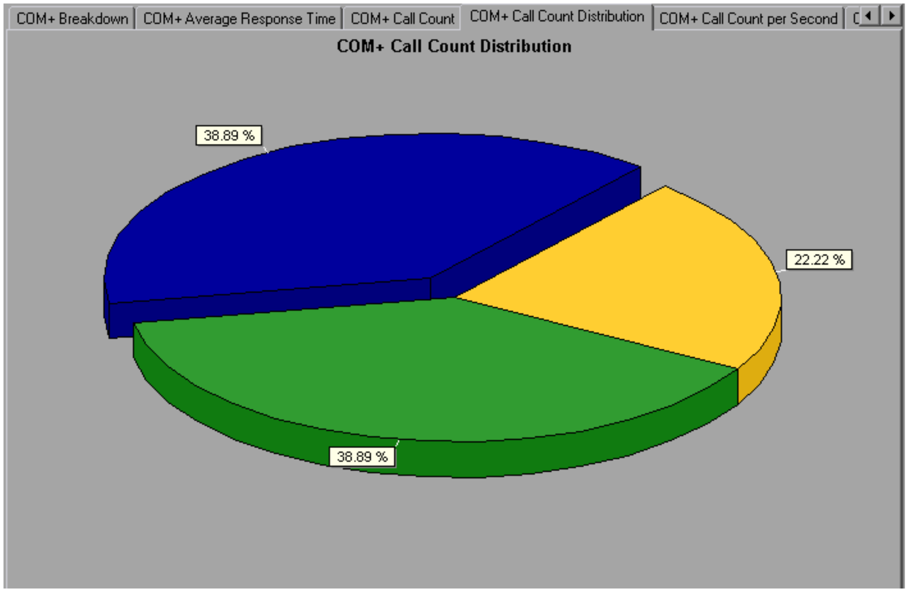
Tips	Sorting List To sort the list by a column, click on the column heading. The list above is sorted by **Average Response Time** which contains the triangle icon specifying a sort in descending order. Viewing COM+ Methods The table initially displays COM+ interfaces, but you can also view the list of COM+ methods. To view the methods of a selected interface, select the **COM+ Methods** option. You can also double-click on the interface row to view the methods. The methods of the specified interface are listed in the **COM+ Method** column.
See also	"Microsoft COM+ Performance Graphs Overview" on page 410



COM+ Call Count Distribution Graph

This graph shows the percentage of calls made to each COM+ interface compared to all COM+ interfaces. It can also show the percentage of calls made to a specific COM+ method compared to other methods within the interface

<table>
<tr><td>Breakdown options</td><td>
The number of calls made to the interface or method is listed in the Call Count column of the COM+ Breakdown Graph table.
Each interface or method is represented by a different colored area on the pie graph. The legend frame (which is found below the graph) identifies the interfaces by color:

<table>
<tr><th>Color</th><th>Scale</th><th>Measurement</th><th>Graph Average</th></tr>
<tr><td></td><td>1</td><td>ContrRnd\COMPlusServer.ConstTime\IDispatch</td><td>70</td></tr>
<tr><td></td><td>1</td><td>ContrRnd\COMPlusServer.ConstTime_ConstTime</td><td>70</td></tr>
<tr><td></td><td>1</td><td>ContrRnd\COMPlusServer.RandomTime_RandomTime</td><td>40</td></tr>
</table>
This legend shows that the green colored area belongs to the COM+ interface IDispatch. Looking at the graph above, we see that 38.89% of calls are made to this interface. The actual figures can be seen in the Call Count column of the COM+ Breakdown Graph table: there are 13 calls to this interface out of a total of 49 calls.
Viewing COM+ Methods
The table initially displays COM+ interfaces, but you can also view the list of COM+ methods by using drill-down or filtering techniques. For more information, see "Filtering and Sorting Graph Data" on page 93 and "Drilling Down in a Graph" on page 119.
</td></tr>
<tr><td>Tips</td><td>To highlight a specific interface line in the graph, select the interface row in the legend.</td></tr>
<tr><td>See also</td><td>"Microsoft COM+ Performance Graphs Overview" on page 410</td></tr>
</table>

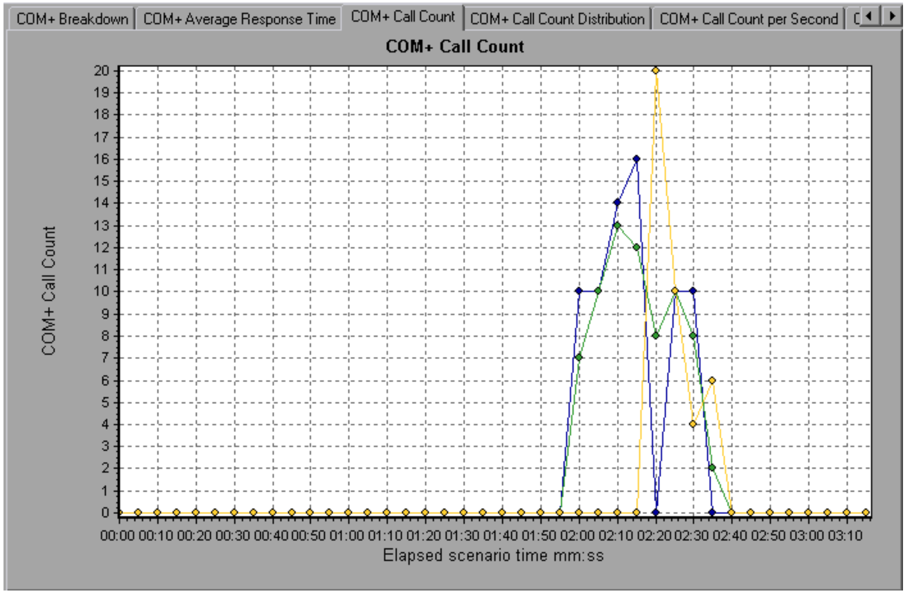

COM+ Breakdown
COM+ Average Response Time
COM+ Call Count
COM+ Call Count Distribution
COM+ Call Count per Second
COM+ Call Count Distribution
38.89 %
22.22 %
38.89 %

COM+ Call Count Graph

This graph displays the number of times COM+ interfaces and methods are invoked during the test.

<table>
<tr><td>X-axis</td><td>Elapsed time from the beginning of the scenario run.</td></tr>
<tr><td>Y-axis</td><td>How many calls were made to a COM+ interface or method.</td></tr>
<tr><td>Breakdown options</td><td>Each interface or method is represented by a different colored line on the graph. The legend frame (which is found below the graph) identifies the interfaces by color:
<table>
<tr><th>Color</th><th>Scale</th><th>Measurement</th><th>Graph Minimum</th><th>Average</th><th>Graph Maximum</th></tr>
<tr><td></td><td>1</td><td>ContrRnd\COMPlusServer.ConstTime\IDispatch</td><td>0</td><td>1.777</td><td>13</td></tr>
<tr><td></td><td>1</td><td>ContrRnd\COMPlusServer.ConstTime_ConstTime</td><td>0</td><td>1.777</td><td>16</td></tr>
<tr><td></td><td>1</td><td>ContrRnd\COMPlusServer.RandomTime_RandomTime</td><td>0</td><td>1.015</td><td>20</td></tr>
</table>
This legend shows that the yellow colored line belongs to the COM+ interface _RandomTime. Looking at the graph above, we see that calls to this interface begin at the beginning of the scenario run. There are 20 calls at the 2:20 minute point.

Viewing COM+ Methods
The table initially displays COM+ interfaces, but you can also view the list of COM+ methods by using drill-down or filtering techniques. For more information, see "Filtering and Sorting Graph Data" on page 93 and "Drilling Down in a Graph" on page 119.</td></tr>
<tr><td>Note</td><td>The call count is computed by multiplying the call frequency by a time interval. As a result, the reported measurement may be rounded.</td></tr>
<tr><td>Tips</td><td>To highlight a specific interface line in the graph, select the interface row in the legend.</td></tr>
<tr><td>See also</td><td>"Microsoft COM+ Performance Graphs Overview" on page 410</td></tr>
</table>

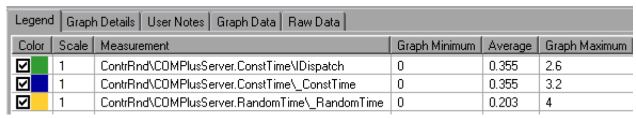

COM+ Breakdown
COM+ Average Response Time
COM+ Call Count
COM+ Call Count Distribution
COM+ Call Count per Second
COM+ Call Count
COM+ Call Count
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
00:00 00:10 00:20 00:30 00:40 00:50 01:00 01:10 01:20 01:30 01:40 01:50 02:00 02:10 02:20 02:30 02:40 02:50 03:00 03:10
Elapsed scenario time mm:ss

COM+ Call Count Per Second Graph

This graph shows the number of times per second a COM+ interface or method is invoked.

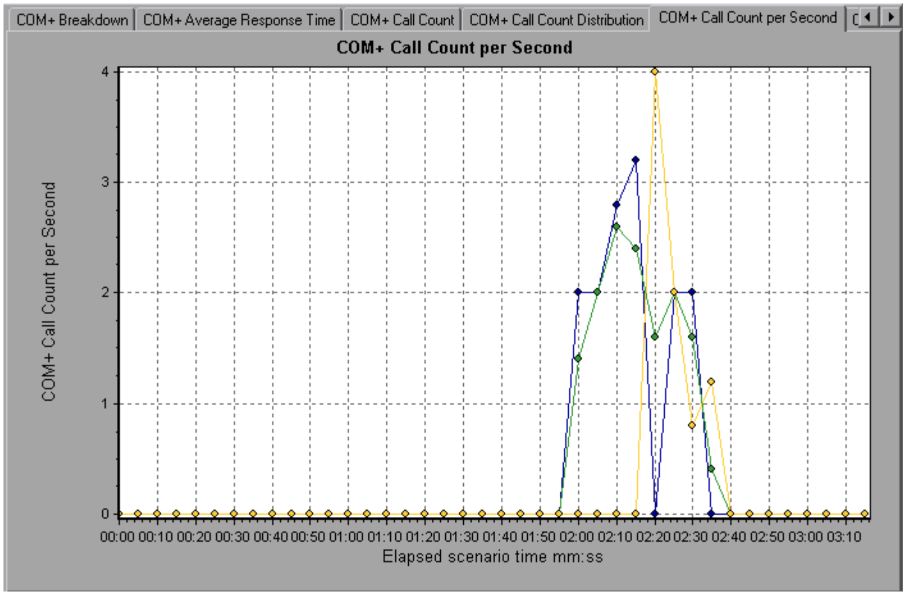
<table>
<tr>
<td>Breakdown options</td>
<td>
This graph is similar to the COM+ Call Count Graph except that the y-axis indicates how many invocations were made to a COM+ interface or method per second.

Each interface or method is represented by a different colored line on the graph. The legend frame (which is found below the graph) identifies the interfaces by color:


<table>
<tr><td>Legend</td><td>Graph Details</td><td>User Notes</td><td>Graph Data</td><td>Raw Data</td><td></td></tr>
<tr><td>Color</td><td>Scale</td><td>Measurement</td><td>Graph Minimum</td><td>Average</td><td>Graph Maximum</td></tr>
<tr><td></td><td>1</td><td>ContrRnd\COMPlusServer.ConstTime\IDispatch</td><td>0</td><td>0.355</td><td>2.6</td></tr>
<tr><td></td><td>1</td><td>ContrRnd\COMPlusServer.ConstTime_ConstTime</td><td>0</td><td>0.355</td><td>3.2</td></tr>
<tr><td></td><td>1</td><td>ContrRnd\COMPlusServer.RandomTime_RandomTime</td><td>0</td><td>0.203</td><td>4</td></tr>
</table>

This legend shows that the green colored line belongs to the COM+ interface IDispatch. Looking at the graph above, we see that calls to this interface begins 1:55 minutes into the scenario run. There is an average of 2.5 calls per second at the 2:10 minute mark.

Viewing COM+ Methods
To view the average response time of the individual methods within a COM+ interface, see "Filtering and Sorting Graph Data" on page 93 and "Drilling Down in a Graph" on page 119.
</td>
</tr>
<tr>
<td>Tips</td>
<td>To highlight a specific interface line in the graph, select the interface row in the legend.</td>
</tr>
<tr>
<td>See also</td>
<td>"Microsoft COM+ Performance Graphs Overview" on page 410</td>
</tr>
</table>

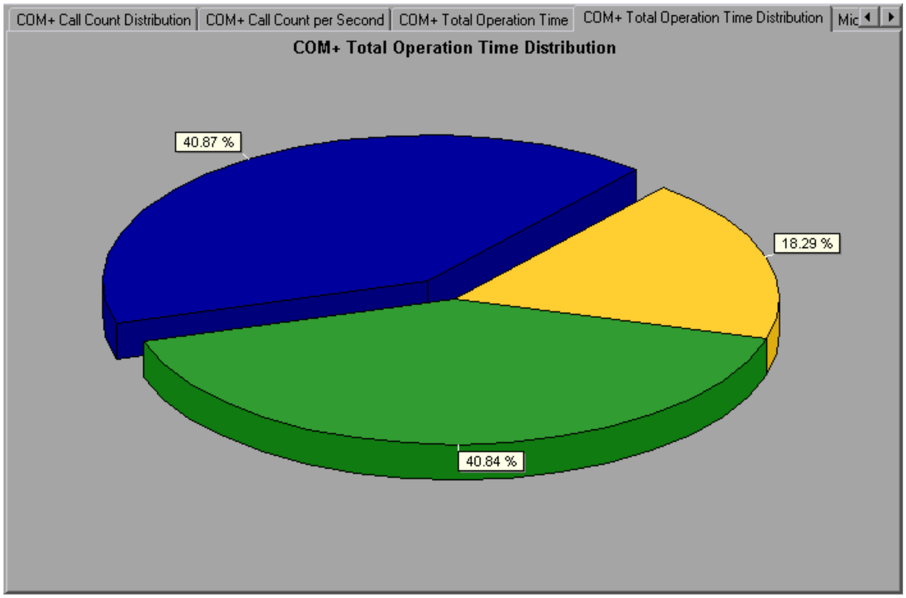

COM+ Breakdown
COM+ Average Response Time
COM+ Call Count
COM+ Call Count Distribution
COM+ Call Count per Second
COM+ Call Count per Second
COM+ Call Count per Second
0
1
2
3
4
00:00 00:10 00:20 00:30 00:40 00:50 01:00 01:10 01:20 01:30 01:40 01:50 02:00 02:10 02:20 02:30 02:40 02:50 03:00 03:10
Elapsed scenario time mm:ss

COM+ Total Operation Time Distribution Graph

This graph shows the percentage of time a specific COM+ interface takes to execute in relation to all COM+ interfaces. It can also show the percentage of time a COM+ method takes to execute in relation to all COM+ methods within the interface.

<table>
<tr><td>Purpose</td><td>Use it to identify those interfaces or methods which take up an excessive amount of time.</td></tr>
<tr><td>Breakdown options</td><td>Each interface or method is represented by a different colored area on the pie graph. The legend frame (which is found below the graph) identifies the interfaces by color:
<table>
<tr><th>Color</th><th>Scale</th><th>Measurement</th><th>Graph Average</th></tr>
<tr><td></td><td>1</td><td>ContrRnd\COMPlusServer.ConstTime\IDispatch</td><td>34.936</td></tr>
<tr><td></td><td>1</td><td>ContrRnd\COMPlusServer.ConstTime_ConstTime</td><td>34.966</td></tr>
<tr><td></td><td>1</td><td>ContrRnd\COMPlusServer.RandomTime_RandomTime</td><td>15.642</td></tr>
</table>
This legend shows that the green colored line belongs to the COM+ interface IDispatch. Looking at the graph above, we see that this interface takes up 40.84% of the COM+ operational time.

Viewing COM+ Methods
To view the average response time of the individual methods within a COM+ interface, see "Filtering and Sorting Graph Data" on page 93 and "Drilling Down in a Graph" on page 119.</td></tr>
<tr><td>Tips</td><td>To highlight a specific interface line in the graph, select the interface row in the legend.</td></tr>
<tr><td>See also</td><td>"Microsoft COM+ Performance Graphs Overview" on page 410</td></tr>
</table>

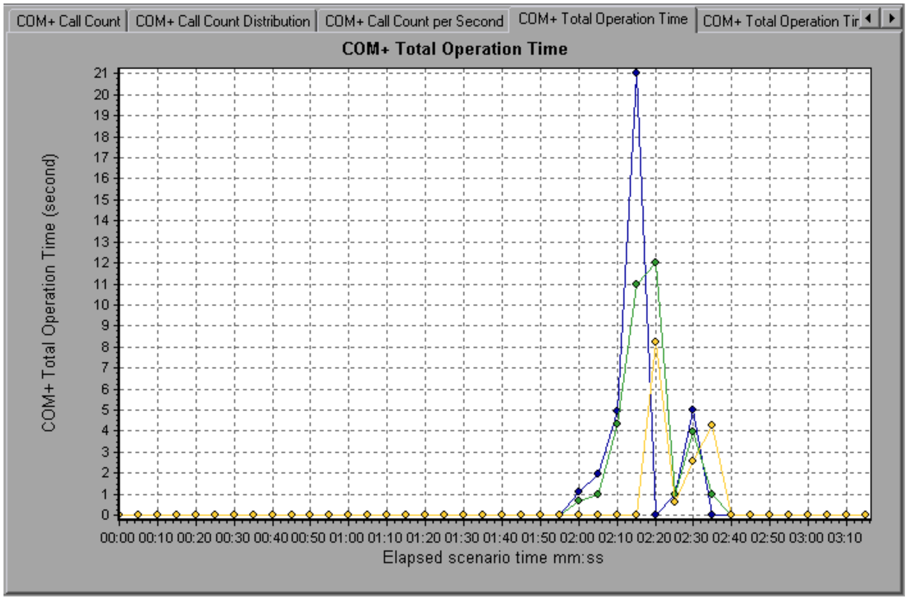

COM+ Call Count Distribution
COM+ Call Count per Second
COM+ Total Operation Time
COM+ Total Operation Time Distribution
Mic
COM+ Total Operation Time Distribution
40.87 %
18.29 %
40.84 %

COM+ Total Operation Time Graph

This graph displays the amount of time each COM+ interface or method takes to execute during the test.

<table>
<tr><td>Purpose</td><td>Use it to identify those interfaces or methods which take up an excessive amount of time.</td></tr>
<tr><td>X-axis</td><td>Elapsed time from the beginning of the scenario run.</td></tr>
<tr><td>Y-axis</td><td>Total time a COM+ interface or method is in operation.</td></tr>
<tr><td>Breakdown options</td><td>Each interface or method is represented by a different colored line on the graph. The legend frame (which is found below the graph) identifies the interfaces by color:
<table>
<tr><th>Color</th><th>Scale</th><th>Measurement</th><th>Graph Minimum</th><th>Average</th><th>Graph Maximum</th></tr>
<tr><td>☑</td><td>1</td><td>ContrRnd\COMPlusServer.ConstTime\IDispatch</td><td>0</td><td>0.887</td><td>12.008</td></tr>
<tr><td>☑</td><td>1</td><td>ContrRnd\COMPlusServer.ConstTime_ConstTime</td><td>0</td><td>0.887</td><td>21.026</td></tr>
<tr><td>☑</td><td>1</td><td>ContrRnd\COMPlusServer.RandomTime_RandomTime</td><td>0</td><td>0.397</td><td>8.24</td></tr>
</table>
This legend shows that the blue colored line belongs to the COM+ interface _ConstTime. Looking at the graph above, we see that throughout the scenario, this interface consumes more time than any other, especially at 2 minutes and 15 seconds into the scenario run, where the calls to this interface take an average of 21 seconds.

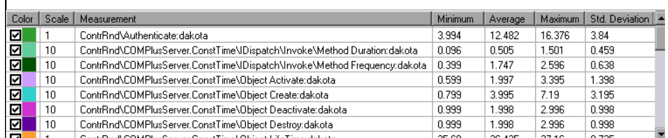
Viewing COM+ Methods
The table initially displays COM+ interfaces, but you can also view the list of COM+ methods by using drill-down or filtering techniques. For more information, see "Filtering and Sorting Graph Data" on page 93 and "Drilling Down in a Graph" on page 119.</td></tr>
<tr><td>Tips</td><td>To highlight a specific interface line in the graph, select the interface row in the legend.</td></tr>
<tr><td>See also</td><td>"Microsoft COM+ Performance Graphs Overview" on page 410</td></tr>
</table>

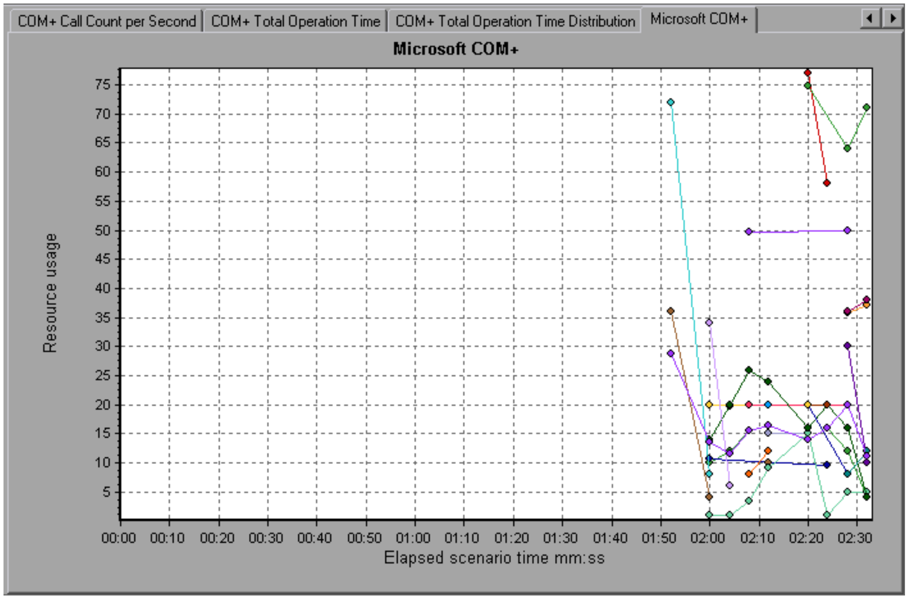


COM+ Call Count
COM+ Call Count Distribution
COM+ Call Count per Second
COM+ Total Operation Time
COM+ Total Operation Tir
COM+ Total Operation Time
COM+ Total Operation Time (second)
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
00:00 00:10 00:20 00:30 00:40 00:50 01:00 01:10 01:20 01:30 01:40 01:50 02:00 02:10 02:20 02:30 02:40 02:50 03:00 03:10
Elapsed scenario time mm:ss

Microsoft COM+ Graph

This graph shows the resource usage of COM+ objects as a function of the elapsed load test scenario time.

<table>
<tr><td>X-axis</td><td>Elapsed time since the start of the run.</td></tr>
<tr><td>Y-axis</td><td>The resource usage of COM+ objects.</td></tr>
<tr><td>Breakdown Options</td><td>Each COM+ object is represented by a different colored line on the graph. The legend frame (which is found below the graph) identifies the objects by color:

<table>
<tr><th>Color</th><th>Scale</th><th>Measurement</th><th>Minimum</th><th>Average</th><th>Maximum</th><th>Std. Deviation</th></tr>
<tr><td></td><td>1</td><td>ContrRnd\Authenticate:dakota</td><td>3.994</td><td>12.482</td><td>16.376</td><td>3.84</td></tr>
<tr><td></td><td>10</td><td>ContrRnd\COMPlusServer.ConstTime\IDispatch\Invoke\Method Duration:dakota</td><td>0.096</td><td>0.505</td><td>1.501</td><td>0.459</td></tr>
<tr><td></td><td>10</td><td>ContrRnd\COMPlusServer.ConstTime\IDispatch\Invoke\Method Frequency:dakota</td><td>0.399</td><td>1.747</td><td>2.596</td><td>0.638</td></tr>
<tr><td></td><td>10</td><td>ContrRnd\COMPlusServer.ConstTime\Object Activate:dakota</td><td>0.599</td><td>1.997</td><td>3.395</td><td>1.398</td></tr>
<tr><td></td><td>10</td><td>ContrRnd\COMPlusServer.ConstTime\Object Create:dakota</td><td>0.799</td><td>3.995</td><td>7.19</td><td>3.195</td></tr>
<tr><td></td><td>10</td><td>ContrRnd\COMPlusServer.ConstTime\Object Deactivate:dakota</td><td>0.999</td><td>1.998</td><td>2.996</td><td>0.998</td></tr>
<tr><td></td><td>10</td><td>ContrRnd\COMPlusServer.ConstTime\Object Destroy:dakota</td><td>0.999</td><td>1.998</td><td>2.996</td><td>0.998</td></tr>
<tr><td></td><td>1</td><td>ContrRnd\COMPlusServer.ConstTime\Object LifeTime:dakota</td><td>35.69</td><td>36.425</td><td>37.16</td><td>0.735</td></tr>
</table>
See the following tables for default counters:
➤ "Authentication Metrics" on page 427
➤ "Application Event" on page 427
➤ "Thread Event" on page 428
➤ "Transaction Events" on page 428
➤ "Object Events" on page 429
➤ "Method Events" on page 430</td></tr>
<tr><td>See also</td><td>"Microsoft COM+ Performance Graphs Overview" on page 410</td></tr>
</table>



Authentication Metrics

Measurement	Description
Authenticate	Frequency of successful method call level authentication. When you set an authentication level for an application, you determine what degree of authentication is performed when clients call into the application.
Authenticate Failed	Frequency of failed method call level authentication.

Application Event

Measurement	Description
Activation	Frequency of application activation or startup.
Shutdown	Frequency of application shutdown or termination.

Thread Event

Measurement	Description
Thread Start	Rate at which single-threaded apartment (STA) thread for application have been started.
Thread Terminate	Rate at which single-threaded apartment (STA) thread for application have been terminated.
Work Enque	Event sent if a work is queued in single thread apartment object (STA). Note: These events are not signaled/sent in Windows Server 2003 and later.
Work Reject	Event sent if a work is rejected from single thread apartment object (STA). Note: These events are not signaled/sent in Windows Server 2003 and later.

Transaction Events

Measurement	Description
Transaction Duration	Duration of COM+ transactions for selected application.
Transaction Start	Rate at which transactions have started.
Transaction Prepared	Rate at which transactions have completed the prepare phase of the two-phase protocol.
Transaction Aborted	Rate at which transactions have been aborted.
Transaction Commit	Rate at which transactions have completed the commit protocol.

Object Events

Measurement	Description
Object Life Time	Duration of object existence (from instantiation to destruction).
Object Create	Rate at which new instances of this object are created.
Object Destroy	Rate at which instances of the object are destroyed.
Object Activate	Rate of retrieving instances of a new JIT-activated object.
Object Deactivation	Rate of freeing JIT-activated object via SetComplete or SetAbort.
Disable Commit	Rate of client calls to DisableCommit on a context. DisableCommit declares that the object's transactional updates are inconsistent and can't be committed in their present state.
Enable Commit	Rate of client calls to EnableCommit on a context. EnableCommit declares that the current object's work is not necessarily finished, but that its transactional updates are consistent and could be committed in their present form.
Set Complete	Rate of client calls to SetComplete on a context. SetComplete declares that the transaction in which the object is executing can be committed, and that the object should be deactivated on returning from the currently executing method call.
Set Abort	Rate of client calls to SetAbort on a context. SetAbort declares that the transaction in which the object is executing must be aborted, and that the object should be deactivated on returning from the currently executing method call.

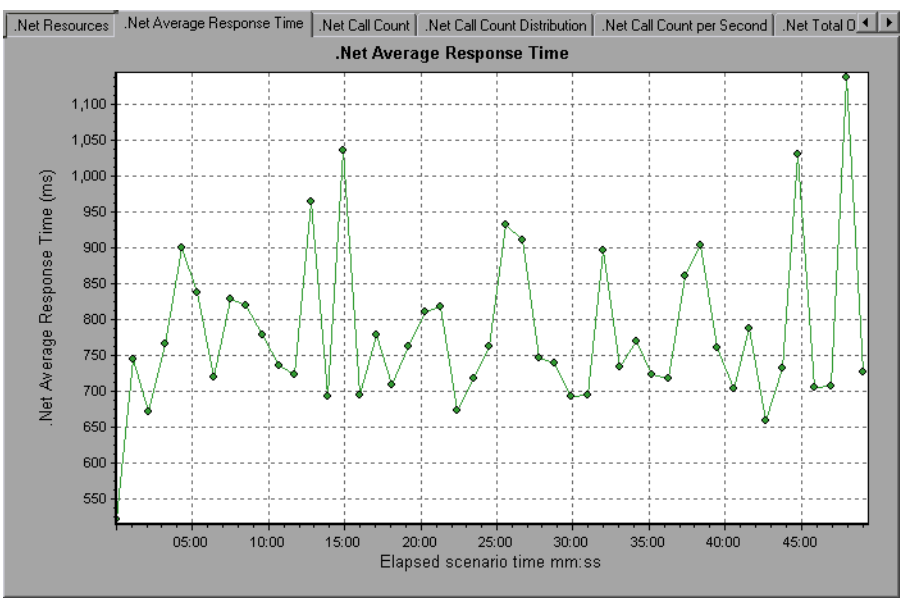
Method Events

Measurement	Description
Method Duration	Average duration of method.
Method Frequency	Frequency of method invocation.
Method Failed	Frequency of failed methods (i.e. methods that return error HRESULT codes).
Method Exceptions	Frequency of exceptions thrown by selected method.

.NET Average Response Time Graph

This graph specifies the average time that .NET classes or methods took to perform during the load test scenario run.

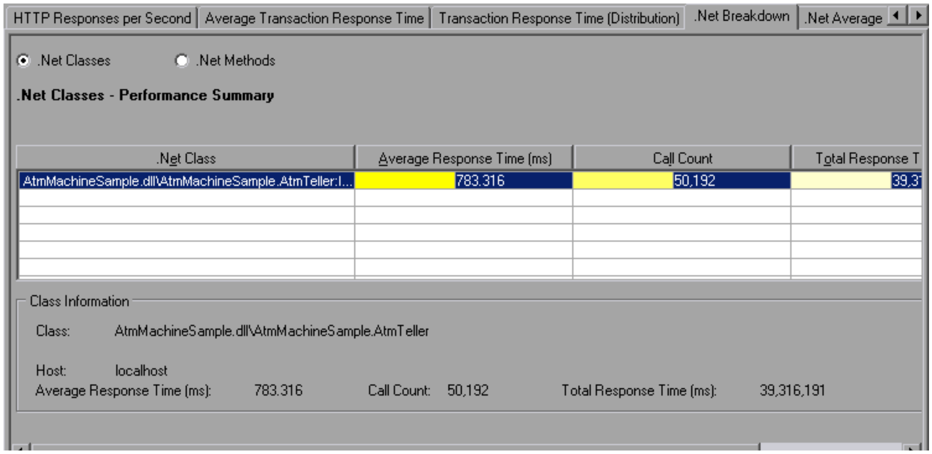
X-axis	Elapsed time from the beginning of the scenario run.
Y-axis	Average response time of a .NET class or method.
Breakdown options	The graph initially displays .NET classes, but you can also view the individual methods within a .NET class by using drill-down or filtering techniques. For more information, see "Filtering and Sorting Graph Data" on page 93 and "Drilling Down in a Graph" on page 119.
Tips	You can change the length of the sample interval. For more information, refer to the *HP LoadRunner Controller User Guide*. **Hint**: To highlight a specific class line in the graph, select the class row in the legend (displayed below the graph).
See also	"Microsoft COM+ Performance Graphs Overview" on page 410



.NET Breakdown Graph

This graph summarizes fundamental result data about .NET classes or methods and presents it in table format.

Purpose	Using the .NET Breakdown table, you can quickly identify the .NET classes or methods which consume the most time during the test. The table can be sorted by column, and the data can be viewed either by .NET class or .NET method.
Breakdown options	The **Average Response Time** column shows how long, on average, a class or method took to perform. The next column, **Call Count**, specifies the number of times the class or method was invoked. The final column, **Total Response Time**, specifies how much time was spent overall on the class or method. It is calculated by multiplying the results from the first two columns together. Classes are listed in the **.NET Class** column in the form **Class:Host**. In the table above, the **AtmMachineSample.AtmTeller** class took an average of 783 seconds to execute and was called 50,912 times. Overall, this class took 39,316 seconds to execute. To sort the list by a column, click the column heading. Each column in the .NET Breakdown graph is graphically represented by another graph. For details, see ".NET Breakdown graph" on page 433: The table initially displays .NET classes, but you can also view the list of .NET methods. To view .NET methods, select the **.NET Methods** option, or double-click the class row. The methods of the specified class are listed in the **.NET Method** column.
See also	"Microsoft COM+ Performance Graphs Overview" on page 410



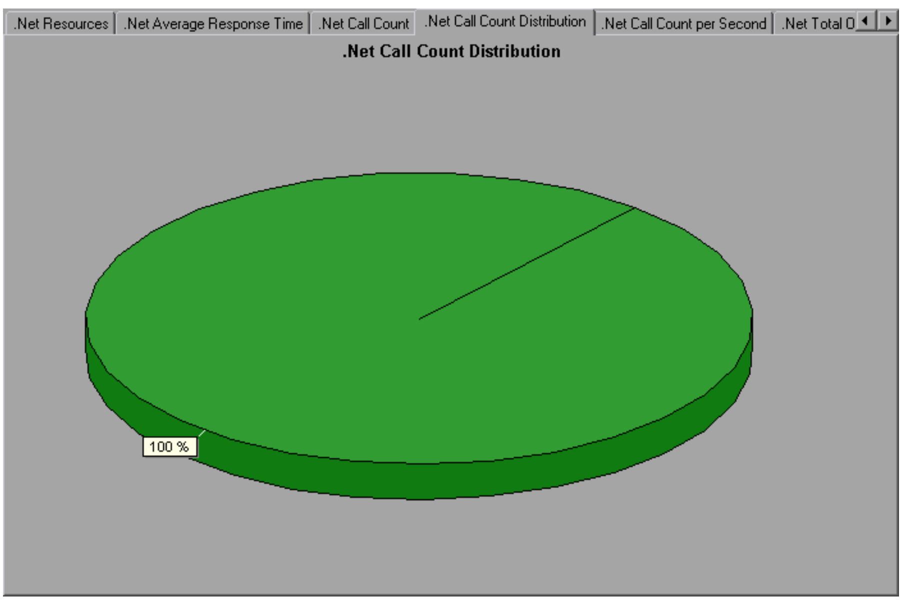
.NET Breakdown graph

.NET Breakdown Column	Graphical Representation
Average Response Time	.NET Average Response Time Graph.
Call Count	.NET Call Count Graph.
Total Response Time	.NET Total Operation Time Distribution Graph.

.NET Call Count Distribution Graph

This graph shows the percentage of calls made to each .NET class compared to all .NET classes. It can also show the percentage of calls made to a specific .NET method compared to other methods within the class

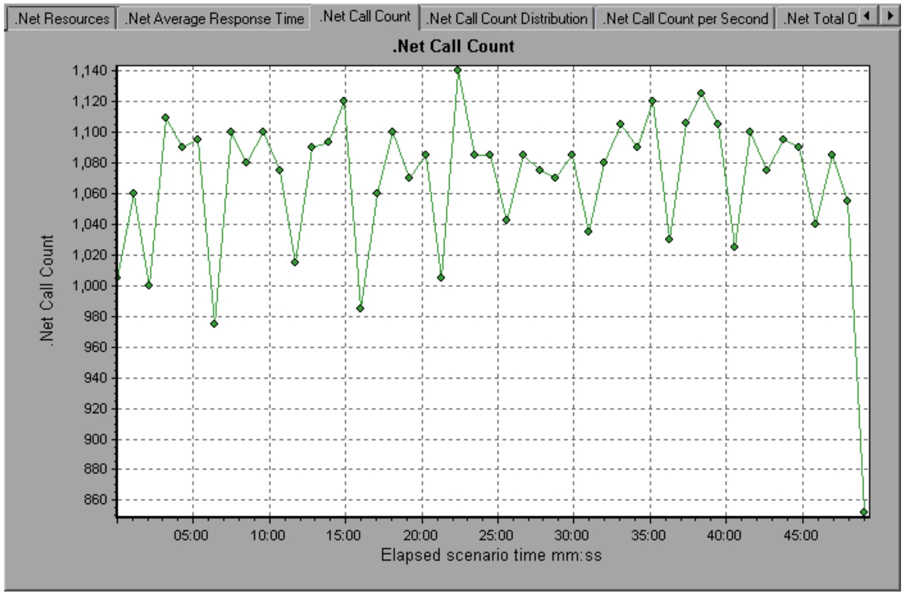
Breakdown options	The number of calls made to the class or method is listed in the **Call Count** column of the .NET Breakdown graph table. The graph initially displays .NET classes, but you can also view the individual methods within a .NET class by using drill-down or filtering techniques. For more information, see "Filtering and Sorting Graph Data" on page 93 and "Drilling Down in a Graph" on page 119.
Tips	To highlight a specific class line in the graph, select the class row in the legend (displayed below the graph).
See also	"Microsoft COM+ Performance Graphs Overview" on page 410



.NET Call Count Graph

This graph displays the number of times that .NET classes and methods are invoked during the test.

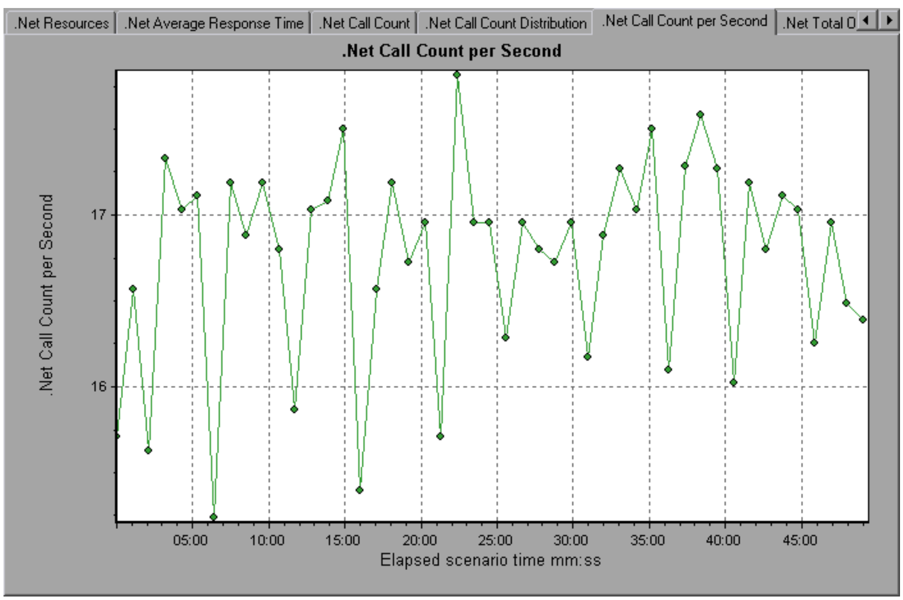
X-axis	Elapsed time from the beginning of the scenario run.
Y-axis	Indicates how many calls were made to a .NET class or method.
Breakdown options	The graph initially displays .NET classes, but you can also view the individual methods within a .NET class by using drill-down or filtering techniques. For more information, see "Filtering and Sorting Graph Data" on page 93 and "Drilling Down in a Graph" on page 119.
Tips	To highlight a specific class line in the graph, select the class row in the legend (displayed below the graph).
Note	The call count is computed by multiplying the call frequency by a time interval. As a result, the reported measurement may be rounded.
See also	"Microsoft COM+ Performance Graphs Overview" on page 410


.Net Resources
.Net Average Response Time
.Net Call Count
.Net Call Count Distribution
.Net Call Count per Second
.Net Total O
.Net Call Count
.Net Call Count
1,140
1,120
1,100
1,080
1,060
1,040
1,020
1,000
980
960
940
920
900
880
860
05:00
10:00
15:00
20:00
25:00
30:00
35:00
40:00
45:00
Elapsed scenario time mm:ss

.NET Call Count per Second Graph

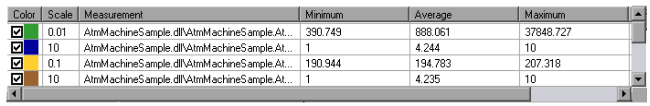
This graph shows the number of times per second that a .NET class or method is invoked.

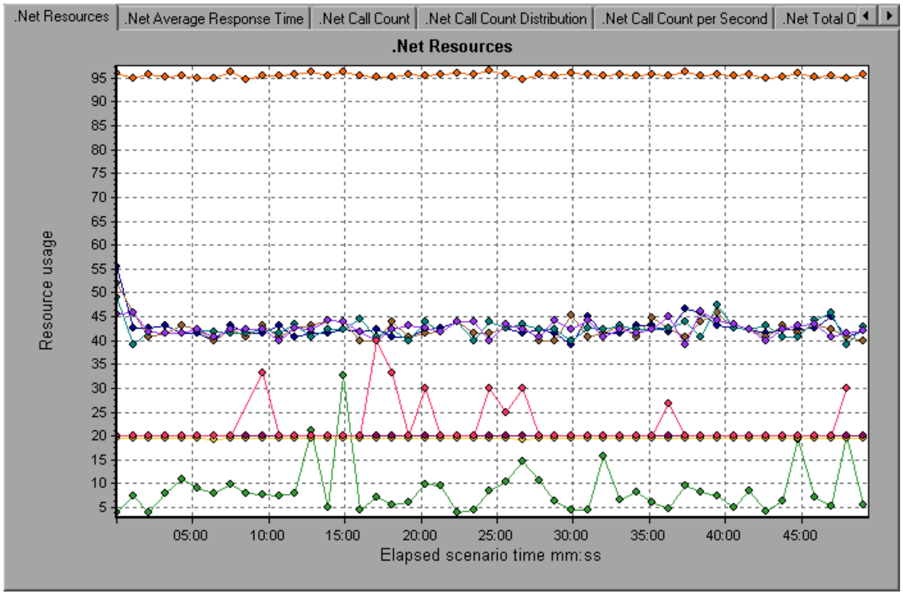
Breakdown options	This graph is similar to the .NET Call Count graph except that the y-axis indicates how many invocations were made to a .NET class or method per second. The graph initially displays .NET classes, but you can also view the individual methods within a .NET class by using drill-down or filtering techniques. For more information, see "Filtering and Sorting Graph Data" on page 93 and "Drilling Down in a Graph" on page 119.
Tips	To highlight a specific class line in the graph, select the class row in the legend (displayed below the graph).
See also	"Microsoft COM+ Performance Graphs Overview" on page 410



.NET Resources Graph

This graph shows the resource usage of .NET methods as a function of the elapsed load test scenario time.

<table>
<tr>
<td>Breakdown options</td>
<td>
Each .NET method is represented by a different colored line on the graph. The legend frame (located below the graph) identifies the methods by color:

<table>
<tr><th>Color</th><th>Scale</th><th>Measurement</th><th>Minimum</th><th>Average</th><th>Maximum</th></tr>
<tr><td></td><td>0.01</td><td>AtmMachineSample.dll\AtmMachineSample.At...</td><td>390.749</td><td>888.061</td><td>37848.727</td></tr>
<tr><td></td><td>10</td><td>AtmMachineSample.dll\AtmMachineSample.At...</td><td>1</td><td>4.244</td><td>10</td></tr>
<tr><td></td><td>0.1</td><td>AtmMachineSample.dll\AtmMachineSample.At...</td><td>190.944</td><td>194.783</td><td>207.318</td></tr>
<tr><td></td><td>10</td><td>AtmMachineSample.dll\AtmMachineSample.At...</td><td>1</td><td>4.235</td><td>10</td></tr>
</table>
You can monitor .NET counters at the application, assembly, class, and method levels. Measurements that take place before the application is fully loaded (such as Assembly Load Time, that measures the time it takes to load an assembly) will not be measured.
The following tables describe the counters that can be measured at each level. All durations are reported in seconds, and all frequencies are reported per five-second polling periods. For example, if 20 events occur in a 5 second polling period, the reported frequency is 4.
➤ "Application Level" on page 439
➤ "Assembly Level" on page 441
➤ "Class Level" on page 441
➤ "Method Level" on page 441
</td>
</tr>
<tr>
<td>See also</td>
<td>"Microsoft COM+ Performance Graphs Overview" on page 410</td>
</tr>
</table>



Application Level

Measurement	Description
Application Lifetime	Monitors the duration of the application in seconds.
Exception Frequency	Monitors the number of exceptions per second, in the five second polling period.
JIT (Just In Time) Duration	Monitors the time (in seconds) it takes for the JIT to compile code.
Thread Creation Frequency	Monitors the number of threads that are created in a polling period.
Thread Lifetime	Monitors the duration of threads.
Domain Creation Frequency	Monitors the number of domain creations in a polling period. (Domains protect areas of code. All applications run in a domain which keeps them encapsulated, so that they cannot interfere with other applications outside the domain.)

Measurement	Description
Domain Load Time	Monitors the time it takes to load a domain. (Domains protect areas of code. All applications run in a domain which keeps them encapsulated, so that they cannot interfere with other applications outside the domain).
Domain Unload Time	Monitors the time it takes to unload a domain. (Domains protect areas of code. All applications run in a domain which keeps them encapsulated, so that they cannot interfere with other applications outside the domain).
Domain Lifetime	Monitors the duration of a domain. (Domains protect areas of code. All applications run in a domain which keeps them encapsulated, so that they cannot interfere with other applications outside the domain).
Module Creation Frequency	Monitors the number of modules that get created in a polling period. (Modules are groups of assemblies that make up a DLL or EXE).
Module Load Time	Monitors the time it takes to load a module. (Modules are groups of assemblies that make up a dll or exe).
Module Unload Time	Monitors the time it takes to unload a module. (Modules are groups of assemblies that make up a dll or exe).
Module Lifetime	Monitors the duration of a module. (Modules are groups of assemblies that make up a dll or exe).
Garbage Collection Duration	Monitors the duration between the start and stop of Garbage Collection.
Garbage Collection Frequency	Monitors the number of breaks for Garbage Collections in a polling period.
Unmanaged Code Duration	Monitors the duration of the calls to unmanaged code.
Unmanaged Code Frequency	Monitors the number of calls to unengaged code in a polling period.

Assembly Level

Measurement	Description
Assembly Creation Frequency	Monitors the number of assembly creations in a polling period. (Assemblies hold the .NET byte code and metadata).
Assembly Load Time	Monitors the time it takes to load an assembly. (Assemblies hold the .NET byte code and metadata).
Assembly Unload Time	Monitors the time it takes to unload an assembly. (Assemblies hold the .NET byte code and metadata).
Assembly Lifetime	Monitors the duration of an assembly. (Assemblies hold the .NET byte code and metadata).

Class Level

Measurement	Description
Class Lifetime	Monitors the duration of a class.
Class Load Time	Monitors the time it takes to load a class.
Class Unload Time	Monitors the time it takes to unload a class.

Method Level

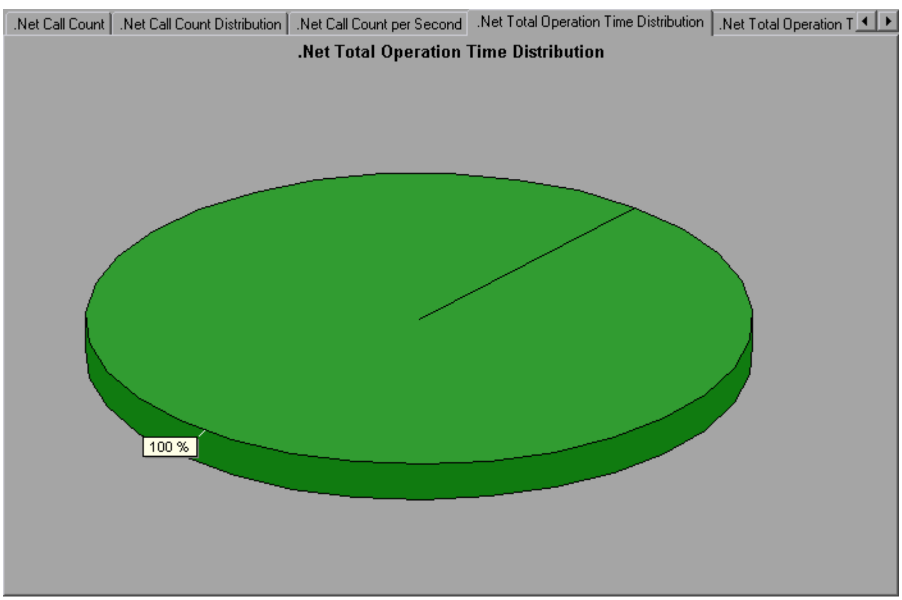
At the method level, the measured time is per method, exclusive of other methods, calls to unmanaged code, and garbage collection time.

Measurement	Description
Method Duration	Monitors the duration of a method.
Method Frequency	Monitors the number of methods called in a polling period.

.NET Total Operation Time Distribution Graph

This graph shows the percentage of time that a specific .NET class took to execute in relation to all the .NET classes. It can also show the percentage of time that a .NET method took to execute in relation to all the .NET methods within the class.

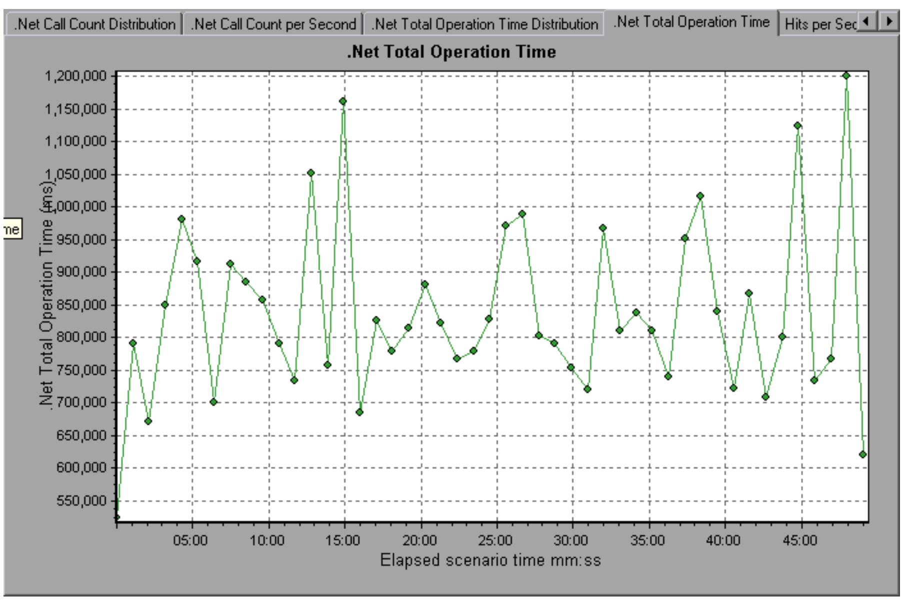
Purpose	Use this graph to identify those classes or methods that take an excessive amount of time.
Breakdown options	The graph initially displays .NET classes, but you can also view the individual methods within a .NET class by using drill-down or filtering techniques. For more information, see "Filtering and Sorting Graph Data" on page 93 and "Drilling Down in a Graph" on page 119.
Tips	To highlight a specific class line in the graph, select the class row in the legend (displayed below the graph).
See also	"Microsoft COM+ Performance Graphs Overview" on page 410



.NET Total Operation Time Graph

This graph displays the amount of time that each .NET class or method took to execute during the test.

Purpose	Use this graph to identify those classes or methods that take an excessive amount of time.
X-axis	Elapsed time from the beginning of the scenario run.
Y-axis	Total time a .NET class or method is in operation.
Breakdown options	The graph initially displays .NET classes, but you can also view the individual methods within a .NET class by using drill-down or filtering techniques. For more information, see "Filtering and Sorting Graph Data" on page 93 and "Drilling Down in a Graph" on page 119.
Tips	To highlight a specific class line in the graph, select the class row in the legend (displayed below the graph).
See also	"Microsoft COM+ Performance Graphs Overview" on page 410



26

Application Deployment Solutions Graphs

This chapter includes:

Concepts

Reference

Concepts

Application Deployment Solutions Graph Overview

LoadRunner's Citrix MetaFrame XP monitor provides you with information about the application deployment usage of the Citrix MetaFrame XP server during a load test scenario execution. In order to obtain performance data, before you execute the scenario you need to activate the online monitor for the server and specify which resources you want to measure.

For more information on activating and configuring the Application Deployment Solutions monitor, refer to the *HP LoadRunner Controller User Guide*.

Reference

Citrix Measurements

Non-Virtual Counters

Measurement	Description
% Disk Time	The percentage of elapsed time that the selected disk drive services read or write requests.
% Processor Time	The percentage of time that the processor executes a non-Idle thread. This counter is a primary indicator of processor activity. It is calculated by measuring the time that the processor spends executing the thread of the Idle process in each sample interval, and subtracting that value from 100%. (Each processor has an Idle thread which consumes cycles when no other threads are ready to run.) It can be viewed as the percentage of the sample interval spent doing useful work. This counter displays the average percentage of busy time observed during the sample interval. It is calculated by monitoring the time the service was inactive, and then subtracting that value from 100%.
File data Operations/sec	The rate that the computer issues Read and Write operations to file system devices. It does not include File Control Operations.

Measurement	Description
Interrupts/sec	The average number of hardware interrupts the processor receives and services per second. It does not include DPCs, which are counted separately. This value is an indirect indicator of the activity of devices that generate interrupts, such as the system clock, the mouse, disk drivers, data communication lines, network interface cards and other peripheral devices. These devices normally interrupt the processor when they have completed a task or require attention. Normal thread execution is suspended during interrupts. Most system clocks interrupt the processor every 10 milliseconds, creating a background of interrupt activity. This counter displays the difference between the values observed in the last two samples, divided by the duration of the sample interval.
Output Session Line Speed	This value represents the line speed from server to client for a session in bps.
Input Session Line Speed	This value represents the line speed from client to server for a session in bps.
Page Faults/sec	A count of the Page Faults in the processor. A page fault occurs when a process refers to a virtual memory page that is not in its Working Set in main memory. A Page Fault will not cause the page to be fetched from disk if that page is on the standby list, and hence already in main memory, or if it is in use by another process with whom the page is shared.

Measurement	Description
Pages/sec	The number of pages read from the disk or written to the disk to resolve memory references to pages that were not in memory at the time of the reference. This is the sum of Pages Input/sec and Pages Output/sec. This counter includes paging traffic on behalf of the system Cache to access file data for applications. This value also includes the pages to/from non-cached mapped memory files. This is the primary counter to observe if you are concerned about excessive memory pressure (that is, thrashing), and the excessive paging that may result.
Pool Nonpaged Bytes	The number of bytes in the Nonpaged Pool, a system memory area where space is acquired by operating system components as they accomplish their appointed tasks. Nonpaged Pool pages cannot be paged out to the paging file, but instead remain in main memory as long as they are allocated.
Private Bytes	The current number of bytes this process has allocated that cannot be shared with other processes.
Processor Queue Length	The instantaneous length of the processor queue in units of threads. This counter is always 0 unless you are also monitoring a thread counter. All processors use a single queue in which threads wait for processor cycles. This length does not include the threads that are currently executing. A sustained processor queue length greater than two generally indicates processor congestion. This is an instantaneous count, not an average over the time interval.
Threads	The number of threads in the computer at the time of data collection. Notice that this is an instantaneous count, not an average over the time interval. A thread is the basic executable entity that can execute instructions in a processor.
Latency – Session Average	The average client latency over the life of a session.

Measurement	Description
Latency – Last Recorded	The last recorded latency measurement for this session.
Latency – Session Deviation	The difference between the minimum and maximum measured values for a session.
Input Session Bandwidth	The bandwidth (in bps) from client to server traffic for a session in bps.
Input Session Compression	The compression ratio for client to server traffic for a session.
Output Session Bandwidth	The bandwidth (in bps) from server to client traffic for a session.
Output Session Compression	The compression ratio for server to client traffic for a session.
Output Session Linespeed	The line speed (in bps) from server to client for a session.

Virtual Channel Counters

All the counters in the following table are measured in bytes per second (bps):

Measurement	Description
Input Audio Bandwidth	The bandwidth from client to server traffic on the audio mapping channel.
Input Clipboard Bandwidth	The bandwidth from client to server traffic on the clipboard mapping channel.
Input COM1 Bandwidth	The bandwidth from client to server traffic on the COM1 channel.
Input COM2 Bandwidth	The bandwidth from client to server traffic on the COM2 channel.
Input COM Bandwidth	The bandwidth from client to server traffic on the COM channel.

Measurement	Description
Input Control Channel Bandwidth	The bandwidth from client to server traffic on the ICA control channel.
Input Drive Bandwidth	The bandwidth from client to server traffic on the client drive mapping channel.
Input Font Data Bandwidth	The bandwidth from client to server traffic on the local text echo font and keyboard layout channel.
Input Licensing Bandwidth	The bandwidth from server to client traffic on the licensing channel.
Input LPT1 Bandwidth	The bandwidth from client to server traffic on the LPT1 channel.
Input LPT2 Bandwidth	The bandwidth from client to server traffic on the LPT2 channel.
Input Management Bandwidth	The bandwidth from client to server traffic on the client management channel.
Input PN Bandwidth	The bandwidth from client to server traffic on the Program Neighborhood channel.
Input Printer Bandwidth	The bandwidth from client to server traffic on the printer spooler channel.
Input Seamless Bandwidth	The bandwidth from client to server traffic on the Seamless channel.
Input Text Echo Bandwidth	The bandwidth from client to server traffic on the local text echo data channel.
Input Thinwire Bandwidth	The bandwidth from client to server traffic on the Thinwire (graphics) channel.
Input VideoFrame Bandwidth	The bandwidth from client to server traffic on the VideoFrame channel.
Output Audio Bandwidth	The bandwidth from server to client traffic on the audio mapping channel.
Output Clipboard Bandwidth	The bandwidth from server to client traffic on the clipboard mapping channel.

Measurement	Description
Output COM1 Bandwidth	The bandwidth from server to client traffic on the COM1 channel.
Output COM2 Bandwidth	The bandwidth from server to client traffic on the COM2 channel.
Output COM Bandwidth	The bandwidth from server to client traffic on the COM channel.
Output Control Channel Bandwidth	The bandwidth from server to client traffic on the ICA control channel.
Output Drive Bandwidth	The bandwidth from server to client traffic on the client drive channel.
Output Font Data Bandwidth	The bandwidth from server to client traffic on the local text echo font and keyboard layout channel.
Output Licensing Bandwidth	The bandwidth from server to client traffic on the licensing channel.
Output LPT1 Bandwidth	The bandwidth from server to client traffic on the LPT1 channel.
Output LPT2 Bandwidth	The bandwidth from server to client traffic on the LPT2 channel.
Output Management Bandwidth	The bandwidth from server to client traffic on the client management channel.
Output PN Bandwidth	The bandwidth from server to client traffic on the Program Neighborhood channel.
Output Printer Bandwidth	The bandwidth from server to client traffic on the printer spooler channel.
Output Seamless Bandwidth	The bandwidth from server to client traffic on the Seamless channel.
Output Text Echo Bandwidth	The bandwidth from server to client traffic on the local text echo data channel.

Measurement	Description
Output Thinwire Bandwidth	The bandwidth from server to client traffic on the Thinwire (graphics) channel.
Output VideoFrame Bandwidth	The bandwidth from server to client traffic on the VideoFrame channel.

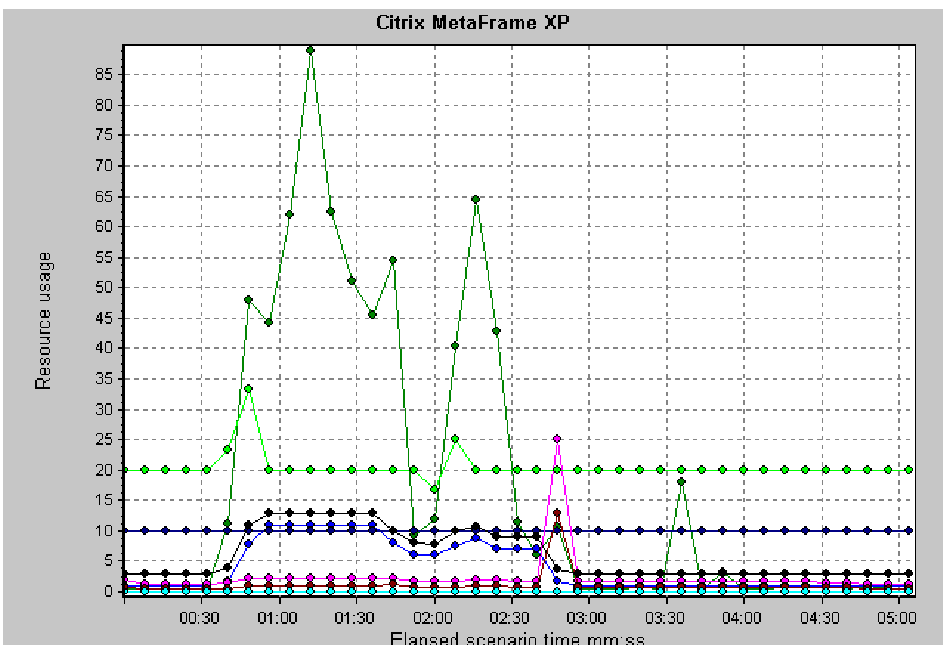
Application Deployment Solutions Graphs User Interface

This section includes:

Citrix MetaFrame XP Graph

This graph is an Application Deployment solution which delivers applications across networks. The Citrix MetaFrame Resource Monitor is an Application Deployment Solution monitor, which provides performance information for the Citrix MetaFrame server.

X-axis	Elapsed time since the start of the run.
Y-axis	The resource usage on the Citrix MetaFrame server.
Note	To obtain data for this graph, you need to enable the Citrix MetaFrame XP monitor (from the Controller) and select the default measurements you want to display, before running the scenario.
See also	"Application Deployment Solutions Graph Overview" on page 446 "Citrix Measurements" on page 447



27

Middleware Performance Graphs

This chapter includes:

Concepts

Reference

Concepts

Middleware Performance Graphs Overview

A primary factor in a transaction's response time is the middleware performance usage. LoadRunner's Middleware Performance monitors provide you with information about the middleware performance usage of the Tuxedo and IBM WebSphere MQ servers during a load test scenario execution. To obtain performance data, you need to activate the online monitor for the server and specify which resources you want to measure before executing the scenario.

For more information on activating and configuring the Middleware Performance monitors, refer to the *HP LoadRunner Controller User Guide*.

Reference

IBM WebSphere MQ Counters

Queue Performance Counters

Measurement	Description
Event - Queue Depth High (events per second)	An event triggered when the queue depth reaches the configured maximum depth.
Event - Queue Depth Low (events per second)	An event triggered when the queue depth reaches the configured minimum depth.
Event - Queue Full (events per second)	An event triggered when an attempt is made to put a message on a queue that is full.
Event - Queue Service Interval High (events per second)	An event triggered when no messages are put to or retrieved from a queue within the timeout threshold.
Event - Queue Service Interval OK (events per second)	An event triggered when a message has been put to or retrieved from a queue within the timeout threshold.
Status - Current Depth	The current count of messages on a local queue. This measurement applies only to local queues of the monitored queue manager.
Status - Open Input Count	The current count of open input handles. Input handles are opened so that an application may "put" messages to a queue.
Status - Open Output Count	The current count of open output handles. Output handles are opened so that an application may "get" messages from a queue.

Channel Performance Counters

Measurement	Description
Event - Channel Activated (events per second)	An event generated when a channel, waiting to become active but inhibited from doing so due to a shortage of queue manager channel slots, becomes active due to the sudden availability of a channel slot.
Event - Channel Not Activated (events per second)	An event generated when a channel attempts to become active but is inhibited from doing so due to a shortage of queue manager channel slots.
Event - Channel Started (events per second)	An event generated when a channel is started.
Event - Channel Stopped (events per second)	An event generated when a channel is stopped, regardless of source of stoppage.
Event - Channel Stopped by User (events per second)	An event generated when a channel is stopped by a user.
Status - Channel State	The current state of a channel. Channels pass through several states from STOPPED (inactive state) to RUNNING (fully active state). Channel states range from 0 (STOPPED) to 6 (RUNNING).
Status - Messages Transferred	The count of messages that have been sent over the channel. If no traffic is occurring over the channel, this measurement will be zero. If the channel has not been started since the queue manager was started, no measurement will be available.
Status - Buffer Received	The count of buffers that have been received over the channel. If no traffic is occurring over the channel, this measurement will be zero. If the channel has not been started since the queue manager was started, no measurement will be available.

Measurement	Description
Status - Buffer Sent	The count of buffers that have been sent over the channel. If no traffic is occurring over the channel, this measurement will be zero. If the channel has not been started since the queue manager was started, no measurement will be available.
Status - Bytes Received	The count of bytes that have been received over the channel. If no traffic is occurring over the channel, this measurement will appear as zero. If the channel has not been started since the queue manager was started, no measurement will be available.
Status - Bytes Sent	The count of bytes that have been sent over the channel. If no traffic is occurring over the channel, this measurement will appear as zero. If the channel has not been started since the queue manager was started, no measurement will be available.

Tuxedo Resources Graph Measurements

The following table describes the default counters that can be measured. It is recommended to pay particular attention to the following measurements: % Busy Clients, Active Clients, Busy Clients, Idle Clients, and all the queue counters for relevant queues.

Monitor	Measurements
Machine	**% Busy Clients.** The percentage of active clients currently logged in to the Tuxedo application server that are waiting for a response from the application server.
	Active Clients. The total number of active clients currently logged in to the Tuxedo application server.
	Busy Clients. The total number of active clients currently logged in to the Tuxedo application server that are waiting for a response from the application server.
	Current Accessers. The number of clients and servers currently accessing the application either directly on this machine or through a workstation handler on this machine.
	Current Transactions. The number of in use transaction table entries on this machine.
	Idle Clients. The total number of active clients currently logged in to the Tuxedo application server that are not waiting for a response from the application server.
	Workload Completed/second. The total workload on all the servers for the machine that was completed, per unit time.
	Workload Initiated/second. The total workload on all the servers for the machine that was initiated, per unit time.

Monitor	Measurements
Queue	**% Busy Servers.** The percentage of active servers currently handling Tuxedo requests.
	Active Servers. The total number of active servers either handling or waiting to handle Tuxedo requests.
	Busy Servers. The total number of active servers currently busy handling Tuxedo requests.
	Idle Servers. The total number of active servers currently waiting to handle Tuxedo requests.
	Number Queued. The total number of messages which have been placed on the queue.
Server	**Requests/second.** The number of server requests handled per second.
	Workload/second. The workload is a weighted measure of the server requests. Some requests could have a different weight than others. By default, the workload is always 50 times the number of requests.
Workstation Handler (WSH)	**Bytes Received/sec.** The total number of bytes received by the workstation handler, per second.
	Bytes Sent/sec. The total number of bytes sent back to the clients by the workstation handler, per second.
	Messages Received/sec. The number of messages received by the workstation handler, per second.
	Messages Sent/sec. The number of messages sent back to the clients by the workstation handler, per second.
	Number of Queue Blocks/sec. The number of times the queue for the workstation handler blocked, per second. This gives an idea of how often the workstation handler was overloaded.

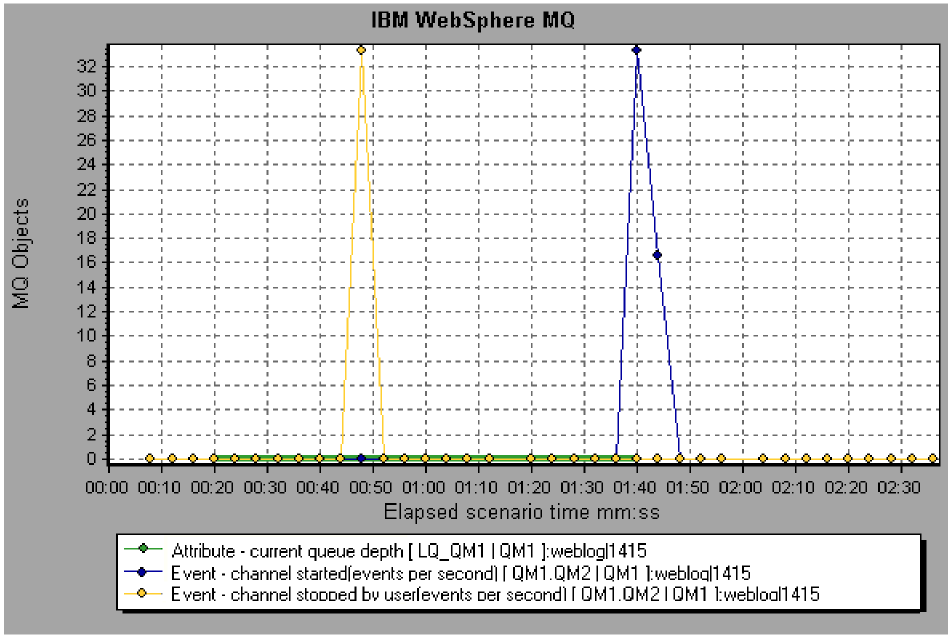
Middleware Performance Graphs User Interface

This section includes (in alphabetical order):

IBM WebSphere MQ Graph Graph

This graph shows the resource usage of IBM WebSphere MQ Server channel and queue performance counters as a function of the elapsed load test scenario time.

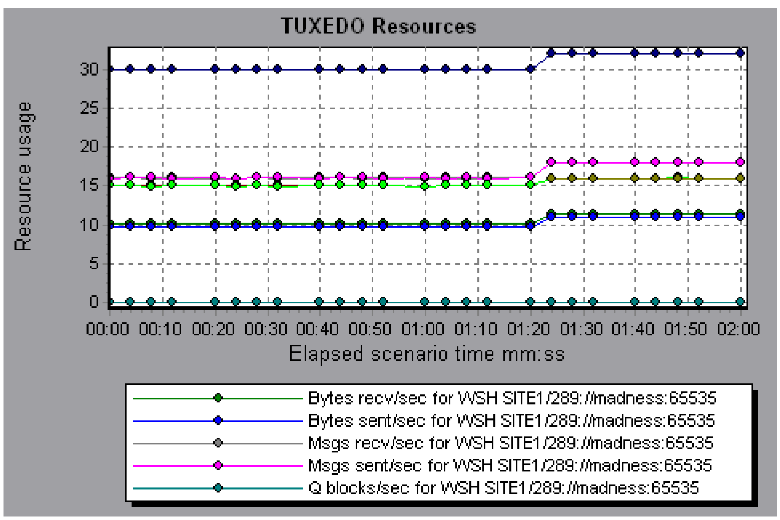
X-axis	Elapsed time since the start of the run.
Y-axis	The resource usage of the IBM WebSphere MQ Server channel and queue performance counters.
Note	To obtain data for this graph, you need to enable the IBM WebSphere MQ monitor (from the Controller) and select the default measurements you want to display, before running the scenario.
See also	"Middleware Performance Graphs Overview" on page 456 "IBM WebSphere MQ Counters" on page 457


IBM WebSphere MQ
MQ Objects
0
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
32
00:00 00:10 00:20 00:30 00:40 00:50 01:00 01:10 01:20 01:30 01:40 01:50 02:00 02:10 02:20 02:30
Elapsed scenario time mm:ss
Attribute - current queue depth [LQ_QM1 | QM1]:weblog|1415
Event - channel started(events per second) [QM1.QM2 | QM1]:weblog|1415
Event - channel stopped by user(events per second) [QM1.QM2 | QM1]:weblog|1415

Tuxedo Resources Graph

This graph provides information about the server, load generator machine, workstation handler, and queue in a Tuxedo system.

X-axis	Elapsed time since the start of the run.
Y-axis	The resource usage on the Tuxedo system.
Note	To obtain data for this graph, you need to enable the TUXEDO monitor (from the Controller) and select the default measurements you want to display, before running the scenario.
See also	"Middleware Performance Graphs Overview" on page 456 "Tuxedo Resources Graph Measurements" on page 460



28

Infrastructure Resources Graphs

This chapter includes:

Concepts

Reference

Concepts

Infrastructure Resources Graphs Overview

LoadRunner's Infrastructure Resources monitor provides you with information about the performance of FTP, POP3, SMTP, IMAP, and DNS Vusers on the network client during load test scenario execution.

Reference

Network Client Measurements

Measurement	Description
Pings per sec	Number of pings per second.
Data transfer bytes per sec	Number of data bytes transferred per second.
Data receive bytes per sec	Number of data bytes received per second.
Connections per sec	Number of connections per second.
Accept connections per sec	Number of connections accepted per seconds.
SSL Connections per sec	Number of SSL connections per second.
SSL Data transfer bytes per sec	Number of SSL data bytes transferred per second.
SSL Data receive bytes per sec	Number of SSL data bytes received per second.
SSL Accept connections per sec	Number of SSL connections accepted per seconds.

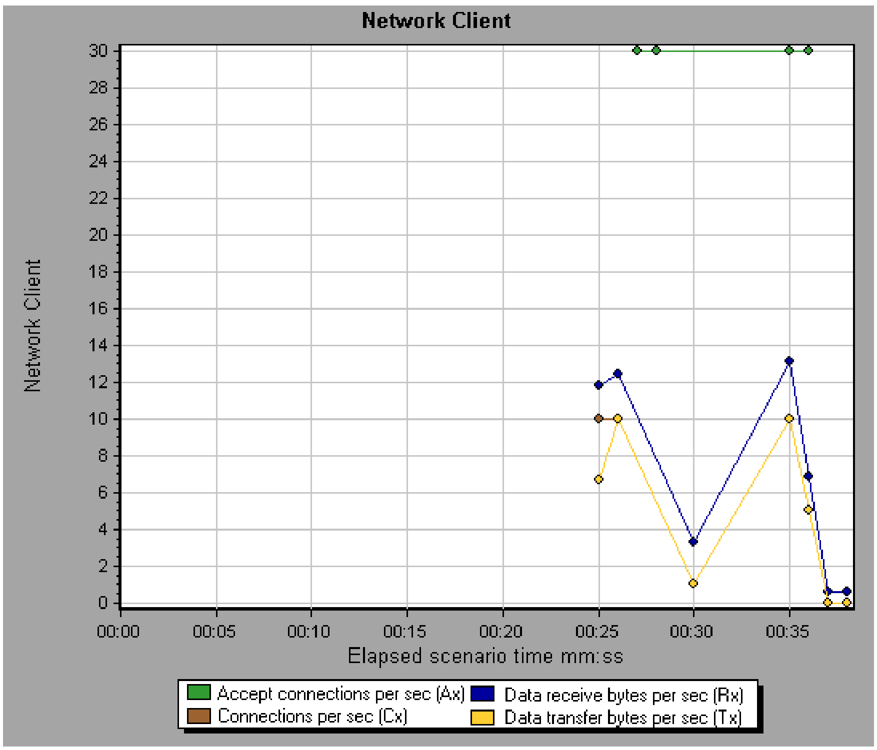
Infrastructure Resources Graphs User Interface

This section includes:

Network Client Graph

This graph displays network client data points for FTP, POP3, SMTP, IMAP, and DNS Vusers during a load test scenario run.

X-axis	Elapsed time since the start of the run.
Y-axis	The resource value of the network client data points..
See also	"Infrastructure Resources Graphs Overview" on page 468



Part III

Analysis Reports

29

Understanding Analysis Reports

This chapter includes:

Concepts

Reference

Concepts

Analysis Reports Overview

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

- Summary Report
- SLA Reports
- Transaction Analysis Report
- HTML report
- Transaction reports

The Summary report provides general information about the scenario run. You can access the Summary report at any time from the Session Explorer.

The SLA report provides an overview of the defined SLAs (Service Level Agreements) with succeeded or failed status.

The Transaction Analysis report provides a detailed analysis of a specific transaction over a specific time period.

You can instruct Analysis to create an HTML report. The HTML report contains a page for each open graph, the Summary report, the SLA report, and the Transaction Analysis report.

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.

Note: SLA reports and Transaction Analysis reports are not available when generating Cross Result graphs. For more information on Cross Result graphs, see "Cross Result and Merged Graphs" on page 153.

Report Templates Overview

You can use Report Templates to create and customize templates which are used when generating reports. Report templates can be used across similar scenario runs and saves time and effort on recreating reports each time.

Using the Report Templates dialog box, you can record document details, define the format of the report, and select the content items to include in the report and configure each content item accordingly.

A list of report templates is displayed in the **Templates** dialog box, under **Rich Reports**. Select this option if you want to generate the report in the load run session in word, excel, HTML or PDF format. For more information on templates, see "Template Dialog Box" on page 90.

Reference

Reports User Interface

The Summary report provides general information about load test scenario execution. This report is always available from the Session Explorer or as a tab in the Analysis window.

The Summary report lists statistics about the scenario run and provides links to the following graphs: Running Vusers, Throughput, Hits Per Second, HTTP Responses per Second, Transaction Summary, and Average Transaction Response Time.

The appearance of the Summary report and the information displayed, will vary depending on whether an SLA (Service Level Agreement) was defined.

An SLA defines goals for the scenario. LoadRunner measures these goals during the scenario run and analyzes them in the Summary report. For more information on defining an SLA, see "SLA Reports" on page 494

Summary report is also provided for Cross Result graphs. For more information about Cross Result graphs, see "Cross Result Graphs Overview" on page 154.

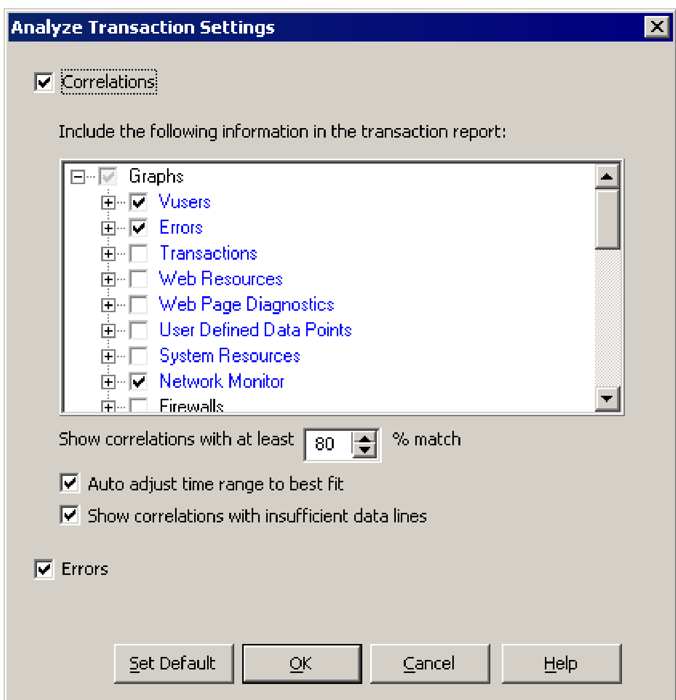
Note: You can save the Summary reports to an Excel file by selecting **View** > **Export Summary to Excel** or by clicking the **Export Summary to Excel** button in the toolbar.

This section includes (in alphabetical order):

- Report <Templates> Window on page 486
- Report Templates General Tab on page 487
- Report Templates Format Tab on page 489
- Report Templates Content Tab on page 491
- SLA Reports on page 494
- Summary Report on page 495
- Transaction Analysis Report on page 503

Analyze Transaction Settings Dialog Box

This dialog box enables you to configure the Transaction Analysis Report to show correlations between the graph of the analyzed transaction and other graphs that you select.



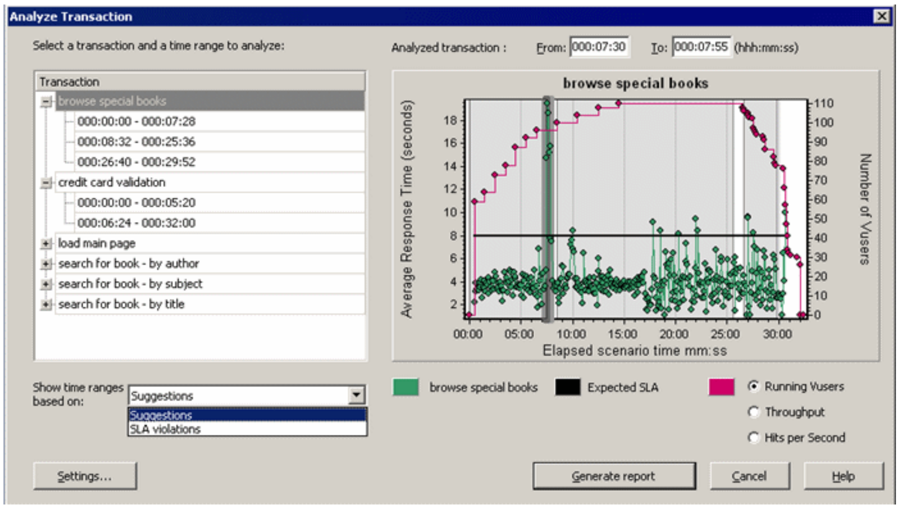
To access	Use one of the following: ➤ **Reports** > **Analyze Transaction** > **Settings** ➤ **Tools** > **Options** > **Analyze Transaction Settings tab**
See also	"Analyze Transactions Dialog Box" on page 479

User interface elements are described below:

UI Elements	Description
Correlations	Defines which graphs you want Analysis to match to the graph of the transaction you selected. Graphs where data is available appear in blue.
Show correlations with at least x% match	The positive or negative percentage correlation between the graph of the analyzed transaction and the graphs selected above. You can change the percentage by entering a value in the box. The default is 20%.
Auto adjust time range to best fit	Analysis adjusts the selected time range to focus on the SLA violations within and around that time period. This option only applies when the Transaction Analysis report is generated directly from the Summary report (from the **X Worst transactions** or **Scenario behavior over time** sections).
Show correlations with insufficient data lines	Displays correlations where one of the measurements contains less than 15 units of granularity.
Errors	Displays errors in the Transaction Analysis Report if selected.

Analyze Transactions Dialog Box

You use the Analyze Transaction dialog box to define the criteria that will be used to analyze the selected transaction in the Transaction Analysis Report. You can analyze a transaction even if you have not defined an SLA.



To access	**Reports > Analyze Transaction** **Summary Report > right-click menu > Add New Item > Analyze Transaction** **Toolbar >** **Summary Report with no SLA > Statistics Summary section > Analyze Transaction tool link**
Note	Analysis data (for example, transactions) that has been excluded by the Summary Filter will not be available for analysis in the Transaction Analysis report.
See also	Filtering and Sorting Graph Data on page 93

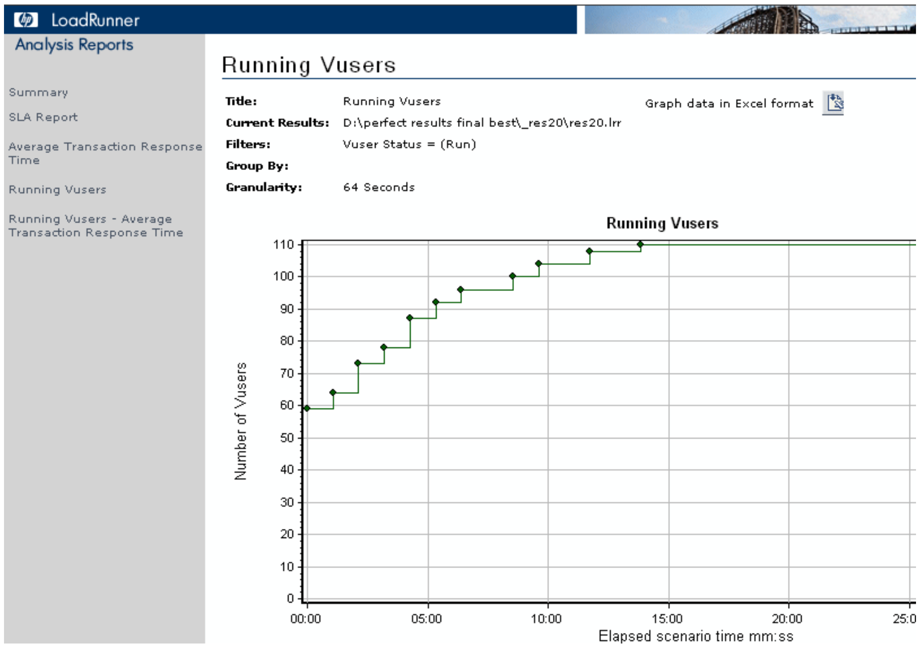
User interface elements are described below:

UI Elements	Description
Show time ranges based on box	Select one of the display options: ➤ **Suggestions.** Lists all transactions and time ranges from the scenario run. ➤ **SLA Violations.** Lists only those transactions and time ranges where the transaction exceeded the SLA. This option does not appear if there were no transactions that exceeded the SLA.
Transaction	Select the transaction to analyze from the Transaction tree.
<**Time Range**>	Select the time range to analyze in one of the following ways: ➤ Select the time range from the Transaction tree. ➤ Enter the time range in the **From** and **To** boxes above the graph. ➤ Select the time range by dragging the bars on the graph.
<**Display options**>	Select one of the following: ➤ **Running Vusers** ➤ **Throughput** ➤ **Hits per Second** The option you select is displayed on the graph and will appear on the snapshot of the graph that appears on the Transaction Analysis Report. Note that your choice only affects the display of the graph and not the calculation for correlations.

UI Elements	Description
Settings	Click **Settings** to define the Analyze Transaction settings in the **Analyze Transaction Settings** dialog box. For more information, see "Analyze Transaction Settings Dialog Box" below. **Note**: You can also define the Analyze Transaction settings in the **Analyze Transaction Settings** tab of the **Options** dialog box (**Tools** > **Options**).
Generate report	The Transaction Analysis Report opens. Once the report has been created, you can access it at any time from the Session Explorer.

HTML Report

Analysis enables you to you to create HTML reports for your load test scenario run. It creates a separate page for each one of the open graphs and reports



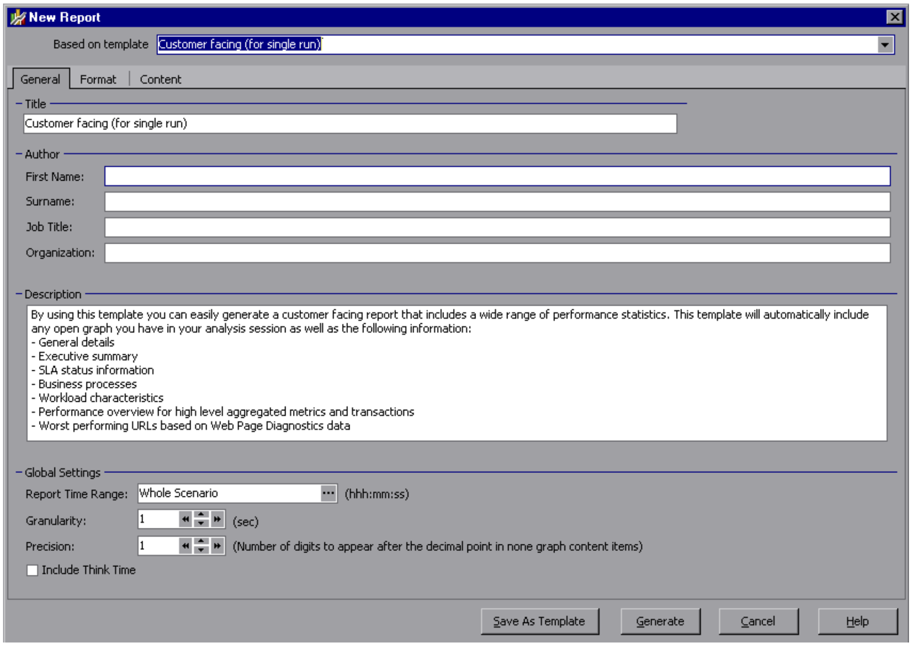
To access	Use one of the following: ➤ **Reports** > **HTML Report** ➤ **Toolbar** > 
Relevant tasks	➤ Open all graphs that you want to include in the report. ➤ Specify a path and file name for the HTML report and click **Save**. Analysis saves a Summary report which has the same name as the file in the selected folder. The rest of the graphs are saved in a folder with the same name as the Summary report's file name. When you create an HTML report, Analysis opens your default browser and displays the Summary report. ➤ 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

User interface elements are described below:

UI Elements	Description
<Graphs> menu left frame	Click the graph link to view an HTML report for that graph.
	You can view an Excel file containing the graph data, by clicking the **Graph data in Excel format** button on the relevant graph page.

New Report Dialog Box

This dialog box enables you to create a report based on the report template selected. You can adjust the report template settings to generate a report that corresponds to the required report layout.



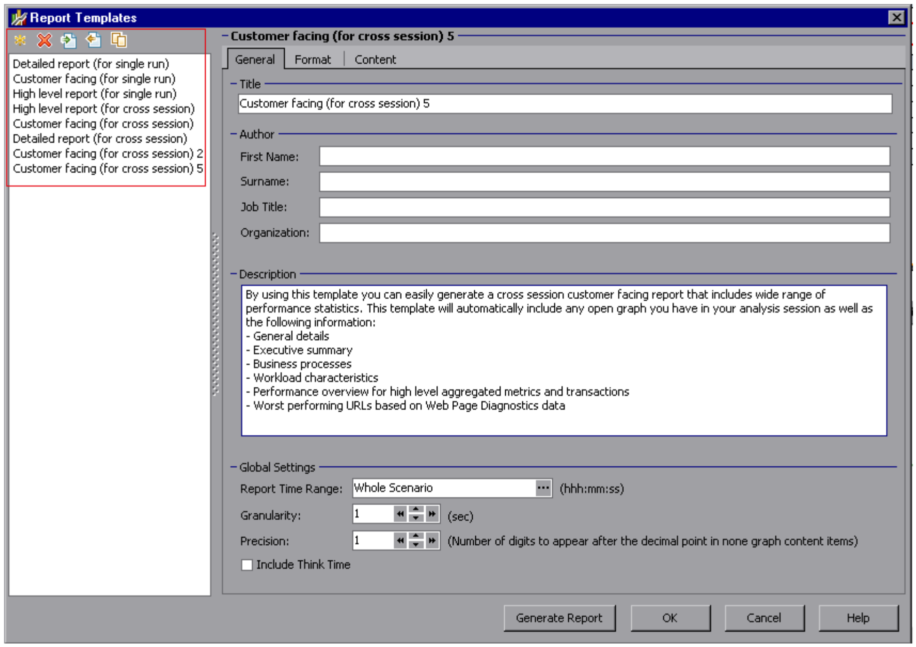
To access	**Reports > New Report**
See also	"Report <Templates> Window" on page 486 "Report Templates General Tab" on page 487 "Report Templates Format Tab" on page 489 "Report Templates Content Tab" on page 491

User interface elements are described below:

UI Elements (A-Z)	Description
Based on Template	Select report template. After a template has been selected, the corresponding settings of the report template appear.
General tab	For user interface details, see "Report Templates General Tab" on page 487.
Format tab	For user interface details, see"Report Templates Format Tab" on page 489.
Content tab	For user interface details, see "Report Templates Content Tab" on page 491.

Report <Templates> Window

This window enables you to add, modify, import, export, or duplicate a report template.



To access	**Reports > Report Templates**
See also	"Report Templates Overview" on page 475

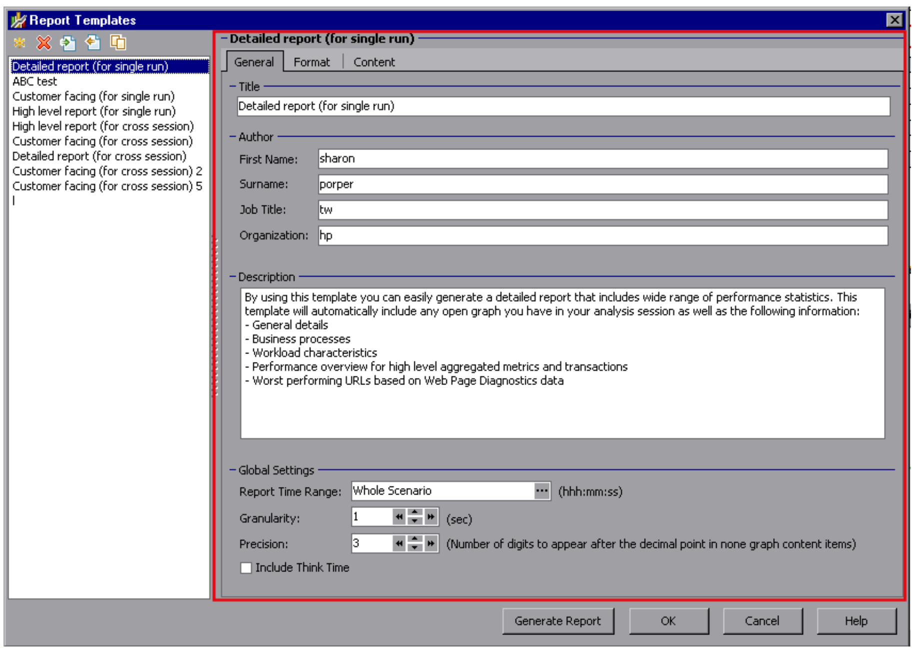
User interface elements are described below:

UI Elements	Description
New	Add a new report template.
Delete	Delete the selected template.
Import	Import report template from selected XML file.

UI Elements	Description
Export	Save the selected template as an XML file to a selected directory.
Duplicate	Create a copy of the selected template.

Report Templates General Tab

This tab enables you to record document details, such as title, author name and title and set global settings, such as Report Time Range and granularity.



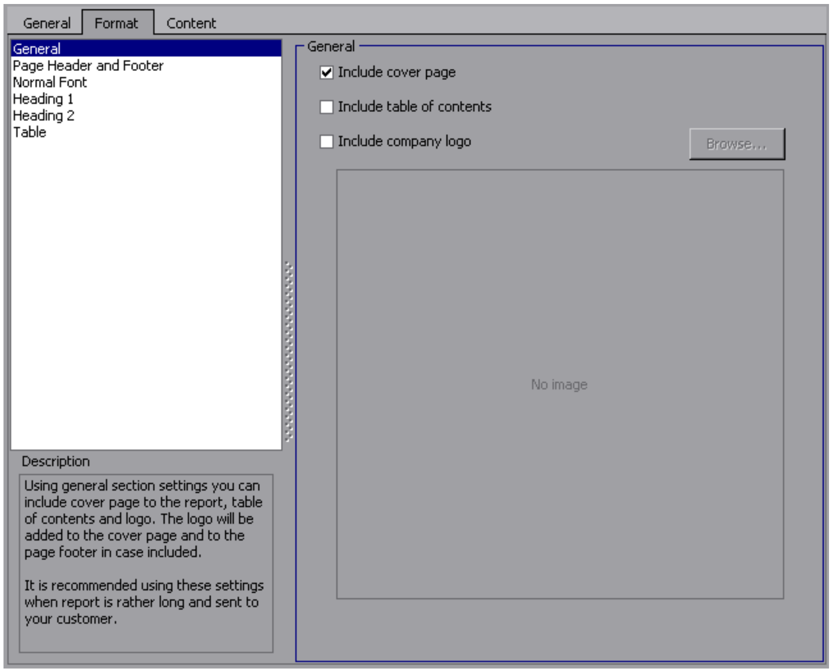
To access	**Reports > Report Templates > General tab**
Important Information	New Report window has the same components as Report Templates.
See also	"Report Templates Overview" on page 475

User interface elements are described below:

UI Elements	Description
Title	A description of the template.
First Name	The first name of the person to display on the report.
Surname	The last name of the person to display on the report.
Job title	The job title of the person to display on the report.
Organization	The name of the organization to display on the report.
Description	You can enter a description and include details of the report template.
Report Time Range	The default setting is Whole Scenario. Click to set the start and end time range of the scenario run time to display on the report.
Granularity	Define granularity settings (in seconds).
Precision	The number of digits to appear after the decimal point in none graph content items.
Include Think Time	Analysis processes data to include think time. This data is then used to generate reports.

Report Templates Format Tab

This tab enables you to define the format of report template.



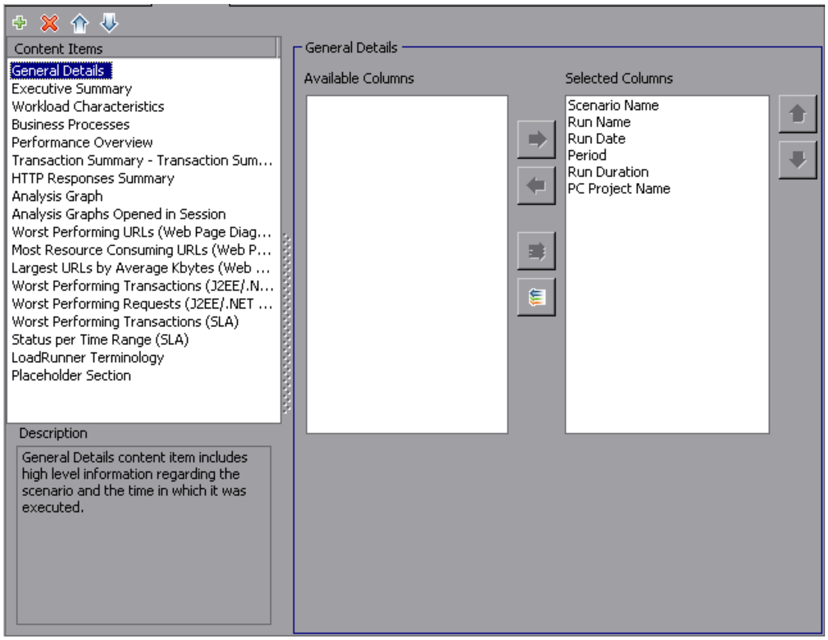
To access	**Reports** > **Report Templates** > **Format tab**
See also	"Report Templates Overview" on page 475

User interface elements are described below:

UI Elements	Description
General	The **General** option enables you to: ➤ include a cover page ➤ include table of contents ➤ include company logo
Page Header and Footer	You can customize the header and footer by using the following options: ➤ Font type, size and color ➤ Bold, italicize, or underline ➤ Right, center or left align ➤ You can add tags, such as date, name or organization. ➤ You can include required details such as page count, date, name, etc. on the left, center or right column.
Normal Font	Select the type of font to use in the report template.
Heading 1/2	Create a style for your headings
Table	Format the table using the following options: ➤ Font type, size and color ➤ Background color ➤ Bold, italicize, or underline ➤ Right, center or left align

Report Templates Content Tab

This tab enables you to select the content items to the report and configure each content item accordingly.



To access	**Reports** > **Report Templates** > **Content tab**
See also	"Report Templates Overview" on page 475

User interface elements are described below:

UI Elements	Description
General Details	Select the session details to display in the report.
Executive Summary	Enter a summary in the open text dialog box.
Workload Characteristics	Select the workload details to display in the report. The workload is determined by the number of transactions, running Vusers load and throughput status on the server.
Business Process	Select script details to display in the report. All the scripts from the scenario are included in the report.
Performance Overview	Select the performance characteristics over the configured elapsed time to display in the report.
Transaction Summary - Transaction Summary	Configure the settings for load test scenario diagnostics data to display. You can set more than one value for the percentile column.
HTTP Responses Summary	Define the time range by which to display HTTP status codes returned from the Web server.
Analysis Graph	Select a graph and configure graph settings to display in the report.
Analysis Graphs Opened in Session	Select the graphs opened in the Analysis session to display in the report.
Worst Performing URLs (Web Page Diagnostics)	Define the number of URLs to display of which take the longest time to load. Based on data from the Web Page Diagnostics graph.
Most Resource Consuming URLs (Web Page Diagnostics)	Define the number of URLs to display of which take up the most computer resources. Based on data from the Web Page Diagnostics graph.
Largest URLs by Average kbytes (Web Page Diagnostics)	Define the number of URLs to display of which are the largest in kbytes and take the most server time to load. Based on data from the Web Page Diagnostics graph.

UI Elements	Description
Worst Performing Transactions (J2EE/ .NET Diagnostics)	Define the number of transactions to display of which the response time takes the longest. Based on data from J2EE/.NET Diagnostics graph.
Worst Performing Requests (J2EE/.NET Diagnostics)	Define the number of transactions to display of requests which take the longest to retrieve data. Based on data from J2EE/.NET Diagnostics graph. **Note:** The request does not have to come from the same transaction.
Worst Performing Transactions (SLA)	Define the number of worst transactions to display which are above the threshold. This data is based on transactions from Average Transaction Response Time (SLA).
Status Per Time Range (SLA)	Shows metrics of SLA status at every time interval. This is relevant to Errors per Second and Average Transaction Response Time (SLA).
Loadrunner Terminology	The terminology content item includes definitions for terms related to LoadRunner objects and graph information.
Placeholder Section	Enter open text dialog.
	Add a content item.
	Delete the selected content item.
	Move up selected content item.
	Move down selected content item.
Parameters	Enter a title, set granularity and percentile used for the selected content item.
Columns	Select the criteria to display in the report for the selected content item.
Include workload scheme graph	A workload scheme graph is included in the report.

UI Elements	Description
Filter	Define filter conditions for selected content item to display in report.
Number of worst elements to show	Shows *x* of the worst elements on the report.

SLA Reports

An SLA (Service Level Agreement) defines goals for the load test scenario. LoadRunner measures these goals during the scenario run and analyzes them in the Summary report. The SLA Report shows the succeeded or failed status of all SLAs that were defined for the scenario run. For more information on defining an SLA, see "SLA Reports" on page 494

Note: Analysis data (for example, transactions) that has been excluded by the Summary Filter will not be available for analysis in the SLA report.

To access	You create the SLA Report in one of the following ways: **Reports** > **Analyze SLA** **Right-click the Summary pane** > **Add New Item** > **Analyze SLA** **Summary Report** >
See also	Defining Service Level Agreements on page 163

User interface elements are described below:

UI Elements	Description
Display of SLA statuses	SLA Status per goal definition ➤ Where the SLA was defined over the whole run, the report displays a single SLA status for each goal definition. SLA status for each transaction per time interval ➤ Where the SLA was defined per time interval within the run, the report displays the status of the SLA for each transaction per time interval. The green squares show time intervals where the transaction performed within the SLA boundary. Red squares where the transaction failed and grey squares show where no relevant SLA was defined. SLA goal definitions ➤ Where the SLA was defined per time interval within the run, a further section appears detailing the goal definitions for the SLA.

Summary Report

The Summary report provides general information about load test scenario execution. It lists statistics about the scenario run and provides links to the following graphs: Running Vusers, Throughput, Hits Per Second, HTTP Responses per Second, Transaction Summary, and Average Transaction Response Time.

To access	**Session Explorer > Reports > Summary Report**
Important information	The Summary report for SAP Diagnostics, J2EE /.NET Diagnostics, and Siebel Diagnostics provides a usage chart that links to and displays each individual transaction's Web, application, and database layers, and provides the total usage time for each transaction.

Relevant tasks	You can save the Summary reports to an Excel file by selecting View > Export Summary to Excel or by clicking  in the toolbar.
See also	The Summary reports for the various diagnostics environments are discussed in detail in the following sections: "SAP Diagnostics Summary Report" on page 589 "J2EE & .NET Diagnostics Graphs Summary Report" on page 624 "Siebel Diagnostics Graphs Summary Report" on page 526

Summary Report with No SLA

User interface elements are described below:

UI Elements	Description
Scenario Details	Shows the basic details of the load test scenario being analyzed.
Statistics Summary	This section shows a breakdown of the transaction statistics and also provides links to the following: ➤ The SLA configuration wizard. For more information on defining an SLA, see "SLA Reports" on page 494 ➤ The Analyze Transaction tool. For more information on analyzing transactions, see "Analyze Transactions Dialog Box" on page 479
Scenario Behavior Over Time	This section displays the average errors per second received by the application under test per time interval. For example, 0 means that on average there were zero errors received per second for that time interval, 0+ means that on average there were slightly more than zero errors received, and so on.

UI Elements	Description
Transaction Summary	This section displays a table containing the load test scenario's diagnostics data. Included in this data is a percentile column (x Percent). This column indicates the maximum response time for that percentage of transactions performed during the run. For example, in the table below, the value in the 88 Percent column for **browse special books** is 8.072. This means that the response time for 88% of the **browse special books** transactions was less than 8.072 seconds. **Note**: You can change the value in the percentile column in one of the following ways: ➤ Open the Options dialog box (**Tools** > **Options**). Click the **General** tab and In the Summary Report section enter the desired percentile in the Transaction Percentile box. ➤ Select View > Summary Filter or click  on the toolbar. The Analysis Summary Filter dialog box opens. In the Additional Settings area enter desired percentile.
HTTP Responses Summary	This section shows the number of HTTP status codes returned from the Web server during the load test scenario, grouped by status code. **Note**: There are additional Diagnostics sections that may appear at the end of the Summary report, depending on the configuration of your system. For more information, see "Summary Report" on page 495.

Summary Report with SLA

User interface elements are described below:

UI Elements	Description
Scenario details	This section shows the basic details of the load test scenario being analyzed.
Statistics Summary	This section shows a breakdown of the transaction statistics.

UI Elements	Description
X Worst Transactions	The X Worst Transactions table shows the worst transactions in terms of how often the transactions exceeded the SLA boundary during the run, and by how much. **Note**: You choose how many transactions are displayed in this table in the Summary Report section on the General tab of the options dialog box. Open the dialog box (**Tools** > **Options**) and enter the number of transactions to display. The default is **5**. You expand a transaction to get more information. When expanded, the following information appears for each transaction: Failure Ratio ➤ The percentage of time intervals where the transaction exceeded the SLA. You can see this graphically in the Scenario Behavior Over Time section below. Failure Value ➤ The average percentage by which the transaction exceeded the SLA over the whole run. Avg exceeding ratio ➤ The average percentage by which the transaction exceeded the SLA over a specific time interval. For example, in the first time interval in the screenshot above, the figure is 4.25%. This means that during that time interval, the transaction may have exceeded the SLA boundary several times, each time by a different percentage margin, the average percentage being 4.25%. Max exceeding ratio ➤ The highest percentage by which the transaction exceeded the SLA over a specific time interval. For example, using the same time interval as above, the transaction may have exceeded the SLA several times, each time by a different percentage margin. The highest percentage being 7.39% Analysis allows you to analyze a specific transaction in more detail. You open the Analyze Transaction tool from this section by clicking the **Analyze Transaction** button. For more information on Transaction Analysis Reports, see "Analyze Transactions Dialog Box" on page 479.

<table>
<tr><th>UI Elements</th><th>Description</th></tr>
<tr><td>Scenario Behavior Over Time</td><td>
This section shows how each transaction performed in terms of the SLA over time intervals. The green squares show time intervals where the transaction performed within the SLA boundary. Red squares show time intervals where the transaction failed and grey squares show where no relevant SLA was defined.
Analysis allows you to analyze a specific transaction in more detail. You open the Analyze Transaction tool from the Scenario Behavior Over Time section in one of the following ways:

Select the transaction to analyze from the list and enter the time interval in the From and To boxes. Then click Analyze Transaction.
Drag the mouse over the desired transaction and time range to analyze. Then click Analyze Transaction.

For more information on Transaction Analysis Reports, see "Analyze Transactions Dialog Box" on page 479.
Note: The time intervals displayed in the Scenario Behavior Over Time section may vary for each interval. The time interval set in the tracking period of the SLA is only the minimum time interval that will be displayed.
It is only the display that varies. The SLA is still determined over the time interval you choose in the Advanced Settings section.
</td></tr>
</table>

UI Elements	Description
Transaction Summary	This section displays a table containing the load test scenario's diagnostics data. Included in this data is a percentile column (x Percent). This column indicates the maximum response time for that percentage of transactions performed during the run. For example, in the table below, the value in the 88 Percent column for **browse special books** is 8.072. This means that the response time for 88% of the **browse special bo. oks** transactions was less that 8.072 seconds. **Note:** You can change the value in the percentile column in the Summary Report section of the General tab of the Options dialog box. Open the dialog box (**Tools** > **Options**) and enter the desired percentage. Alternatively, you can also change the value in the Summary Filter (View > Summary Filter)
HTTP Responses Summary	This section shows the number of HTTP status codes returned from the Web server during the load test scenario, grouped by status code. **Note:** There are additional Diagnostics sections that may appear at the end of the Summary report, depending on the configuration of your system. For more information, see "Summary Report" on page 495.

Summary reports for Cross Result Graphs

User interface elements are described below:

UI Elements	Description
<**graphs**>	Displays summary information for the scenarios that you are comparing. The information is displayed in a way that enables you to compare data from the different scenarios. Includes the same type of information as the regular Summary report except for the following: ➤ SLA information ➤ Diagnostics information ➤ Scenario behavior over time

Transaction Analysis Report

This report enables you to individually examine each of the transactions from the load test scenario run.

To access	**Reports** > **Analyze Transaction** >**Generate report button**

User interface elements are described below:

UI Elements	Description
Observations	This section shows both positive and negative correlations between the graph of the transaction being analyzed, and other graphs based on the settings you chose in the Analyze Transaction Dialog Box. When two graphs are correlated, it means that their behavior matches each other by a certain percentage. To view the correlating graph, select one of the results and then click the View Graph icon at the bottom of the section. The graph comparison opens. You can return to the Transaction Analysis Report from the graph comparison at anytime by clicking the Back to <transaction name> icon on the toolbar. **Note**: The correlations are automatically calculated according to a default ratio of 20%. You can adjust this ratio by clicking the arrows next to the percentage. Then click Recalculate.
Errors	This section is divided into two sub-sections. ➤ **Application Under Test errors.** Shows errors that occurred during the transaction that were direct results of Vuser activity. ➤ **All errors.** Shows Application Under Test errors as well as errors that were not related to Vuser activity, and which affected your system and not the application under test.

UI Elements	Description
Observation Settings	This section displays a summary of the settings that were selected in the Advanced Settings section of the Analyze Transaction dialog box.
Graph	The Graph section displays a snapshot of selected transaction and time range for analysis merged with the display option you selected (Running Vusers, Throughput, or Hits per Second). Note that it is only a snapshot and can not be manipulated like normal graphs.

Part IV

Working with Diagnostics

30

Siebel Diagnostics Graphs

This chapter includes:

Concepts

Tasks

Reference

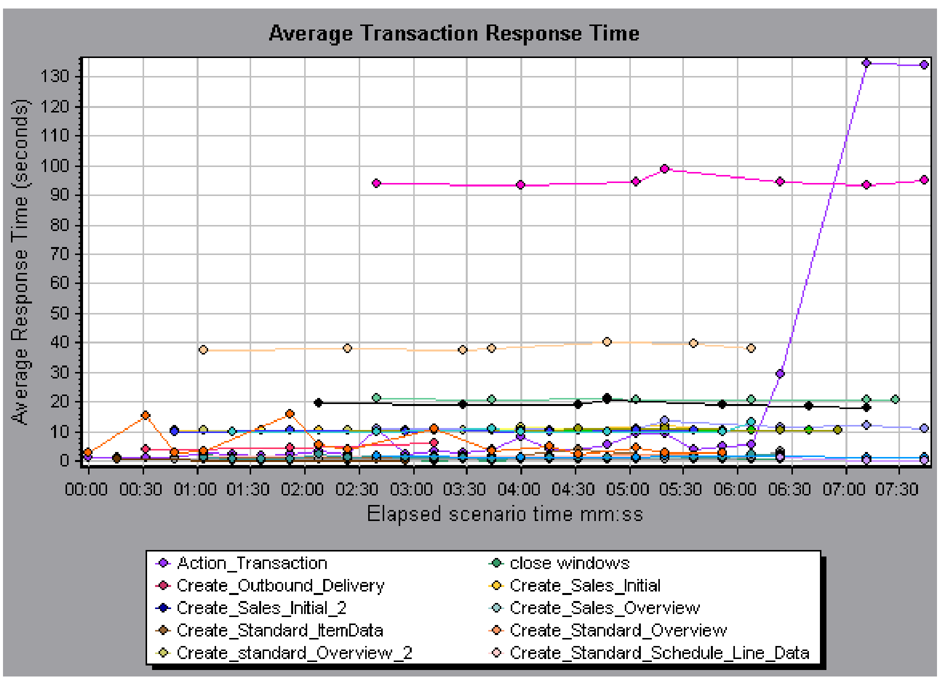
Concepts

Siebel Diagnostics Graphs Overview

Siebel Diagnostics graphs enable you to trace, time, and troubleshoot individual transactions through Web, application, and database servers.

To analyze where problems are occurring, you correlate the data in the Siebel Diagnostics graphs with data in the Transaction Response Time graphs.

You begin analyzing these graphs with the transaction graphs that display the average transaction response time during each second of the load test scenario run. For example, the following Average Transaction Response Time graph demonstrates that the average transaction response time for the **Action_Transaction** transaction was high.



Using the Siebel Diagnostics graphs, you can pinpoint the cause of the delay in response time for this transaction.

Alternatively, you can use the Summary Report to view individual transactions broken down into Web, application, and database layers, and the total usage time for each transaction. For more information, see "Siebel Diagnostics Graphs Summary Report" on page 526.

Note: A measurement that is broken down in the Average Transaction Response Time graph will be different from the same measurement broken down in the Siebel Diagnostics graph. This is because the Average Transaction Response Time graph displays the average transaction response time, whereas the Siebel Diagnostics graph displays the average time per transaction event (sum of Siebel Area response time).

Tasks

How to Enable Siebel Diagnostics

To generate Siebel diagnostics data, you must first install the ERP/CRM Mediator (Mediator). For information on installing the Mediator, refer to the *HP LoadRunner Installation Guide*.

The Mediator is used to gather and correlate offline diagnostics data from the Siebel server. The Mediator processes the diagnostics data, and then passes it to the Controller.

Note: The Mediator must reside in the same LAN as the Siebel server.

To obtain diagnostics data for these graphs, you need to set up the Siebel Diagnostics module before running the scenario, and specify the sampling percentage of diagnostics data to include in the diagnostics graphs. For more information on configuring Siebel Diagnostics, refer to the *HP LoadRunner Controller User Guide*.

- The settings that you configure are per scenario. All scripts in the scenario will run under the same diagnostics configuration.
- To ensure that valid diagnostics data is generated, manually define the transactions in the Vuser script rather than using automatic transactions. Make sure to disable the following options in the Run-Time Settings' **General : Miscellaneous** node: **Define each action as a transaction** and **Define each step as a transaction**.

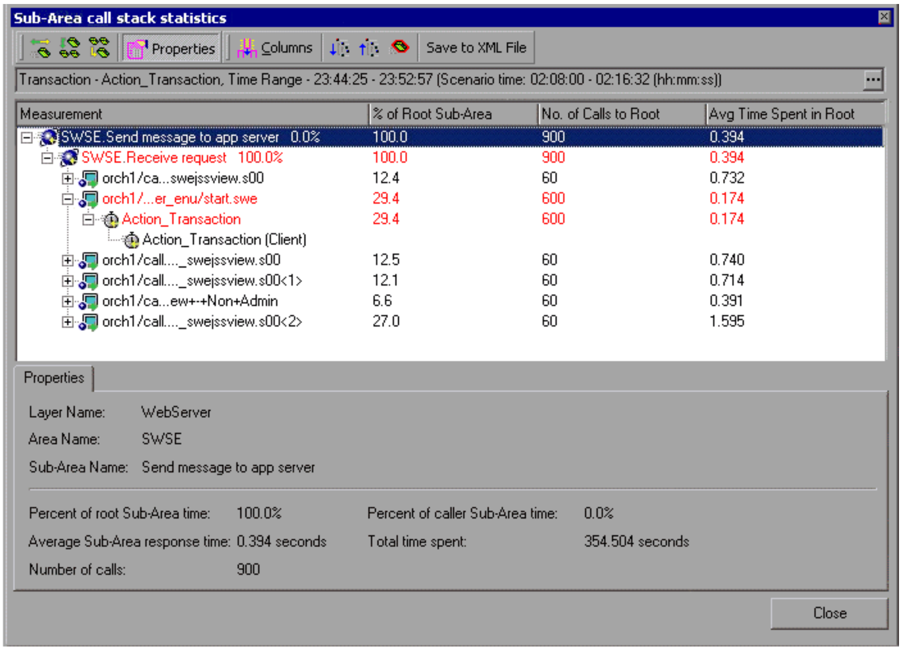
Reference

Siebel Diagnostics User Interface

This section includes (in alphabetical order):

Call Stack Statistics Window

This window enables you to view which components called the selected component.



To access	Analysis window > <Siebel> graph > right click sub-area and select **Siebel Diagnostics** > **Show Sub-Area Call Stack Statistics**
See also	"Siebel Diagnostics Graphs Overview" on page 508

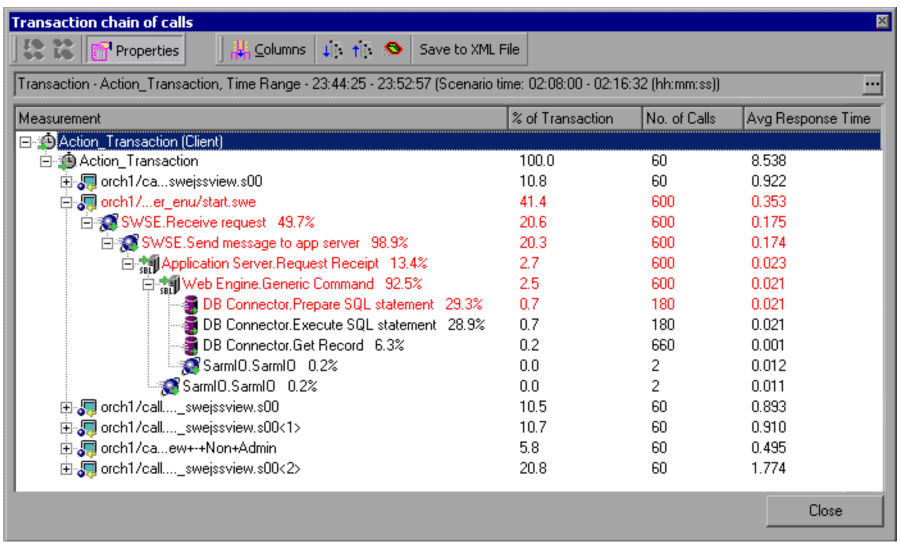
User interface elements are described below:

UI Elements	Description
Measurement	Name of the sub-area, displayed as **AreaName:SubAreaName**. In the case of a database call, query information is also displayed. The percent shown indicates the percentage of calls to this component from its child.
% of Root Sub-Area	Displays the percentage of sub-area time in relation the total root sub-area time.
No. of Calls to Root	Displays the amount of times this transaction or sub-area was executed.
Avg Time Spent in Root	Time spent in root is the time that the sub-area spent in the root sub-area/area/transaction. Average Time Spent in Root time is the total time spent in the root divided by the number of instances of the sub-area.
STD Time Spent in Root	The standard deviation time spent in the root.
Min Time Spent in Root	The minimum time spent in the root.
Max Time Spent in Root	The maximum time spent in the root.
% of Called	Displays the percentage of sub-area time in relation the child sub-area time.
Total Time Spent in Root	Displays the total sub-area execution time, including the child execution time.
	**Expand All.** Expands the entire tree.
	**Collapse All.** Collapses the entire tree.
	**Expand Worst Path.** Expands only the parts of the path on the critical path.
Save to XML File	Saves the tree data to an XML file.

UI Elements	Description
Properties	**Properties** Area. Displays the full properties of the selected sub-area.
SQL Query	**SQL Query.** Displays the SQL query for the selected sub-area. (For Database only.

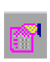
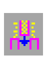
Chain of Calls Window

This window enables you to view the components that the selected transaction or sub-area called. The following figure shows all the calls in the critical path of the parent **Action_Transaction** server-side transaction are displayed.



To access	➤ Use one of the following:To view transaction call chains - right click a component and select **Siebel Diagnostics** > **Show Chain of Calls** ➤ To view sub-area statistics - right click sub-area and select **Show Sub-Area Chain of Calls**
Note	Each red node signifies the most time consuming child to its parent.

User interface elements are described below:

UI Elements	Description
	**Switch to Sub-Area Chain of Calls.** When the sub-area call stack statistics data is displayed, this displays the sub-area chain of calls data (only if the root is a sub-area).
	**Switch to Sub-Area Call Stack Statistics.** When the sub-area chain of calls data is displayed, this displays the sub-area call stack statistics data (only if the root is a sub-area).
	**Show Sub-Area Chain of Calls.** Displays the Sub-Area Chain of Calls window.
	**Show Sub-Area Call Stack Statistics.** Displays the Sub-Area Call Stack Statistics window.
	**Properties.** Hides or displays the properties area (lower pane).
	**Columns.** Enables you to select the columns shown in the Calls window. To display additional fields, drag them to the desired location in the Calls window. To remove fields, drag them from the Calls window back to the Columns chooser.
Measurement	Name of the sub-area, displayed as **AreaName:SubAreaName**. In the case of a database call, query information is also displayed. The percent shown indicates the percentage of calls to this component from its parent.

UI Elements	Description
% of Transaction/ Root Sub-Area	Displays the percentage of sub-area time in relation the total transaction/root sub-area time.
No of Calls	Displays the amount of times this transaction or sub-area was executed.
Avg Response Time	Response time is the time from the beginning of execution until the end. Average response time is the total response time divided by the number of instances of the area/sub-area.
STD Response Time	The standard deviation response time.
Min Response Time	The minimum response time.
Max Response Time	The maximum response time.
% of Caller	Displays the percentage of sub-area time in relation the parent sub-area time.
Total time	Displays the total sub-area execution time, including the child execution time.

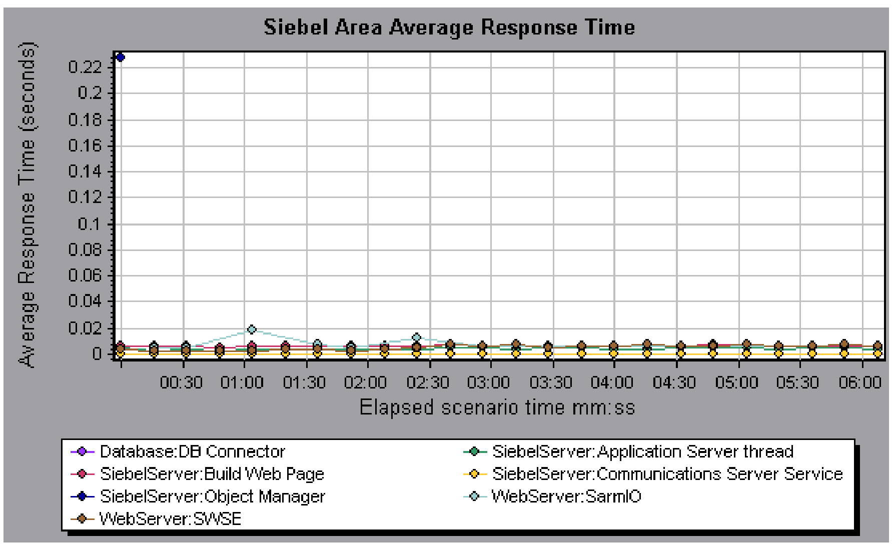
Siebel Area Average Response Time Graph

This graph displays the average response time for the server side areas, computed as the total area response time divided by the number of area calls.

Purpose	For example, if an area was executed twice by one instance of transaction A, and once by another instance of the same transaction, and it took three seconds for each execution, then the average response time is 9/3, or 3 seconds. The area time does not include calls made from the area to other areas.
X-axis	Elapsed time since the start of the run.
Y-axis	Average response time (in seconds) per area.

Breakdown options	For breakdown options, see "Siebel Breakdown Levels" on page 520
Tips	You can filter the Siebel graphs by the following fields: ➤ **Transaction Name.** Shows data for the specified transaction. ➤ **Scenario Elapsed Time.** Shows data for transactions that ended during the specified time. For more information on filtering, see "Filtering and Sorting Graph Data" on page 93.
See also	"Siebel Breakdown Levels" on page 520

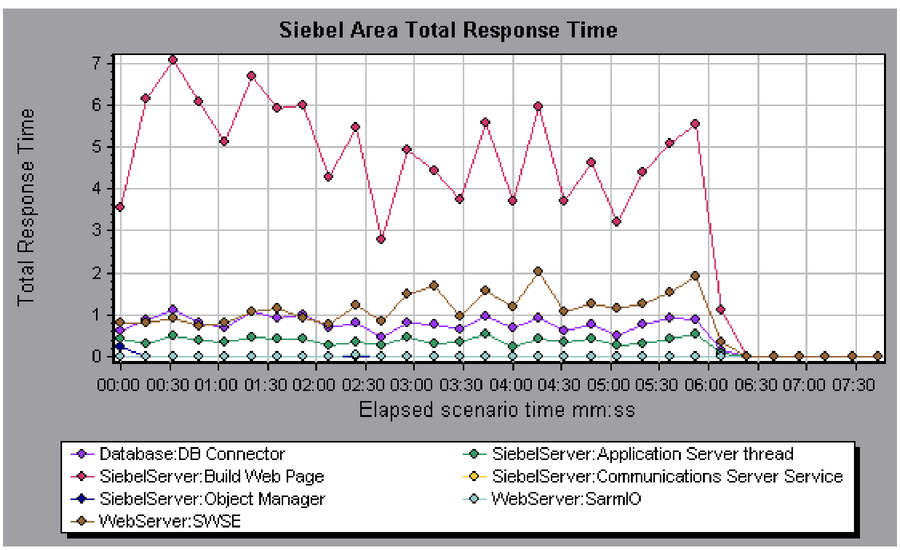
Example



Siebel Area Call Count Graph

This graph displays the number of times that each Siebel area is called.

X-axis	Elapsed time since the start of the run.
Y-axis	The call count.
Breakdown options	For breakdown options, see "Siebel Breakdown Levels" on page 520
Tips	You can filter the Siebel graphs by the following fields: ➤ **Transaction Name.** Shows data for the specified transaction. ➤ **Scenario Elapsed Time.** Shows data for transactions that ended during the specified time. For more information on filtering, see"Filtering and Sorting Graph Data" on page 93
See also	"Siebel Diagnostics Graphs Overview" on page 508

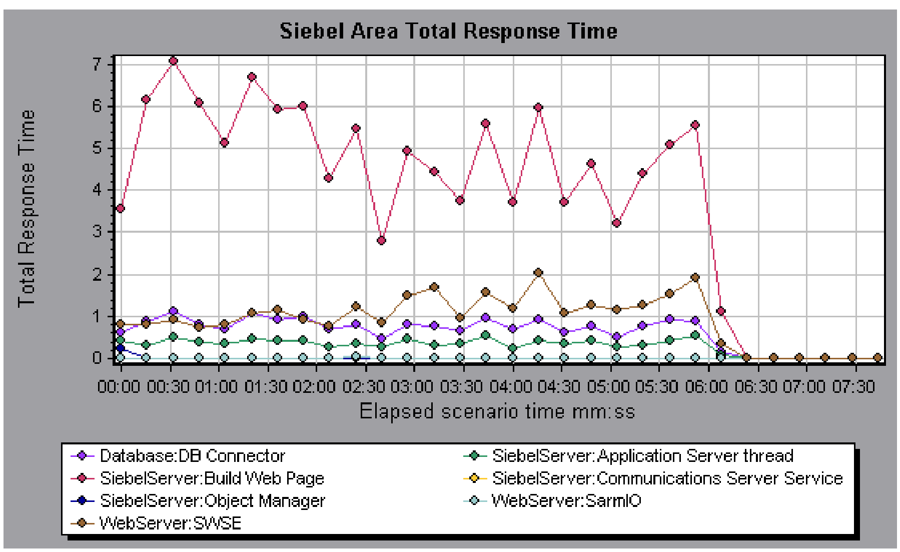


Siebel Area Total Response Time Graph

This graph displays the total response time of each Siebel area.

X-axis	Elapsed time since the start of the run.
Y-axis	Average response time (in seconds) per area.
Breakdown options	For breakdown options, see "Siebel Breakdown Levels" on page 520
Tips	You can filter the Siebel graphs by the following fields: ➤ **Transaction Name.** Shows data for the specified transaction. ➤ **Scenario Elapsed Time.** Shows data for transactions that ended during the specified time. For more information on filtering, see"Filtering and Sorting Graph Data" on page 93
See also	"Siebel Diagnostics Graphs Overview" on page 508

Example

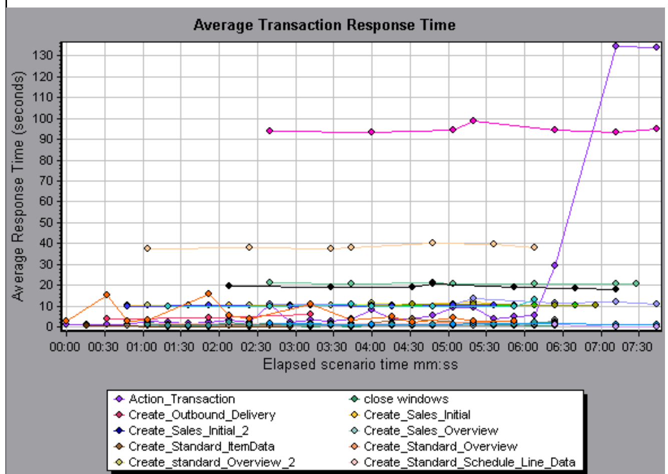


Siebel Breakdown Levels

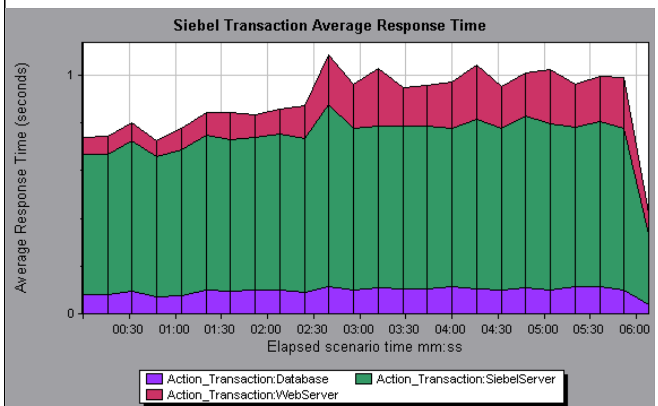
You can break down Siebel layers into areas, sub-areas, servers, and scripts to enable you to pinpoint the exact location where time is consumed.

To access	Use one of the following to access breakdown options: ➤ <**Siebel Diagnostics Graphs**> > **View** > **Siebel Diagnostics** ➤ <**Siebel Diagnostics Graphs**> > select transaction > short-cut menu > **Siebel Diagnostics** ➤ See toolbar options for each breakdown level
Important Information	The breakdown menu options and buttons are not displayed until an element (transaction, layer, area, sub-area) is selected.
See also	"Siebel Diagnostics Graphs Overview" on page 508

Siebel Breakdown Levels are described below:

<table>
<tr><td>Transaction Level</td><td>The following figure displays the top level Average Transaction Response Time graph. The graph displays several transactions.

</td></tr>
</table>

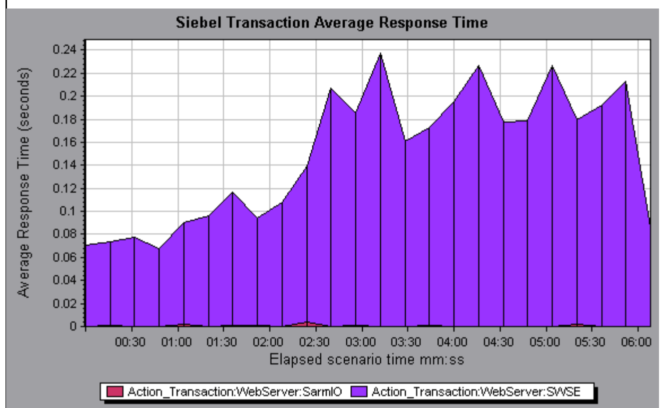
<table>
<tr>
<td>**Layer Level**</td>
<td> **Siebel Layer Breakdown** button shows the breakdown of the selected transaction.
 **Undo Siebel Layer Breakdown** returns the graph to the transaction level.

In the following figure, the Action_Transaction transaction has been broken down to its layers (Siebel Database, Application, and Web).

</td>
</tr>
</table>

<table>
<tr>
<td>Area Level</td>
<td>
 Siebel Area Breakdown button breaks the data down to its Siebel areas.

 Undo Siebel Area Breakdown button returns the graph to the layer level.

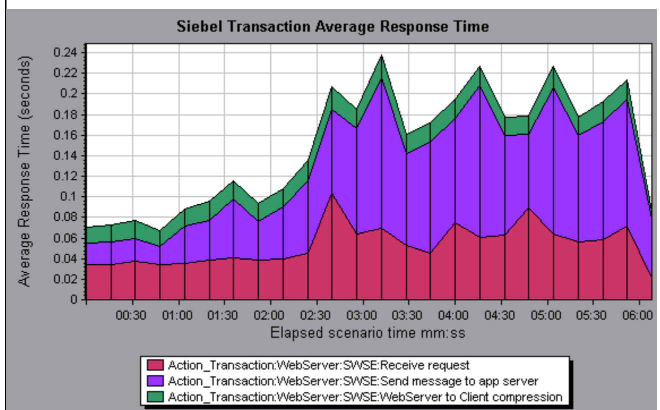
In the following the Web layer of the Action_Transaction transaction has been broken down to its Siebel areas.

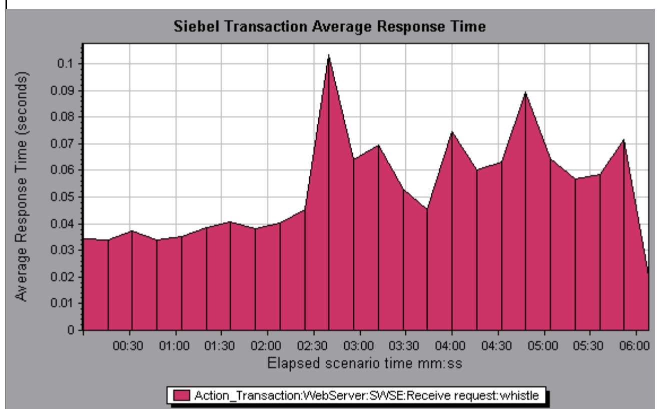


</td>
</tr>
<tr>
<td>Script Level</td>
<td>
 Siebel Script Breakdown button breaks the data down to its Siebel scripts. You can only break down to the script level from the scripting engine area.

 Undo Siebel Script Breakdown button returns the graph to the sub-area level.

You can break a transaction down further to its Siebel script level. You can only break down to the script level from the scripting engine area.
</td>
</tr>
</table>

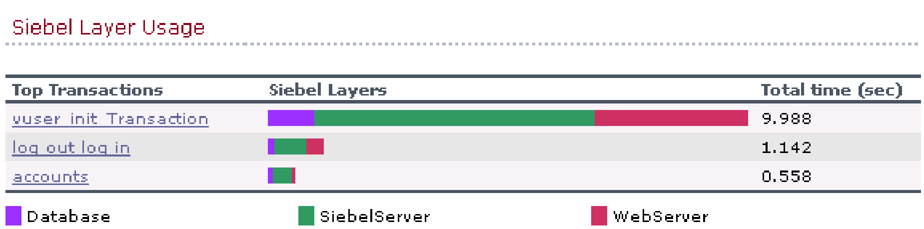
Sub-Area Level	 **Siebel Sub-Area Breakdown** button breaks the data down to its Siebel sub-areas. You can only break down to the sub-area level from the area level.  **Undo Siebel Sub-Area Breakdown** button returns the graph to the area level. In the following figure, the area level of the Action_Transaction transaction has been broken down to its Siebel sub-area. 

Server Level	 **Siebel Server Breakdown** button to group the data by Siebel server.  **Undo Siebel Server Breakdown** button ungroups data in the graph. In the following figure, the Action_Transaction;WebServer:SWSE:Receive Request transaction has been broken down to its Siebel servers. Server level breakdown is usual for pin pointing overloaded servers and for load balancing. 
See also	"Siebel Diagnostics Graphs Overview" on page 508

Siebel Diagnostics Graphs Summary Report

The Siebel Usage section of the Summary Report provides a usage chart for the Siebel layer breakdown. This report is available from the Session Explorer or as a tab in the Analysis window.

Breakdown options	The Siebel Layer Usage section breaks the individual transactions into: ➤ Web Server ➤ Siebel Server ➤ Database Layers ➤ Total usage time for each transaction
Tips	To view server side diagnostics data from the Summary Report, click the Siebel layer on which you want to perform transaction breakdown. The Siebel Transaction Response Time graph opens displaying the breakdown of the selected transaction.
Note	If you do not see diagnostics data on the Summary Report, check if you are using a user-defined template. To view relevant data, choose a different template from the list of templates, or create and apply a new template. For more information about using templates, see "Template Dialog Box" on page 90.
See also	"Siebel Diagnostics Graphs Overview" on page 508

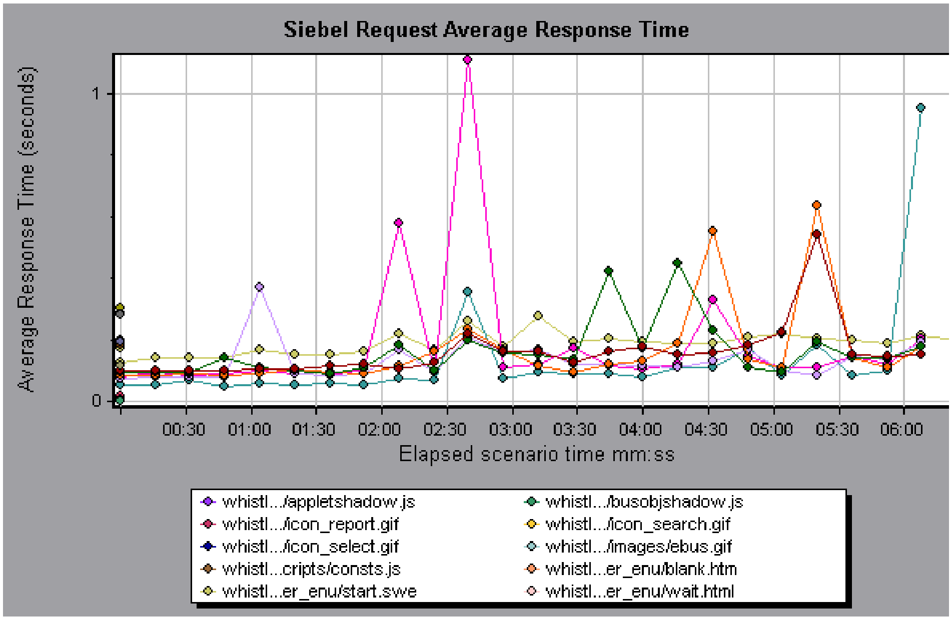


Siebel Request Average Response Time Graph

This graph displays the response time per HTTP request.

Purpose	The time is computed as the total request response time divided by the total number of instances of the specific request. For example, if a request was executed twice by one instance of transaction A, and once by a second instance of transaction A, and it took three seconds to execute each request, then the average response time is 9/3, or 3 seconds. The request time does not include the nested calls from within each request.
X-axis	Elapsed time since the start of the run.
Y-axis	Average response time (in seconds) per area.
Breakdown options	For breakdown options, see "Siebel Breakdown Levels" on page 520
Tips	You can filter the Siebel graphs by the following fields: ➤ **Transaction Name.** Shows data for the specified transaction. ➤ **Scenario Elapsed Time.** Shows data for transactions that ended during the specified time. For more information on filtering, see "Filtering and Sorting Graph Data" on page 93
See also	"Siebel Diagnostics Graphs Overview" on page 508

Example



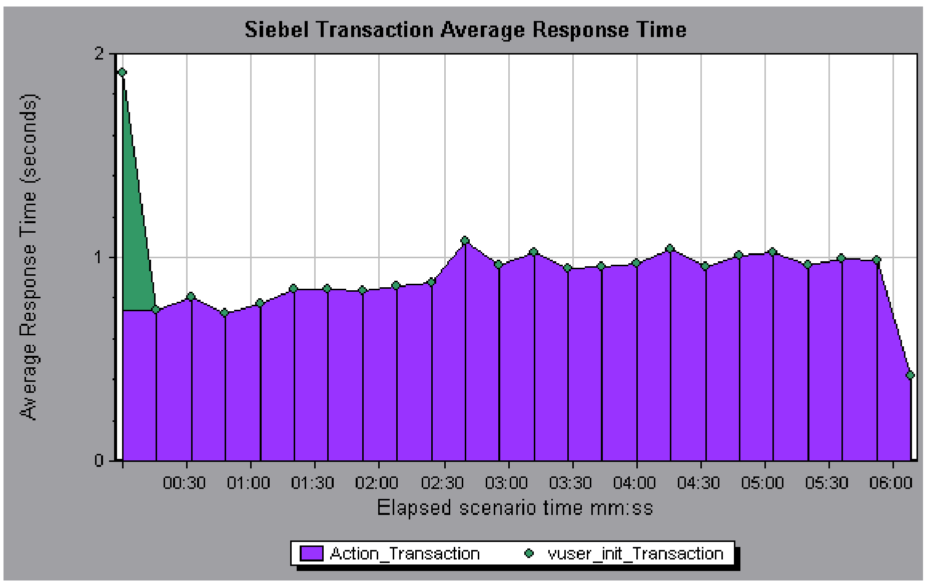
Siebel Transaction Average Response Time Graph

This graph displays the server response time for the selected area (layer, area, or sub-area) within each transaction, computed as the total response time for that layer or area divided by the total number of relevant transactions.

X-axis	Elapsed time since the start of the run.
Y-axis	Average response time (in seconds) per area.
Breakdown options	For breakdown options, see "Siebel Breakdown Levels" on page 520

Tips	You can filter the Siebel graphs by the following fields: ➤ **Transaction Name.** Shows data for the specified transaction. ➤ **Scenario Elapsed Time.** Shows data for transactions that ended during the specified time. For more information on filtering, see "Filtering and Sorting Graph Data" on page 93
See also	"Siebel Breakdown Levels" on page 520

Example



31

Siebel DB Diagnostics Graphs

This chapter includes:

Concepts

Tasks

Reference

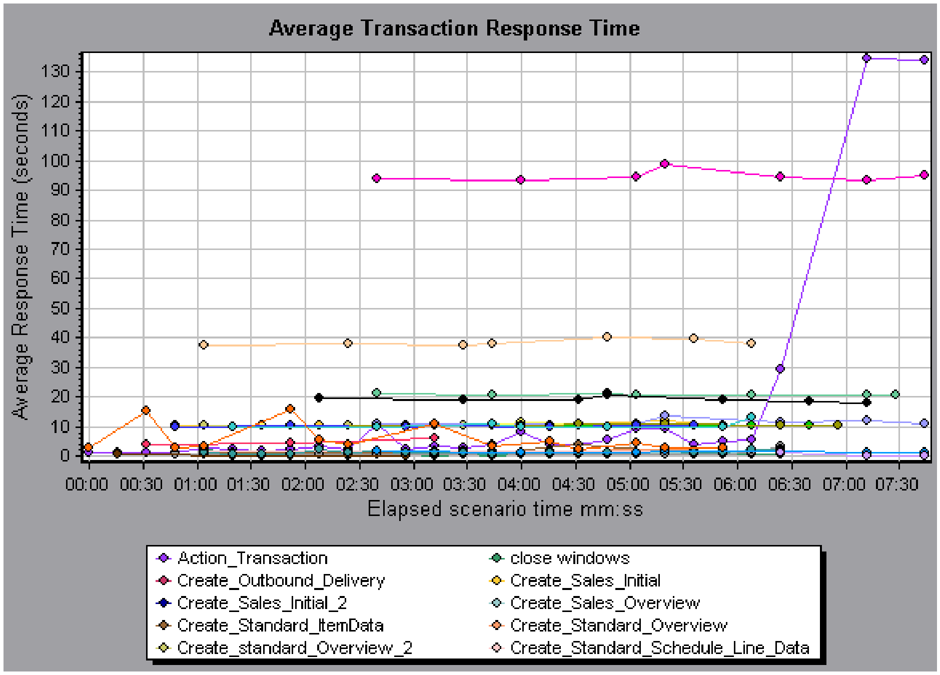
Concepts

Siebel DB Diagnostics Graphs Overview

Siebel DB Diagnostics graphs provide you with performance information for SQLs generated by transactions on the Siebel system. You can view the SQLs for each transaction, identify the problematic SQL queries of each script, and identify at what point problems occurred.

To analyze where problems are occurring, you correlate the data in the Siebel DB Diagnostics graphs with data in the Transaction Response Time graphs.

You begin analyzing these graphs with the transaction graphs that display the average transaction response time during each second of the load test scenario run. For example, the following Average Transaction Response Time graph demonstrates that the average transaction response time for the **query_for_contact** transaction was high.



Using the Siebel DB Diagnostics graphs, you can pinpoint the cause of the delay in response time for this transaction.

Note: A measurement that is broken down in the Average Transaction Response Time graph will be different from the same measurement broken down in the Siebel DB Side Transactions graph. This is because the Average Transaction Response Time graph displays the average transaction time, whereas the Siebel DB Side Transactions graph displays the average time per transaction event (sum of SQL component response times).

Tasks

How to Enable Siebel DB Diagnostics

To generate Siebel DB diagnostics data, you must first install the ERP/CRM Mediator. For information on installing the Mediator, refer to the *HP LoadRunner Installation Guide*.

The Mediator is used to gather and correlate offline diagnostics data from the Siebel server. The Mediator processes the diagnostics data, and then passes it to the Controller.

Note: The Mediator must reside in the same LAN as the Siebel server.

To obtain diagnostics data for these graphs, you need to set up the Siebel DB Diagnostics module before running the scenario, and specify the sampling percentage of diagnostics data to include in the diagnostics graphs. For more information on configuring Siebel DB Diagnostics, refer to the *HP LoadRunner Controller User Guide*.

Note:

- You should not use the Data Time Range feature (**Tools** > **Options** > **Result Collection** > **Data Time Range**) when analyzing Siebel DB Diagnostics graphs since the data may be incomplete.
- The settings that you configure are per scenario. All scripts in the scenario will run under the same diagnostics configuration.
- To ensure that valid diagnostics data is generated, manually define the transactions in the Vuser script rather than using automatic transactions. Make sure to disable the following options in the Run-Time Settings' **General : Miscellaneous** node: **Define each action as a transaction** and **Define each step as a transaction**.

How to Synchronize Siebel Clock Settings

This task describes how to synchronize the Load Generator and Siebel application server clocks to ensure that the correlation of SQLs to transactions is correct.

1. Choose **Tools** > **Siebel Database Diagnostics Options**.
2. Select **Apply Application Server time settings.**
3. Click **Add** and enter the information as described in "Siebel Database Diagnostics Options Dialog Box" on page 544.

4 Click **OK** to save the data and close the dialog box.

Note: You must reopen the results file for time synchronization to take effect.

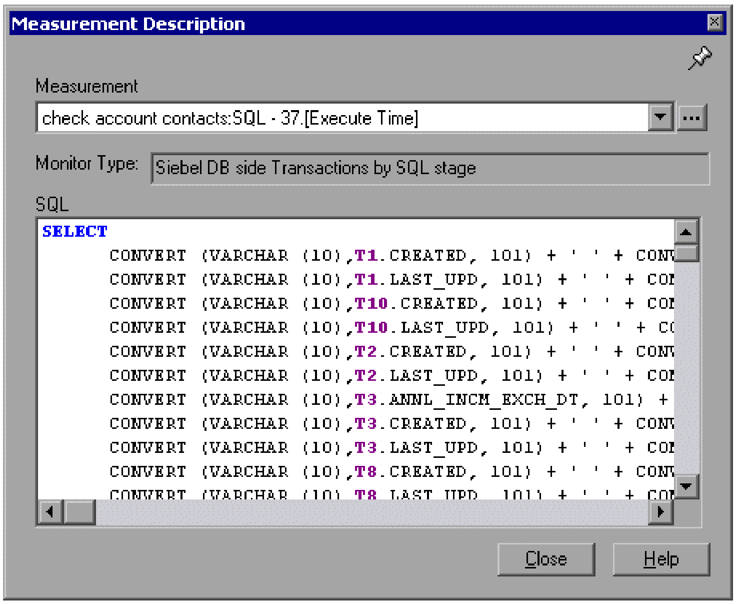
Reference

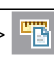
Siebel DB Diagnostics Graphs User Interface

This section includes (in alphabetical order):

Measurement Description Dialog Box

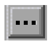
You can view the full SQL statement for a selected SQL element by choosing **Show measurement description** from the Legend window. The Measurement Description dialog box opens displaying the name of the selected measurement and the full SQL statement.



To access	Legend window > 
See also	"Siebel Database Breakdown Levels" on page 539

User interface elements are described below:

UI Elements	Description
	Break the data down to a lower level.
	Return to the previous level.

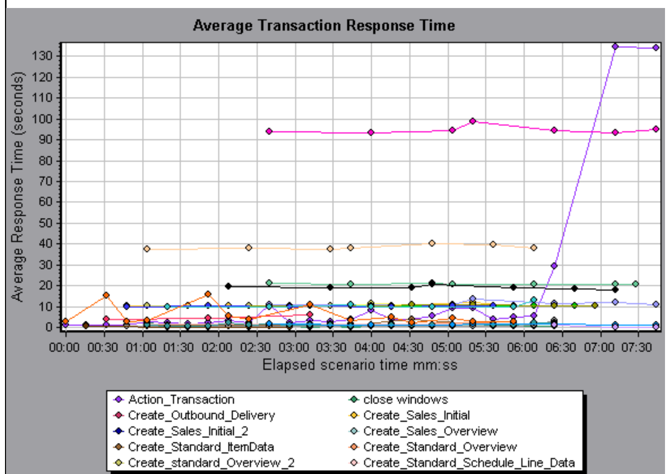
UI Elements	Description
	To keep the focus on the Measurement Description dialog box, click the **Stay on Top** button. This enables you to view the full SQL statement of any measurement by selecting it in the Legend window. Click the button again to remove the focus.
	Click the **Breaking Measurement** button to display the Transaction Name and SQL Alias Name of the selected measurement.

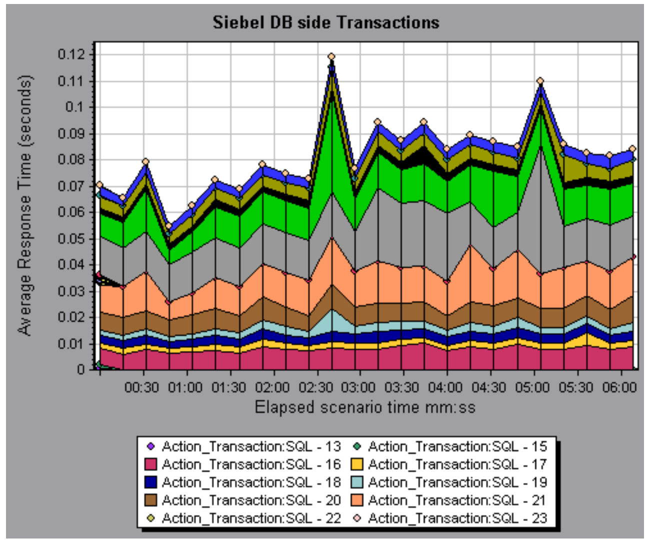
Siebel Database Breakdown Levels

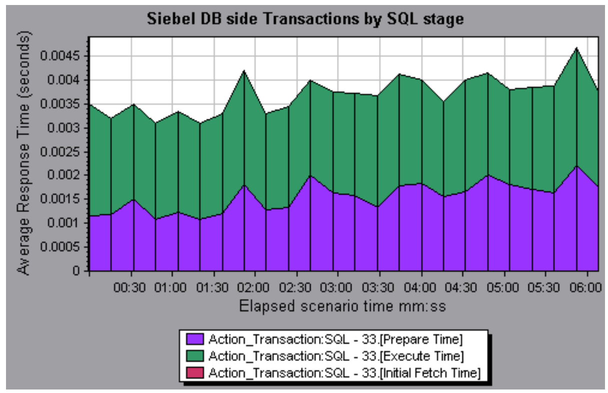
You can break down Siebel layers into areas, sub-areas, servers, and scripts to enable you to pinpoint the exact location where time is consumed.

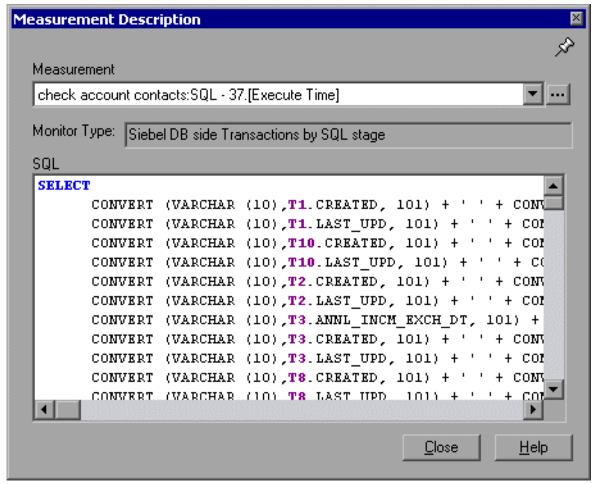
To access	Use one of the following to access breakdown options: ➤ **<Siebel DB Diagnostics Graphs>** > **View** > **Siebel DB Diagnostics** ➤ **<Siebel DB Diagnostics Graphs>** > select transaction > short-cut menu > **Siebel DB Diagnostics** ➤ See toolbar options for each breakdown level
Important Information	The breakdown menu options and buttons are not displayed until a transaction is selected.
See also	"Siebel DB Diagnostics Graphs Overview" on page 532

Siebel Breakdown Levels are described below:

Transaction Level	The following figure displays the top level Average Transaction Response Time graph. The graph displays several transactions. You can break this graph down to show the SQL statements and the SQL stages level. 

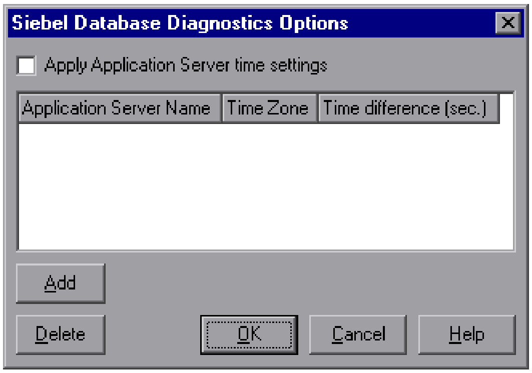
SQL Statements Level	 **Siebel SQL Statements Breakdown** button shows the breakdown of the selected transaction. In the following figure, the Siebel DB Side Transactions graph displays the Action_Transaction broken down to its SQL statements. 

SQL Stages Level	 **Measurement Breakdown** button breaks the data down to a lower level.  **Undo Breakdown Measurement** button returns to the previous level. In the following figure, the Siebel DB Side Transactions by SQL Stage graph displays Action_Transaction:SQL-33 broken down to its SQL stage: Prepare, Execute, and Initial Fetch. 

Show measurement description	You can view the full SQL statement for a selected SQL element by choosing **Show measurement description** from the Legend window. The Measurement Description dialog box opens displaying the name of the selected measurement and the full SQL statement. 
See also	"Siebel DB Diagnostics Graphs Overview" on page 532

Siebel Database Diagnostics Options Dialog Box

This dialog box enables you to synchronize the Load Generator and Siebel application server clocks.



To access	**Tools > Siebel Database Diagnostics Options**
Note	You must reopen the results file for time synchronization to take effect.
See also	"How to Synchronize Siebel Clock Settings" on page 535

User interface elements are described below:

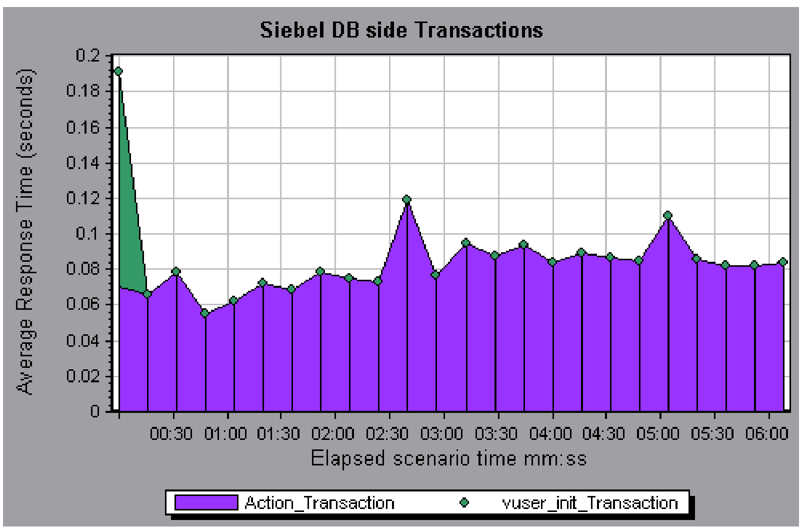
UI Elements	Description
Apply Application Server time settings	Enables the synchronized time settings option.
Application Server Name	Enter the name of the Siebel application server.
Time Zone	Enter the time zone of the Siebel application server (GMT or Local). GMT means the application server time is reported in GMT time, and local means the application server time is reported in local time.

UI Elements	Description
Time Difference (sec.)	Enter the time difference (in seconds) between the load generator and the Siebel application server. Use the minus sign ("-") if the time on Siebel application server is ahead of the load generator. For example, if the application server time is two minutes ahead of the load generator time, enter -120 in the time difference field.
Add	Enables you to add an application server's time settings to the list.
Delete	Deletes the server breakdown time settings from the list.

Siebel DB Side Transactions Graph

This graph displays the average transaction execution time in the Siebel database.

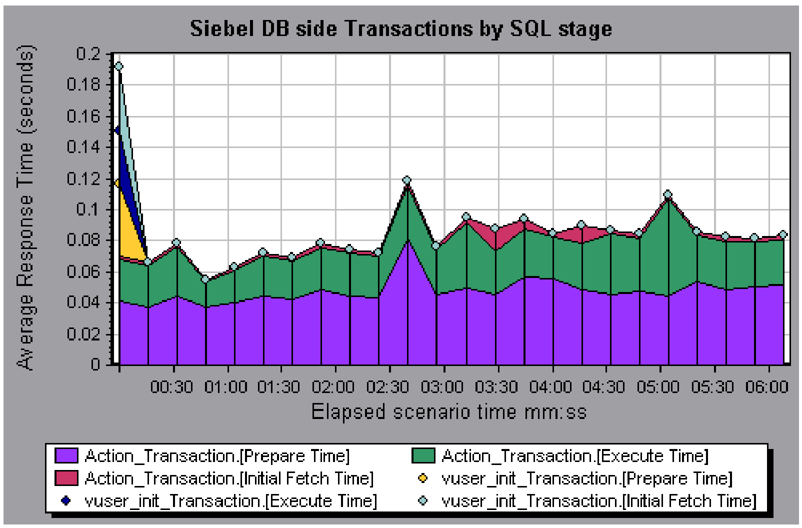
X-axis	Elapsed time since the start of the run.
Y-axis	Average response time (in seconds) of each transaction.
Breakdown options	You can break down a transaction in the Siebel DB Side Transactions graph to view its SQL statements. In the following figure, the **Action_Transaction** transaction is broken down to its SQL statements. to see an example of the Siebel DB Side Transactions graph.
See also	"Siebel DB Diagnostics Graphs Overview" on page 532



Siebel DB Side Transactions by SQL Stage Graph

This graph displays the time taken by each SQL, grouped by SQL stage: Prepare, Execute, and Initial Fetch.

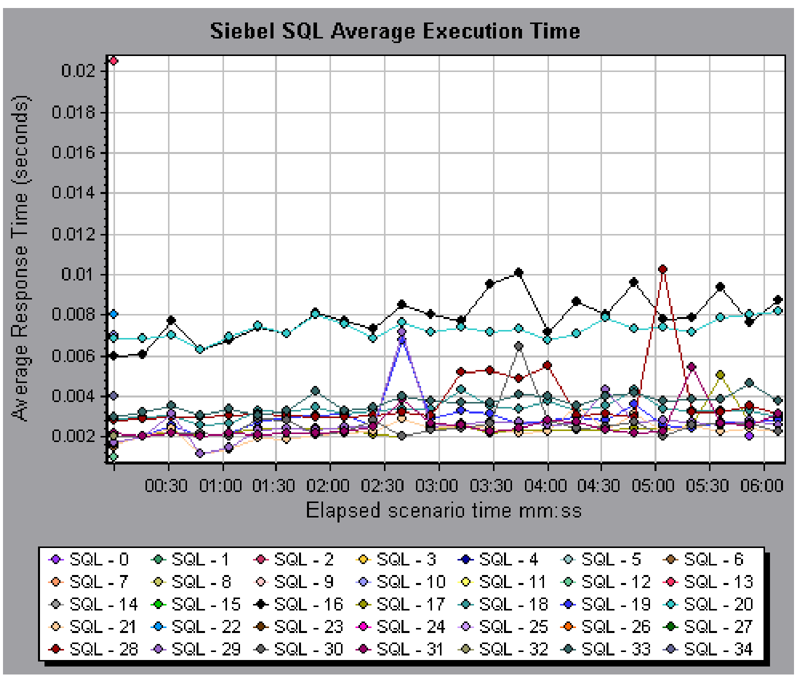
X-axis	Elapsed time since the start of the run.
Y-axis	Average time (in seconds) taken to perform each SQL stage.
Breakdown options	"Siebel Database Breakdown Levels" on page 539
See also	"Siebel DB Diagnostics Graphs Overview" on page 532



Siebel SQL Average Execution Time Graph

This graph displays the average execution time of each SQL performed in the Siebel database.

Purpose	This enables you to quickly identify problematic SQLs regardless of the transaction that produced them. You can then choose **Show measurement description** from the Legend window to view the full SQL statement. The SQL statements are listed by a numeric ID.
X-axis	Elapsed time since the start of the run.
Y-axis	Average response time (in seconds) of each SQL.
Breakdown options	"Siebel Database Breakdown Levels" on page 539
See also	"Siebel DB Diagnostics Graphs Overview" on page 532



32

Oracle 11i Diagnostics Graphs

This chapter includes:

Concepts

Tasks

Reference

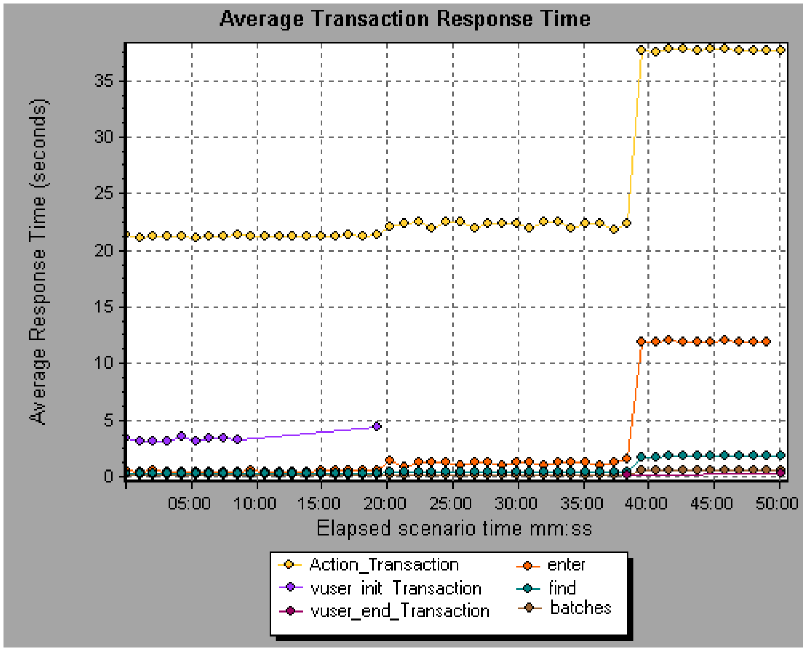
Concepts

Oracle 11i Diagnostics Graphs Overview

Oracle 11i Diagnostics graphs provide you with performance information for SQLs generated by transactions on the Oracle NCA system. You can view the SQLs for each transaction, identify the problematic SQL queries of each script, and identify at what point problems occurred.

To analyze where problems are occurring, you correlate the data in the Oracle 11i Diagnostics graphs with data in the Transaction Response Time graphs.

You begin analyzing these graphs with the transaction graphs that display the average transaction response time during each second of the load test scenario run. For example, the following Average Transaction Response Time graph demonstrates that the average transaction response time for the **enter** transaction was high.



Using the Oracle 11i Diagnostics graphs, you can pinpoint the cause of the delay in response time for this transaction.

Note:

- A measurement that is broken down in the Average Transaction Response Time graph will be different from the same measurement broken down in the Oracle 11i Side Transactions graph. This is because the Average Transaction Response Time graph displays the average transaction time, whereas the Oracle 11i Side Transactions graph displays the average time per transaction event (sum of SQL component response times).
- **vuser_init** and **vuser_end** actions in Oracle cannot be broken down. For more information, refer to the *HP Virtual User Generator User Guide*.

Tasks

How to Enable Oracle 11i Diagnostics

To generate Oracle 11i Diagnostics data, you must first install the ERP/CRM Mediator. For information on installing the Mediator, refer to the *HP LoadRunner Installation Guide*.

The Mediator is used to gather and correlate offline diagnostics data from the Oracle server. The Mediator processes the diagnostics data, and then passes it to the Controller.

Note: The Mediator must reside in the same LAN as the Oracle server.

To obtain diagnostics data for these graphs, you need to set up the Oracle 11i Diagnostics module before running the scenario, and specify the sampling percentage of diagnostics data to include in the diagnostics graphs. For more information on configuring Oracle 11i Diagnostics, refer to the *HP LoadRunner Controller User Guide*.

Note:

- The settings that you configure are per scenario. All scripts in the scenario will run under the same diagnostics configuration.
- To ensure that valid diagnostics data is generated, manually define the transactions in the Vuser script rather than using automatic transactions. Make sure to disable the following options in the Run-Time Settings' **General : Miscellaneous** node: **Define each action as a transaction** and **Define each step as a transaction**.
- If the Oracle 11i trace cannot be enabled automatically using the built-in mechanism, you can enable it manually in the Vuser script using the **nca_set_custom_dbtrace** and **nca_set_dbtrace_file_index** functions. This may occur if you are using a custom application that does not have a standard UI.
- You should not use the Data Time Range feature (**Tools** > **Options** > **Result Collection** > **Data Time Range**) when analyzing Oracle 11i Diagnostics graphs since the data may be incomplete.

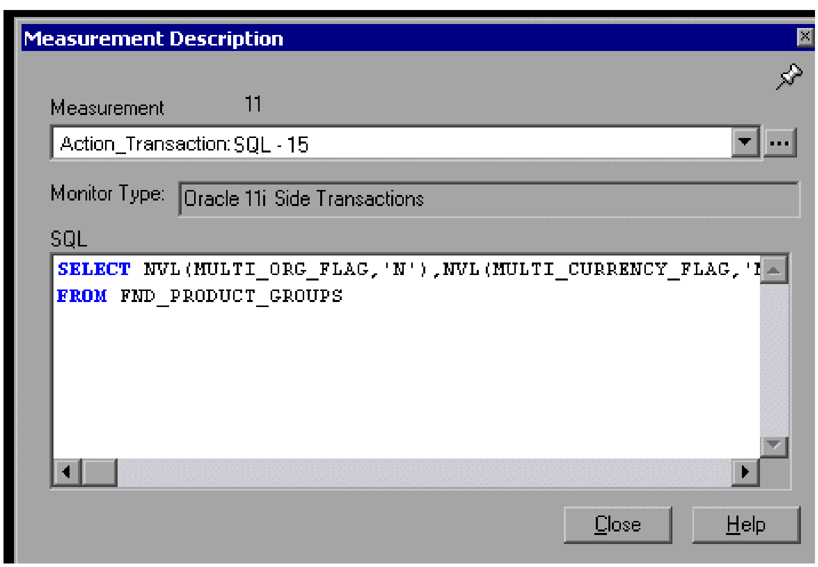
Reference

Oracle 11i Diagnostics Graphs User Interface

This section includes (in alphabetical order):

Measurement Description Dialog Box

This dialog box enables you to view the full SQL statement for a selected SQL element.



To access	Legend window > 
See also	➤ "Oracle 11i Diagnostics Graphs Overview" on page 550 ➤ "Oracle Breakdown Levels" on page 557

User interface elements are described below:

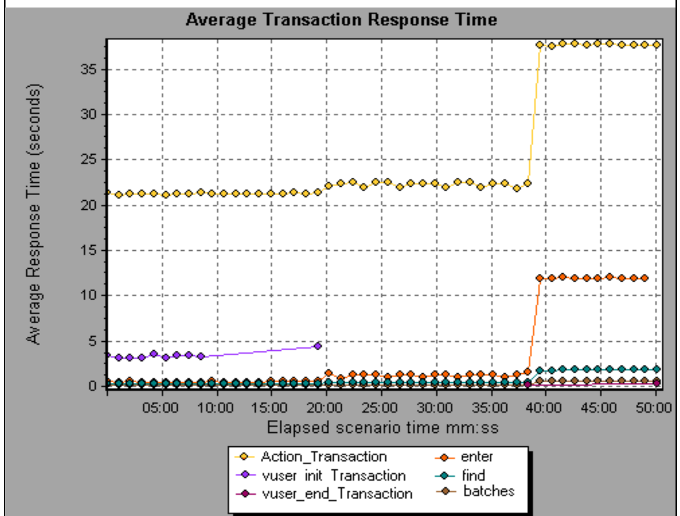
UI Elements	Description
	To keep the focus on the Measurement Description dialog box, click the Stay on Top button. This enables you to view the full SQL statement of any measurement by selecting it in the Legend window. Click the button again to remove the focus.
	Click the Breaking Measurement button to display the Transaction Name and SQL Alias Name of the selected measurement.

Oracle Breakdown Levels

After you have enabled Oracle 11i Diagnostics on the Controller machine and run the load test scenario, you can view the diagnostics data.

To access	Use one of the following to access breakdown options:" ➤ <**Oracle Diagnostics Graphs**> > **View** > **Oracle Diagnostics** ➤ <**Oracle Diagnostics Graphs**> > select transaction > short-cut menu > **Oracle Diagnostics** ➤ See toolbar options for each breakdown level
Important Information	The breakdown menu options and buttons are not displayed until a transaction is selected.
See also	"Oracle 11i Diagnostics Graphs Overview" on page 550

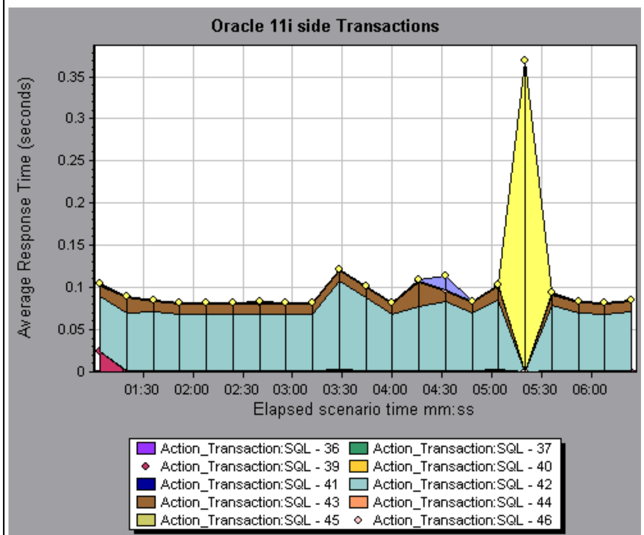
Oracle Breakdown Levels are described below:

<table>
<tr>
<td>Transaction Level</td>
<td>
The following figure illustrates the top level Average Transaction Response Time graph. The graph displays several transactions.
You can break this graph down to show the SQL statements and the SQL stages level.


</td>
</tr>
</table>

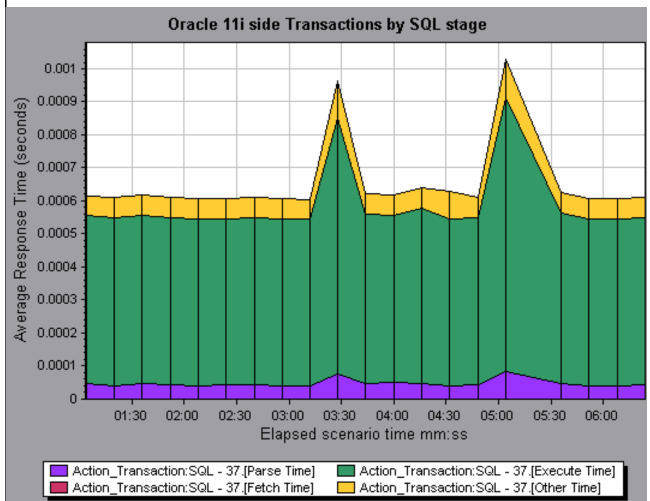
<table>
<tr>
<td>SQL Statements Level</td>
<td>
Oracle SQL Statement Breakdown button shows the breakdown of the selected transaction.

In the following figure, the Oracle 11i Side Transactions graph displays the Action_Transaction transaction broken down to its SQL statements.



</td>
</tr>
</table>

<table>
<tr>
<td>SQL Stages Level</td>
<td>In the following figure, the Oracle 11i Side Transactions by SQL Stage graph displays Action_Transaction:SQL-37 broken down to its SQL stages: Parse Time, Execute Time, Fetch Time, and Other Time. Other Time includes other database time such as bind time.

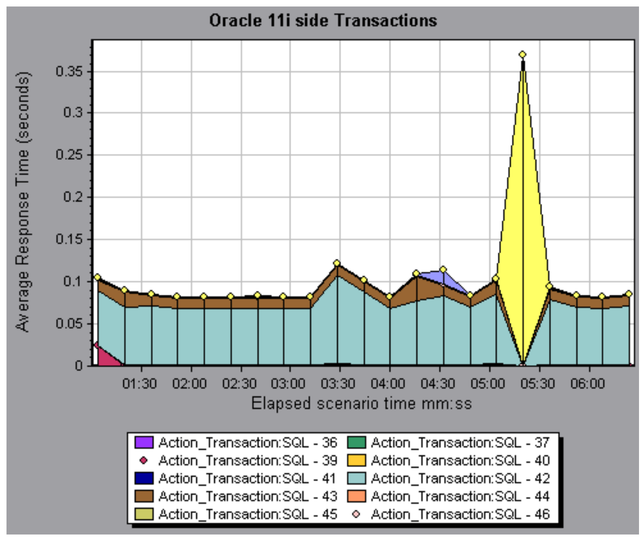


</td>
</tr>
<tr>
<td></td>
<td>You can break the data down to a lover level.</td>
</tr>
<tr>
<td></td>
<td>Enables you to return to a previous level.</td>
</tr>
</table>

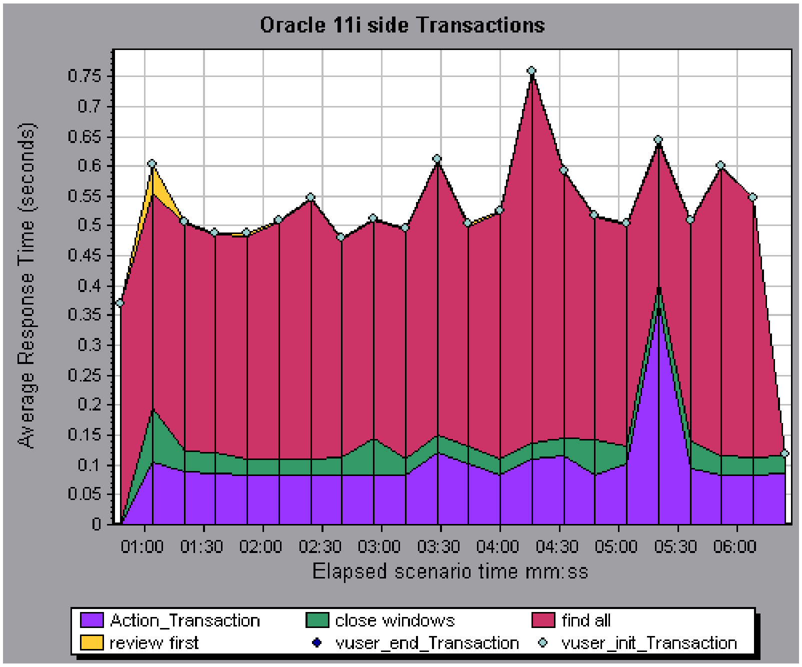
Oracle 11i Side Transactions Graph

This graph displays the average transaction execution time in the Oracle database.

<table>
<tr><td>X-axis</td><td>Elapsed time of the scenario run.</td></tr>
<tr><td>Y-axis</td><td>Response time (in seconds) of each transaction.</td></tr>
<tr><td>Breakdown options</td><td>You can break down a transaction in the Oracle 11i Side Transactions graph to view its SQL statements. In the following figure, the Action_Transaction transaction is broken down to its SQL statements.



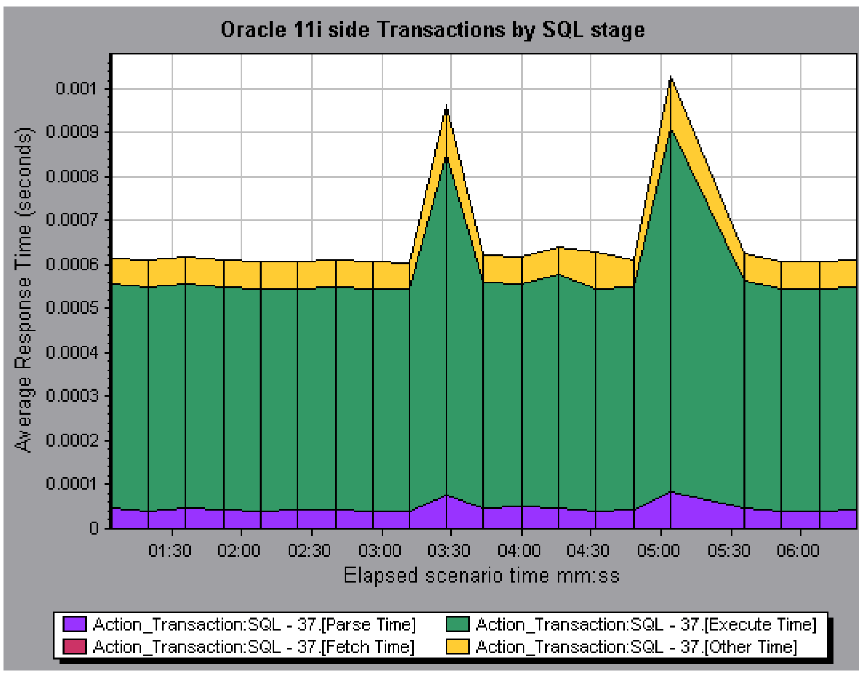
To break the displayed elements down further, see "Oracle Breakdown Levels" on page 557.</td></tr>
<tr><td>See also</td><td>"Oracle 11i Diagnostics Graphs Overview" on page 550</td></tr>
</table>


Oracle 11i side Transactions
Average Response Time (seconds)
0.75
0.7
0.65
0.6
0.55
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
01:00
01:30
02:00
02:30
03:00
03:30
04:00
04:30
05:00
05:30
06:00
Elapsed scenario time mm:ss
Action_Transaction
close windows
find all
review first
vuser_end_Transaction
vuser_init_Transaction

Oracle 11i Side Transactions by SQL Stage Graph

This graph displays the time taken by each SQL, divided by the SQL stages: Parse Time, Execute Time, Fetch Time, and Other Time. Other Time includes other database time such as bind time.

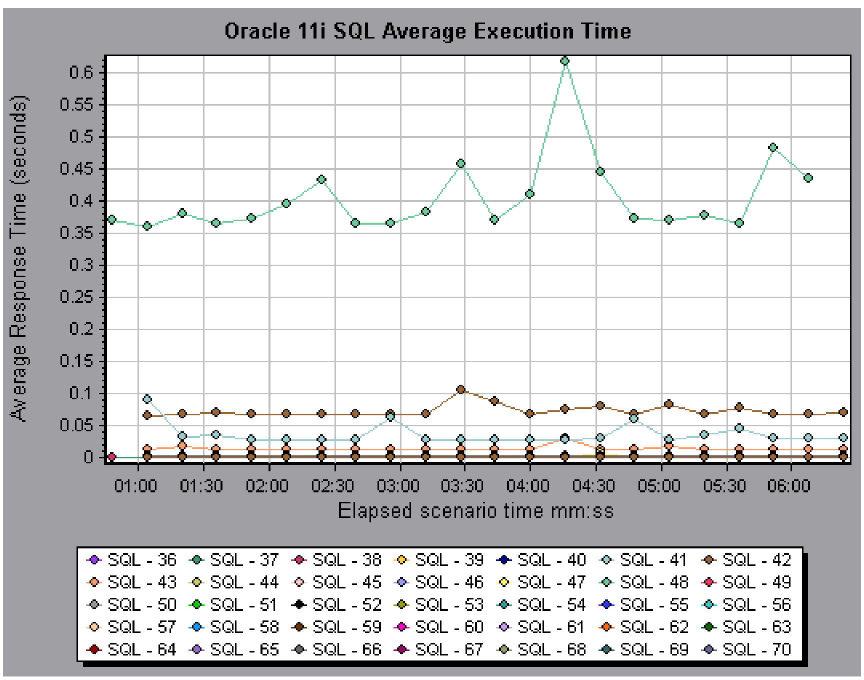
X-axis	Elapsed time since the scenario run.
Y-axis	Average response time (in seconds) of each SQL stage.
Breakdown options	"Oracle Breakdown Levels" on page 557
See also	"Oracle 11i Diagnostics Graphs Overview" on page 550



Oracle 11i SQL Average Execution Time Graph

This graph displays the average execution time of each SQL performed in the Oracle database.

Purpose	The graph enables you to quickly identify problematic SQLs regardless of the transaction that produced them.
X-axis	Elapsed time since the scenario run.
Y-axis	Average response time (in seconds) of each SQL.
Breakdown options	"Oracle Breakdown Levels" on page 557
Tips	You can then select **Show measurement description** from the Legend window to view the full SQL statement.
Note	The SQL statements are shortened to a numeric indicator.
See also	"Oracle 11i Diagnostics Graphs Overview" on page 550



33

SAP Diagnostics Graphs

This chapter includes:

Concepts

Tasks

Reference

Concepts

SAP Diagnostics Graphs Overview

SAP Diagnostics enables you to pinpoint the root cause of a certain problem (i.e. DBA, Network, WAS, Application, OS/HW) quickly and easily, and engage with the relevant expert only, without having to present the problem to a whole team of people.

Using SAP Diagnostics, you can create graphs and reports, which you can present to the relevant expert when discussing the problems that occurred.

SAP Diagnostics also allow an SAP performance expert (in one of the areas of expertise) to perform the required root-cause analysis more quickly and easily.

Tasks

How to Enable SAP Diagnostics

To generate SAP diagnostics data, you must first install the ERP/CRM Mediator. For information on installing the Mediator, refer to the *HP LoadRunner Installation Guide*.

The Mediator component is used to gather and correlate offline diagnostics data from the SAP server. The Mediator processes the diagnostics data, and then passes it to the Controller.

To obtain diagnostics data for these graphs, you need to set up the SAP Diagnostics module before running the load test scenario, and specify the sampling percentage of diagnostics data to include in the diagnostics graphs. For more information on configuring SAP Diagnostics, refer to the *HP LoadRunner Controller User's Guide*.

Important Notes

- In order for SAP Diagnostics to perform effectively, it relies on sound and consistent performance of the SAP software. Malformation or inappropriate configuration of the SAP software could cause inaccuracies in SAP Diagnostics data collection.
- To ensure that valid diagnostics data is generated, manually define the transactions in the Vuser script rather than using automatic transactions. Make sure to disable the following options in the Run-Time Settings' **General: Miscellaneous** node: **Define each action as a transaction** and **Define each step as a transaction**.
- The ERP/CRM Mediator requires an installation of SAPGUI 6.20 or 6.40.

- If no connection can be established between the ERP/CRM Mediator and one of the SAP application servers, no work process or OS monitor data is collected for that server. However, as long as there is a connection to one of the servers, statistical records for response time breakdown are still available.
- The settings that you configure are per scenario. All scripts in the scenario will run under the same diagnostics configuration.

How to Configure SAP Alerts

SAP Diagnostics comes with a set of alert rules with pre-defined threshold values.

When you open a LoadRunner results file (.lrr) in Analysis, these alert rules are applied to the load test scenario results, and if a threshold value is exceeded, Analysis generates an alert that there is a problem.

Before opening a LoadRunner results file, you can define new threshold values for the alert rules using the Alerts Configuration dialog box. Then, when you open the results file, the customized alert rules are applied.

Note: When an Analysis session is open, the Alerts Configuration dialog box is not editable. To edit thresholds in the Alerts Configuration dialog box, close all open sessions.

This task describes how to define threshold values for alert rules when analyzing load test scenario results.

1. Close all open Analysis sessions.
2. From the **Tools** menu, select **SAP Diagnostics Alerts Configuration**.
3. The **Generate alert if** column lists the rules. Set the threshold for each rule in the **Threshold** column.

4. By default, all pre-defined alert rules are enabled. To disable an alert rule, clear the check box next to that rule.
5. Click **OK** to apply your changes and close the Alerts Configuration dialog box.

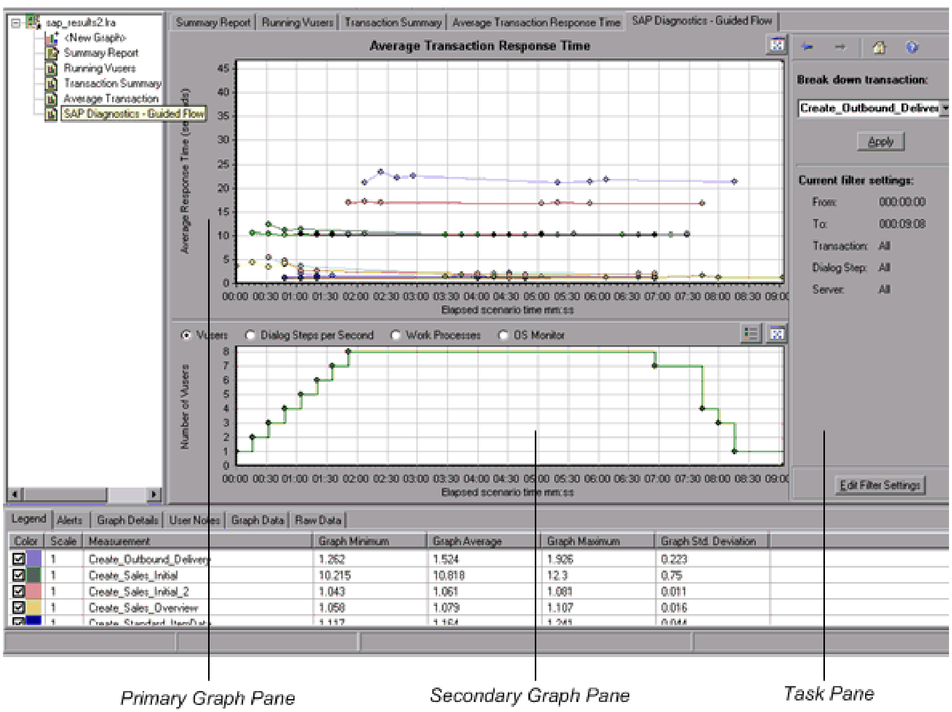
Note: Modifying the alert rules does not affect the results of a saved Analysis session. You need to re-analyze the results in order for new settings to take effect.

Reference

SAP Diagnostics - Guided Flow Tab

You open the SAP Diagnostics graphs from the Analysis Summary Report or from **Session Explorer** > **Graphs** > **SAP Diagnostics - Guided Flow.**

This tab remains open throughout the Analysis application flow, and its content varies according to the breakdown flow.

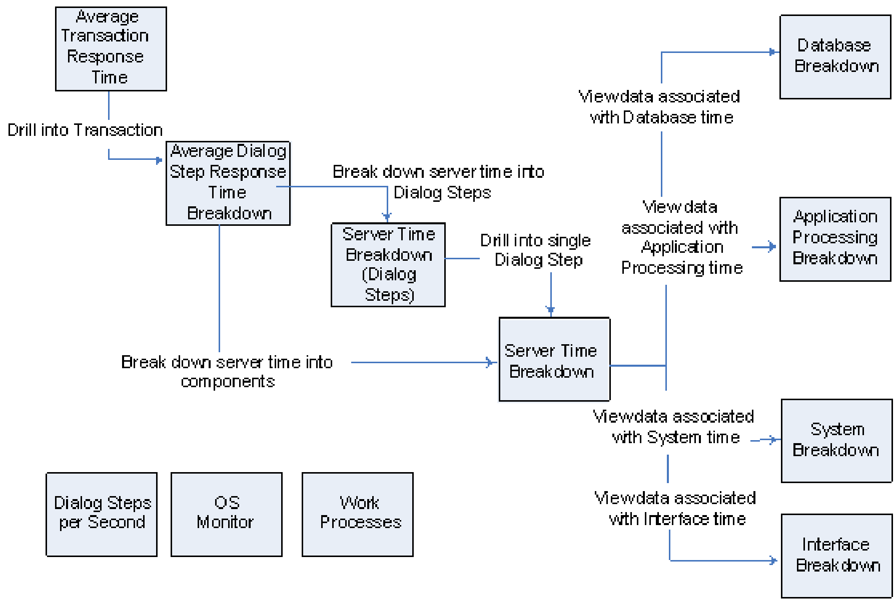


User interface elements are described below:

UI Elements	Description
Primary Graph Pane	The upper pane of the SAP Diagnostics - Guided Flow tab is referred to as the *primary graph pane*. This pane displays graphs of the transactions and their broken down dialog steps or components, and other associated resources. You break down the graphs displayed in this pane using the breakdown options provided in the right pane of the guided flow (see "SAP Breakdown Task Pane" on page 581). You can open the displayed graph in full view by clicking the **Enlarge Graph** button in the top right corner of this pane. An enlarged version of the graph opens in a new tab.
Secondary Graph Pane	The lower pane of the SAP Diagnostics - Guided Flow tab is referred to as the *secondary graph pane* and displays graphs showing secondary information supporting the graph displayed in the primary graph pane. To see the legend for the graph displayed in this pane, click the **Graph Legend** button in the top right corner. To see all the data in the Legend, scroll along the horizontal scroll bar. You can open the displayed graph in full view by clicking the **Enlarge Graph** button in the top right corner of this pane. An enlarged version of the graph opens in a new tab.
Task Pane	The pane on the right side of the SAP Diagnostics - Guided Flow tab is referred to as the *task pane*. You use the task pane to choose the level of breakdown you want to view, to filter and group transaction and server information, and to navigate backwards and forwards through the broken down graphs. For more information see "SAP Breakdown Task Pane" on page 581.

Application Flow

The following diagram depicts the general flow of SAP Diagnostics:



The main view of SAP Diagnostics displays all of the transactions in a scenario run for which there is SAP diagnostics data. Each transaction can be broken down into server-time components, or first into the dialog steps that comprise a transaction, and then into server-time components. The server components can further be broken down into sub-components or other related data.

There are 3 independent/parallel views: Dialog Steps per Second, OS Monitor, and Work Processes. These do not generally participate in the breakdown flow, and you may choose to display or hide them.

SAP Diagnostics User Interface

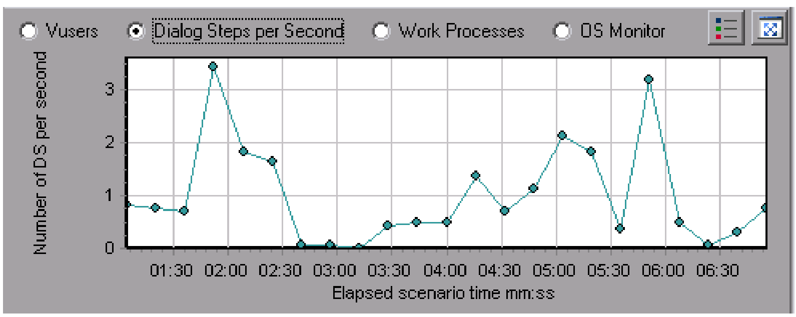
This section includes (in alphabetical order):

Dialog Steps per Second Graph

This graph represents the number of dialog steps that ran on all the servers during each second of the load test scenario run.

X-axis	Elapsed scenario time (in hh:mm:ss).
Y-axis	Number of dialog steps per second.
See also	"SAP Breakdown Task Pane" on page 581 "Vuser Graphs" on page 233 "Work Processes Graph" on page 593 "OS Monitor Graph" on page 574

Example



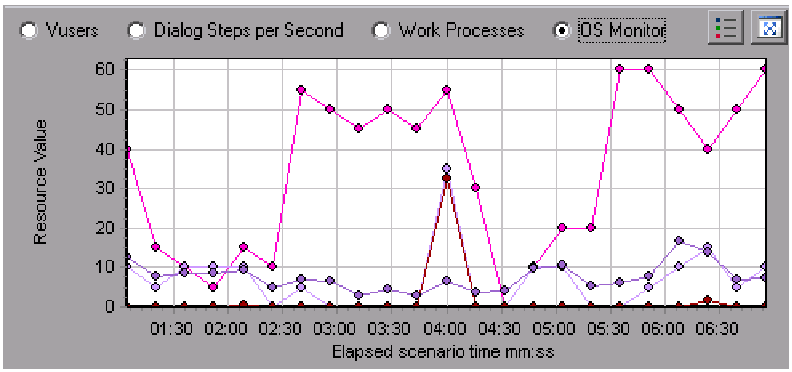
OS Monitor Graph

This graph represents the operating system resources that were measured throughout the load test scenario run.

X-axis	Elapsed scenario time (in hh:mm:ss).
Y-axis	Resource value.

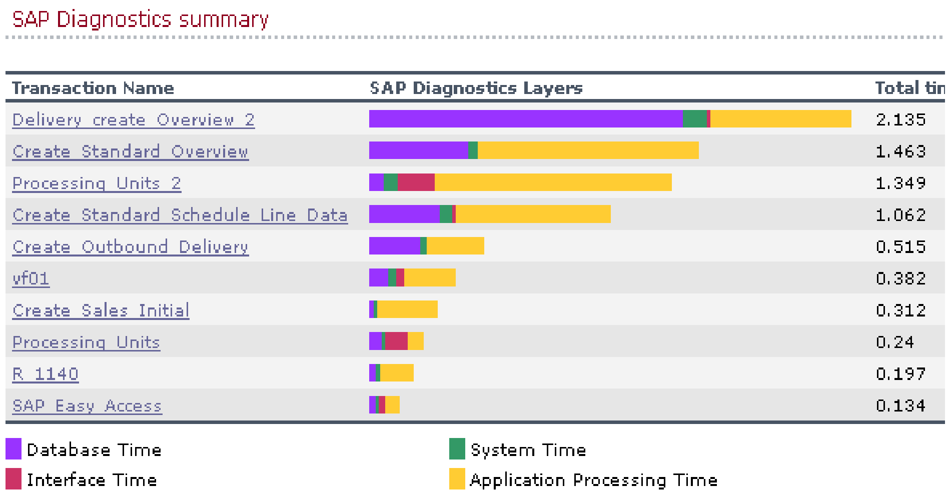
Note	This graph is available only when a single server filter is applied.
See also	"SAP Breakdown Task Pane" on page 581 "Dialog Steps per Second Graph" on page 574 "Work Processes Graph" on page 593

Example



SAP Alerts Configuration Dialog box

This dialog box enables you to define threshold values for alert rules used when opening the results file (.lrr) in Analysis.



To access	Tools > SAP Diagnostics alerts configuration
Important information	Modifying the alert rules does not affect the results of a saved Analysis session. You need to re-analyze the results in order for new settings to take effect.
See also	"SAP Diagnostics Graphs Overview" on page 566

User interface elements are described below:

UI Elements	Description
Enabled	By default, all pre-defined alert rules are enabled. To disable an alert rule, clear the check box next to that rule.
Generate alert if	The **Generate alert if** column lists the rules.
Threshold	Set the threshold for each rule in the **Threshold** column.

SAP Alerts Window

This Window displays a list of alerts related to the data displayed in the current graph(s) shown in the Analysis window.

To access	**Windows > SAP Alerts**
See also	"SAP Alerts Window" on page 577 "How to Configure SAP Alerts" on page 568

User interface elements are described below:

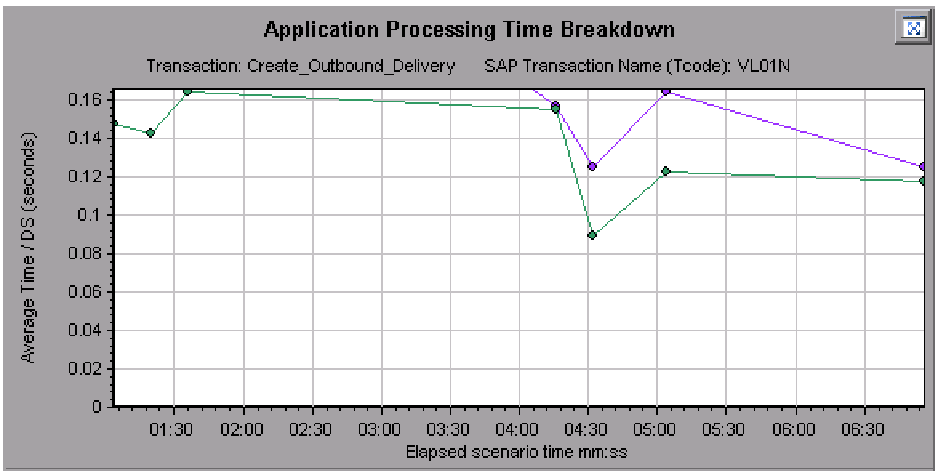
UI Elements	Description
Type	Displays one of the following icons indicating the type of alert: **Standard Alert.** This alert is generated in the context of a transaction and/or server if the conditions of a pre-defined alert rule are met. **Major Alert**. There are two types of alerts: ➤ **General Application Problem Alert.** If a standard alert was generated in the context of a transaction, and the same alert was generated in the context of all other transactions running in the same time frame, then a major alert of this type is generated, indicating that there is a general application problem. **Note:** If a Dialog Step filter is applied (for a single dialog step), then this alert is not generated. ➤ **Server-Specific Problem Alert.** This alert is generated for a specific server if a certain measurement on that server exceeds its threshold, while the overall server performance for that measurement is satisfactory. This type of alert indicates that there is a server related problem. **Note:** Server-Specific Problem alerts are generated only when the current server context is "All Servers”.
Time interval	The time interval during which the problem occurred.
Transaction/Server	The name of the transaction and server where problem occurred.
Description	A description of the alert.
Recommended Step	Recommends what to do in order to understand the problem on a deeper level.
Action	A link to a graph representing the data described in the alert, allowing for a more graphic display of the alert. Double-click the link to open the graph.

SAP Application Processing Time Breakdown Graph

This graph displays the behavior of resources associated with application processing time, namely ABAP time and CPU time.

X-axis	Elapsed load test scenario time (in hh:mm:ss).
Y-axis	Average time per dialog step (in seconds).
See also	"SAP Breakdown Task Pane" on page 581 "SAP Secondary Graphs" on page 592

Example



SAP Primary Graphs

You view the SAP Diagnostics graphs in the primary graph pane.

You can open the graph in full view by clicking in the top right corner of the primary graph pane. An enlarged version of the graph opens in a new tab.

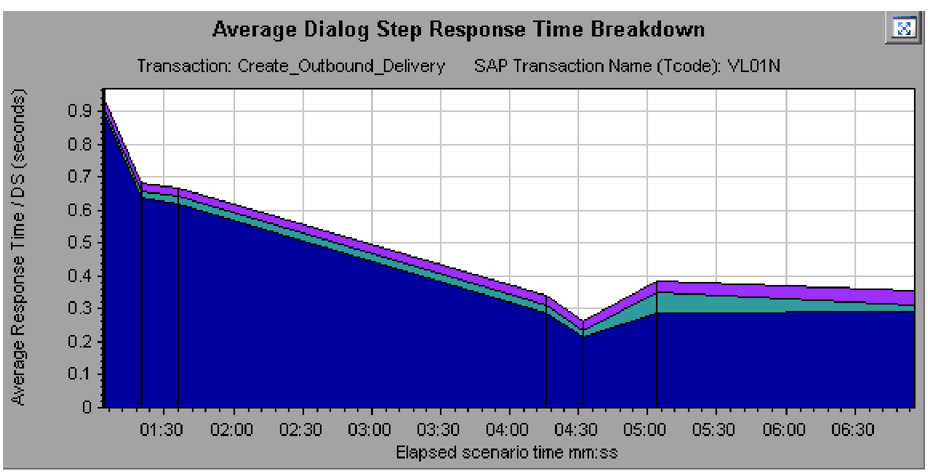
To filter or group data displayed in these graphs, see "Current filter settings" on page 583.

SAP Average Dialog Step Response Time Breakdown Graph

This graph represents a breakdown of the average dialog step response time of a specific transaction. The graph displays the Network Time, Server Response Time, (including the GUI time), and Other Time (the time taken for the client to process the dialog step) of a single transaction.

X-axis	Elapsed time since the start of the run (in hh:mm:ss).
Y-axis	The average response time divided by the number of dialog steps (in seconds).
Breakdown options	Components This option opens the "SAP Server Time Breakdown Graph" on page 587 Dialog Steps This option opens the "SAP Server Time Breakdown (Dialog Steps) Graphs" on page 586
See also	"SAP Breakdown Task Pane" on page 581 "SAP Secondary Graphs" on page 592 "Edit filter settings" on page 584

Example



SAP Average Transaction Response Time Graph

This graph displays all the SAP-related transactions in the load test scenario.

X-axis	Elapsed time since the start of the run.
Y-axis	Average response time (in seconds) of each transaction
Breakdown graph	"SAP Average Dialog Step Response Time Breakdown Graph" on page 580
Tips	Select a transaction in one of the following ways: ➤ Select the transaction from the Break down transaction: list in the task pane. ➤ Highlight the transaction by selecting the line representing it in the graph. ➤ Select the transaction from the graph legend. This highlights the line in the graph.
See also	"SAP Breakdown Task Pane" on page 581 "SAP Secondary Graphs" on page 592 "Edit filter settings" on page 584

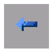
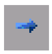
SAP Breakdown Task Pane

The task pane enables you to choose the level of breakdown you want to view, to filter and group transaction and server information, and to navigate backwards and forwards through the broken down graphs.

To access	**Session Explorer** > **Graphs** > **SAP Diagnostics** > **SAP Diagnostics - Guided Flow**
See also	"SAP Breakdown Task Pane" on page 581

SAP Breakdown Toolbar

User interface elements are described below

UI Elements (A-Z)	Description
	**Back.** Click to view previous breakdown graph, or to ungroup grouped data.
	**Next.** Click to view next breakdown graph.
	**Home.** Click to return to the initial SAP Average Transaction Response Time graph.
	**Help.** Click to get help on the breakdown options.

Breakdown Options

To break down SAP diagnostics data, choose the breakdown and filter options from the task pane.

User interface elements are described below:

UI Elements	Description
Break down transaction	Select a transaction from this list to display the average dialog step response time breakdown.
Break down server time into	Displays the breakdown options for the Average Dialog Step Response Time Breakdown graph. ➤ Select **Components** to view a breakdown of the transaction's server components, namely database time, interface time, application processing time, and system time. ➤ Select **Dialog Steps** to view a breakdown of the transaction's dialog steps.
Break down dialog step <dialog step>	Break down a dialog step into its server-time components, namely database time, interface time, application processing time, and system time.

UI Elements	Description
View data associated with <component>	Break down a server-time component (database time; interface time; application processing time; system time) to view data associated with it.
No available breakdown	There are no further breakdown options.
Apply	Click to apply the selected breakdown option.

Current filter settings

This section displays the filter/grouping settings of the graph currently displayed in the primary graph pane.

User interface elements are described below:

UI Elements	Description
From/To	Enter values (in hh:mm:ss) to filter the graph over a specified time interval.
Transaction	Displays the name of the transaction represented in the graph.
Dialog Step	Displays the name of the dialog step represented in the graph.
Server	Displays the name of the server represented in the graph.

Edit filter settings

Click this button to modify filter or grouping settings. When you click **Edit Filter Settings** the filter/grouping options become editable.

User interface elements are described below:

UI Elements	Description
Filter	Use this option to filter the current graph by time interval, transaction, dialog step, and/or server. ➤ **From/To.** Enter values (in hh:mm:ss) to filter the graph over a specified time interval. ➤ **By Transaction.** Filter the graph to display information about a specific transaction by selecting the transaction from the list. ➤ **By Dialog Step.** Filter the graph to display information about a specific dialog step by selecting the dialog step from the list. ➤ **By Server.** Filter the graph to display information about a server by selecting the server name from the list. **Note:** Only servers associated with the data displayed in the current graph are listed in the By Server list

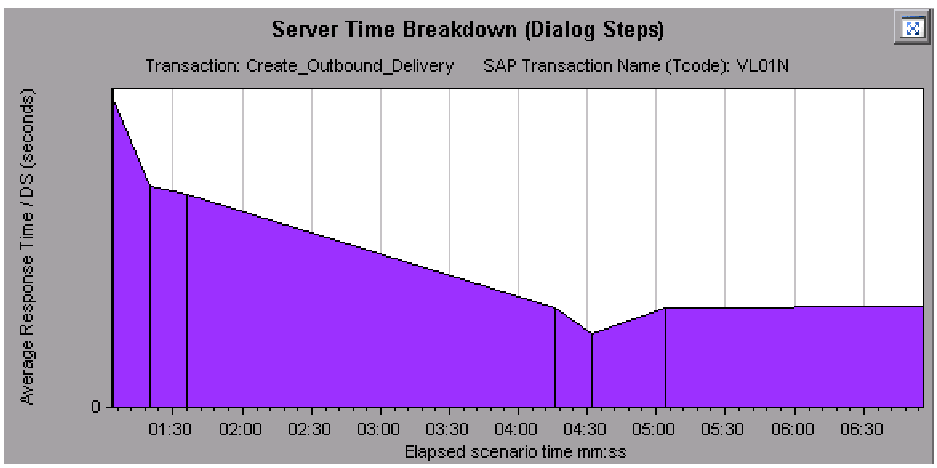
UI Elements	Description
Group	Use this option to group the data represented in the graph by transaction or by server. Select a transaction, component or subcomponent from the list. ➤ **By Transaction**. Select this check box to group by transaction. ➤ **By Server**. Select this check box to group by server. **Note:** After applying grouping to a graph, you need to ungroup the data in order to apply further breakdown options. To ungroup grouped data, click the **Back** button on the toolbar. **Important**! When you open a saved session, the **Back** is disabled. If you have grouped data, you need to click the Home button, or open a new SAP Diagnostics - Guided Flow tab to restart SAP breakdown.
OK	Click **OK** to apply the chosen filter/grouping settings. The **Current filter settings** area displays the chosen settings in non-editable mode. **Notes**: ➤ Global filtering is enabled when viewing SAP Diagnostics graphs (special SAP view) but cannot be applied on these graphs. ➤ Local filtering is disabled in the SAP Diagnostics - Guided Flow tab. To apply local filters to a SAP Diagnostics graph displayed in the Guided Flow tab, open the graph in a new tab by clicking the Enlarge Graph button.

SAP Server Time Breakdown (Dialog Steps) Graphs

This graph displays the dialog steps of a particular transaction.

X-axis	Elapsed time since the start of the run (in hh:mm:ss).
Y-axis	The average response time per dialog step (in seconds).
Breakdown graph	"SAP Server Time Breakdown Graph" on page 587
See also	"SAP Breakdown Task Pane" on page 581 "SAP Secondary Graphs" on page 592 "Edit filter settings" on page 584

Example

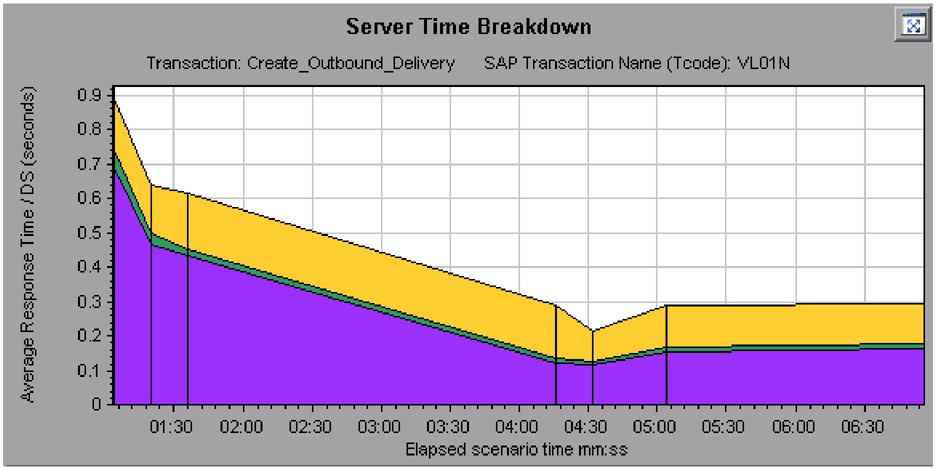


SAP Server Time Breakdown Graph

This graph represents the server-time components of a single transaction, namely database time, application processing time, interface time, and system time.

X-axis	Elapsed time since the start of the run (in hh:mm:ss).
Y-axis	Represents the average response time per dialog step (in seconds).
Breakdown graphs	➤ "SAP Database Time Breakdown Graph" on page 588 ➤ "SAP Application Processing Time Breakdown Graph" on page 579 ➤ "SAP System Time Breakdown Graph" on page 592 ➤ "SAP Interface Time Breakdown Graph" on page 591
Tips	In the task pane, select a component from the **View data associated with** box.
See also	"SAP Breakdown Task Pane" on page 581 "SAP Secondary Graphs" on page 592 "Edit filter settings" on page 584

Example

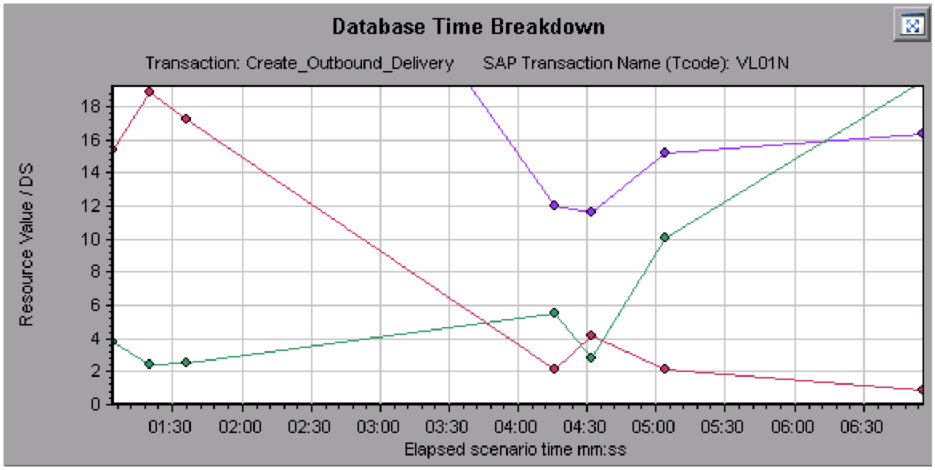


SAP Database Time Breakdown Graph

This graph displays the behavior of resources associated with database time, namely time taken to access a record, database time, and the number of records accessed per dialog step.

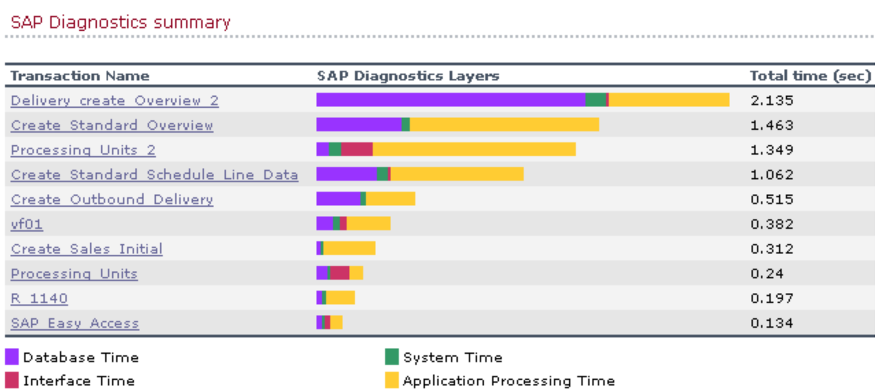
X-axis	Elapsed time since the start of the run (in hh:mm:ss).
Y-axis	Represents the resource value per dialog step (in msec).
Tips	You can open the graph in full view by clicking in the top right corner of the primary graph pane. An enlarged version of the graph opens in a new tab.
See also	"SAP Breakdown Task Pane" on page 581 "SAP Secondary Graphs" on page 592

Example



SAP Diagnostics Summary Report

This report displays a list of major alerts generated when opening the Analysis session, and a summary of the SAP diagnostics data.



To access	Use one of the following: ➤ **Session Explorer** > **Reports** > **Summary Report** > **Major Alerts** ➤ **Session Explorer** > **Reports** > **Summary Report** > **SAP Diagnostics Summary**
Note	If you do not see diagnostics data on the Summary Report, check if you are using a user-defined template. To view relevant data, choose a different template from the list of templates, or create and apply a new template. For more information about using templates, see "Template Dialog Box" on page 90.
See also	"SAP Diagnostics Graphs Overview" on page 566

SAP Diagnostics Summary

User interface elements are described below:

UI Elements	Description
Transaction	Individual transactions. You can click a transaction name to display the server time breakdown for that transaction.
SAP Diagnostics Layers	Relative server-time breakdown in layers. Click a layer to display data associated with the component.
Total time	Total usage time for each transaction.

Major Alerts

User interface elements are described below:

UI Elements	Description
Time Interval	The time during which the problem occurred.
Transaction/Server	Which transaction and server were involved.

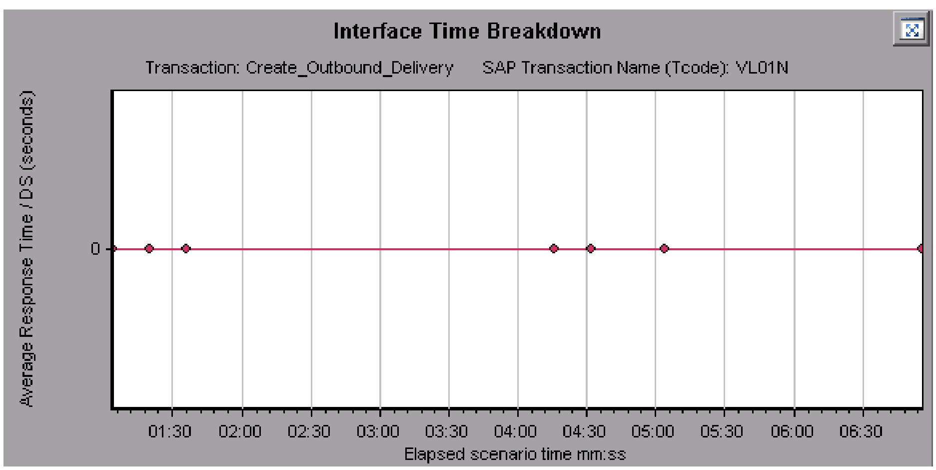
UI Elements	Description
Description	A description of the alert.
Action	This column provides a link to a graphic depiction of the problem.

SAP Interface Time Breakdown Graph

This graph displays the behavior of resources associated with interface time, namely GUI time, RFC time, and roll-wait time.

X-axis	Elapsed load test scenario time (in hh:mm:ss)
Y-axis	Average response time per dialog step (in seconds).
See also	"SAP Breakdown Task Pane" on page 581 "SAP Secondary Graphs" on page 592

Example

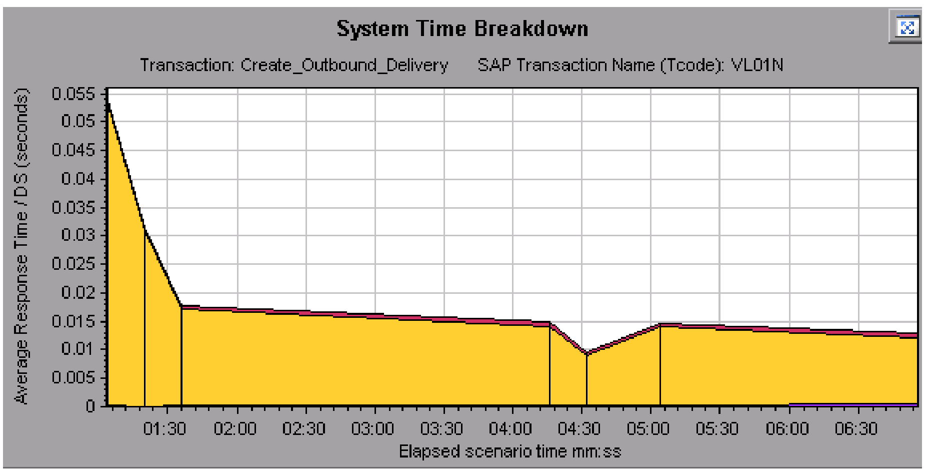


SAP System Time Breakdown Graph

This graph displays the behavior of the sub-components of the system time component, namely the dispatcher wait time, the load and generation time, and the roll-in and roll-out times.

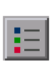
X-axis	Elapsed load test scenario time (in hh:mm:ss)
Y-axis	Average response time per dialog step (in seconds)
See also	"SAP Breakdown Task Pane" on page 581 "Secondary Graph Pane" on page 571

Example



SAP Secondary Graphs

The secondary graph pane of the SAP Diagnostics - Guided Flow tab displays graphs that support the graph displayed in the primary graph pane. You can correlate over time only one graph displayed in the secondary graph pane.



To see the legend for the graph displayed in this pane, click the **Graph Legend** button in the top right corner. To see all the data in the Legend, scroll along the horizontal scroll bar.

You can open the displayed graph in full view by clicking the **Enlarge Graph** button in the top right corner of this pane. An enlarged version of the graph opens in a new tab.

You view the following graphs in the secondary graph pane:

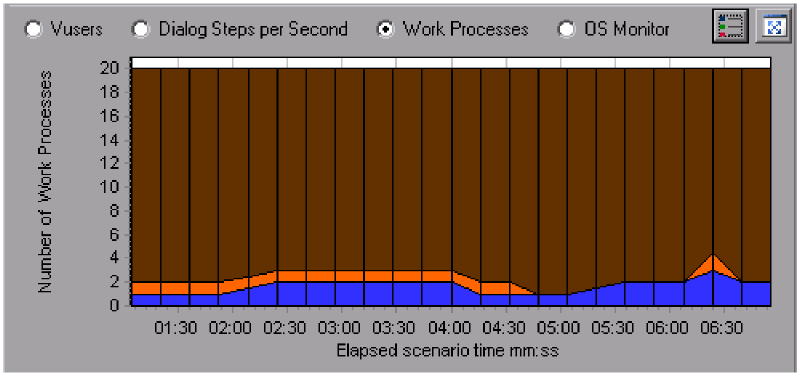
- Vuser Graphs.
- Dialog Steps per Second Graph
- Work Processes Graph
- OS Monitor Graph

Work Processes Graph

This graph represents the number and distribution of work processes that ran throughout the load test scenario.

X-axis	Elapsed scenario time (in hh:mm:ss).
Y-axis	Number of work processes.
Note	This graph is available only when a single server filter is applied.
See also	"SAP Breakdown Task Pane" on page 581 "Vuser Graphs" on page 233 "Dialog Steps per Second Graph" on page 574 "OS Monitor Graph" on page 574

Example


Vusers
Dialog Steps per Second
Work Processes
OS Monitor
Number of Work Processes
20
18
16
14
12
10
8
6
4
2
0
01:30 02:00 02:30 03:00 03:30 04:00 04:30 05:00 05:30 06:00 06:30
Elapsed scenario time mm:ss

34

J2EE & .NET Diagnostics Graphs

This chapter includes:

Concepts

Tasks

Reference

Concepts

J2EE & .NET Diagnostics Graphs Overview

The J2EE & .NET Diagnostics graphs in LoadRunner Analysis enable you to trace, time, and troubleshoot individual transactions and server requests through J2EE & .NET Web, application, and database servers. You can also quickly pinpoint problem servlets and JDBC calls to maximize business process performance, scalability, and efficiency.

The J2EE & .NET Diagnostics graphs are comprised of two groups:

➤ **J2EE & .NET Diagnostics Graphs.** These graphs show you the performance of requests and methods generated by virtual user transactions. They show you the transaction that generated each request.

➤ **J2EE & .NET Server Diagnostics Graphs.** These graphs show you the performance of all the requests and methods in the application you are monitoring. These include requests generated by virtual user transactions and by real users.

Tasks

How to Enable Diagnostics for J2EE & .NET

To generate Diagnostics for J2EE & .NET data, you must first install HP Diagnostics.

Before you can use HP Diagnostics with LoadRunner, you need to ensure that you have specified the Diagnostics Server details in LoadRunner. Before you can view Diagnostics for J2EE & .NET data in a particular load test scenario, you need to configure the Diagnostics parameters for that scenario. For more information about Configuring HP Diagnostics to work with LoadRunner, refer to the *HP LoadRunner Controller User Guide*.

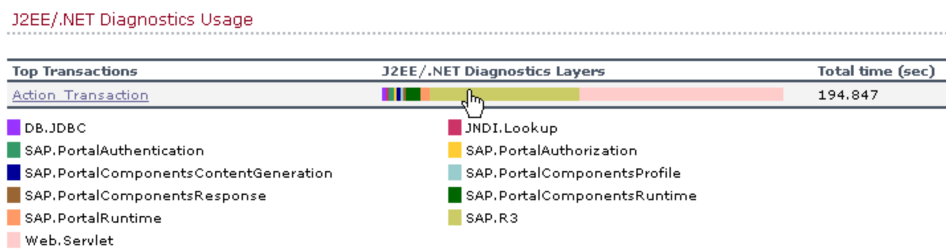
Note: To ensure that valid J2EE/.NET diagnostics data is generated during the scenario run, you must manually mark the beginning and end of each transaction in the Vuser script, rather than using automatic transactions.

Viewing J2EE to SAP R3 Remote Calls

The *Remote Function Call (*RFC) protocol in SAP allows communication to take place between SAP J2EE and SAP R3 environments. When remote calls take place between SAP J2EE and SAP R3 environments, Analysis displays information about the RFC functions, including the name of each function.

You view information about RFC functions by breaking down the SAP R3 layer. You can view the RFC function information in a graph display or in the Chain Of Calls window.

1 Go to the **J2EE/.Net Diagnostics Usage** section of the Summary Report. Next to the relevant transaction, click the color representing the **SAP.R3** layer.

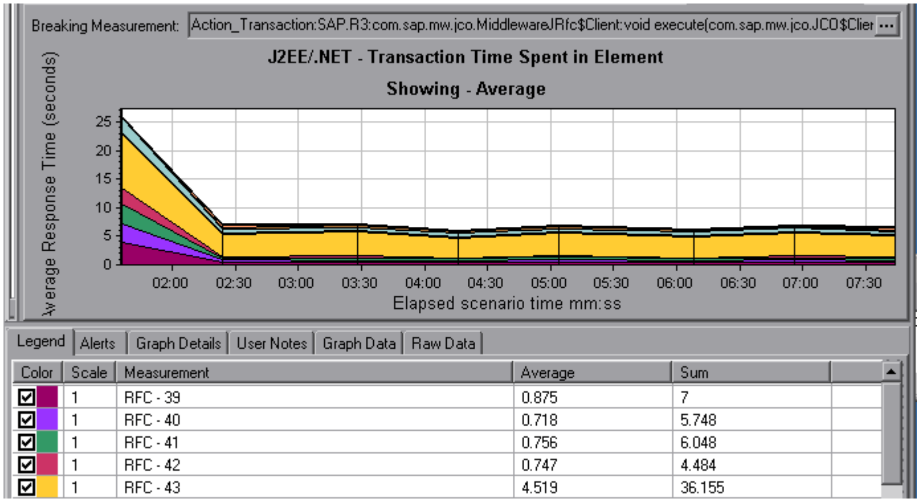


The J2EE/.NET - Transaction Time Spent in Element graph opens, representing the SAP.R3 layer.

2 Right click the graph and choose **J2EE/.NET Diagnostics** > **Break down the class to methods**.

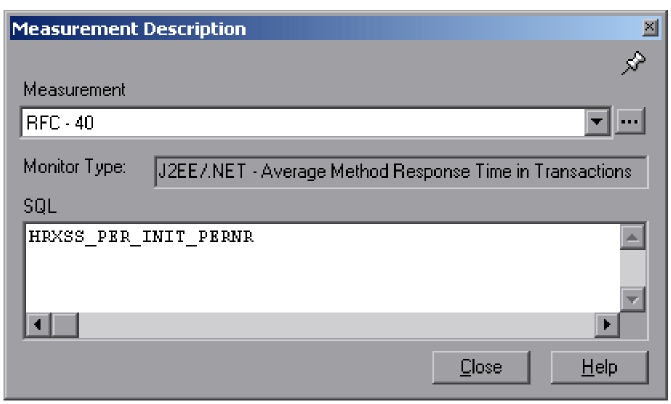
3 Break down the graph further by right clicking the graph and choosing **J2EE/.NET Diagnostics** > **Break down the method to SQLs**.

The graph is broken down into the different RFC functions.



4 To view the name of each RFC function, right click an RFC measurement in the **Measurement** column in the graph legend and choose **Show measurement description**.

The Measurement Description dialog box opens. The name of the RFC function is displayed in the **SQL** box.



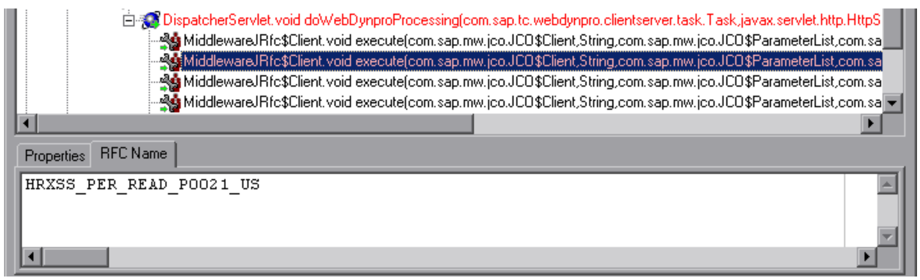
To view RFC function information in the Chain Of Calls window:

1 Go to the **J2EE/.Net Diagnostics Usage** section of the Summary Report. Next to the relevant transaction, click the color representing the **SAP.R3** layer.

The J2EE/.NET - Transaction Time Spent in Element graph opens, representing the SAP.R3 layer.

2 Right click the graph and choose **J2EE/.NET Diagnostics** > **Show chain of calls**.

The Transaction chain of calls window opens. When you click any of the RFC functions, in the **Measurement** column, the name of the function is displayed in the lower pane in the **RFC Name** tab.



Reference

J2EE & .NET Diagnostics Data

The J2EE & .NET Diagnostics graphs provide an overview of the entire chain of activity on the server side of the system. At the same time, you can break down J2EE/.NET layers into classes and methods to enable you to pinpoint the exact location where time is consumed. In addition, you can view custom classes or packages that you set the J2EE/.NET probe to monitor. You can also view the transaction chain of calls and call stack statistics to track the percentage of time spent on each part of the transaction.

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

This section also includes:

➤ "Example Transaction Breakdown" on page 601

➤ "Using the J2EE & .NET Breakdown Options" on page 607

➤ "Viewing Chain of Calls and Call Stack Statistics" on page 610

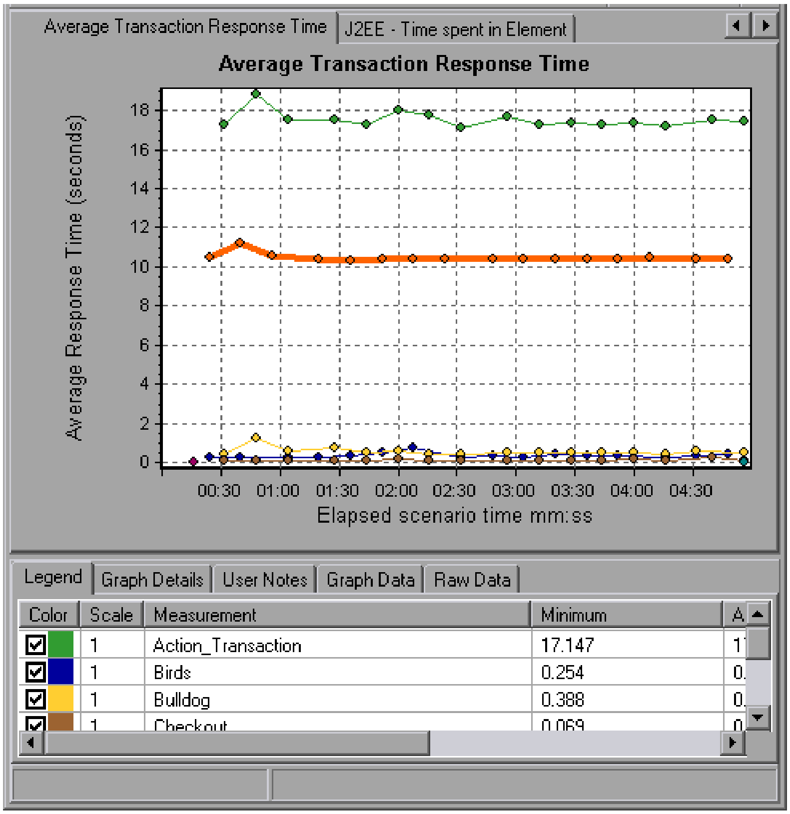
➤ "Understanding the Chain of Calls Window" on page 613

Example Transaction Breakdown

The following graphs illustrate the breakdown of a transaction to its layers, classes, and methods.

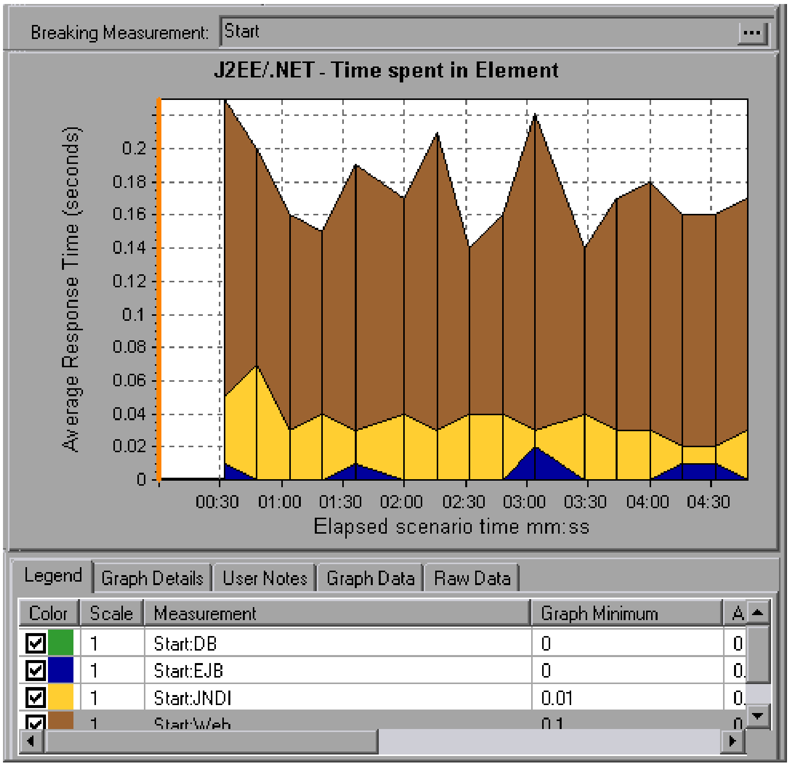
Transaction Level

The following figure displays the top level Average Transaction Response Time graph. The graph displays several transactions: **Birds**, **Bulldog**, **Checkout**, **Start**, etc.



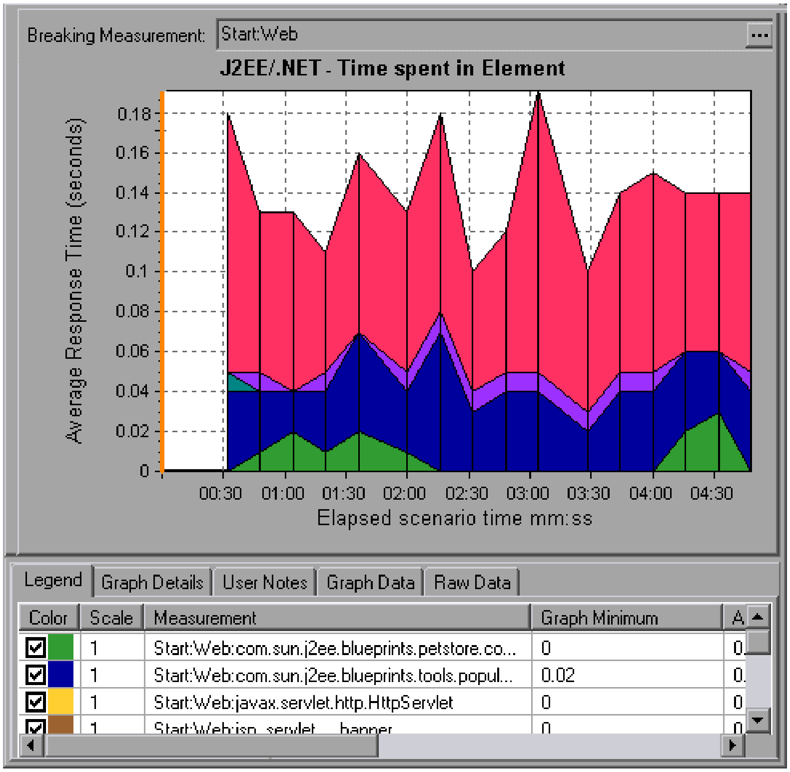
Layer Level

In the following figure, the **Start** transaction has been broken down to its layers (DB, EJB, JNDI, and Web). In J2EE/.NET transactions, the Web layer is generally the largest.



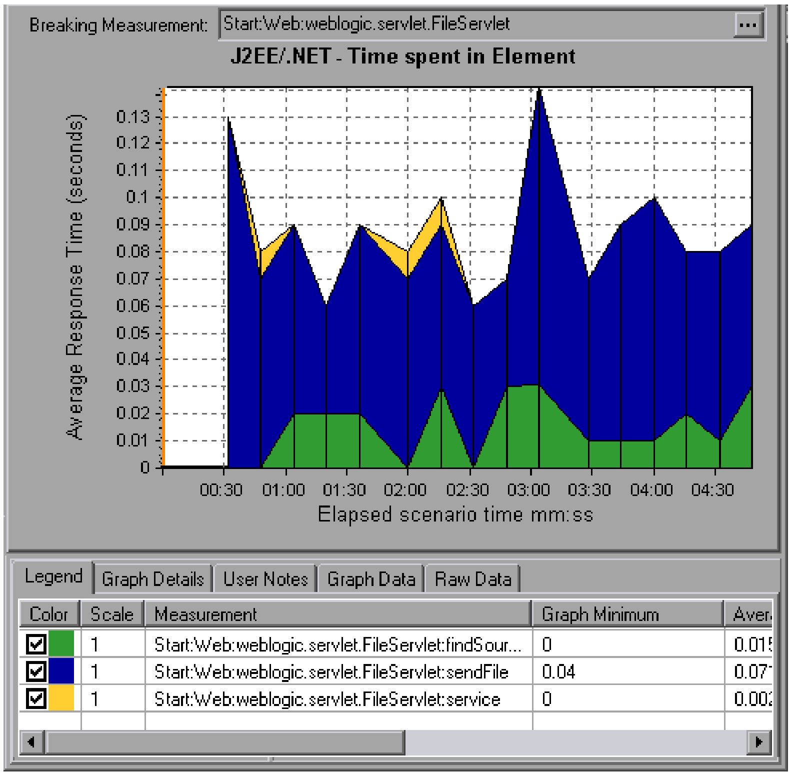
Class Level

In the following figure, the Web layer of the **Start** transaction has been broken down to its classes.



Method/Query Level

In the following figure, the **weblogic.servlet.FileServlet** component of the **Web** layer of the **Start** transaction has been broken down to its methods.



Note: Some JDBC methods can invoke SQLs which can be broken down further. In this case there is another level of breakdown, that is SQL Statements. For the methods that can not be further broken down into SQL statements when reaching this level of breakdown, you see **NoSql**.

Cross VM Analysis

When a server request makes a remote method invocation, the J2EE & .NET Diagnostics graphs display certain measurements relating to the classes and methods involved in these requests. These measurements are displayed at a layer, class and method level. The VM making the call is referred to as the *caller VM*, and the VM that executes the remote call is the *callee VM*.

The measurements are described below:

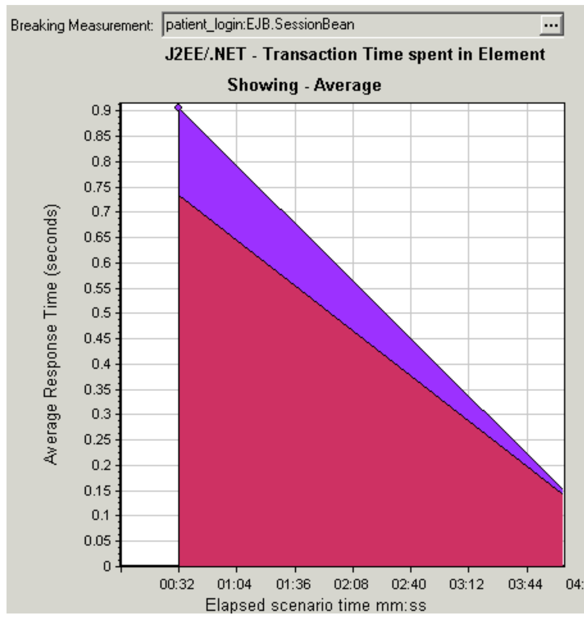
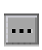
Measurements	Description
Cross VM Layer	A measurement that represents a dummy layer that integrates the data from the remote classes and methods in server requests that take place across two or more virtual machines.
Remote-Class	A measurement that represents a dummy class that integrates the data from the remote methods in server requests that take place across two or more virtual machines.
Remote-Class: Remote Method	A measurement that represents a dummy method. Remote-Class: Remote Method measures the total time, call count, exclusive latency, minimum and maximum values, standard deviation, and so on of the methods that are executed remotely, relative to the caller virtual machine.

Note: Since this data is measured on the caller virtual machine the exclusive latency will include all of the time required for making the remote method invocation such as network latency.

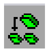
Using the J2EE & .NET Breakdown Options

J2EE & .NET breakdown options are described.

To access	Use one of the following to access breakdown options:" ➤ <**J2EE & .NET Graphs**> > **View** > **J2EE & .NET Diagnostics** ➤ <**J2EE & .NET Diagnostics Graphs**> > select transaction > short-cut menu > **J2EE & .NET Diagnostics** ➤ See toolbar options for each breakdown level
Notes	➤ The breakdown menu options and buttons are not displayed until an element (transaction, server request, layer) is selected. ➤ If there is no URI in the SQL, URI-None appears in front of the full measurement description in the Measurement Description dialog box.
See also	"J2EE & .NET Diagnostics Graphs Overview" on page 596

UI Elements (A-Z)	Description
<Right-click> transaction in Average Response TIme Graph	Choose J2EE/.NET Diagnostics > Show Server Requests. A new graph opens showing the breakdown of the selected transaction. The name of the transaction is displayed in the Breaking Measurement box.  You can view the full SQL statement for a selected SQL element by choosing **Show measurement description** from the Legend window right-click menu. The Measurement Description dialog box opens displaying the name of the selected measurement and the full SQL statement.
	To view transaction properties for the breakdown measurement, click the **Breaking Measurement** button. To disable this feature, choose **View** > **Display Options**, and clear the **Show Breaking Measurement** check box.

UI Elements (A-Z)	Description
	Select **View** > **J2EE**/**.NET Diagnostics** > **Break down the server request to layers**, or click the measurement breakdown button in the toolbar above the graph. **Note**: The option in the J2EE/.NET Diagnostics menu, and the tool tip of the measurement breakdown button, vary according to the element that you want to break down. For example, when you select a server request, the menu option and tool tip are **Break down server request to layers**.
	Select **View** > **J2EE**/**.NET Diagnostics** > **Show VM**, or click the **Show VM** button in the toolbar above the graph. This breaks the data down to the application host name (VM).
	Select **View** > **J2EE**/**.NET Diagnostics** > **Undo Break down the server request to layers**, or click the **Undo** <*Measurement Breakdown*>button in the toolbar above the graph. **Note:** The option in the J2EE/.NET Diagnostics menu, and the tool tip of the measurement breakdown button, vary according to the element whose breakdown you want to undo. For example, when you select a layer, the menu option and tool tip are Undo break down server request to layers.

UI Elements (A-Z)	Description
	Select **View** > **J2EE**/**.NET Diagnostics** > **Hide VM**, or click the **Hide VM** button in the toolbar above the graph.
	Display the chain of call or call stack statistics in the measurements tree window: Drag the orange time line on to the graph to the time specifying the end of the period for which you want to view data, and choose View > J2EE/.NET Diagnostics > Show Chain of Calls, or click the Show Chain of Calls button in the toolbar above the graph. **Note:** A measurement that is broken down in the Average Method Response Time in Transactions graph will be different from the same measurement broken down in the J2EE/.NET - Transaction Time Spent in Element graph. This is because the J2EE/.NET - Average Method Response Time in Transactions graph displays the average transaction time, whereas the J2EE/.NET - Transaction Time Spent in Element graph displays the average time per transaction event (sum of method execution time).

Viewing Chain of Calls and Call Stack Statistics

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

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

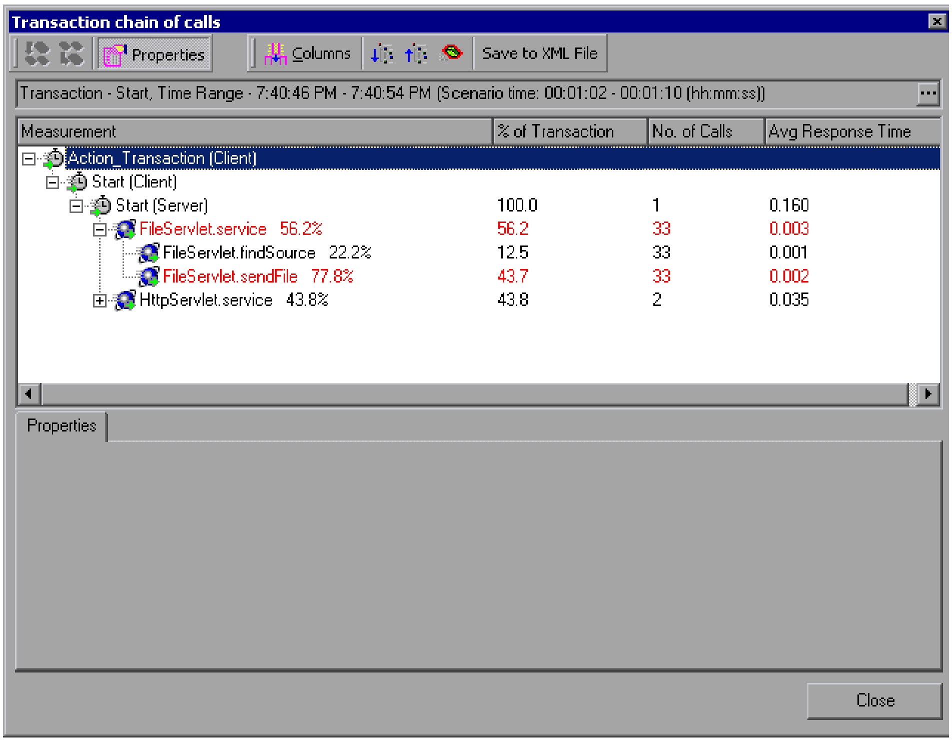
Chain of call and call stack statistics data are shown in the measurements tree window. The title of the window changes depending on which kind of data you are viewing.

- To set the point to which the measurements tree window relates, you must drag the orange time line to the desired spot.
- To view transaction call chains, right-click a component and choose **J2EE/.NET Diagnostics** > **Show Chain of Calls**. The Chain of Calls window opens displaying the chain of calls from the parent transaction downwards.

➤ To view method statistics, in the Chain of Calls window right-click a method and choose **Show Method Chain of Calls** or **Show Method Call Stack Statistics**.

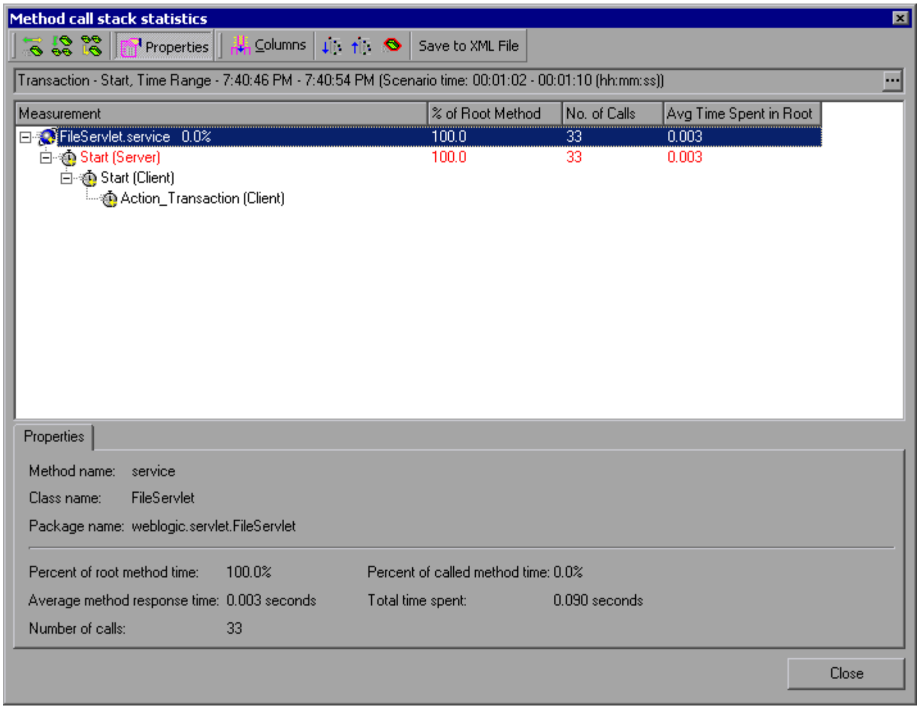
The Chain of Calls Windows

You use the Chain of Calls window to view the components that the selected transaction or method called. In the following figure, all the calls in the critical path of the Start server-side transaction are displayed.



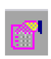
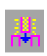
Note: Each red node signifies the most time consuming child of its parent.

You use the Call Stack Statistics window to view which components called the selected component. In the following figure, the **FileServlet.service** was called by Start (Server), which was called by Start (Client), and so on, down to the transaction at the bottom of the chain.



Understanding the Chain of Calls Window

User interface elements are described below:

UI Elements (A-Z)	Description
	**Switch to Method Chain of Calls.** When the call stack statistics data is displayed, displays the method chain of calls data (only if the root is a method).
	**Switch to Method Call Stack Statistics.** When the method chain of calls data is displayed, displays the method call stack statistics data (only if the root is a method)
	**Show Method Chain of Calls.** Displays the Chain of Calls window.
	**Show Method Call Stack Statistics.** Displays the Call Stack Statistics window.
	**Properties.** Hides or displays the properties area (lower pane).
	**Columns.** Enables you to select the columns shown in the Calls window. To display additional fields, drag them to the desired location in the Calls window. To remove fields, drag them from the Calls window back to the Columns chooser.
	**Expand All.** Expands the entire tree.
	**Collapse All.** Collapses the entire tree.
	**Expand Worst Path.** Expands only the parts of the path on the critical path.
Save to XML File	Saves the tree data to an XML file.
Method Properties	Area. Displays the full properties of the selected method.
SQL Query	Displays the SQL query for the selected method. (For Database only.)The following columns are available in the Chain of Calls window:

The following columns are available in the Chain of Calls window:

Column	Description
Measurement	Name of the method, displayed as **ComponentName:MethodName**. In the case of a database call, query information is also displayed. The percent shown indicates the percentage of calls to this component from its parent.
% of Root Method	Percentage of the total time of the method from the total time of the root tree item.
No of Calls	Displays the amount of times this transaction or method was executed.
Avg Response Time	Response time is the time from the beginning of execution until the end. Average response time is the total response time divided by the number of divided by the number of instances of the method.
STD Response Time	The standard deviation response time.
Min Response Time	The minimum response time.
Max Response Time	The maximum response time.
% of Caller	Displays the percentage of method time in relation the parent method time.
Total time	Displays the total method execution time, including the child execution time.

The following columns are available in the Call Stack Statistics window:

Column	Description
Measurement	Name of the method, displayed as **ComponentName.MethodName**. In the case of a database call, query information is also displayed. The percent shown indicates the percentage of calls to this component from its child.
% of Root Method	Percentage of the total time of the transaction (or method) from the total time of the root tree item.

Column	Description
No. of Calls to Root	Displays the amount of times this transaction or method was executed.
Avg Time Spent in Root	Time spent in root is the time that the sub-area spent in the root sub-area/area/transaction. Average Time Spent in Root time is the total time spent in the root divided by the number of instances of the method.
STD Time Spent in Root	The standard deviation time spent in the root.
Min Time Spent in Root	The minimum time spent in the root.
Max Time Spent in Root	The maximum time spent in the root.
% of Called	Displays the percentage o method time in relation the child method time.
Total Time Spent in Root	Displays the total method execution time, including the child execution time.

Graph Filter Properties

You can filter the J2EE & .NET Diagnostics graphs so that the displayed data is more suitable to your needs. You can filter using the following methods:

- Before opening a graph, enter filter criteria in the **Graph Properties** box of the **Open Graph** dialog box. For more information, see "Open a New Graphs Dialog Box" on page 43.
- From an open graph, enter filter criteria in the **Filter condition** fields in a filter dialog box. For more information, see "Filter Dialog Boxes" on page 109 and "Drilling Down in a Graph" on page 119.

User interface elements are described below:

UI Elements (A-Z)	Description
Class Name	Shows data for specified classes.
Layer Name	Shows data for specified layers.
Scenario Elapsed Time	Shows data for transactions that ended during the specified time.
SQL Logical Name	Shows data for specified SQL logical names. Due to the length of some SQL names, after you choose an SQL statement it is assigned a "logical name." This logical name is used in the filter dialog, legend, grouping, and other places in place of the full SQL statement. You can view the full SQL statement in the Measurement Description dialog box (**View > Show Measurement Description**).
Transaction Name - J2EE/.NET	Shows data for a specified transaction.

Some JDBC methods have the ability to invoke SQL's (each method can invoke several different SQL's) so there is another level of breakdown which is the SQL statements.

Note: For the methods that do not have SQL statement when reaching this level of breakdown you see **NoSql**.

J2EE & .NET Diagnostics User Interface

This section includes (in alphabetical order):

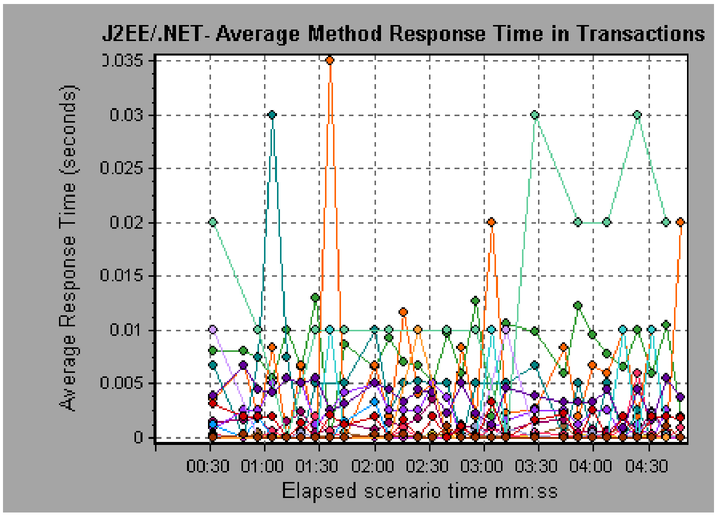
J2EE/.NET - Average Method Response Time in Transactions Graph

This graph displays the average response time for the server side methods, computed as Total Method Response Time/Number of Method calls. For example, if a method was executed twice by an instance of transaction A and once by another instance of the same transaction, and it took three seconds for each execution, the average response time is 9/3, or 3 seconds. The method time does not include calls made from the method to other methods.

X-axis	Elapsed time.
Y-axis	Average response time (in seconds) per method

Breakdown options	"Using the J2EE & .NET Breakdown Options" on page 607
See also	"J2EE & .NET Diagnostics Graphs Overview" on page 596

Example

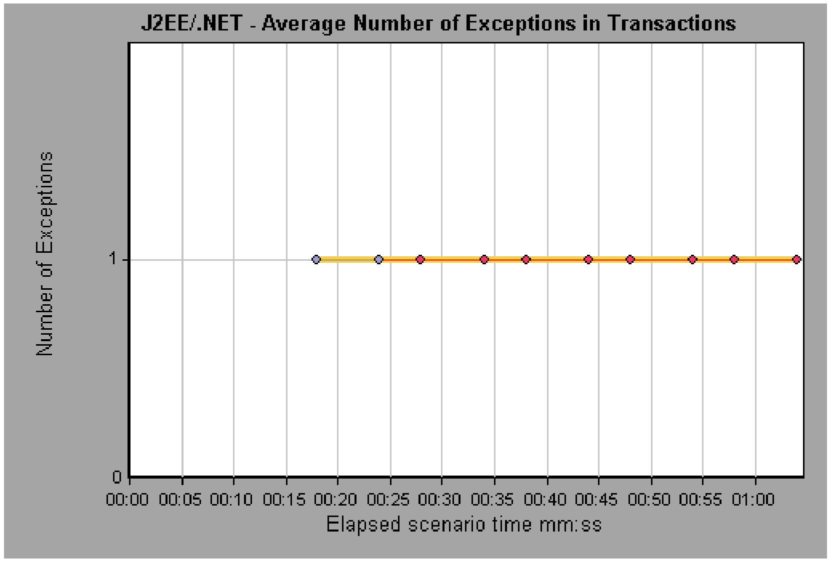


J2EE/.NET - Average Number of Exceptions in Transactions Graph

This graph displays the average number of code exceptions per method, transaction, or request that were monitored during the selected time range.

X-axis	Elapsed time.
Y-axis	Represents the number of events.
Breakdown options	To break the displayed elements down further, see "Using the J2EE & .NET Breakdown Options" on page 607.
See also	"J2EE & .NET Diagnostics Graphs Overview" on page 596

Example

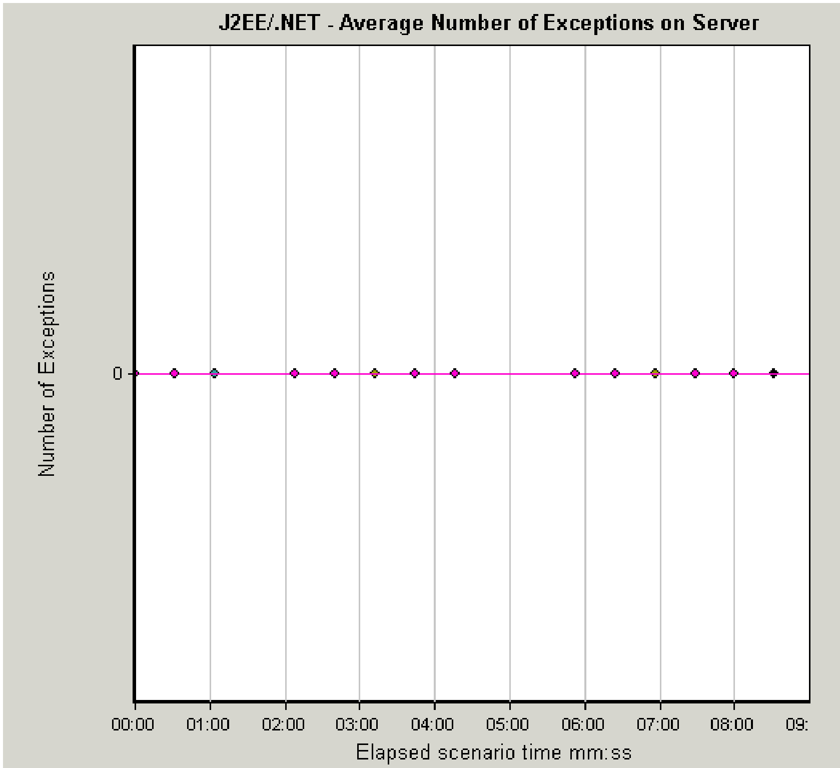


J2EE/.NET - Average Number of Exceptions on Server Graph

This graph displays the average number of code exceptions per method that were monitored during the selected time range.

X-axis	Elapsed time of the scenario run.
Y-axis	Number of events.
Breakdown options	"Using the J2EE & .NET Breakdown Options" on page 607
See also	"J2EE & .NET Diagnostics Graphs Overview" on page 596

Example

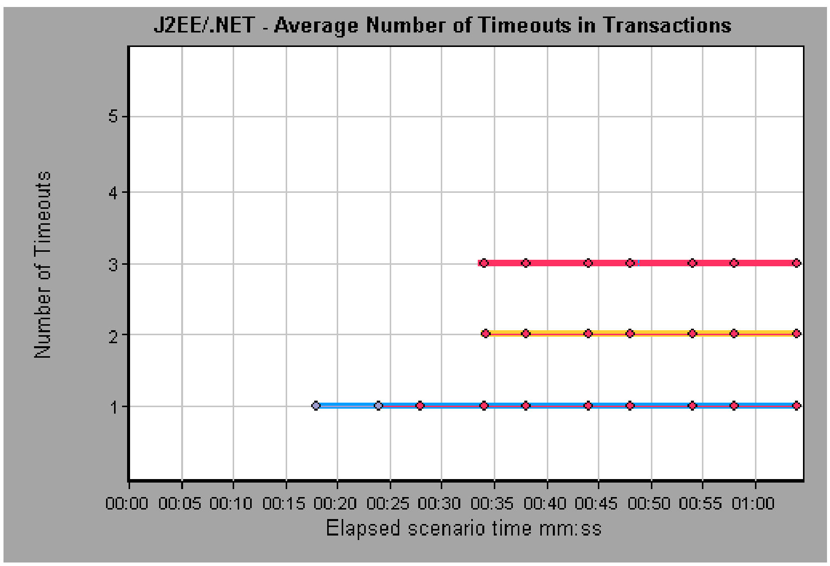

J2EE/.NET - Average Number of Exceptions on Server
Number of Exceptions
0
00:00
01:00
02:00
03:00
04:00
05:00
06:00
07:00
08:00
09:
Elapsed scenario time mm:ss

J2EE/.NET - Average Number of Timeouts in Transactions Graph

This graph displays the average number of timeouts per method, transaction, or request that were monitored during the selected time range.

X-axis	Elapsed time since the scenario run.
Y-axis	Represents number of events.
Breakdown options	"Using the J2EE & .NET Breakdown Options" on page 607
See also	"J2EE & .NET Diagnostics Graphs Overview" on page 596

Example

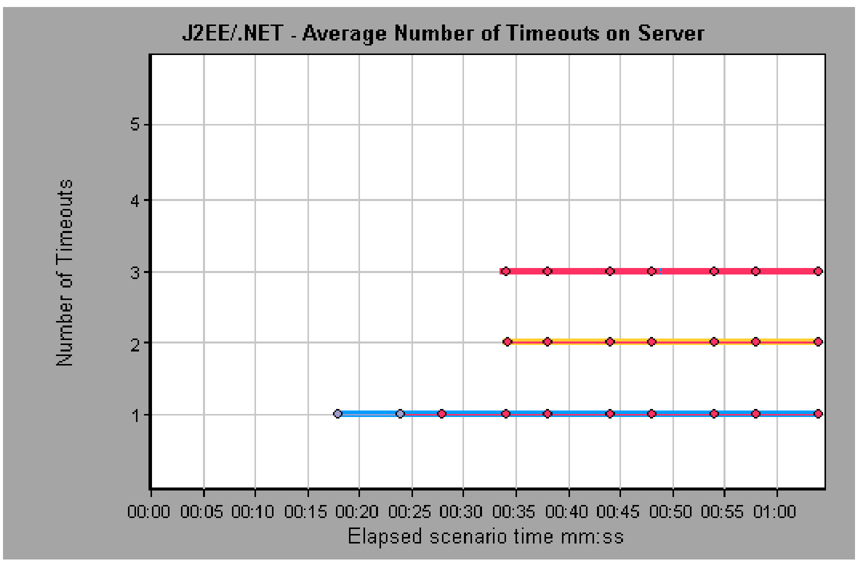


J2EE/.NET - Average Number of Timeouts on Server Graph

This graph displays the average number of timeouts per method that were monitored during the selected time range.

X-axis	Elapsed time since the scenario run.
Y-axis	Number of events.
Breakdown options	"Using the J2EE & .NET Breakdown Options" on page 607
See also	"J2EE & .NET Diagnostics Graphs Overview" on page 596

Example

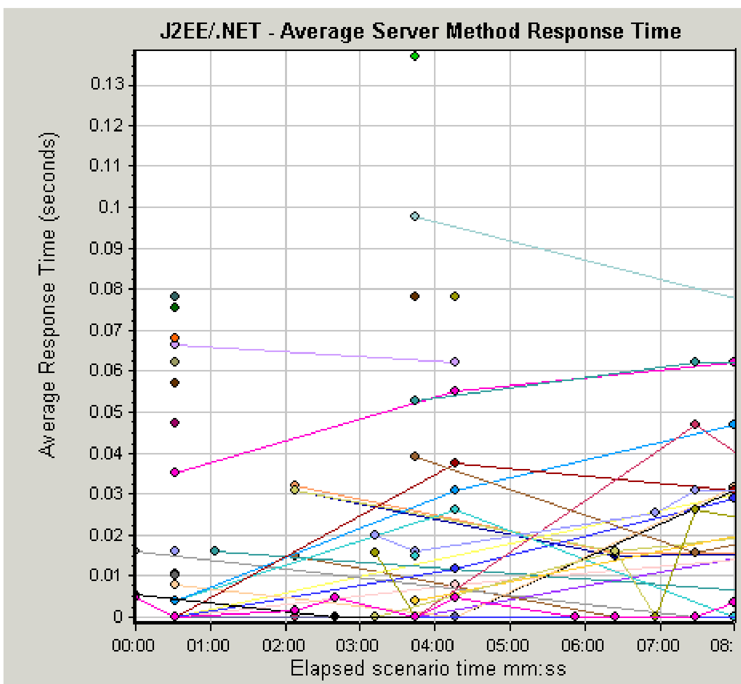


J2EE/.NET - Average Server Method Response Time Graph

This graph displays the average response time for the server side methods, computed as Total Method Response Time/Number of Method calls.

X-axis	Elapsed time since the scenario run.
Y-axis	Average response time (in seconds) per method.
Breakdown options	"Using the J2EE & .NET Breakdown Options" on page 607
Note	The method time does not include calls made from the method to other methods.
See also	"J2EE & .NET Diagnostics Graphs Overview" on page 596

Example

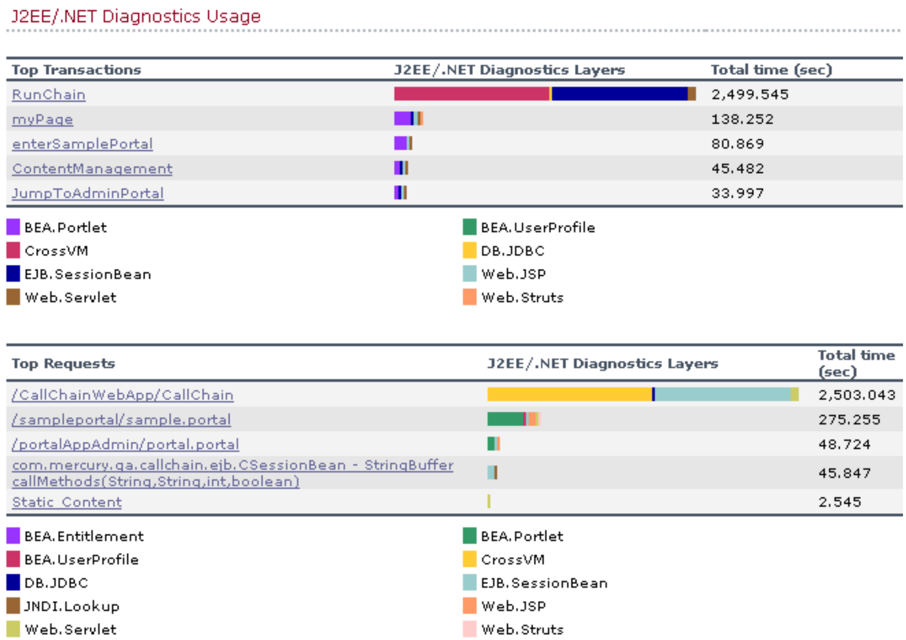


J2EE & .NET Diagnostics Graphs Summary Report

The J2EE & .NET Diagnostics Usage section of the Summary report provides general information about load test scenario execution and a usage chart for the J2EE & .NET Diagnostics and server request layers.

Breakdown options	The J2EE & >NET Diagnostics usage section breaks transactions and server requests into: ➤ Web server activity (Servlets and JSPs data) ➤ Application server activity (JNDIs) ➤ Back-end activity of database requests (JDBC methods and SQL queries) ➤ Total usage time for each transaction and request
Tips	Click on transaction In the J2EE/.NET Diagnostics Usage section of the Summary Report, click the transaction or J2EE /.NET layer on which you want to perform breakdown. The J2EE/.NET - Transaction Time Spent in Element graph or J2EE/.NET - Server Request Time Spent in Element graph opens. Click on layer Clicking a layer displays the specific layer breakdown in the transaction or server request. For more information, see "J2EE/.NET - Transaction Time Spent in Element Graph" on page 639 and "J2EE/.NET - Average Number of Timeouts on Server Graph" on page 622.
Note	If you do not see diagnostics data on the Summary Report, check if you are using a user-defined template. To view relevant data, choose a different template from the list of templates, or create and apply a new template. For more information about using templates, see "Template Dialog Box" on page 90.
See also	"J2EE & .NET Diagnostics Graphs Overview" on page 596

Example



J2EE/.NET - Method Calls per Second in Transactions Graph

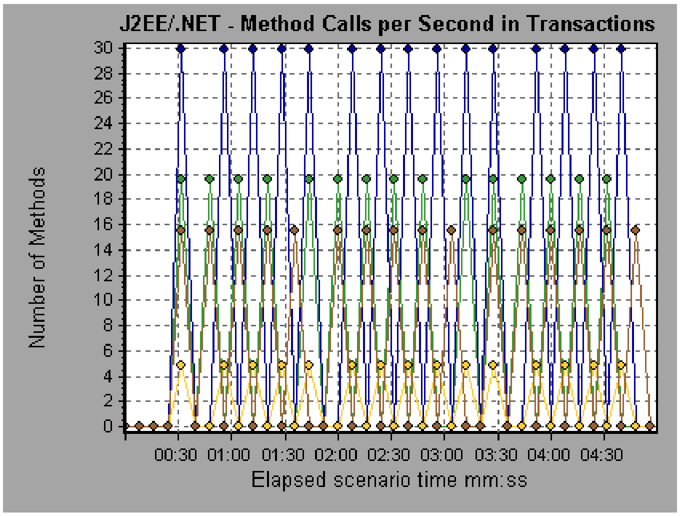
This graph displays the number of completed sampled transactions during each second of a load test scenario run.

The number of transactions included in the sample is determined by the sampling percentage set in the Diagnostics Distribution dialog box in the Controller (**Diagnostics** > **Configuration**). For more information, refer to the *HP LoadRunner Controller User Guide*.

X-axis	Elapsed time.
Y-axis	Represents the number of completed sampled transactions per second

Breakdown options	To break the displayed elements down further, see "Using the J2EE & .NET Breakdown Options" on page 607.
See also	"J2EE & .NET Diagnostics Graphs Overview" on page 596

Example



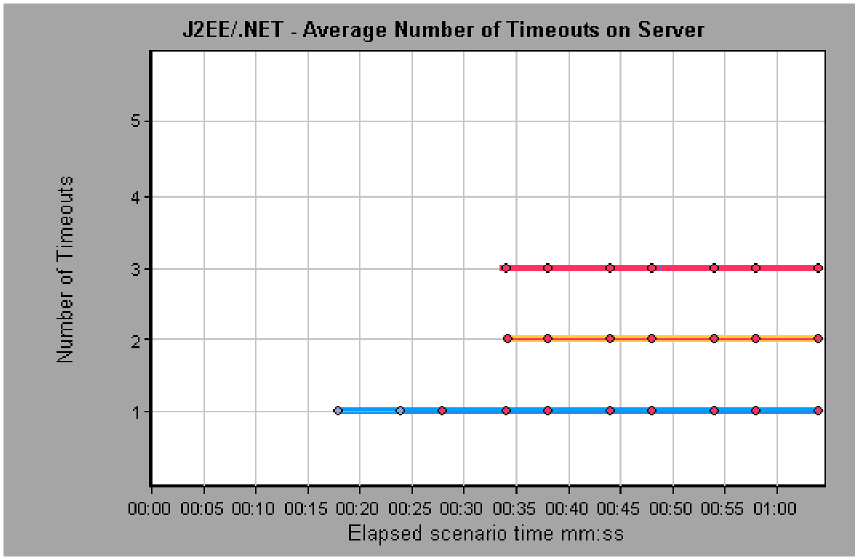
J2EE/.NET - Probes Metrics Graph

This graph displays performance metrics collected by HP Diagnostics probes. Metrics include JVM related data such as Heap usage and Garbage Collection, application server specific metrics, JDBC (Java Database Connectivity) metrics, and more.

X-axis	Elapsed time since the scenario run.
Y-axis	Resource usage. The following probe metric data is provided for offline analysis: ➤ **HeapUsed** ➤ **GC Collections/sec** ➤ **GC time Spent in Collections** To include additional Probe metric data in offline Analysis, you use the Diagnostics configuration file, **etc./offline.xml**. For more information, see the *HP Diagnostics Installation and Configuration Guide*.
Data Grouping	By default, the data in the graph is grouped by Category Name (the Diagnostics metric category name) and Probe Name. As a result, the default format for the measurement name is the graph is: <Name of metric from Diagnostics (unit of metric)>:<Diagnostics metric category name>:<Probe name> If the measurement unit is a count, no unit name is displayed in parentheses.

Important Information	By default, the following probe metric data is provided for offline analysis: **HeapUsed**, **GC Collections/sec**, and **GC time Spent in Collections**. To include additional Probe metric data in offline Analysis, you use the Diagnostics configuration file, **etc/offline.xml**. For more information, see the *HP Diagnostics Installation and Configuration Guide*. For example, for the following measurement name: ➤ the name of the metric is **GC Time Spent in Collections**. ➤ the value is measured as a percentage. ➤ the metric category name is **GC**. ➤ the Probe name is **MyJBossDev**. In addition to the regular Analysis filter criteria, you can also filter and group by the Diagnostics metrics collector name and the host name
Note	You need to synchronize the operating system time settings on the Controller machine and the Diagnostics Servers to ensure accurate display of the elapsed scenario time in the Probe Metrics graph.
See also	"J2EE & .NET Diagnostics Graphs Overview" on page 596

Example

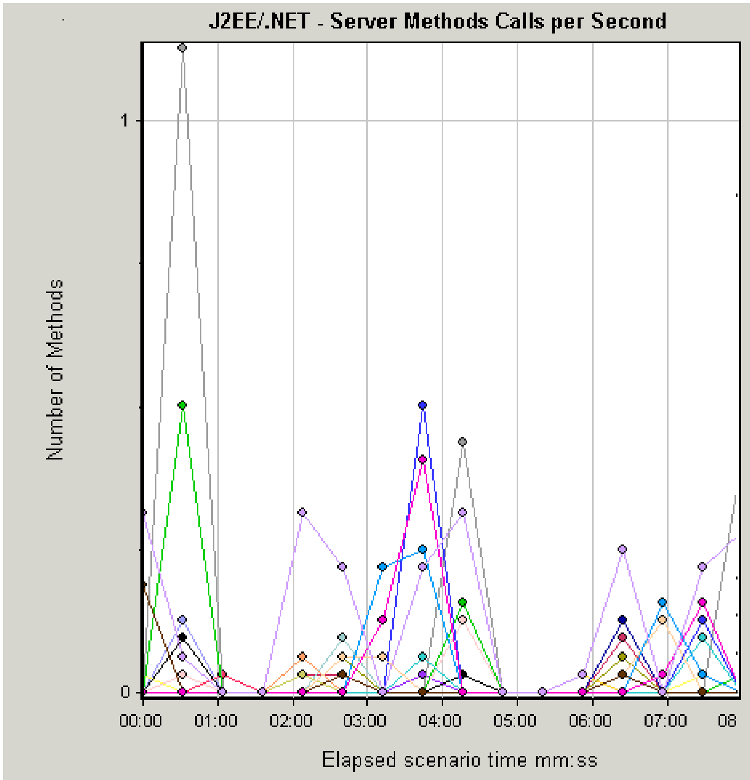


J2EE/.NET - Server Methods Calls per Second Graph

This graph displays the number of completed sampled methods during each second of a load test scenario run.

X-axis	Elapsed time of the scenario run.
Y-axis	Number of completed sampled methods per second.
Breakdown options	"Using the J2EE & .NET Breakdown Options" on page 607
Note	The number of methods included in the sample is determined by the sampling percentage set in the Diagnostics Distribution dialog box in the Controller (**Diagnostics** > **Configuration**). For more information, refer to the *HP LoadRunner Controller User Guide*.
See also	"J2EE & .NET Diagnostics Graphs Overview" on page 596

Example

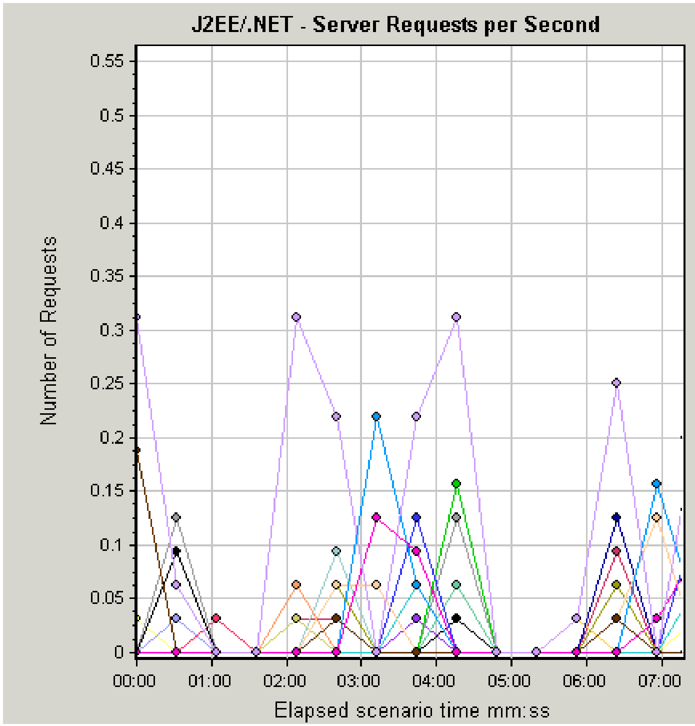

J2EE/.NET - Server Methods Calls per Second
Number of Methods
1
0
00:00
01:00
02:00
03:00
04:00
05:00
06:00
07:00
08
Elapsed scenario time mm:ss

J2EE/.NET - Server Requests per Second Graph

This graph displays the number of completed sampled requests during each second of a load test scenario run.

X-axis	Elapsed time of the scenario run.
Y-axis	Number of completed sampled requests per second.
Breakdown options	"Using the J2EE & .NET Breakdown Options" on page 607
Note	The number of requests included in the sample is determined by the sampling percentage set in the Diagnostics Distribution dialog box in the Controller (**Diagnostics** > **Configuration**). For more information, refer to the *HP LoadRunner Controller User Guide*.
See also	"J2EE & .NET Diagnostics Graphs Overview" on page 596

Example

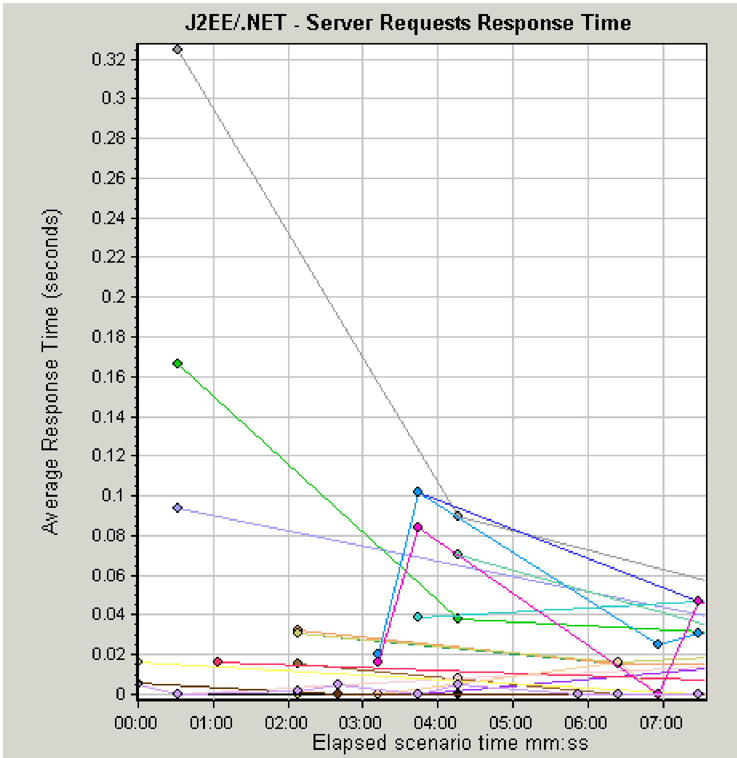

J2EE/.NET - Server Requests per Second
Number of Requests
0.55
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
00:00
01:00
02:00
03:00
04:00
05:00
06:00
07:00
Elapsed scenario time mm:ss

J2EE/.NET - Server Request Response Time Graph

This graph displays the server response time of requests that include steps that cause activity on the J2EE/.NET backend.

X-axis	Elapsed time of the scenario time.
Y-axis	Average time (in seconds) taken to perform each request.
Breakdown options	"Using the J2EE & .NET Breakdown Options" on page 607
Note	The reported times, measured from the point when the request reached the Web server to the point it left the Web server, include only the time that was spent in the J2EE/.NET backend.
See also	"J2EE & .NET Diagnostics Graphs Overview" on page 596

Example

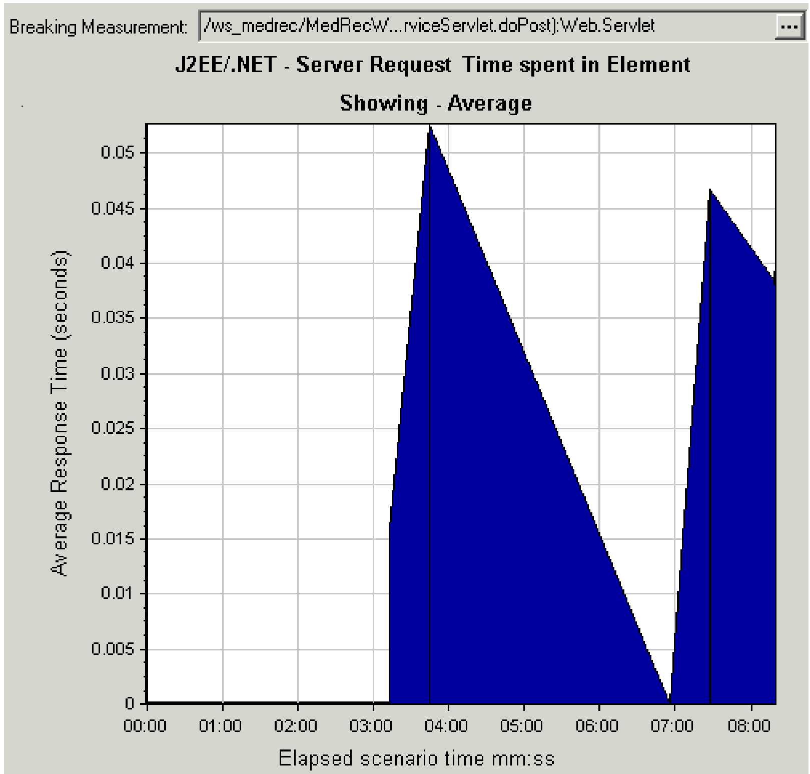

J2EE/.NET - Server Requests Response Time
Average Response Time (seconds)
0.32
0.3
0.28
0.26
0.24
0.22
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
00:00
01:00
02:00
03:00
04:00
05:00
06:00
07:00
Elapsed scenario time mm:ss

J2EE/.NET - Server Request Time Spent in Element Graph

This graph displays the server response time for the selected element (layer, class, or method) within each server request.

Purpose	The time is computed as Total Response Time/Total Number of Server Requests. For example, if a method was executed twice by an instance of server request A and once by another instance of the same server request, and it took three seconds for each execution, the average response time is 9/2, or 4.5 seconds. The server request time does not include the nested calls from within each server request.
X-axis	Elapsed time of the scenario run.
Y-axis	Average response time (in seconds) per element within the server request.
Breakdown options	"Using the J2EE & .NET Breakdown Options" on page 607
Filtering properties	The display of the graph is determined by the Graph Properties selected when the graph is opened, as described: None ➤ Time spent in each server request Server request ➤ Filtered by server request. Grouped by layer. Server request and layer ➤ Filtered by server request and layer. Grouped by class. Server request, layer, and class ➤ Filtered by server request, layer, and class. Grouped by method.
Tips	To obtain data for this graph, you must first install HP Diagnostics. Before you can view Diagnostics for J2EE & .NET data in a particular load test scenario, you need to configure the Diagnostics parameters for that scenario, as described in the *HP LoadRunner Controller User Guide*.
See also	"J2EE & .NET Diagnostics Graphs Overview" on page 596

Example



Breaking Measurement:
/ws_medrec/MedRecW...rviceServlet.doPost):Web.Servlet
J2EE/.NET - Server Request Time spent in Element
Showing - Average
Average Response Time (seconds)
0.05
0.045
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
00:00
01:00
02:00
03:00
04:00
05:00
06:00
07:00
08:00
Elapsed scenario time mm:ss

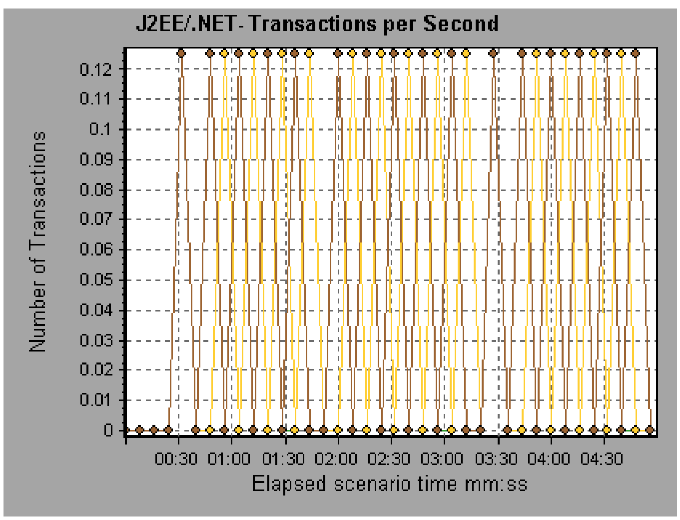
J2EE/.NET - Transactions per Second Graph

This graph displays the number of completed sampled transactions during each second of a load test scenario run.

The number of transactions included in the sample is determined by the sampling percentage set in the Diagnostics Distribution dialog box in the Controller (**Diagnostics** > **Configuration**). For more information, refer to the *HP LoadRunner Controller User Guide*.

X-axis	Elapsed time.
Y-axis	Number of completed sampled transactions per second
Breakdown options	To break the displayed elements down further, see "Using the J2EE & .NET Breakdown Options" on page 607.
See also	"J2EE & .NET Diagnostics Graphs Overview" on page 596

Example

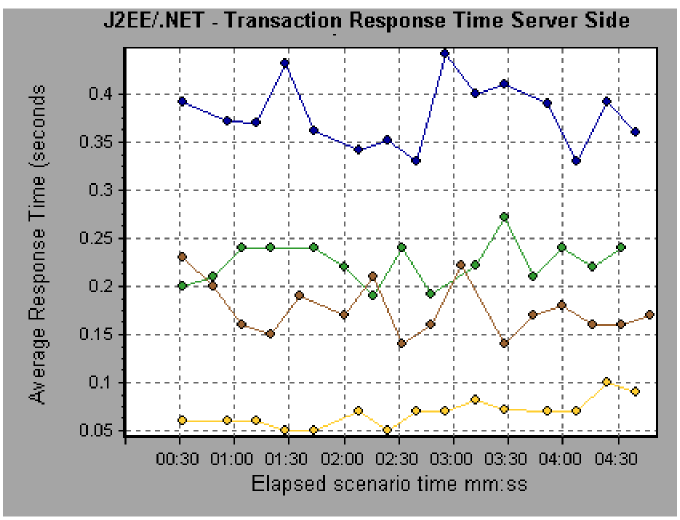


J2EE/.NET - Transaction Response Time Server Side Graph

This graph displays the transaction server response time of transactions that include steps that cause activity on the J2EE/.NET backend. The reported times, measured from the point when the transaction reached the Web server to the point it left the Web server, include only the time that was spent in the J2EE/.NET backend.

X-axis	Elapsed time.
Y-axis	Average response time (in seconds) of each transaction
Breakdown options	"Using the J2EE & .NET Breakdown Options" on page 607
See also	"J2EE & .NET Diagnostics Graphs Overview" on page 596

Example

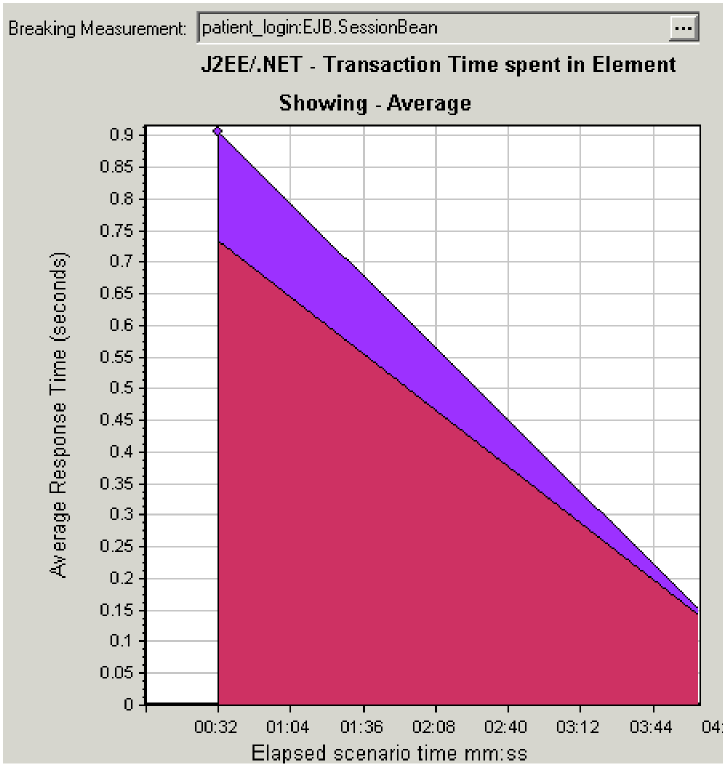


J2EE/.NET - Transaction Time Spent in Element Graph

This graph displays the server response time for the selected element (layer, class, or method) within each transaction.

X-axis	Elapsed time.
Y-axis	Average response time (in seconds) per element within the transaction.
Breakdown options	The display of graph data is determined by the graph properties selected when the graph was opened, as described in the following table: For information on filtering on graph data, see "Graph Data Display" on page 640 You can break down the displayed elements. For more information, see "Using the J2EE & .NET Breakdown Options" on page 607.
Tips	To obtain data for this graph, you must enable the J2EE & .NET Diagnostics module (from the Controller) before running the load test scenario.
Note	The time is computed as Total Response Time/Total Number of Transactions. For example, if a method was executed twice by an instance of transaction A and once by another instance of the same transaction, and it took three seconds for each execution, the average response time is 9/2, or 4.5 seconds. The transaction time does not include the nested calls from within each transaction.
See also	"J2EE & .NET Diagnostics Graphs Overview" on page 596 "Filtering and Sorting Graph Data" on page 93

Example



Graph Data Display

If you filter by these properties...	The graph data is displayed like this
None	Time spent in each transaction.
Transaction	Filtered by transaction. Grouped by layer.
Transaction and layer	Filtered by transaction and layer. Grouped by class.
Transaction, layer, and class	Filtered by transaction, layer, and class. Grouped by method.

Index

H

T

U

V

W

X

Y