



Console User's Guide

Version 12.0

RADVIEW

The software supplied with this document is the property of RadView Software and is furnished under a licensing agreement. Neither the software nor this document may be copied or transferred by any means, electronic or mechanical, except as provided in the licensing agreement. The information in this document is subject to change without prior notice and does not represent a commitment by RadView Software or its representatives.

WebLOAD Console User's Guide

© Copyright 2018 by RadView Software. All rights reserved.

November, 2018, RadView Publication Number WL-OSSPRO-1013-CUG10

WebLOAD, TestTalk, Authoring Tools, ADL, AppletLoad, and WebExam, are trademarks or registered trademarks of RadView Software IBM, and OS/2 are trademarks of International Business Machines Corporation. Microsoft Windows, Microsoft Windows NT, Microsoft Word for Windows, Microsoft Internet Explorer, Microsoft Excel for Windows, Microsoft Access for Windows and Microsoft Access Runtime are trademarks or registered trademarks of Microsoft Corporation. SPIDERSESSION is a trademark of NetDynamics. UNIX is a registered trademark of AT&T Bell Laboratories. Solaris, Java and Java-based marks are registered trademarks of Sun Microsystems, Inc. HP-UX is a registered trademark of Hewlett-Packard. SPARC is a registered trademark of SPARC International, Inc. Netscape Navigator and LiveConnect are registered trademarks of Netscape Communications Corporation. Any other trademark name appearing in this book is used for editorial purposes only and to the benefit of the trademark owner with no intention of infringing upon that trademark.

For product assistance or information, contact:	
Toll free in the US: Fax: World Wide Web	1-888-RadView +1-908-864-8099 www.RadView.com
North American Headquarters:	International Headquarters:
RadView Software Inc. 991 Highway 22 West, Suite 200 Bridgewater, NJ 08807 Email: info@RadView.com Phone: 908-526-7756 Fax: 908-864-8099 Toll Free: 1-888-RadView	RadView Software Ltd. 13 Haamal Street, Park Afek Rosh Haayin, 4809249, Israel Email: info@RadView.com Phone: +972-3-915-7060 Fax: +972-3-915-7011

Table of Contents

Chapter 1. Introduction.....	1
WebLOAD Documentation.....	1
Icons and Typographical Conventions.....	3
Where to Get More Information.....	4
Online Help	4
Technical Support Website.....	4
Technical Support.....	4
Chapter 2. Introducing WebLOAD	5
Chapter 3. Getting Started.....	7
Welcome to WebLOAD	7
How Does WebLOAD Work?.....	8
WebLOAD Architecture	9
What is the Product of Your WebLOAD Testing?	10
HTTP Client.....	11
WebLOAD Features	12
How are Tests Defined?.....	18
Configuring WebLOAD	19
Required Software for each Load Session Component	20
Example of a Test Configuration.....	20
Sharing a script	21
Working with a Firewall.....	22
Managing the Communication Between the Systems in your Load Session.....	24
Working with TestTalk	24
Pinging a Host.....	26
Troubleshooting Communications	27
Chapter 4. WebLOAD Quick Start	29
Quick Start Steps.....	29

Creating a Script	29
Configuring a Load Template using WebLOAD Wizard.....	34
Opening WebLOAD Wizard.....	34
Selecting a Script.....	34
Selecting a Host.....	35
Scheduling the Test	35
Selecting the Performance Measurements	36
Setting Session Options	38
Running the Test.....	38
Analyzing Test Results	39
Viewing the Results in Real Time	40
Creating Integrated Reports.....	40
Viewing Results with WebLOAD Analytics.....	41
Chapter 5. WebLOAD Console Features	43
The WebLOAD Console	43
Opening the Console.....	44
Console Elements	45
Title Bar	45
Console System Button.....	45
Quick Access Toolbar.....	47
Ribbon	47
Right-click Menus.....	48
Status Bar	48
The Session Tree	48
The Results Window	60
Event Viewer	66
Log Window.....	66
The WebLOAD Console Startup Dialog Box.....	67
Edit Mode and Run Modes of Operation.....	68
The Console in Edit Mode.....	68
The Console in Run Mode.....	69
The Console Ribbon Options	70
File Tab.....	70
Home Tab	71
Session Tab	72
Tools Tab.....	74
Help Drop Down.....	76
About Button.....	76
Chapter 6. Creating Load Templates.....	77
WebLOAD Console Options for Creating Load Templates.....	77

Chapter 7. Planning a Test Session	79
Load Session Workflow	79
Forming a Test Plan.....	81
Planning a Test Program	82
Creating the Test Scripts.....	82
Examining Your Test Resources.....	84
Starting WebLOAD Console	84
Chapter 8. Connecting to WebRM.....	87
How Does WebRM Work.....	88
Requesting Resources from WebRM	88
Updating the Number of Requested Resources.....	89
Chapter 9. Managing Load Templates and Load Sessions	91
Managing Load Templates.....	91
Creating a New Load Template	92
Saving a Load Template	92
Opening a Saved Load Template	92
Saving Additional Information with the Load Template	93
Creating a Mix.....	95
Managing Load Sessions	96
Saving a Load Session.....	96
Saving Additional Information for the Load Session.....	96
Chapter 10. Creating Load Templates with the WebLOAD Wizard	99
Using the WebLOAD Wizard to Create Load Templates.....	99
The WebLOAD Wizard Workflow	100
Opening the WebLOAD Wizard	100
Selecting a script or Mix.....	102
Selecting a Script.....	103
Selecting a Mix	104
Creating a Mix.....	105
Selecting Host Computers.....	107
Load Machines	108
Probing Client Machines	108
Setting Up Cloud Computers	108
Adding Host Computers	111
Designating a Host Computer as a Load Machine	112
Designating all Host Computers as Load Machines	112
Designating a Host Computer as a Probing Client Machine.....	112

Replacing a Load Machine or Probing Client Machine	113
Deleting a Host Computer	113
Scheduling the Load Session.....	114
Collectively Distributing the Load	115
Individually Scheduling the Load.....	118
Scheduling Models.....	121
Selecting the Performance Measurements	130
Completing the WebLOAD Wizard	131
After Completing the WebLOAD Wizard	132
Adding a New Script/Mix to the Template.....	132
Editing the Load Template.....	132
Viewing the WebLOAD Wizard Default Report	132
Modifying the Host Selection	133
Modifying the Load Schedule.....	133
Chapter 11. Creating Load Templates with Goal-Oriented Test.....	135
Using the Goal-Oriented Test	135
How does the Goal-Oriented Test Work?	136
The Goal-Oriented Test Wizard Workflow	137
Opening the Goal-Oriented Test Wizard	138
Selecting a Script or Mix	139
Running Single Scripts.....	140
Running a Mix of Scripts	141
Defining a Mix	142
Selecting Performance Measurements.....	145
Selecting Host Computers.....	146
Adding Host Computers to the Host Computers List	147
Designating a Host Computer as a Load Machine	148
Designating all Host Computers as Load Machines	149
Designating a Host Computer as a Probing Client Machine.....	149
Deleting a Host Computer	149
Configuring Probing Clients.....	150
Defining Performance Goals.....	151
About the Goal Definition Dialog Box.....	151
Adding Performance Goals.....	153
Deleting a Goal.....	157
Setting and Using Defaults.....	157
Defining the Acceleration Rate	158
Defining WebLOAD Console's Action Once the Goals are Reached.....	160
Finishing the Goal-Oriented Test Wizard	161

Viewing the Goal-Oriented Test Default Report.....	162
--	-----

Chapter 12. Configuring a Load Template Manually..... 165

The Manual Test Configuration Workflow	165
Starting a New Template Manually.....	166
Selecting a Script.....	166
Creating Load Templates with One or More Single Scripts	167
Selecting Hosts.....	168
Opening the Host Selection Dialog Box	168
Defining Host Computers	169
Defining Load Machines.....	170
Defining Probing Client Machines.....	170
Deleting a Host Computer	171
Scheduling the Test Session	171
Creating a Load Schedule Manually.....	171
Creating a Load Schedule with the Load Profiler.....	173
Completing Your Manual Load Template Configuration.....	175

Chapter 13. Setting Global Options..... 177

How to Set Global Options	177
The General Tab.....	179
The Export Tab.....	183
The File Locations Tab	186
The Data Drilling Tab	189
The Advanced Tab	193
The Database Tab	194
The Notification Mail Tab	196
The Dynatrace Tab	197
The Defect Tracking Tab.....	198
Tracking Defects	199
The SLA Manager.....	204
Accessing the SLA Manager	204
Configuring the SLA Manager Rules and Actions	204

Chapter 14. Setting Script Options..... 211

About Script Options	211
Modifying Script Options.....	213
Setting the SSL Options	213
Setting Sleep Time Playback Settings	215
Configuring Round Termination Settings	217

Setting the Browser Cache.....	219
Configuring Browser Parameter Settings	221
Setting the Browser Type and Version	223
Setting Load Generator and Probing Client Max Connections.....	224
Setting Redirection Limits	225
Simulating DNS Cache	225
Simulating SSL Cache	226
Enabling Persistent Connection (Keep-Alive)	226
Enabling GZip Support.....	227
Setting Character Encoding.....	228
Setting the HTTP Parameters.....	229
Setting Authentication	232
Bandwidth Settings	234
Enabling Network Emulation.....	237
Installing ipfw on a Windows Load Generator.....	237
Installing ipfw on a Linux CentOS 7 Load Generator	238
Diagnostic Settings	240
Enabling Syntax Checking	242
Log Report Diagnostics.....	243
Enabling enhanced RadView Support Diagnostic.....	244
Enabling Show Performance Warnings.....	244
Functional Testing Options.....	245
Setting Automatic Data Collection.....	246
Setting Page Time Verifications.....	247
Inserting a Text Verification Test	248
Pass / Fail Definition Settings	249
Configuring Reset Mechanism Settings	250
Configuring the Java Options	252
Chapter 15. Functional Verification Testing	255
The Functional Testing Tab.....	255
Opening the Functional Testing Tab.....	256
Selecting the Tests to Perform.....	258
User-Defined Transactions and Verification	259
Including a User-Defined Transaction in your script.....	259
User-Defined Transactions and Verification Syntax	259
User-Defined Transaction Example	260
Verification Function Return Values	261
Reporting Events	262
Tracking Reasons for Transaction Failure.....	262
Viewing Verification Test Results	264

Viewing Error Information in the Log Window	265
Viewing Error Information in the Event Viewer.....	266
Printing the Contents of the Log Window	267
Saving the Contents of the Log Window	267
Data Drilling.....	267
Chapter 16. Running Tests and Analyzing Test Results.....	269
Explanation of Creation and Analysis of Performance Reports	269
Chapter 17. Running a Load Session	271
Starting a Load Session.....	271
Running a Load Session from the System Memory.....	271
Launching WebLOAD Console through the Command Line Interface	272
Establishing Communication.....	275
Errors in Communication.....	275
The Console Screen in Session Mode.....	277
Viewing the Session Tree in Session Mode.....	278
WebLOAD Console Operating Commands	278
Starting a Test Session.....	279
Pausing a Test Session	279
Resuming a Test Session.....	279
Stopping a Load Session Component.....	280
Stopping a Test Session.....	280
Manually Shutting Down Cloud Machines	280
Throttle Control – Changing the Load Size Dynamically During Runtime.....	281
Opening Throttle Control.....	281
Activating Throttle Control.....	282
Deactivating Throttle Control.....	282
Chapter 18. Working with Git.....	283
Prerequisites to working with Git.....	283
Setting up Access to Git Local and Remote Repositories	284
Performing a Commit action.....	285
Performing a Commit Dir action.....	286
Performing a Push action	286
Performing a Pull action.....	287
Opening the Git UI.....	288
Chapter 19. Viewing the Log Window.....	289
Setting the Log Window Options.....	289

Opening the Log Window	290
Viewing the Entire Log Message	290
Sorting Error Messages	291
Opening the Log Window Menu	291
Printing Error Messages Recorded in the Log Window	291
Viewing the Message Status	292
Clearing the Log Window	293

Chapter 20. WebLOAD Console Performance Reports..... 295

Viewing Test Results on the WebLOAD Dashboard.....	295
Viewing Test Results on the WebLOAD Transactions Dashboard	297
Opening Reports.....	298
Opening an Existing Report.....	301
Editing a Report.....	301
Renaming a Report.....	302
Deleting a Report.....	303
Configuring an Integrated Report.....	303
Selecting Measurements for the Integrated Report	304
Changing the Reports Parameters.....	305
Importing Report Parameters from a Saved Load Template	306
Load Generator Performance Statistics	307
Regression Testing.....	308
Configuring an Integrated Report to Include Data from Previously Saved Load Sessions	309
Removing a Previously Saved Load Session from the Integrated Report Configuration	309
Viewing Integrated Reports	310
Changing the Scale of a Report for a Selected Measurement	312
Viewing the True Value vs. Relative Value of a Measurement.....	313
Integrated Report Toolbar	313

Chapter 21. WebLOAD Statistics Reports 317

Statistical Measurements and Their Values.....	317
Changing the Length of the Reporting Interval	318
Current Value Computation	319
Statistics Definitions.....	319
Statistics for Virtual Clients and Probing Clients.....	330
Summary Reports.....	330
Using the Statistics Report.....	332
Statistics Report-Summary by Script	333
Statistics Report-Summary by Total	335
Statistics Report-Summary by Measurement	337

Statistics Report-History by Measurement and Script 339
 Statistics Report-History by Measurement and Total 340

Chapter 22. Data Drilling Reports 343

Data Drilling Reports 343
 Opening a Data Drilling Report 344
 Viewing the Transaction Grid..... 344
 Viewing the Transaction Reason Failure Grid 345
 Viewing the Instance Grid..... 346
 Viewing the Parent Transaction Instance Tree 347
 Viewing the Child Transaction Instance Tree 348

Chapter 23. Exporting Performance Reports 351

Report Export / Import Options 351
 Exporting Selective WebLOAD Report Data..... 352
 Exporting Performance Reports to Excel 353
 Exporting Performance Reports to a Tab File..... 355
 Exporting Performance Reports to HTML..... 356
 Exporting All Statistical Data..... 357
 Exporting All Statistics to Excel..... 358
 Exporting All Statistics to a Tab File..... 359
 Exporting All Statistics to HTML 359
 Customizing the Export File 360

Chapter 24. Performance Measurements Manager 363

Using the Performance Measurements Manager..... 363
 About the Performance Measurements Manager 363
 Opening the Performance Measurements Manager..... 364
 The Performance Measurements Manager Main Window..... 365
 The PMM Configuration Tree Structure 368
 Navigating the PMM Configuration Tree..... 370
 Using PMM Configuration Files 371
 Creating a New Configuration File..... 371
 Opening / Importing Saved PMM Configuration Files..... 372
 Assigning a PMM Configuration to a WebLOAD Console Template 373
 Saving and Exporting PMM Configurations 373
 Specifying the Data Sources, Hosts, and Measurements to Monitor 375
 Adding Data Sources 375
 Adding Measurements to a Configured Data Source 383
 Adding Data Sources to a Configured Host 384
 Deleting a Data Source..... 384
 Deleting a Monitored Host..... 384

Deleting a Measurement.....	385
Configuring Administrator Privileges for Perfmon (Windows).....	385
Enabling Data Sources Monitoring.....	387
Enabling Application Server Monitoring.....	387
Enabling Database Monitoring.....	395
Enabling Systems Monitoring.....	399
Enabling Web Servers Monitoring.....	399
Enabling Monitoring of Application Servers that use Generic Monitoring Protocols...	402
Monitoring Generic SNMP Data Sources.....	403
Remote Performance Measurement Manager	404
Setting the Remote Machine Load Generator.....	404
Setting the Console.....	406
Importing External Statistics.....	407
Appendix A. Integrating with AppDynamics	409
Configuring WebLOAD Scripts for AppDynamics.....	409
Configuring AppDynamics to Recognize WebLOAD Transactions.....	410
Configuring AppDynamics Lite to Recognize WebLOAD Transactions	410
Configuring AppDynamics Pro to Recognize WebLOAD Transactions.....	412
Viewing WebLOAD Performance in AppDynamics.....	417
Appendix B. Integrating with Dynatrace	419
Monitoring a WebLOAD Load Session in Dynatrace	419
Viewing in Dynatrace a WebLOAD Point of Interest	421
Specifying the Dynatrace System Profile	422
Viewing in Dynatrace the Transactions Related to WebLOAD Errors.....	423
Appendix C. Integrating with Nagios.....	425
Instructing PMM to Collect Data from Nagios	425
Selecting the Nagios Statistics to Display in Reports	430
Appendix D. Integrating with New Relic	433
Prerequisites	433
Get or Generate a New Relic API Key	433
Instructing PMM to Collect Data from New Relic.....	435
Selecting the New Relic Statistics to Display in Reports.....	440
Appendix E. WebLOAD Console Error Messages.....	443

Message Severity	443
Success / Info Message	443
Minor Error.....	443
Error.....	443
Severe Error	444
WebLOAD Console Error Messages.....	444
Appendix F. Reference Materials.....	449
WebLOAD Console Files.....	449
Appendix G. Glossary	451
Index	465

Introduction

Welcome to WebLOAD, the premier performance, scalability, and reliability testing solution for internet applications.

WebLOAD is easy to use and delivers maximum testing performance and value. WebLOAD verifies the scalability and integrity of internet applications by generating a load composed of Virtual Clients that simulate real-world traffic. Probing Clients let you refine the testing process by acting as a single user that measures the performance of targeted activities, and provides individual performance statistics of the internet application under load.

This section provides a brief introduction to WebLOAD technical support, including both documentation and online support.

IMPORTANT NOTE: In previous WebLOAD versions, a WebLOAD script was called an “Agenda”. From version 12.0, it is referred to simply as a script. Wherever “Agenda” is still displayed, we are referring to the WebLOAD script.

WebLOAD Recorder used to be called WebLOAD IDE.

WebLOAD Documentation

WebLOAD is supplied with the following documentation:

WebLOAD™ Installation Guide

Instructions for installing WebLOAD and its add-ons.

WebLOAD™ Recorder User’s Guide

Instructions for recording, editing, and debugging load test Scripts to be executed by WebLOAD to test your Web-based applications.

WebLOAD™ Console User's Guide

A guide to using WebLOAD console, RadView's load/scalability testing tool to easily and efficiently test your Web-based applications. This guide also includes a quick start section containing instructions for getting started quickly with WebLOAD using the RadView Software test site.

WebLOAD™ Analytics User's Guide

Instructions on how to use WebLOAD Analytics to analyze data and create custom, informative reports after running a WebLOAD test session.

WebRM™ User's Guide

Instructions for managing testing resources with the WebLOAD Resource Manager.

WebLOAD™ Scripting Guide

Complete information on scripting and editing JavaScript Scripts for use in WebLOAD and WebLOAD Recorder.

WebLOAD™ JavaScript Reference Guide

Complete reference information on all JavaScript objects, variables, and functions used in WebLOAD and WebLOAD Recorder test Scripts.

WebLOAD™ Extensibility SDK

Instructions on how to develop extensions to tailor WebLOAD to specific working environments.

WebLOAD™ Automation Guide

Instructions for automatically running WebLOAD tests and reports from the command line, or by using the WebLOAD plugin for Jenkins

WebLOAD™ Cloud User Guide

Instructions for using RadView's WebLOAD Cloud to view, analyze and compare load sessions in a web browser, with full control and customization of the display.

The guides are distributed with the WebLOAD software in online help format. The guides are also supplied as Adobe Acrobat files. View and print these files using the Adobe Acrobat Reader. Install the Reader from the Adobe website <http://www.adobe.com>.



Icons and Typographical Conventions

Before you start using this guide, it is important to understand the terms, icons, and typographical conventions used in the documentation.

For more information on specialized terms used in the documentation, see *Glossary* (on page 451).

The following icons appear next to the text to identify special information.

Table 1: Icon Conventions

Icon	Type of Information
	Indicates a note.
	Indicates a feature that is available only as part of a WebLOAD Add-on.

The following kinds of formatting in the text identify special information.

Table 2: Typographical Conventions

Formatting convention	Type of Information
Special Bold	Items you must select, such as ribbon options, command buttons, or items in a list.
<i>Emphasis</i>	Use to emphasize the importance of a point or for variable expressions such as parameters.
CAPITALS	Names of keys on the keyboard. for example, SHIFT, CTRL, or ALT.
KEY+KEY	Key combinations for which the user must press and hold down one key and then press another, for example, CTRL+P, or ALT+F4.

Where to Get More Information

This section contains information on how to obtain technical support from RadView worldwide, should you encounter any problems.

Online Help

WebLOAD provides a comprehensive on-line help system with step-by-step instructions for common tasks.

You can press the **F1** key on any open dialog box for an explanation of the options or select **Help > Contents** to open the on-line help contents and index.

Technical Support Website

The technical support page on our website provides:

- The option of opening a ticket
- Links to WebLOAD documentation

Technical Support

For technical support in your use of this product, contact:

North American Headquarters	International Headquarters
e-mail: support@RadView.com Phone: 1-888-RadView (1-888-723-8439) (Toll Free) 908-526-7756 Fax: 908-864-8099	e-mail: support@RadView.com Phone: +972-3-915-7060 Fax: +972-3-915-7011



Note: We encourage you to use e-mail for faster and better service.

When contacting technical support please include in your message the full name of the product, as well as the version and build number.

Introducing WebLOAD

The WebLOAD introductory material gives you an overall view of WebLOAD and consists of the following chapters:

- *Getting Started* (on page 7) describes WebLOAD, how WebLOAD works, the WebLOAD components, and features.
- *WebLOAD Console Features* (on page 43) describes the features available through the Console ribbon and button.

Getting Started

WebLOAD accurately simulates Internet users' behavior and models real-life demands on your Web application to predict capacity requirements, report bottlenecks, and report weak links in your application before deployment.

This chapter introduces WebLOAD, the WebLOAD test components and an overview of WebLOAD's capabilities and features.

Welcome to WebLOAD

WebLOAD delivers the following for maximum testing performance, ease of use and value:

- Full client-side JavaScript support.
- Document Object Model (DOM) access.
- JavaScript-based test scripting.
- Seamless integration with Web application servers.
- Automated operation.
- Single-point management and real-time graphical and statistical reporting.

WebLOAD verifies the scalability and integrity of Web applications by generating a load composed of Virtual Clients that simulate real-world traffic. Probing Clients let you refine the testing process by acting as a single user that measures the performance of targeted activities, and provides individual performance statistics of the Web application under load.

- **WebLOAD is powerful.** WebLOAD unifies performance, scalability and integrity testing into a single process for accurate and timely verification of your Web application, under load defined conditions. WebLOAD provides you with unmatched flexibility for defining and parameterizing Virtual Clients such as connection speed, browser types, multithreading, and SSL encryption strength, for absolute control while simulating real-world conditions.
- **WebLOAD is easy to use.** Recording a test script is as easy as using a browser, with WebLOAD Recorder recording each step. WebLOAD Recorder requires no

programming and is operated using an intuitive, graphical user interface to create visual scripts. These visual scripts can later be enhanced by modifying the automatically generated script. Scripts developed in WebLOAD Recorder can be run in WebLOAD to emulate Virtual Clients for load testing, or can be run in WebLOAD Console to emulate Virtual Clients for load testing.

- **WebLOAD is efficient.** WebLOAD delivers unmatched levels of load generation through its optimized architecture and use of operating systems. You can freely incorporate additional systems for virtually unlimited testing loads.
- **WebLOAD is simple.** WebLOAD automatically finds the performance threshold of your Web application through Goal-Oriented testing. Simply define target performance parameters and WebLOAD generates and increases the number of Virtual Clients accessing the application until performance levels can no longer be sustained.

How Does WebLOAD Work?

WebLOAD tests Web applications by generating Virtual Clients that simulate real-world loads. Virtual Clients emulate the actions of human users by performing typical actions on your Web application. By increasing the number of Virtual Clients, you increase the load on the system. You create visual JavaScript-based test scripts that define the behavior of the Virtual Clients and WebLOAD executes these test scripts monitoring the application response graphically and statistically, and presenting the test results in real time.

WebLOAD incorporates functional verification into the scalability testing process allowing you to accurately verify the scalability and integrity of your Web applications at the per-client, per-transaction, and per-instance level under defined load conditions. WebLOAD saves the test results, including data from the Load Machines (described in the following paragraphs) and the hosting hardware's performance monitor. You can view all or part of the data in real time, or after the test session is complete, in tabular format or in graphical format.

The WebLOAD test process includes:

1. Planning the test
2. Creating scripts
3. Creating Load Templates
4. Running the test
5. Analyzing the test results

This guide describes steps 1, 3, 4, and 5 in the test creation process in detail. Information on performing step 2, Creating scripts is available in the *WebLOAD Recorder User's Guide*.

WebLOAD Architecture

The following diagram illustrates the configuration for a typical WebLOAD test.

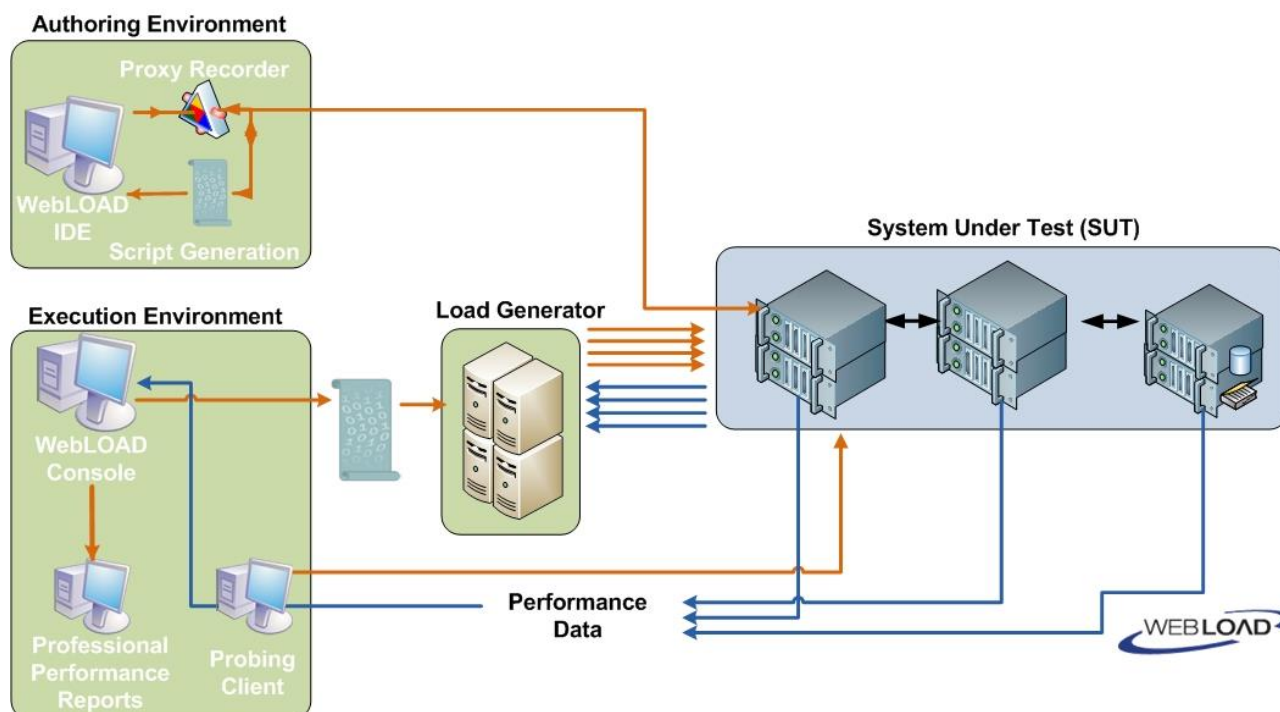


Figure 1: Typical WebLOAD Architecture

The *Console* sets up, runs and controls a test session. At the Console, you can:

- Define the hosts participating in the load test
- Specify the test scripts that the load test executes
- Schedule tests
- View performance reports

Load Machines are hosts, which run Load Generator software simulating multiple *Virtual Clients* simultaneously accessing your System Under Test (SUT). One Load Machine can run multiple Load Generators.

Load Generators execute tests that “bombard” the System Under Test with a large load, to enable complete scalability and stress testing.

The tests consist of multiple simultaneous requests made by Virtual Clients (which emulate Web browsers) to Web servers. Load Machines can run multiple threads.

The *System Under Test* (SUT) is where the Web system under test resides. The SUT does *not* require that WebLOAD software be installed on it.

The *Probing Client Machines* are also hosts. They run Probing Client software, which simulates a single Virtual Client accessing the SUT and runs at the same time as Load Machines, to further measure the performance of the SUT under load. WebLOAD returns exact values for Probing Client performance, in contrast to Load Machines, which return average values of Virtual Client performance.

Additional WebLOAD components are:

- **WebLOAD Recorder** – A tool for recording, creating, and authoring protocol test scripts for the WebLOAD environment. For more information, see the *WebLOAD Recorder User's Guide*.
- **TestTalk** – A network agent that facilitates communication between the Console and hosts, either Load Machines or Probing Client Machines. TestTalk *must* be installed on both the Console and the hosts. TestTalk runs in the background and does not interfere with any other program running on your system.
- **WebLOAD Analytics** – enables you to generate a variety of reports based on performance data and statistics gathered by WebLOAD. Use WebLOAD Analytics to select a Load Session and generate predefined or custom reports displaying the results of the test.
- **WebRM** – distributes and circulates WebLOAD testing resources (Virtual Clients and Probing Clients) amongst users on a “need to use” basis. WebRM is packaged with a maximum number of Virtual Clients, Probing Clients and WebLOAD Console seats, as defined by the WebLOAD license.

What is the Product of Your WebLOAD Testing?

WebLOAD continuously samples the activity while a Load Session is in progress, and saves a wide variety of measurement data. You can view all or part of the data in real time or after the Load Session is complete. Data can be displayed in tabular format or in graphical format. At the conclusion of a Load Session, data can automatically be exported to a spreadsheet program, HTML or as a tab-delimited file to a database application. Reports can also be created using WebLOAD Analytics.

HTTP Client

Using the HTTP client testing mode, testers can simulate the user scenario at the protocol level. Each Virtual Client sends HTTP methods (and/or other protocol calls) to the server based on the recorded script. This is a low-level, “nuts-and-bolts” approach that focuses on specific internal implementation commands, such as HTTP transactions.

Scripts created for HTTP Client mode focus on HTTP transactions. HTTP Client scripts are created through the WebLOAD Recorder. The WebLOAD Recorder is an easy-to-use tool that facilitates the writing of test scripts for HTTP Client mode. The WebLOAD Recorder records your actions as you navigate through a Web application, automatically generating a script that reflects your actions in JavaScript, writing the GET and POST HTTP protocol commands that correspond to your browser activities. You can then use these scripts to test the performance of your Web application during WebLOAD test sessions.

Working in HTTP Client mode provides users with the most efficient, portable, powerful work mode. HTTP Client mode is *portable to many operating systems*, including all different flavors of *Windows, Linux, Solaris*, and so on. HTTP Client mode runs so *efficiently* that users can achieve large benchmarks by simulating extremely heavy test loads of *thousands* of clients. HTTP Client mode also provides the most *comprehensive, detailed statistical analysis and test reports*.

Working in HTTP Client mode does require a certain level of programming sophistication, since it can require hands-on programming of the JavaScript scripts with HTTP protocol code, as described in the *WebLOAD JavaScript Reference Guide*. HTTP Client mode scripts are designed to be run only on WebLOAD.

HTTP Client Feature Summary

The following table summarizes the features of the HTTP Client mode:

Table 3: HTTP Client Feature Summary

Feature	HTTP Client
Average Load per Machine* *single CPU, 1GB memory	Up to 1000
JavaScript scripts	YES
Complex Objects (Flash, ActiveX, etc.)	YES, if using HTTP
Single Script for WebLOAD Recorder	YES
Authoring Tool	YES
Portable to UNIX, etc.	YES




Feature	HTTP Client
Detailed Statistics	YES
HTTP Manipulations	YES
Backward Compatibility (V5.0)	YES

WebLOAD Features

Using WebLOAD, you have access to all the features described in this section, including multiple functionality, protocols, and supported environments.

Table 4: WebLOAD Features

Feature	Benefit(s)
<i>Supported Protocols</i>	
HTTP 1.0 and 1.1 Support	Comprehensive support for HTTP 1.0 and 1.1 including cookies, proxies, SSL, client certificates, authentications, persistent connections, and chunked transfer coding.
SSL Updates	Support for the new SSL standard (TLS 1.0).
SSL Bit Limit	Enables you to set a limit on the maximum number of bits available to Virtual Client for SSL encryption.
Custom HTTP Headers	Define, edit, and manipulate HTTP headers.
Custom User-Agents	Various hand-sets can be emulated by defining custom user-agents.
AJAX Support	Supports automatic recording of AJAX calls into the test script, enabling debugging and full access to all request data (headers and body), both in the script and during run time. WebLOAD supports various formats: XML, JSON, other text-based formats, and binary data. In your script, you can define AJAX requests to be executed concurrently, better simulating the way they are executed by a web browser.
SOAP/Web Services Support	Built-in support for recording Web service calls into the script, providing you with access to both the SOAP envelope and payload for debugging and parameterization.
Authentication Methods	Native support for NTLM and Kerberos.
Comprehensive Support for Dynamic Content	WebLOAD delivers comprehensive support for dynamic applications, including Web 2.0 technologies such as AJAX.
Internet Protocol Support	Native support for FTP, SMTP, POP3, IMAP, NNTP, LDAP, TCP, UDP, and Telnet provides support for non-HTTP protocols.
ODBC Support	Native support from WebLOAD Java Script for ODBC commands to be integrated in a script. Specific support for Oracle, MS SQL and MySQL databases.

Feature	Benefit(s)
 Multimedia protocols support	WebLOAD Multimedia Add-on, supporting RTSP/RTP streaming protocol, for generating scripts for testing multimedia websites.
 Radius protocol support	Supports adding Radius capabilities to WebLOAD scripts, for a complete simulation of a user-scenario that includes authentication for a Radius server.
 Adobe Flex support	The WebLOAD Adobe Flex Add-on supports recording and executing an Adobe Flex script with AMF 0 / 3 protocol.

Supported Programming Interfaces

XML Document Object Model support	WebLOAD provides full support for using the XML Document Object Model. Using XML DOM objects, WebLOAD Scripts are able to both access XML site information and generate new XML data to send back to the server for processing, taking advantage of all the additional benefits that XML provides.
COM Support	WebLOAD Java Script provides direct object access to any component that has a COM wrapping and an Idispatch interface.
Java from JavaScript	WebLOAD supports full Java access from your JavaScript scripts. Full Java support means that your WebLOAD Scripts not only test access time to an HTML page, they can also invoke and run Java classes used by the Java applications embedded within an HTML page.

Script Authoring

Automatic Script Recording	Automatically generate the script by recording protocol level traffic using a proxy-based recorder. Automatic proxy settings available for leading browsers, such as Microsoft Internet Explorer and Mozilla Firefox. The script is generated during recording. Additional information on the recorded pages can be viewed in the Browser View, HTTP Headers View, HTML View, and DOM View.
Sleep Recording	Enables you to automatically record “think” time in the script to provide a more realistic simulation of users accessing your Web application.
JavaScript Script Definition (as part of WebLOAD Recorder)	Delivers the familiarity and scripting power of a full programming language for maximum flexibility, including writing, editing and extending.
Script Authoring (as part of WebLOAD Recorder)	Automatically generates test scripts in JavaScript from a standard Web browser.
Advanced Script Debugging	A fully functional debugger that enables the execution of scripts. The debugger supports break-points, step-by-step debugging, watch, and variable windows. In addition, there are the following views for execution output: Browser View, HTTP Headers View, HTML View, DOM View and Log View.

Feature	Benefit(s)
Data Driven Load Testing	Use real data to emulate realistic scenarios where many different users are accessing your Web application simultaneously. For example, you can use different user-names and passwords for different Virtual Clients.
Include Files	Provides modularity by enabling you to reference JavaScript source code in multiple scripts using the IncludeFile command.
Copy Files	Enables you to copy files from the Console to a Load Machine. The CopyFile command is important if your test uses a script that references auxiliary files such as input files, output files, or other files, while the script and its auxiliary files reside on the Console.
Output Files	Enables you to save script output for later study and analysis. The wIOutputFile object lets you write script output messages to an output file.
Input Files	Enables you to parameterize your script to create variance in the business process that is simulated by Virtual Clients. Use the GlobalInputFile Building Block and the wIInputFile object to read values from external files and use them as input to your application.
Correlation Engine	A rule-based correlation engine, which automatically correlates dynamic parameters – such as session ID – in your script immediately after recording.
Global User Variables	Global User Variables enable you to share data between Load Generators and Load Machines.
<i>Load Test Definition</i>	
Easy User Interface Settings for Runtime Options for each Script	Enables you to customize the runtime options through the Console User Interface.
Save Test Scenarios	Saves test scenarios for accurate before/after comparisons and for re-executing test sessions.
Comprehensive Performance Measurements	Over 25 performance metrics for maximum testing flexibility and accuracy.
First Byte Statistic	First Byte is the time it takes a Virtual Client to receive the first byte of data.
User-defined Timers	Measure specific activities for identifying performance bottlenecks.
Automatic Transactions	All transactions in the script are automatically turned into Named Transactions, eliminating the need to manually create timers for all transactions.
Automatic named transaction enhancements - Object level and HTTP level	WebLOAD automatically creates a sub transaction for all content and automatically breaks down every HTTP command into s elements (open, send, wait, receive).
User-Defined Transactions	Defines specific user-activities in the application as transactions and measures their performance.

Feature	Benefit(s)
Performance Threshold	Transactions can be defined as failed, according to the amount of time the transaction takes.
Real-Time Server Side Statistics	Using WebLOAD you can monitor a variety of server-side applications, databases, systems, and Web-server statistics. You can configure the WebLOAD Performance Measurement Monitor (PMM) to display real-time performance statistics for services available from the server including the percentage of CPU usage, memory usage, etc.
Supported Server Side Statistics for Open Interoperability	Server side statistics can be collected over generic protocols and interfaces such as SNMP, JMX, Perfmon (Windows) and a Generic Unix performance collector. Specific pre-configured collectors exist for the following servers, using the generic protocols: BEA WebLogic, IBM WebSphere 5, Microsoft Active Server Pages, Microsoft ASP .NET, Oracle, Microsoft SQL Server, MySQL, Windows NT/2000/XP, Unix platforms, Windows .NET Framework, Microsoft IIS, Apache, Sun One (former iplanet 6), and Microsoft Media Server.
Enhanced Statistic Setting	ESS provides additional statistical information on the WebLOAD metrics.
Sleep Time Control Settings	Easily controls sleep behavior on the fly to save script development time.
Pass/Fail Definitions	Set rules to define the success or failure of test sessions.
Script Level Options	Unique options can now be set for each script in the test session.
Load Profiler	Create schedules based on pre-defined visual templates for easier and faster scheduling.
Global Functional Verification	Global Functional Verification enables you to select verification tests to run on all pages returned from the server from a list of predefined tests.
Browser Cache	Imitates the Web browser cache activity to provide a realistic simulation of Web activity.
Variable Connection Speed	Enables you to define a load as coming from different connection speeds (i.e., 14400, 28800...).
File Locations	Supplies a default directory for files used and generated by WebLOAD, including: <ul style="list-style-type: none"> • Sessions • Templates • IncludeFiles • CopyFiles • Scripts

Feature	Benefit(s)
<i>Load Test Execution</i>	
Single-Point Management	Console for managing WebLOAD setup, testing, monitoring and reporting.
Throttle Control	Vary the number of Virtual Clients on the fly to create “what-if” scenarios as you change the load conditions.
Probing Client	Simulates a single user-scenario and gets user-level measurements and statistics for better analysis of your system’s performance.
Goal-Oriented testing	The Goal-Oriented Test enables the user to specify performance thresholds and then automatically raise the user load settings until the application reaches the desired performance goals.
Unlimited Load Generation	Optimized, distributed architecture for unlimited load generation with minimal resources, using multiple Load Machines.
Efficient Load Generation	Optimized architecture, keeping Virtual Client memory footprint and CPU utilization to the minimum, and facilitating maximal number of Virtual Clients per Load Machine.
Support for Keep-Alive	Supports persistent connections with Web applications for accurate simulation of Web browser behaviors.
Supports Multithreading	Provides accurate real-world simulation of Web browser behavior.
Synchronization Point	Enables you to ensure that all Virtual Clients execute a given command simultaneously.
Diagnostics Options	The Diagnostic options can be enabled when developing a script or for tracking problems in existing scripts.
Command Line Activation	Allows you to initiate testing through a command line interface. You can enter the WebLOAD launch command into either a batch job or an external script. WebLOAD runs directly, without user intervention, using the parameters specified.
Remote PMM	Enables collecting performance statistics from SUT machines that are behind a firewall.
Integration with defect tracking systems	Allows opening bugs from WebLOAD and posting them into an external defect tracking tool.
SLA Manager	Supports sending error messages and optionally also notifications (via email) when pre-defined conditions occur during the execution of the load test.
<i>Load Test Analysis & Reports</i>	
Real-time Dashboard	View test execution statistics in real-time for a quick and automated overview of test performance.
Real-Time Analysis	Graphical and statistical analysis of test sessions while the test session is running.

Feature	Benefit(s)
Real-Time Reports and Graphs	WebLOAD's Integrated Reports let you see the results of your test in real-time. Test results can be viewed on the test, script, page and object level. Reports include over 75 metrics including response time, hit per seconds, pages per second etc.
Data Drilling	When testing the integrity of your Web application, it is important to verify whether a given transaction succeeded or failed, as well as to determine the reasons for the failure. Data Drilling enables you to display a detailed description of all user-defined and named transactions to the instance level.
Real-Time Transactions Dashboard	Graphical analysis of transaction statistics in real-time.
Error Path Zoom In/Out and Coloring	The path of the failed transaction is colored and expanded in the Data Drilling reports to emphasize the failed transaction. A zoom in/out feature offers the ability to expand or collapse the Data Drilling tree according to the error path.
Regression Testing	Regression testing enables you to compare Load Sessions while tests are running, or after the Load Session is complete.
Microsoft Excel Integration	Automatically exports all test session data to Microsoft Excel for easy storage and reference.
Export Load Session data	Additional support for exporting all test session data to HTML and tabbed file formats.
WebLOAD Analytics	WebLOAD Analytics provides comprehensive reports including failure rates. It automates the process of manually creating reports from WebLOAD performance data. Easily and quickly view these pre-configured reports to gain better insight to performance and scalability issues of your Web applications.
Comprehensive Test Session Reports	Save and compare graphical and statistical test reports for measuring the performance of an application over time.
HTML reports	Export WebLOAD's reports to HTML format.
Predefined Default Reports	WebLOAD supplies six predefined reports to provide you with a fast and easy way to view the information of interest to you in real time without having to spend time configuring the measurements to appear in the reports.

Feature	Benefit(s)
<i>Extensibility</i>	
WebLOAD SDK	<p>WebLOAD SDK enables external developers and users to add functionality to WebLOAD in the following areas:</p> <ul style="list-style-type: none"> • Protocol driver extension • PMM extension • Toolbox (Building Block) extension • User-agent extension • Report Template extension • Object (Java) connector extension • JavaScript language extension
<i>Various Productivity Features</i>	
Enterprise WebLOAD resource pooling and sharing (using WebRM)	<p>Addresses corporate functional and performance testing efforts by organizing, managing and directing WebLOAD resources (Virtual Clients, Probing Clients and WebLOAD Consoles).</p> <p>WebRM enables multiple users involved in various stages of application development and testing to share testing resources. By distributing WebLOAD testing resources each developer can run a test session to simulate, validate, and pinpoint where performance problems occur at any stage of the development life cycle, thus eliminating design flaws and ensuring product quality.</p>

How are Tests Defined?

WebLOAD tests are configured by creating Load Templates, which define the events to occur in the test session. Using the WebLOAD Wizard, you can quickly and easily define basic Load Templates that include:

- The scripts (test scripts) to run
- The machines on which to run the test
- The number of Virtual Clients to run
- The test schedule

After creating a basic Load Template with the WebLOAD Wizard you can modify the template to include advanced features, such as connection speed, browser type, and pass/fail definitions, through the WebLOAD Console ribbon.

Configuring WebLOAD

A Load Session includes the following components:

Table 5: Load Session Components

Component	Description
Console	The Console controls set up, manages a Load Session, and displays the Load Session results. It is recommended that the Console reside on its own separate system and not be installed on systems running Load Generators or Probing Clients.
SUT (System Under Test)	The Web application to be tested. WebLOAD can test any Web application that can be accessed through a Web browser, on the Internet or intranet. The SUT requires no special software.
Load Machine	The host(s) that runs Load Generators. The Load Generators bombard the SUT with user-defined transactions for testing scalability and integrity of the SUT.
Probing Client Machine (optional)	The host(s) that run Probing Clients. A Probing Client runs at the same time as load testing. WebLOAD generates exact values for the Probing Client performance as opposed to averages for Load Generator performance.
WebRM	WebRM distributes and circulates WebLOAD testing resources (Virtual Clients, Probing Clients, and WebLOAD Consoles) amongst users on a "need to use" basis.

The following diagram shows a simple WebLOAD test configuration. The Load Machine and the Probing Client Machine both access the SUT. The Console controls the test from a separate system. All three computers are networked.

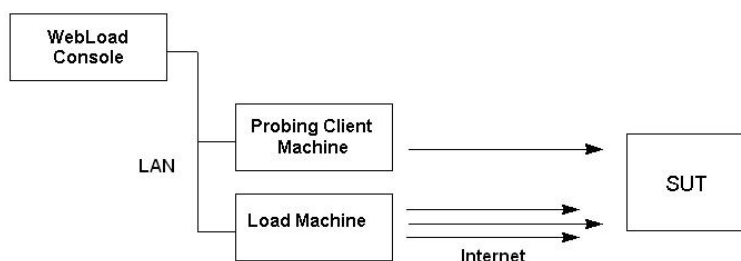


Figure 2: Simple Test Configuration Diagram

Required Software for each Load Session Component

Table 6: Load Session Component Required Software

Test Session Component	Required Software for Running Test Session Component
Console	<ul style="list-style-type: none"> • TestTalk • Console
SUT (System Under Test)	None
Load Generator	TestTalk
Probing Client (optional)	TestTalk
Shared disk (optional)	None

Example of a Test Configuration

The following diagram illustrates a typical WebLOAD test configuration.

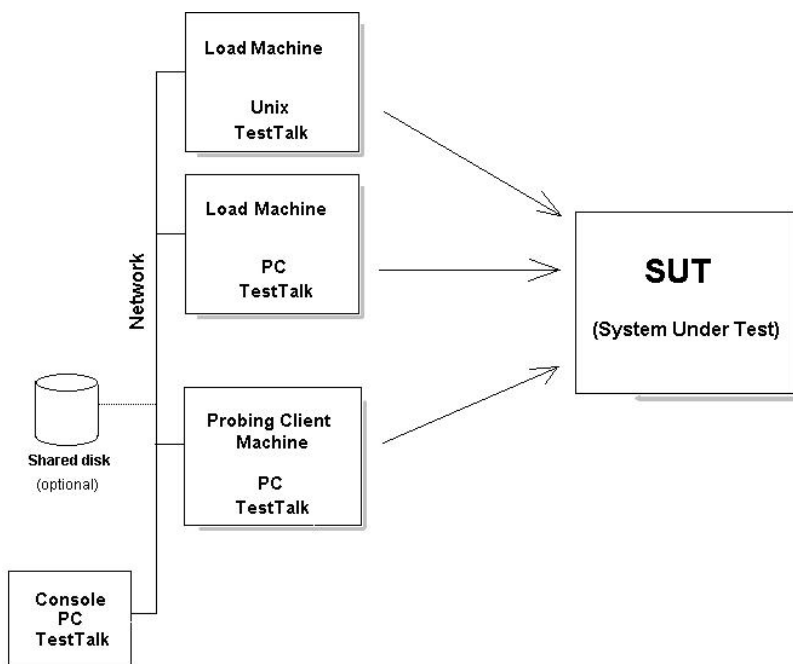


Figure 3: Typical Test Configuration Diagram

This test session has three participating host computers. Two hosts act as *Load Machines*, and one host is a *Probing Client Machine*. In this configuration, the Console is run on a separate system. All systems are on a network with a shared disk that contains the script.

The Console

In the diagram above, the Console is a node on the network along with the host computers in the test session, and the shared disk. The Console communicates with the hosts that are testing the SUT through the TestTalk network agent. The Console controls the test setup, management and reporting.



Note: Running a Load Machine on the same system running the Console software is not recommended.

The Load Machine

Web application tests can be run from either one Load Machine, or from several hosts simultaneously. The test load can be greater if the load is divided between multiple systems. For example, if one host can simulate 100 simultaneous Virtual Clients, a second host with the same capacity can simulate an equal number of Virtual Clients. When the two hosts run simultaneously, they can simulate double that number.

Probing Client

The Probing Client resides on the network with the other hosts in the test session. The Probing Client tests the performance of the SUT while it is being bombarded by the Load Machines. In this test session there is one Probing Client, however WebLOAD can support multiple Probing Clients in a test session. WebLOAD generates exact values for the Probing Client performance as opposed to averages for Load Machine performance.

Sharing a script

You can increase the size of the test load by distributing the load generation tasks over multiple Load Machines. You may want all of these Load Machines to test the SUT using the same script. There are several ways to share a script over multiple systems.

Shared Disk

You can configure a shared disk that can be accessed by all the systems on the network, as shown in the previous diagram. The script can then reside on the shared disk, and can be used by all systems for testing. Thus there is only one copy of the script, and all the relevant hosts run it from the shared disk.

In this configuration, you must make sure that in the script configuration all hosts refer to the shared disk and are mapped to the same logical name. On one host the logical name of the shared disk may be “K” while on another host the logical name may be “L”. Each host must configure the script using the correct logical name.

Copying the Test Program to Each Host

Copy the script to each host participating in the test. Verify that the most updated copy of the script resides on each system.

Working with a Firewall

If your machine configuration is similar to the one illustrated below, no special WebLOAD configuration is necessary.

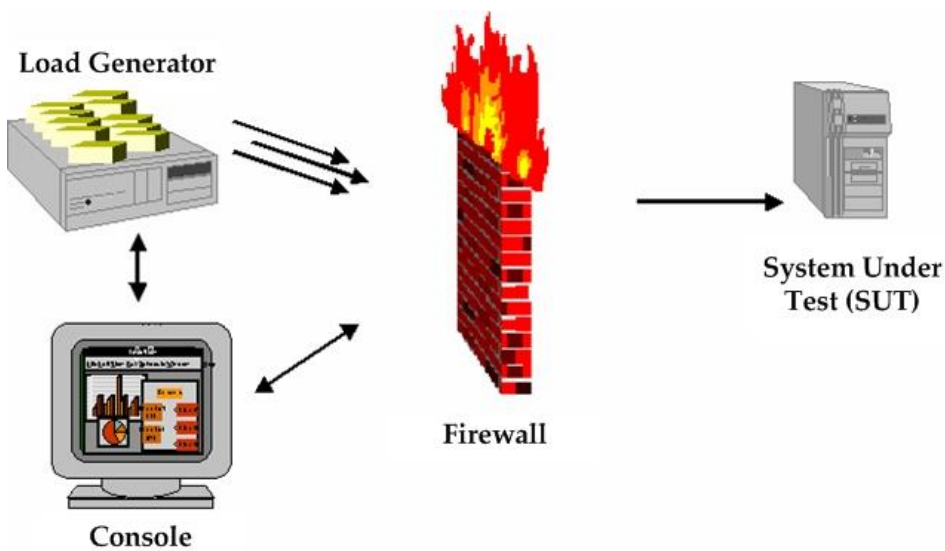


Figure 4: Firewall Configuration

In some cases you may want to use machine resources in the test session that reside on different sides of a firewall. In this case, you must change your initial settings to enable WebLOAD to accommodate this unique configuration.

The following diagrams illustrate situations where the Load Machine and the Console are on opposite sides of the firewall.

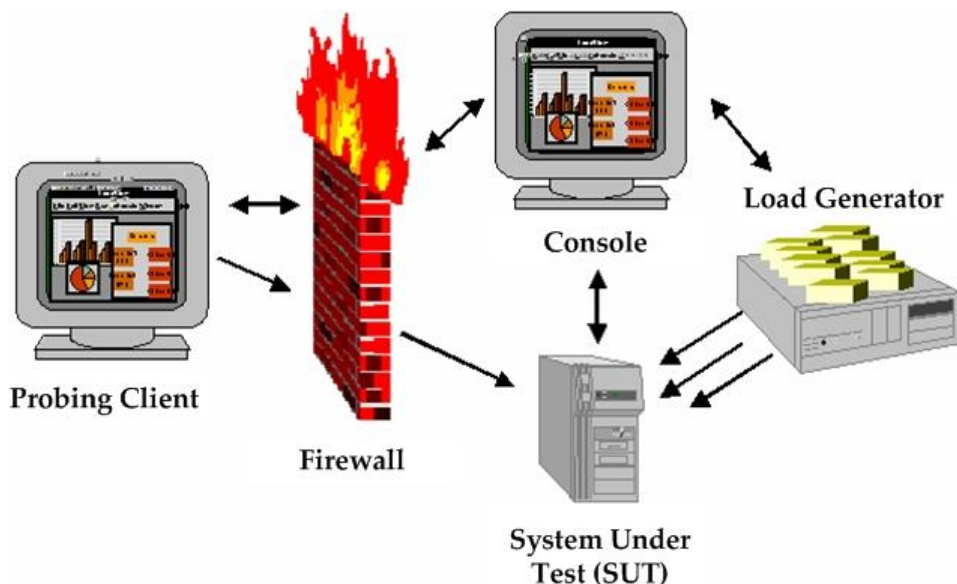


Figure 5: Probing Client and Console Across Firewall

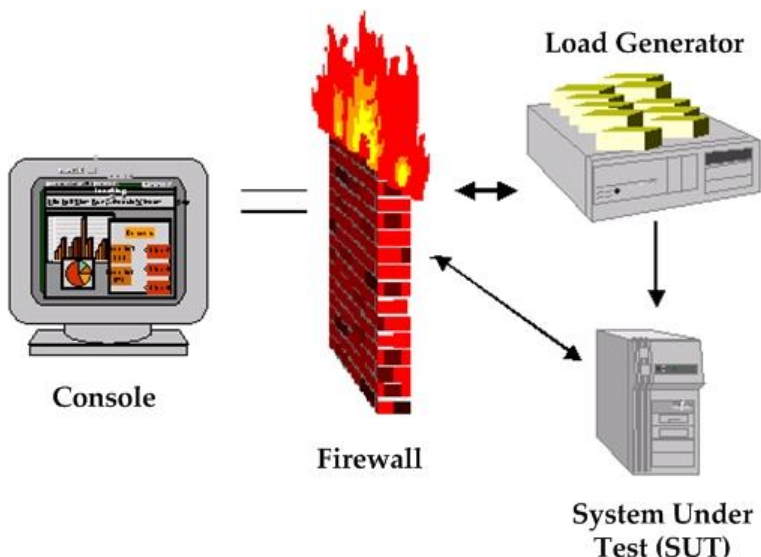


Figure 6: Load Machine and Console Across Firewall

The following configuration is required to enable data transfer through the firewall. Generally, the rules that need to be added to the firewall are:

- Allow outgoing TCP from the Console all ports to Probing Client port 9000.
- Allow incoming TCP from Probing Client port 9000 to Console all ports (options=ackonly).

Managing the Communication Between the Systems in your Load Session

The WebLOAD Console must be able to communicate with all of the hosts in your test session, those that run WebLOAD modules and those that run components of the SUT. All of the hosts must be on a common Local Area Network (LAN) or Wide Area Network (WAN). In addition to the physical network, all of the hosts running a WebLOAD module must be running TestTalk, the WebLOAD network agent, which is installed with WebLOAD.

Working with TestTalk

TestTalk enables communication between the hosts in your Load Session. Each of the hosts participating in the test session must be running TestTalk in the background for the test to execute successfully.

TestTalk is automatically run when you start the Console. While TestTalk is running, the TestTalk icon is displayed in the Status bar.



Caution: Do not close TestTalk in the middle of a test session. This can cause unexpected behavior. When shutting down the system, be sure to shut down the Console first and only then shut down TestTalk.

Viewing the Network Configuration

To view the network configuration:

- Right-click the TestTalk icon in the tray and select Net Configuration.

The following TestTalk message box appears.

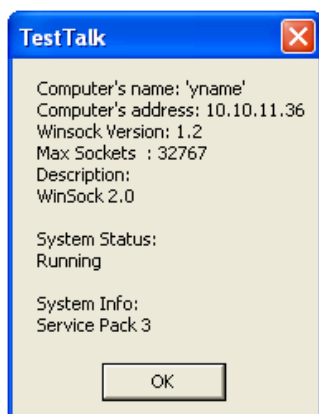


Figure 7: TestTalk Message Box

The information displayed in the TestTalk message box includes:

Table 7: TestTalk Message Box Items

Item	Description
Computer's name	The name of your computer. You must know the name of each of the hosts to configure your test session. If you are not familiar with the name of your host, open TestTalk at that host and view the name in this message box.
Computer's address	The IP address. You need this information to ping a host if you want to verify that it is connected to the network.

Required Connections

WebLOAD can only run a test when the hosts participating in the test session are properly connected. Each participating host must have two types of connections. If either of these connections is not complete, you may experience problems running WebLOAD.

- Physical connection

The hosts in the test session must all be physically connected to the network. They must be turned on and the connection must be live.

- Connected by network agent

All of the hosts in the test session must be running TestTalk as a background process. Only the host controlling the tests must be running the Console.

Testing Your Connection

To test your connection:

1. Ping the host to test its physical connection to the network. This is described in *Managing the Communication Between the Systems in your Load Session* (on page 24). If the host is not connected properly to the network, connect it and rerun WebLOAD.
2. If the ping is acknowledged, verify that TestTalk is running on the host. This is done by checking the actual host to verify that the TestTalk program is functioning. If TestTalk is not running, open TestTalk by selecting **Start > Programs > RadView > WebLOAD > TestTalk** and retry the program.

TestTalk runs in the background. Error messages are displayed on the program window. Check this window for any error messages that may explain the problem.

3. If there is still a problem with the host, you may be experiencing problems not related to TestTalk. Restart the system and rerun WebLOAD.

Pinging a Host

You can test a host's physical connection to the Console by pinging the remote host. The response that the Console receives from the ping indicates the status of the connection to the remote computer.

To ping a host:

- From the operating system, type **ping systemname**.

Example:

```
c:> ping systemname
```

Acknowledgments from a Ping

You can receive several types of acknowledgments from a ping:

- Ping `systemname` with 32 bytes of data.

```
reply from 192.114.35.1: bytes = 32 time = 1ms TTL = 255
```

```
reply from 192.114.35.1: bytes = 32 time = 1ms TTL = 255
```

```
reply from 192.114.35.1: bytes = 32 time = 1ms TTL = 255
```

```
reply from 192.114.35.1: bytes = 32 time = 10ms TTL = 255
```

This acknowledgment indicates that the host is connected properly. If you receive this response, and WebLOAD continues to return an error, test the TestTalk network agent.

- Bad IP address `systemname`.
Indicates that the remote host is not connected properly. The host must be physically connected to the network, and configured properly.
- Request timed out.
Indicates that the IP address is known to the Console, however the host is not currently working. The host must be turned on and tested again.

Troubleshooting Communications

On occasion, you may have trouble with a WebLOAD host. Refer to the following table for possible solutions.

Table 8: Troubleshooting Communications

Problem	Solution
<p>At the beginning of a test session, the host prompts: All hosts are stopped.</p> <p>This indicates that none of the hosts are accessible and the Domain Name Server (DNS) cannot recognize a host name.</p>	<p>Test your connection, as described in <i>Testing Your Connection</i> (on page 26).</p>
<p>At the beginning of a test session, a host prompts: Load Session Setup -timed out, not completed.</p> <p>This indicates that at least one host is not accessible and the Domain Name Service (DNS) cannot recognize a host name.</p>	<p>Test your connection, as described in <i>Testing Your Connection</i> (on page 26).</p>
<p>Your system starts, but the test session times out during operation.</p>	<p>Test your connection, as described in <i>Testing Your Connection</i> (on page 26).</p> <p>Check the names of the computers in the TestTalk Help menu, under Net Configuration. Verify that there is complete consistency between the names of the computers and the names used in WebLOAD.</p>
<p>WebLOAD reports unexpected results from a test.</p>	<p>Access the System Under Test (SUT). If you have a good connection, the test results are accurate.</p>

Problem	Solution
The system issues the message: Error loading script...	The Console must specify the complete path of the script that is used by the remote host. For example, if the script is on a shared disk, the name of the disk in the path specified by the Console, must be the same name used by the remote host.

WebLOAD Quick Start

This section shows you how you can get started quickly using WebLOAD by designing and executing a performance test of the RadView Software test site at <http://www.webloadmpstore.com>. Then you can refer to the complete WebLOAD documentation set to learn more techniques for using WebLOAD.

Quick Start Steps

In this Quick Start, you perform the following steps:

1. Create a script (see *Creating a* on page 29).
2. Configure a Load Template using the WebLOAD Wizard (see *Configuring a Load Template using WebLOAD Wizard* on page 34).
3. Configure Session options (see *Setting Session Options* on page 38).
4. Run a test (see *Running the Test* on page 38).
5. Analyze test results (see *Analyzing Test Results* on page 39).



Note: We recommend that you follow the steps in the order they appear. All examples are interrelated and dependent on earlier steps.

Creating a Script

The first step in creating a script is to record your actions as you interact with your Web application.

To create a script:

1. Start WebLOAD Recorder by selecting **Start > All Programs > RadView > WebLOAD > WebLOAD Recorder**.

WebLOAD Recorder opens.

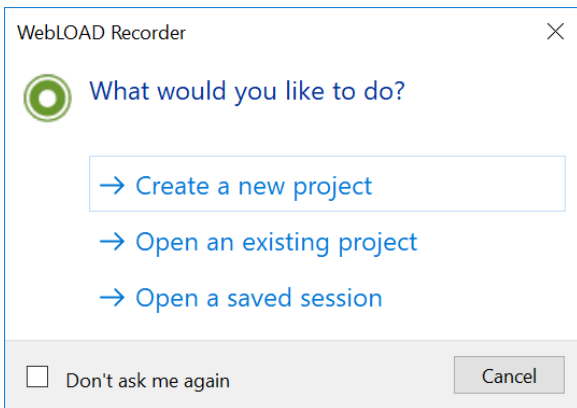


Figure 8: WebLOAD Recorder Dialog Box

2. Select **Create a new project**, and click **OK**.

The WebLOAD Recorder main window opens in Editing Mode for you to begin creating your script.

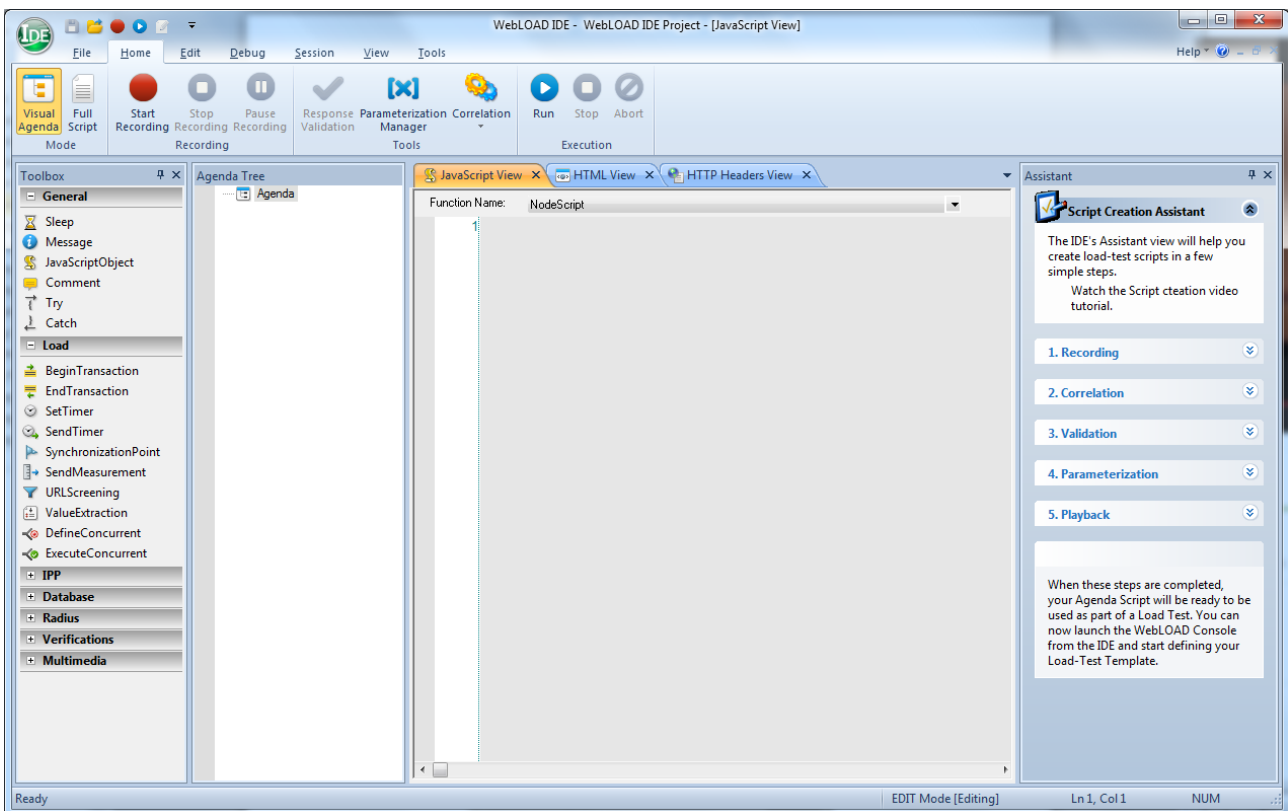



Figure 9: WebLOAD Recorder Main Window in Editing Mode

3. In the main window, in Editing Mode, click the **Start Recording**  button in the Home tab of the WebLOAD Recorder ribbon to begin recording.

The Recording dialog box appears.

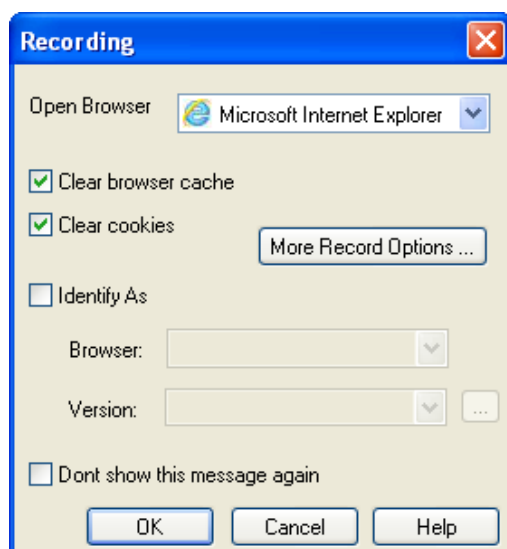


Figure 10: WebLOAD Recorder Recording Dialog Box

4. Set the desired recording parameters, as described in the *WebLOAD Recorder User's Guide*.
5. Click **OK**.

WebLOAD Recorder begins recording all of the actions you perform in the browser, as indicated by the recording notification in the WebLOAD Recorder status bar.

EDIT Mode [Recording]

Figure 11: Status Bar

A browser window opens.

6. In the address bar, enter the Web address <http://www.webloadmpstore.com> to go to the WebLOAD test site.

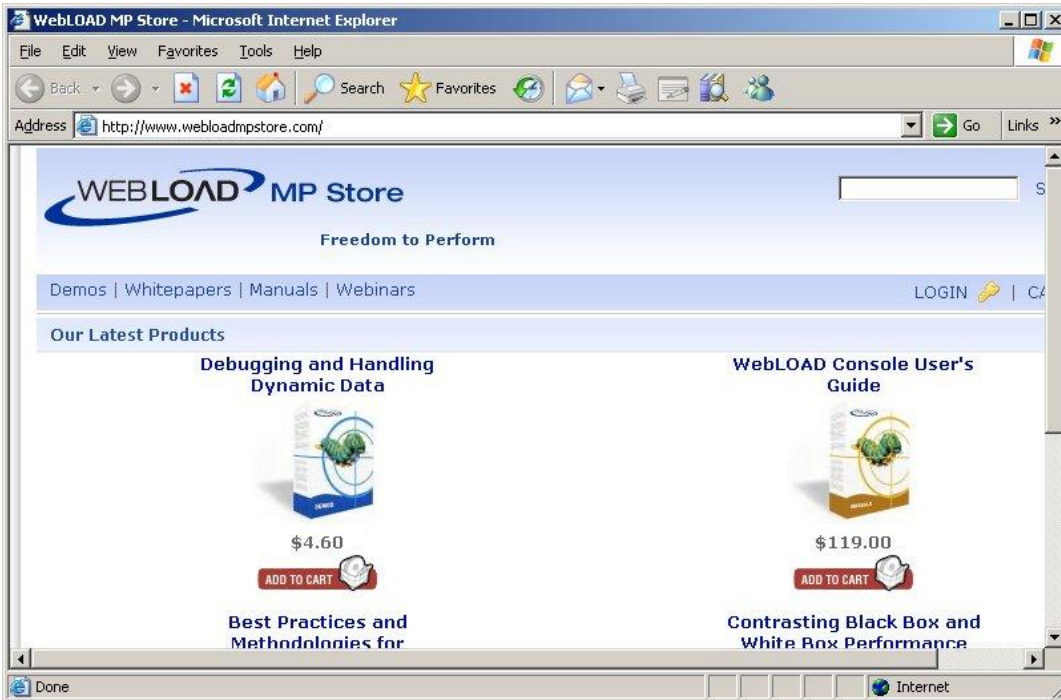


Figure 12: WebLOAD Test Site Home Page

7. Navigate through the site, performing the actions you want to test.

For example:

- a. Click a product to view the product details.
- b. Click **Add to Cart** to add the product to your shopping chart.

Your actions are recorded and appear in the Script Tree as you navigate the site. (If you see additional nodes in the Script Tree with different URLs, this may be traffic generated by browser plug-ins or extensions, for example, third-party toolbars.)

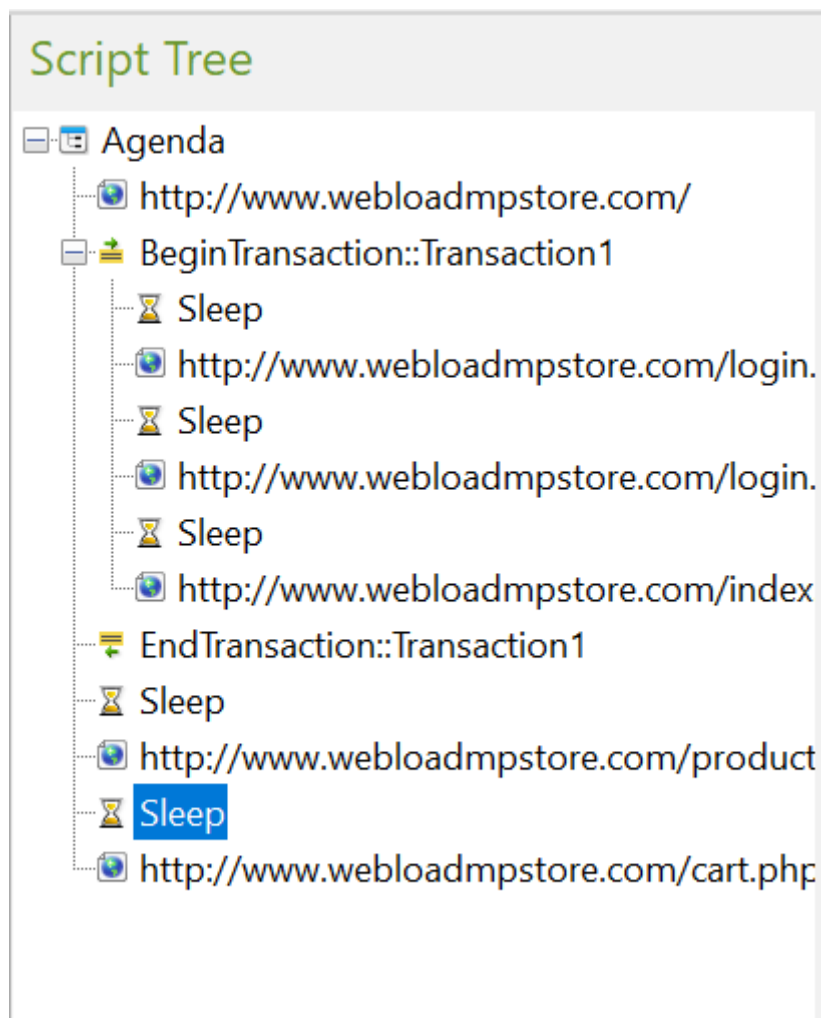




Figure 13: The Script Tree

8. Click the **Stop Recording**  button in the **Home** tab of the WebLOAD Recorder ribbon to stop the recording.
9. Click the **Save**  button in the Quick Access toolbar or in the **File** tab of the WebLOAD Recorder ribbon to save your script.
10. Type `QuickStart` as the name of the script in the Save As dialog box and click **Save**.

The script is saved with the extension `*.wlp`.

You now have a basic script that can be used in a WebLOAD test configuration. For complete information on creating, editing and modifying scripts, and adding functionality to your script, see the *WebLOAD Recorder User's Guide*.

Configuring a Load Template using WebLOAD Wizard

Using the Quick Start, you will learn how to perform the following steps in order to configure a Load Template with the WebLOAD Wizard in the WebLOAD Console:

1. Opening the WebLOAD Wizard.
2. Selecting a script and a Performance Statistic to monitor.
3. Selecting a host on which to run the Load Machine.
4. Scheduling the test and selecting the number of Virtual Clients to run.
5. Setting Session options.

For more details about each of these steps, see *Creating Load Templates with the WebLOAD Wizard* (on page 99).

Opening WebLOAD Wizard

The first step in defining a Load Template with the WebLOAD Wizard is defining the type of script to run. The script is the test script that defines the actions that the Virtual Clients perform during the test. WebLOAD supports both single scripts (lone test scripts) and Mix scripts (a group of scripts simulating groups of users performing different activities on the application simultaneously). In this Quick Start, we will create a template running a single script. For information on creating templates with multiple scripts or Mix scripts, see *Creating Load Templates with the WebLOAD Wizard* (on page 99).


To open WebLOAD Wizard:

1. Start the WebLOAD Console.
Select **Start > All Programs > RadView > WebLOAD > WebLOAD Console**.
The WebLOAD Console opens, and the WebLOAD Console Startup dialog box appears.
2. Select **Create a new Template using WebLOAD Wizard**, and click **OK**.
The WebLOAD Wizard opens.
3. On the Welcome screen, click **Next**. The script/Mix Type screen appears.

Selecting a Script

To select a script and a performance statistic to monitor:

1. On the Test Type screen, select **Single Script**, and click **Next**.

2. On the script/Mix Selection screen, click the  button and browse to the location of the `QuickStart.wlp` script created earlier using WebLOAD Recorder. Select the script and click **Open**.
3. On the script / Mix Selection screen, click **Next**. The Host Selection screen appears.

Selecting a Host

For the steps in this Quick Start, you use your own machine as the host Load Machine to generate Virtual Clients. The number of Virtual Clients that can be generated depends on the power of the host machine.

To select a host:

1. On the Host Selection screen, note that your computer is automatically listed as the default Load Machine.
2. Click **Next** to accept the default selection.

Scheduling the Test

When you schedule the test, you define the load to be generated throughout the test and the total testing time. The Load Profiler provides a quick and easy way to select different testing patterns.

To schedule the test:

1. On the Schedule screen, select **Manual**, and click **Next**.
The wizard progresses to the Schedule Manually screen.
2. Click your computer.
The Load Profiler buttons becomes enabled.
3. Click **Load Profiler**.
The Load Profiler opens.
4. Select **Linear** from the **Select scheduling model** drop-down list.
5. Enter 10 in the **Concluding number of Virtual Clients** field and accept the defaults for the **Total time in minutes** and **Starting number of Virtual Clients** fields.
The test will run for 10 minutes, starting with a load of 5 Virtual Clients and increasing to a maximum of 10.
6. Click **Replace**.
The Linear Schedule model now appears in the row below your host computer name in the Schedule Manually screen.

- Click **Next**. The Performance Measurements Manage screen appears.

Selecting the Performance Measurements

The Performance Measurements Manage screen enables defining which statistics to monitor during the test.



Note: In this simple example, we will monitor the CPU usage on your system during runtime. During a real test, you would probably monitor the CPU performance of your application server.

- In the Performance Measurements Manage page, click **Add Monitors**.
The Performance Measurements Manager window appears.

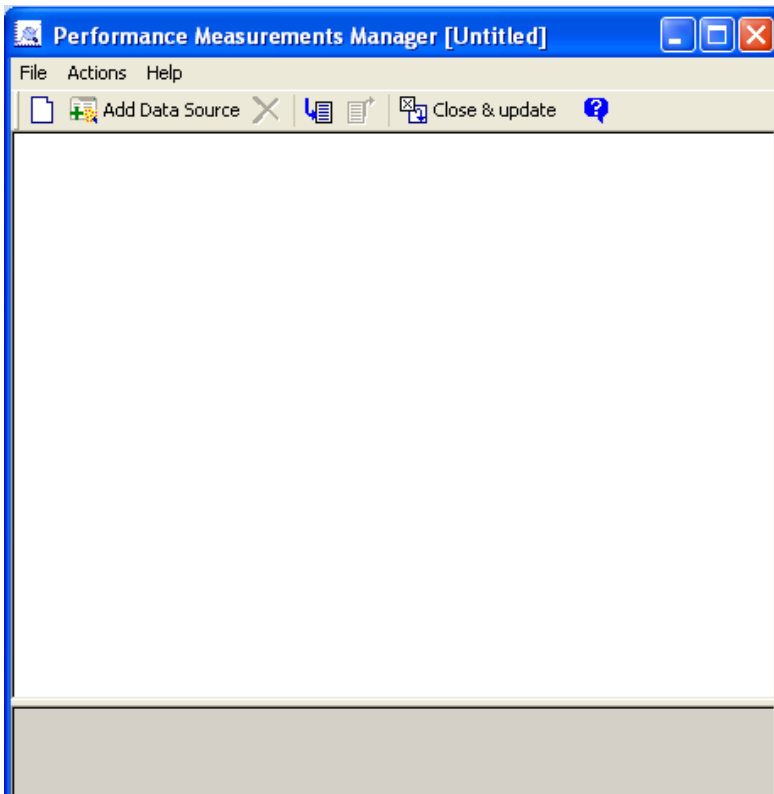



Figure 14: Performance Measurements Manager (Untitled)

- Click the  **Add data source** toolbar button.
The Performance Measurements Manager Wizard opens.
- On the Welcome screen, click **Next**.
The Selecting a Data Source window appears.
- Click the plus sign (“+”) adjacent to **System** to expand the tree, select **Windows platform**, and click **Next**.

5. Click **Browse** to browse available hosts. In the Host selection dialog box, highlight the name of your computer in the list, and click **Select**.
6. Click **Next**.
WebLOAD attempts to connect to the selected host and collect data. A tree of all available measurements to monitor appears.
7. Expand the tree to select the measurement you want. Click **Processor** ► **_Total**, select the checkbox adjacent to **% Processor Time**, and click **Next**.
8. Click **Finish**.

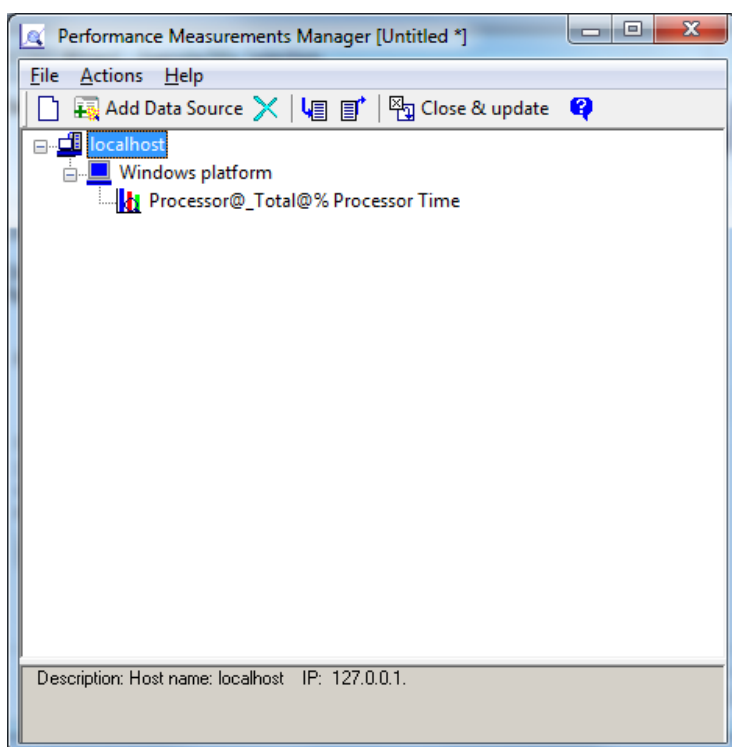



Figure 15: The Performance Measurements Manager

9. Click the  **Close & update** toolbar button to save the configuration with the template and return to the WebLOAD Wizard.
10. Click **Next**.
11. Clear the **To run this Load Template immediately, select this checkbox and then click Finish** checkbox, and click **Finish**.

You have now created a Load Template using WebLOAD Wizard.

Setting Session Options

Before you run the template you created, you can select options for verifications to execute during runtime. You can set WebLOAD to perform the following tasks:

- Collect data at the page level.
- Verify the time required to receive a page (default 20 seconds).
- Verify that the links in your Web application act as expected.

When you select page verification, WebLOAD collects transaction data under the Title name, in this case <http://www.webloadmpstore.com>. You can observe statistics, such as the response times for this page, as the load increases. You can also gather performance data about the objects on a page by setting WebLOAD to collect object level data. For example, you can determine whether pictures or other embedded objects are taking a long time to download.

To set options for these verification tests:

1. In the WebLOAD Console, select **Current Session Options** in the **Tools** tab of the ribbon.
2. Select the **Functional Testing** tab.
3. Under **Automatic Data Collection**, select the **All Pages** checkbox.
4. Under **Page Time**, check the **Verification** checkbox, enter 20 in the **Max Time** field, and select **Minor Error** from the **Severity Level** drop-down list.

If the time required to receive a page exceeds 20 seconds, a Minor Error message appears in the Log Window and the event is available in the WebLOAD Data Drilling report.

5. Under **Verification**, select the **Verification** checkbox, and enter 5 in the % field.

Five percent of the Virtual Clients will perform the selected verification tasks. This percentage lets you balance data gathering with the need to maintain performance during the test.

6. Click **OK**.

Running the Test

You are now ready to start the test. As you run the test, you can view the data gathered by WebLOAD in real time.

To run the test:

1. Select **Start Session** in the **Session** tab of the ribbon or in the Quick Access toolbar.

- At the prompt, click **Yes** to save the template. Type the filename `QuickStart.tpl`, and click **Save**. You can now reuse this template, with its script, schedule and options at a later time.

The test begins.

Analyzing Test Results

You can view test results in real time while the test is running or after the test session has completed. Once the test begins, the WebLOAD Wizard default report appears in the Results window.

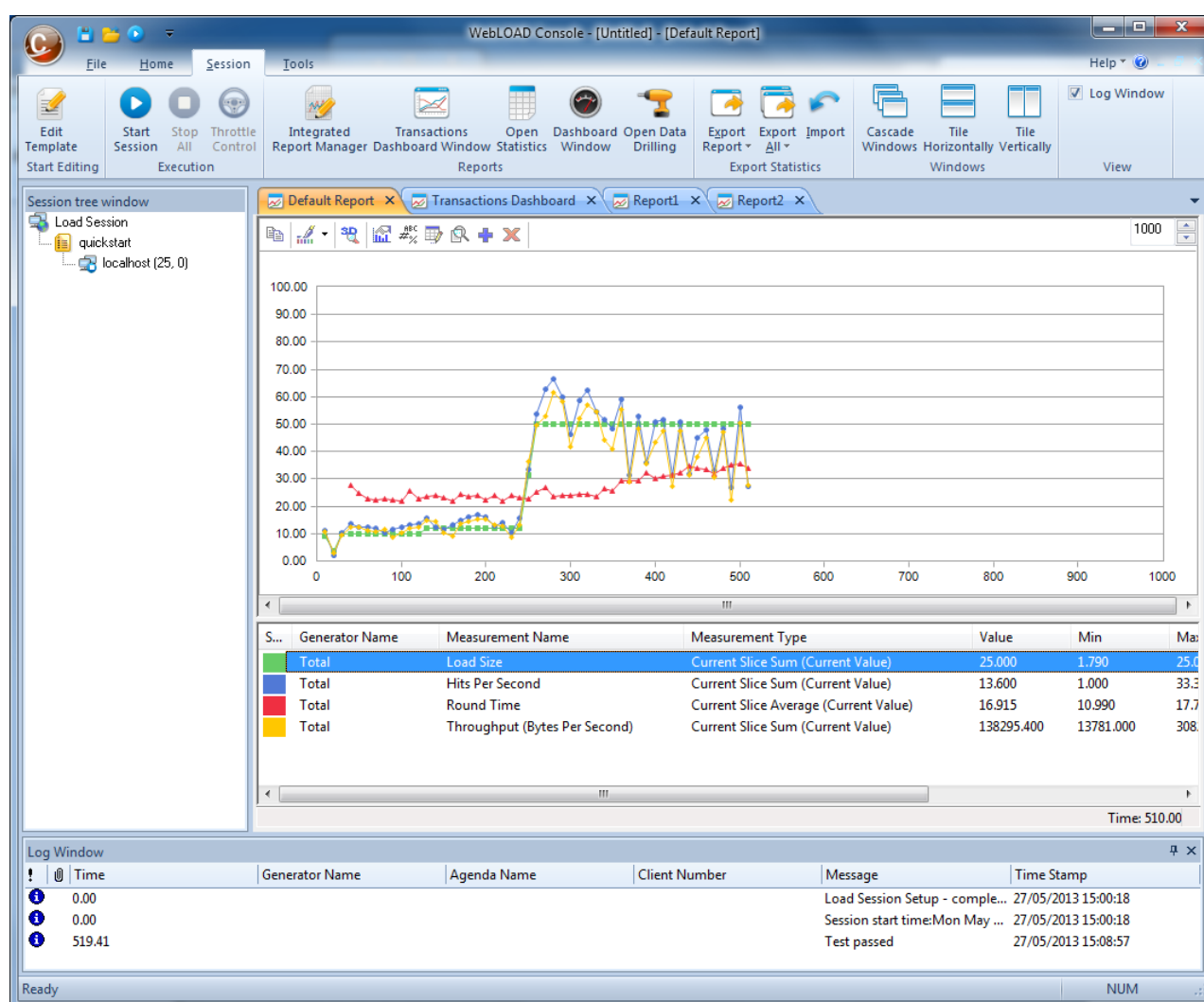


Figure 16: The WebLOAD Console – WebLOAD Default Report

Viewing the Results in Real Time

Data on load size, hits per second, round time (one run through the script) and throughput (bytes per second) appear in the Chart View window, with a color legend underneath the chart. Each data point represents a 20 second slice. Data is gathered every 20 seconds and is then either averaged (for hits, round time and throughput) or summed (for load size). The graphs are all normalized to scale so they can be viewed in a single Chart View window. You can access several kinds of test details:

- Hover over any data point to see actual values.
- Select **Dashboard Window** in the **Session** tab of the ribbon to view a quick overview of the status of the test including the following:
 - Elapsed time
 - Current load size (number of running Virtual Clients)
 - Rounds executed
 - Failed rounds
 - Failed hits
 - Hits per second
 - Pages per second
 - Throughput
- Select **Open Statistics** in the **Session** tab of the ribbon. A Statistics tab appears in the results window. Scroll through the data and double-click a cell to drill-down to further detail.

Each of these views opens in a new tab window. Close all windows except the Default Report tab.

Creating Integrated Reports

Create an Integrated report to view the results for additional metrics gathered by WebLOAD. This can be done while the current session is still running.

To create an Integrated Report:

1. In the WebLOAD Console, select **Integrated Report Manager** in the **Session** tab of the ribbon.
The Integrated Reports dialog box appears.
2. In the Integrated Reports dialog box, click **New**.
3. In the tree, expand the QuickStart::Total branch.

- Click the plus sign (“+”) adjacent to Hit Time to expand the branch. Then expand the Current Slice branch in the same way. Check the checkbox adjacent to Average (Current Value).

Checkmarks appear next to Hit Time, Current Slice, and Average (Current Value).

- Expand Successful Hits and then expand Current Slice. Check the checkbox adjacent to Sum (Current Value).
- Expand Successful Connections and then expand Current Slice. Select the checkbox adjacent to Sum (Current Value).
- Scroll to the PM@your-system-name branch at the bottom of the tree (PM stands for performance monitor). Click the adjacent plus sign (“+”) to expand the branch. Expand Processor:%Processor Time:_Time and then expand Current Slice. Check the checkbox adjacent to Average (Current Value).

This choice displays the statistics on CPU processor usage that were collected during the load test.

- Click **OK**.

The Save dialog box appears.

- Type `MyReport` as the name of the report, click **Predefined Report**, and click **OK**.

The new report is added to the list of Predefined Reports.

- To open the newly created report, select **MyReport** in the list of Predefined Reports, and click **Open**.

In the WebLOAD Console Results window, the MyReport tab appears displaying the report data. The name of the tab is the name of the report you selected.

- If the test is still running, select **Stop All** in the **Session** tab of the ribbon.
- Select **Save** in the **File** tab of the ribbon to save the data from the test along with the template information (script, options and schedule). Specify a filename (QuickStart) and location and click **Save**.

Viewing Results with WebLOAD Analytics



WebLOAD Analytics creates analysis reports based on your session data. Even after you complete a session, you can use this data to create both predefined reports and custom reports of your own. You can create portfolios, which are collections of custom reports to run on other sessions. You can also export reports to PDF, Excel, RTF, HTML, or TXT formats to share with others, store with test plans, or mail to team members.

To view results with WebLOAD Analytics:

- Select **WebLOAD Analytics** in the **Tools** tab of the ribbon.

Data for the last session saved is automatically selected and imported into the WebLOAD Analytics sessions database.

WebLOAD Analytics automatically creates a new report based on the Summary Portfolio and generates the report. You can optionally perform any of the following:

- **Modify the report** – Select the **Report Settings** tab and edit the parameters you wish to change.
 - **Print the report** – Select the **Report View** tab from the Reports area. Click  to open the Print dialog and print the report.
 - **Publish the report** – Click  to publish your report as the default format specified in the Preferences window. Select a location and name for your report and click **Save**. The report is published in the format chosen and saved to the location specified in the Preferences window.
2. Close WebLOAD Analytics.
 3. Close the WebLOAD Console.

This concludes our Quick Start guide for WebLOAD. For additional information, refer to RadView's website at <http://www.radview.com/support>.

WebLOAD Console Features

The WebLOAD Console is where you perform the majority of your load / performance testing activities.

The WebLOAD Console

The Console consists of the following:

- Console System button
- Quick Access toolbar
- Ribbon
- Session Tree
- Results window
- Status bar

The Console manages the Load Session. It also:

- Configures Load Session hosts and scripts.
- Schedules Load Session scripts.
- Monitors the application's performance and integrity under the generated load.
- Manages the Load Session as it is running, allowing you to pause, stop and continue Load Session components as needed.
- Displays the current performance of the SUT and provides a final performance report for Virtual Clients and Probing Clients.
- Manages exporting of test reports.

The WebLOAD Console component controls your test session.

From the Console, you can:

- Specify the program (script) that the test session runs.
- Select the number of Virtual Clients to run.

- Define the hosts participating in the Load Session.
- Schedule tests.
- View performance reports.

Opening the Console

To open the Console:

- Select **Start > All Programs > RadView > WebLOAD > WebLOAD Console**,
-Or-

From your desktop, click the **Console** icon.

The WebLOAD Console Startup dialog box appears, displaying a variety of shortcuts to the various WebLOAD Console components.

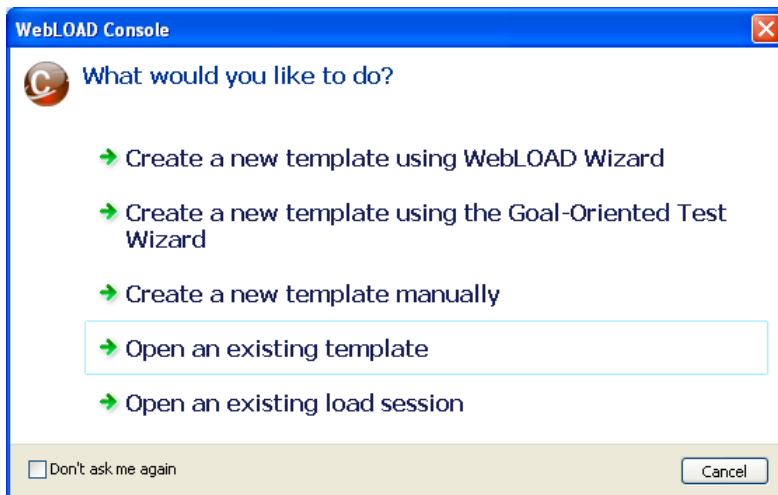


Figure 17: WebLOAD Console Startup Screen

After you select an option and start working, the WebLOAD main screen appears.

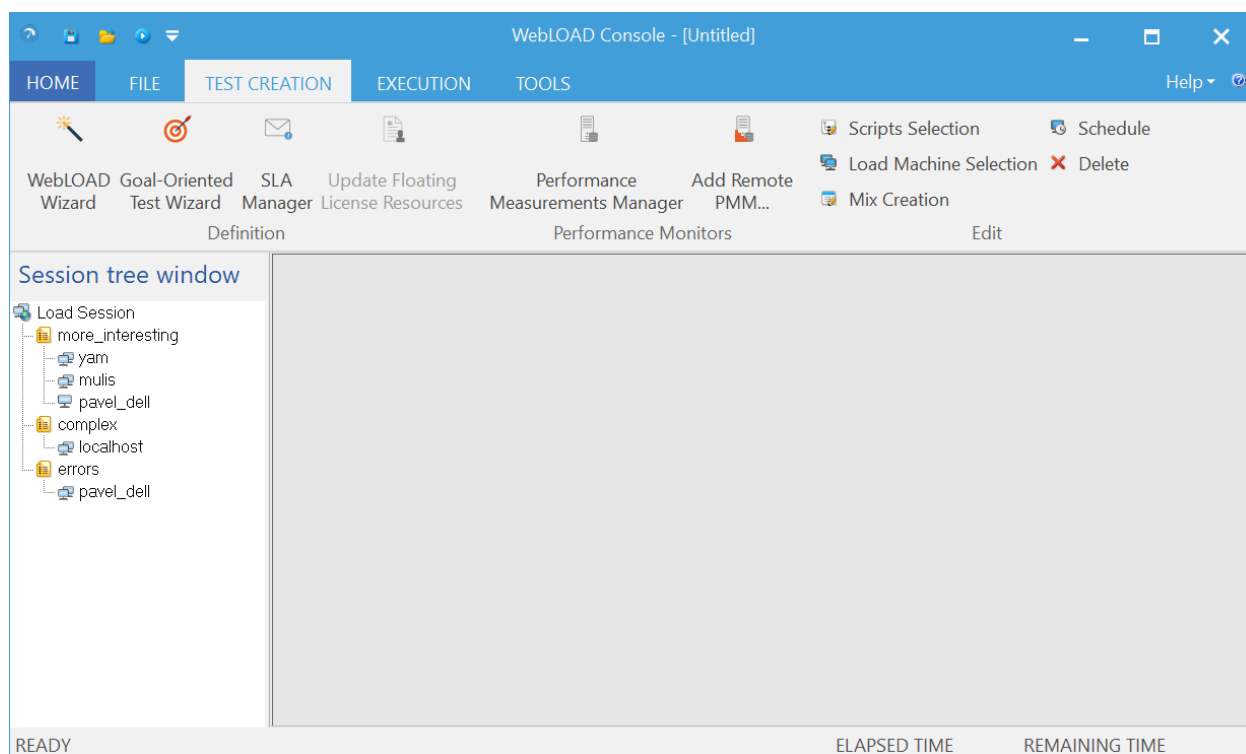


Figure 18: WebLOAD Console Main Window

The Console is operated through its Console System button and ribbon at the top of the screen. Tests are configured through a series of dialog boxes.

Console Elements

When you first start WebLOAD Console, you see the Console window interface in Edit mode. The Console window displays different components in the two different modes:

- Edit Mode where you create your templates.
- Run Mode where you run your sessions and see the results.

The Console window interface consists of the following elements.

Title Bar

The title bar displays the name and path of the current template or session.

Console System Button

The Console System button provides access to the most commonly used tasks.



Figure 19: Console System Button

Table 9: Console System Button Menu Options

Option	Description
New	Starts a new Load Template. A new empty session Tree appears.
Open	Enables opening any of the following: <ul style="list-style-type: none"> • A saved Load Session. A Load Session is saved as a file of type *.ls. The Console displays a Session Tree window initialized to the settings saved in the file. • A Load Template file of the type *.tpl. A Load Template file contains test configuration, the assigned scripts, the testing schedule, and the report configuration, without the test session results.
Save	Saves a Load Template or Load Session. When a Load Session is saved, it includes the Load Template and the data accumulated running the test.
Save As	Enables performing a Save or Save As operation If the session tree is currently displaying a Load Session, the Save operation saves the current Load Session configuration and results to a file of type *.ls. The following information is saved: <ul style="list-style-type: none"> • Selected hosts, both Probing Clients and Load Machines • Assigned scripts for each host • Test session schedule • Report configuration • Test session results (Performance Report data) If the session tree is currently displaying a Load Template, the following information is saved: <ul style="list-style-type: none"> • Selected hosts, both Probing Clients and Load Machines • Assigned scripts for each host • Test session schedule • Report configuration
Goal-Oriented Test Wizard	Opens the Goal-Oriented Test wizard. The Goal-Oriented Test wizard enables you to run goal-seeking performance testing.
WebLOAD Wizard	Opens the WebLOAD Wizard. This wizard walks you through the WebLOAD Console configuration process.

Option	Description
Edit Template	Switches from Run mode to Edit mode.
Global Options	Opens the Global Options dialog box. Use the global options to customize WebLOAD Console to your specifications, see <i>Setting Global Options</i> (on page 177).
Default Options	Opens the Default Options dialog box. Use the default settings to customize the script settings to your specifications. When building a new template or session, or editing an existing one, each new single script or script participating in a Mix is automatically assigned these defaults, see <i>Setting Script Options</i> (on page 211).
Current Session Options	Opens the Current Session Options dialog box. Use the session settings to customize the current session to your specifications. This option will update the options for all scripts in the current session, see <i>Setting Script Options</i> (on page 211).
Script Options	Opens the Script Options dialog box. Use the session settings to customize the current script to your specifications, see <i>Setting Script Options</i> (on page 211).
Print	Prints the active chart or grid displayed in the Results window, with all contents and formats.
Additional Information	Displays additional information for the active template / Session.
<i>Recent files</i>	Lists the template / Session files you have most recently opened in the WebLOAD Console. To quickly reopen one of these files, click it.
Exit	Closes the Console.

Quick Access Toolbar

The Quick Access toolbar provides shortcuts for your most common tasks. By default, the toolbar provides shortcuts to the Save, Open and Start Session commands. You can customize the toolbar to add any other WebLOAD Console command.



Figure 20: Quick Access Toolbar

Ribbon

The ribbon displays the commands you use to work with WebLOAD Console. In addition to the standard File tab, various tabs are provided to access functions specific to template creation and session management, such as the Session tab.

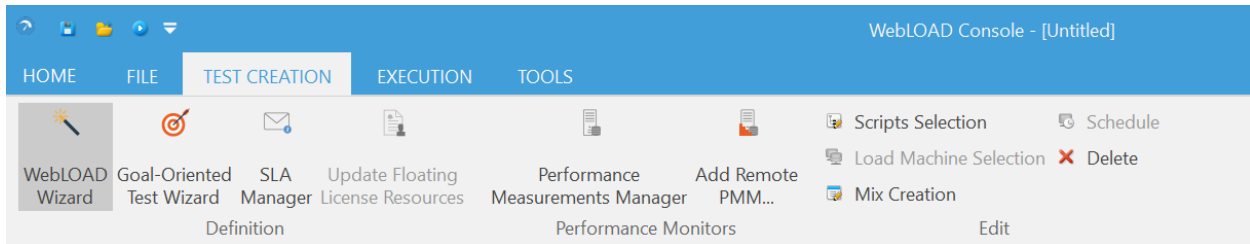


Figure 21: Console Ribbon

Right-click Menus

The right-click menus provide shortcuts to frequently performed actions.

The specific list of shortcuts available depends on the area where you click the right mouse button.

For example, the right-click menu displayed when you right-click the Load Session icon in the Session Tree in edit mode enables you to:

- Open the WebLOAD Wizard.
- Modify script selections.
- Delete the Load Template.
- Access the default and current session options.
- Access the SLA Manager.

Status Bar

At the bottom of the Console window, the status bar displays information about operations in progress.

While a session is running, the status bar displays continually updated information about the Elapsed time since the session started running, and the Remaining time till session end (if known).

The Session Tree

The Session Tree displays in the left pane of the Console screen, and gives you a complete graphical overview of the test session including the scripts run, and the hosts running each script. The icons adjacent to the tree items enable you to view your test activity at a glance.

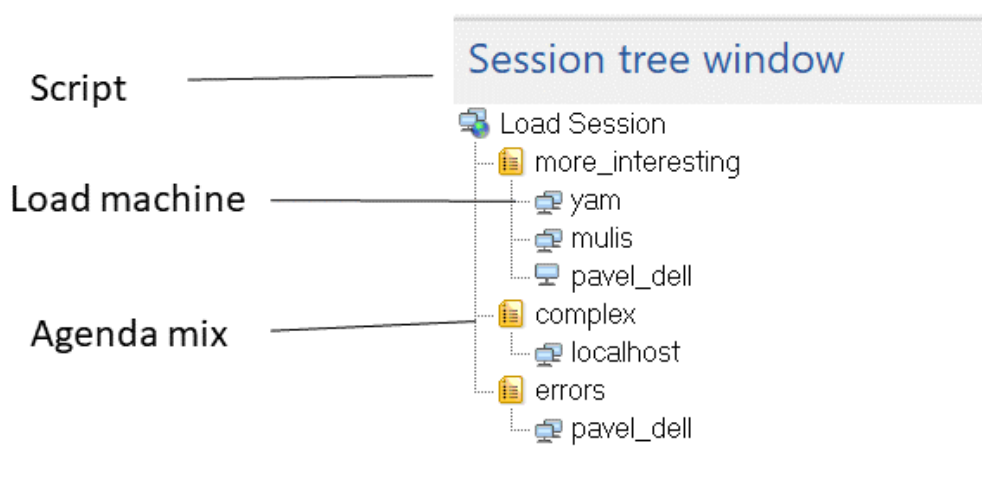
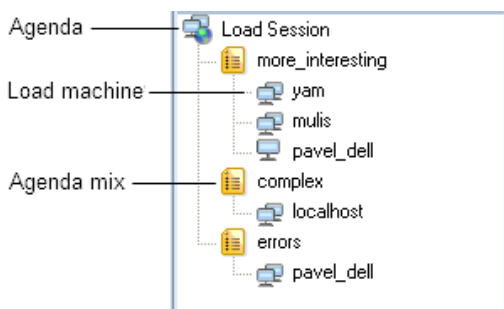


Figure 22: Session Tree

During the test configuration phase, the Session Tree is a convenient interface for editing the test session. Clicking the **WebLOAD Wizard** icon in the Home tab of the ribbon starts the WebLOAD Wizard. Double-clicking a tree node opens the dialog box for configuring that tree item. Right-clicking a tree item prompts a menu that enables you to perform a number of functions on the test session. As you change your configuration through the Console configuration dialog boxes, the changes that you make are reflected in the Session Tree.

While your test session is running, the Session Tree displays its progress. The Session Tree displays the scripts configured for running, the participants in the test, and their status. For more information on the status of the test, see *Status Icons* (on page 51).

The Session Tree Structure

- The “root” of the tree, in the upper-most line, displays the Load Session icon.
- The first level branches display the scripts to be run during the test session. The scripts may be single scripts or a Mix of scripts. The icons displayed next to the script name indicate its status. For more information on the status of Session components, see *Status Icons* (on page 51).
- The second level branches present the hosts participating in the test session. These hosts can be Load Machines or Probing Clients. A different icon represents each host type. The icons displayed adjacent to the hosts indicate their status. Several hosts can run the same script.

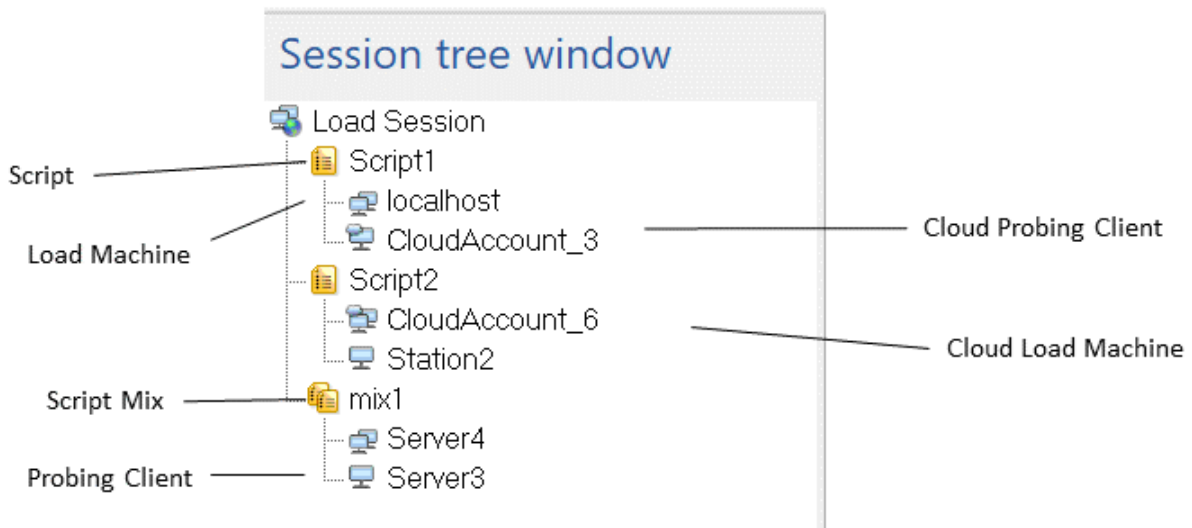









Figure 23: Session Tree Structure

Session Tree Icons

WebLOAD Console displays icons on each line of the Session Tree, making it easy to view your test elements and activity.

The Session Tree icons are described in the following table:

Table 10: Session Tree Icons













Icon	Session Component
	Load Session – The root of the Session Tree.
	Script – A single test script.
	Mix of scripts – A combination of test scripts.
	Load Machine – A machine used for generating Virtual Clients.
	Cloud Load Machine – A cloud machine used for generating Virtual Clients.
	Probing Client Machine – A machine used for generating Probing Clients.
	Cloud Probing Client Machine – A cloud machine used for generating Probing Clients.

Status Icons

During runtime, status icons appear adjacent to each Load Session item (hosts, scripts) in the Session Tree, indicating the item’s status.

The status icons are described in the following table.

Table 11: Status Icons

Icons	Cloud Icons	Description
		No current activity.
		Initialization completed successfully. Ready to run.
		Currently running.
		Temporarily paused. Remainder of Load Session continues running as usual.
		Currently stopped.
		An error has occurred.

Navigating the Session Tree

You can move through the Session Tree by expanding and collapsing nodes.

The following table lists the keystrokes you can use to move in the Session Tree:

Table 12: Session Tree Keystrokes

Key	Result
Up arrow	Moves the selection up one item in the Session Tree.
Down arrow	Moves the selection down one item in the Session Tree.
Right arrow	Expands the selected item. If the selected item does not contain hidden items, behaves like down arrow.
Left arrow	Collapses the selected item. If the selected item does not contain exposed items, behaves like up arrow.
Page Up / Home	Moves the selection to the first item in the Session Tree.
Page Down	Moves the selection to the bottom item visible in the pane.
End	Moves the selection to the last item in the Session Tree.
Plus sign "+" on the numeric keypad	Expands the selected node.
Minus sign "-" on the numeric keypad	Collapses the selected node.
Asterisk "*" on the numeric keypad	Expands the entire tree below the selected node.

The Session Tree in Edit Mode

In edit mode, the Session Tree provides a complete graphical overview of the test session.

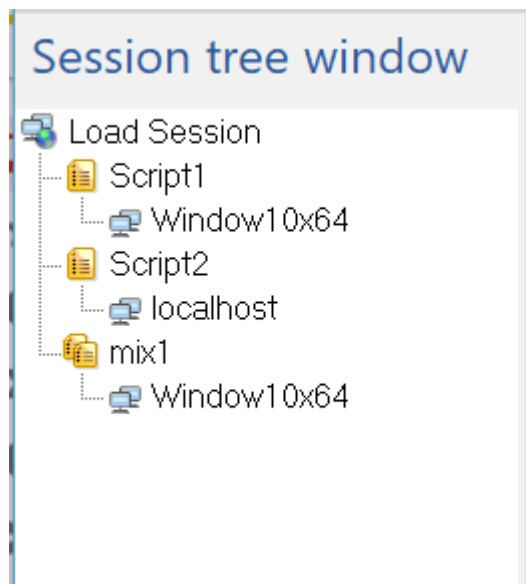









Figure 24: Edit Mode Session Tree

The following is a line-by-line explanation of the above Session Tree:

Table 13: Edit Mode Session Tree

Line Num	Icon	Explanation
1	 Load Session	Indicates a Session Tree.
2	 Script1	The script file to be run during the Load Session.
3	 Windows7x64	Describes the Load Machine for this test session. The Load Machine icon is followed by the host name (Windows7x64). A test session may include multiple Load Machines.

Line Num	Icon	Explanation
4	 Script2	The script file to be run during the Load Session.
5	 localhost	Describes the Probing Client Machine host for this test session. The Probing Client icon is followed by the Host name (localhost). A test session may include multiple Probing Client hosts.
6	 Mix 1	The Mix file to be run during the Load Session.
7	 Windows7x64	Describes the Load Machine for this test session. The Load Machine icon is followed by the host name (Windows7x64).

Right-Click Options in Edit Mode

Using the Session Tree you can define and modify Load Templates.

The following options can be accessed by right-clicking a Session Tree component in edit mode:

Table 14: Edit Mode Session Tree Options

Tree Item	Options	Description
Load Session	Open Wizard	Opens the WebLOAD Wizard.
	Modify Scripts Selection	Opens the script / Mix Selection dialog box to modify the selections.
	Delete	Deletes the selected WebLOAD Console component and all sub-components.
	Default Options	Opens the Default Options dialog box and enables setting new default options for each script added to a Load Session.
	Current Session Options	Opens the Current Session Options dialog box and enables setting options for each script currently defined in the Load Session.
	SLA Manager	Opens the SLA Manager dialog box for defining the conditions under which you want to receive log messages and optionally also an email notification.

Tree Item	Options	Description
Script/Mix	Open Wizard	Opens the WebLOAD Wizard.
	Modify Load Machine Selection	Opens the Host Selection dialog box and modify the Load Machines and Probing Client Machines defined for the session.
	Modify Load Machine Schedule	Opens the Schedule Manually dialog box and modify the test schedule.
	Open	Opens the script in WebLOAD Recorder for editing.
	Delete	Deletes the script.
	Rename	Renames the script.
	View JavaScript	Opens a window displaying the JavaScript for the script.
	Copy Load Machine and Schedule	Available for script only. Copies the Load Machines and the load schedule defined for this script to the clipboard.
	Paste Load Machine and Schedule	Available for script only. Pastes the Load Machines and the load schedule from the clipboard.
	Script Options	Available for script only. Opens the Script Options dialog box. For more information on these settings, see <i>Setting Script Options</i> (on page 211).
	Mix Options	Available for Mix only. Opens the Mix Creation dialog box, enabling you to select a script from the Mix for which to change the options.
Load Machine and Probing Client Machine	Modify Schedule	Opens the Schedule Manually dialog box enabling you to modify the scheduling parameters.
	Replace	Replaces the host (from the host list).
	Delete	Deletes the host.
	Copy Schedule	Copies the schedule defined for the host to the clipboard.
	Paste Schedule	Pastes the schedule from the clipboard.

The Session Tree in Run Mode

The Session Tree displays the complete configuration of the current Load Session. Status icons reflect the status of your Load Session.

Through the Session Tree you can see:

- The scripts running in the test session.
- The hosts running each script.
- Each script and host's current operating status.

Icons are displayed on each line of the tree, making it easy to view the test activity.

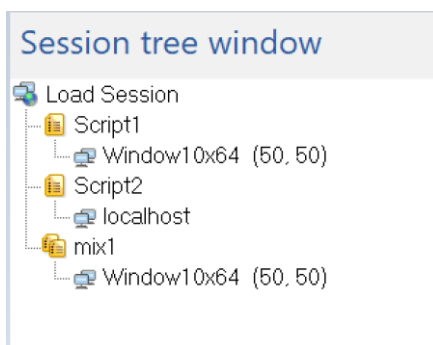









Figure 25: Run Mode Session Tree

The following is a line-by-line explanation of the above Session Tree:

Table 15: Run Mode Session Tree

Line Num	Icon	Explanation
1	 Load Session	Indicates a Session Tree.
2	 Script1	The script file to be run during the Load Session.

Line Num	Icon	Explanation
3	 Windows7x64 (100, 100)	<p>Describes the Load Machine for this test session.</p> <p>The Load Machine icon, along with its status (currently running), is followed by the host name (Windows7x64).</p> <p>A test session may include multiple Load Machines.</p> <p>Two numbers follow each Load Machine entry:</p> <ul style="list-style-type: none"> • The first number is the maximum load that script schedules it to generate over the course of a test session. • The second number is the current load being generated. <p>In this example, Windows7x64 is running under Script1, and has a maximum scheduled load of 100 Virtual Clients and is currently generating a load of 100 Virtual Clients.</p>
4	 Script2	The script file to be run during the Load Session.
5	 localhost	Describes the Probing Client Machine host for this test session. The Probing Client icon, along with its status (currently running), is followed by the Host name (localhost). A test session may include multiple Probing Client hosts.
6	 Mix 1	The Mix file to be run during the Load Session.
7	 Windows7x64 (100, 100)	<p>Describes the Load Machine for this test session. The Load Machine icon, along with its status (currently running), is followed by the host name (Windows7x64).</p> <p>Two numbers follow each Load Machine entry:</p> <ul style="list-style-type: none"> • The first number is the maximum load that Mix schedules it to generate over the course of a test session. • The second number is the current load being generated. <p>In this example, Windows7x64 is running under Mix1, and has a maximum scheduled load of 100 Virtual Clients and is currently generating a load of 100 Virtual Clients.</p>

To the right of each Load Machine two numbers are displayed in parentheses, for example, sharony (25, 45). These display the load size currently being generated; followed by the maximum load size (number of Virtual Clients) the script is scheduled to generate. The current load size number will change according to the load schedule defined in the Schedule dialog box. Before a test session begins, the 'current' load size being generated is 0. In most modes of operation, the current load size will never exceed the maximum load size scheduled for that Script listed to the right of the current value.



Note: Using Throttle Control, you can increase the current load size to exceed the maximum load size - but not while running the Goal-Oriented Test Session.

In run mode you can right-click an item in the Session Tree to view the test parameters defined in the session.

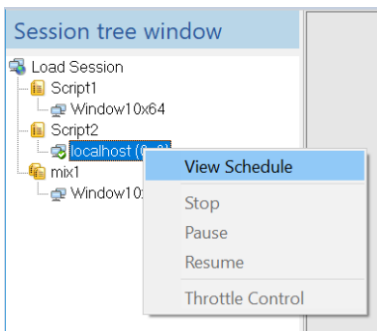


Figure 26: Session Test Parameters Menu

Right-Click Options in Run Mode

While in run mode, the following options can be accessed through the right mouse button:

Table 16: Run Mode Session Tree Options

Tree Item	Options	Description
Load Session	View Scripts Selection	Opens the script / Mix Selection dialog box.
	Stop All	Controls the test in progress.
	View Goals	Opens the View Goals message box (available only during the Goal-Oriented Test Sessions) to display the goals defined in the test session.
	Reset all Load Generators	Resets the Load Generators.

Tree Item	Options	Description
	Pause	Pauses the test session in progress.
	Resume	Resumes execution of tests that have been paused.
	Throttle Control	Opens the Throttle Control dialog box, enabling you to dynamically change the load generated on the fly. Throttle Control for the Load Session divides the new VC number evenly across all scripts.
	Default Options	Opens the Default Options dialog box enabling you to view the default options for each script added to a Load Session.
	Current Session Options	Opens the Default Options dialog box enabling you to view the options for each script currently defined in the Load Session.
	SLA Manager	Opens the SLA Manager dialog box for defining the conditions under which you want to receive log messages and optionally also an email notification.
Script / Mix	View Load Machine Selection	Opens the Load Machine Selection window.
	View Load Machine Schedule	Opens the Load Machine Schedule window displaying the schedule defined for the script.
	Stop	Controls the test in progress, enabling you to stop running the selected script.
	Pause	Pauses the selected script while the test is in progress.
	Resume	Resumes execution of the selected Scripts that have been paused.
	Throttle Control	Opens the Throttle Control dialog box, enabling you to dynamically change the load generated on the fly. Throttle Control for the script/Mix divides the new VC number evenly across all Load Generator machines.
	Open	Opens the script in WebLOAD Recorder.
	View Java Script	Displays the JavaScript code in an external window.
	Script / Mix Options	Opens the View Script / Mix dialog box, enabling you to edit the script options.

Tree Item	Options	Description
Load Machine	View Schedule	Opens the Load Machine Schedule dialog box displaying the schedule defined for the Load Machine.
	Stop	Controls the test in progress, enabling you to stop running the selected Load Machine.
	Pause	Pauses the selected Load Machine while the test is in progress.
	Resume	Resumes execution of the selected Load Machine while the test is in progress.
	Throttle Control	Opens the Throttle Control dialog box, enabling you to dynamically change the load generated on the fly. Throttle Control for the Load Machine only changes the number of VCs for this Load Generator machine.



Note: In edit mode, the items in the pop-up menus are used to edit the test session configuration. See *The Session Tree in Edit Mode* (on page 53).

The Results Window

The Results window displays all of the reports opened during a test session. Use the tabs located at the top and the bottom of the window to view different reports. The following reports are available in the Results window.

Table 17: Results Window

Report	Description
Integrated Reports	A report that can be configured to combine Performance data from the Load Machines, Probing Clients, Performance data on the SUT gathered from Performance Manager, and existing performance data from previously saved Load Sessions and external files.
Statistic Reports	A real-time monitor that displays the results of your test while you run it that can be saved and exported.
WebLOAD Dashboard	Provides a summary of the performance statistics generated during the test session.
Transactions Dashboard	A real-time monitor that displays the transactions statistics in your test.

Report	Description
Data Drilling Reports	Provides a detailed description of all user-defined and named transactions to the instance level, including all hit failures and the reasons for them.
Viewer	Provides a graphic presentation of event failures displaying the actual events.

Integrated Reports

WebLOAD Integrated Reports provide both a graphical and statistical view of the performance of your application as it is being tested. Integrated Reports can be viewed while the test is in progress or saved for later analysis.

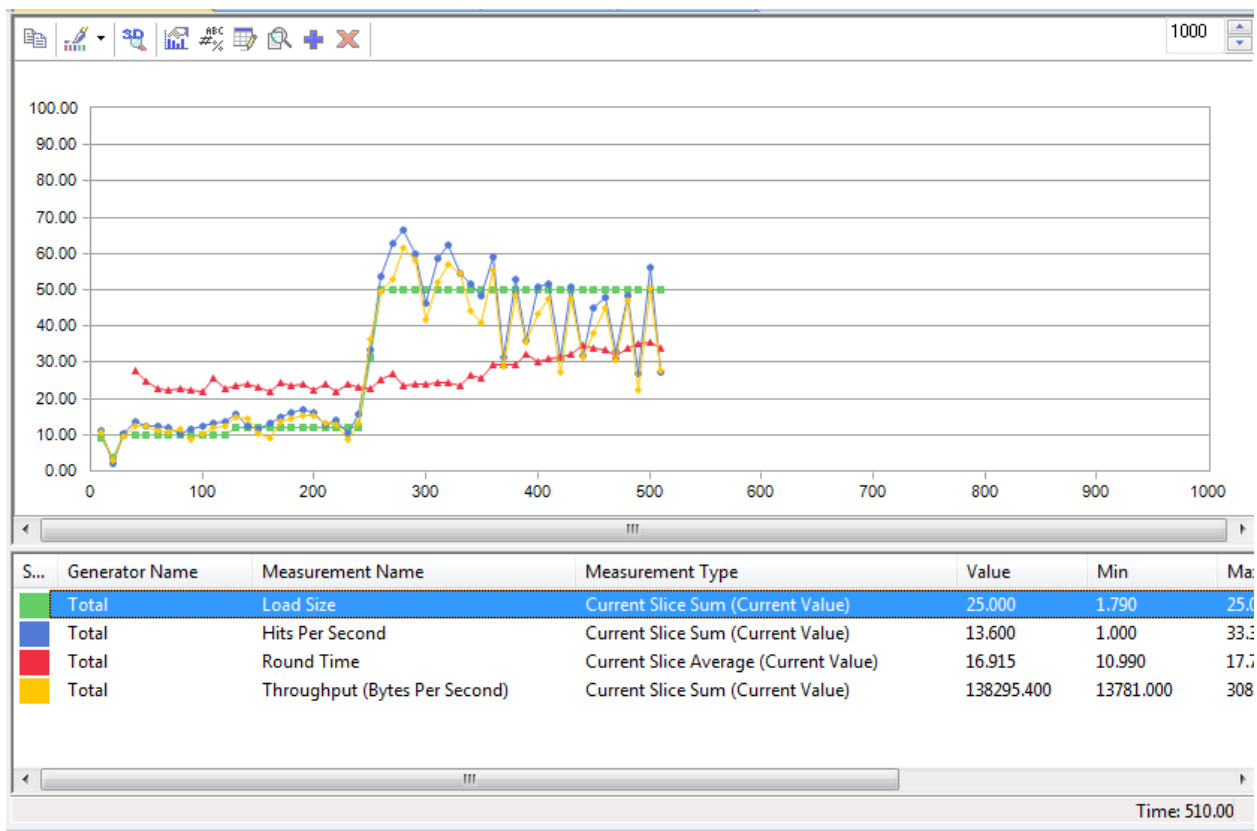


Figure 27: Integrated Report

Statistics Reports

WebLOAD collects approximately 35 different statistics during a test. Statistics Reports display the values for all of them.

Measurements (Current Values)	Total	Quickstart:Total	Quickstart@jryakovb	PM@JRYAAKOVB
Load Size	1.000	1.000	1.000	-
Round Time	-	-	-	-
Rounds	0.000	0.000	0.000	-
Successful Rounds	0.000	0.000	0.000	-
Rounds Per Second	0.000	0.000	0.000	-
Successful Rounds Per	0.000	0.000	0.000	-
Page Time	-	-	-	-
Pages	0.000	0.000	0.000	-
Pages Per Second	0.000	0.000	0.000	-
DNS Lookup Time	-	-	-	-
Hit Time	-	-	-	-
Hits	0.000	0.000	0.000	-
Successful Hits	0.000	0.000	0.000	-
Hits Per Second	0.000	0.000	0.000	-
Successful Hits Per Second	0.000	0.000	0.000	-
Attempted Connections	0.000	0.000	0.000	-
Successful Connections	0.000	0.000	0.000	-
Connect Time	-	-	-	-
Process Time	-	-	-	-
Receive Time	-	-	-	-
Send Time	-	-	-	-
Time To First Byte	-	-	-	-
Throughput (Bytes Per	0.000	0.000	0.000	-
Response Time	-	-	-	-
Response Data Size	0.000	0.000	0.000	-
Responses	0.000	0.000	0.000	-
Content Size	-	-	-	-
GET_http://ad.doubleclick.ne	-	-	-	-
GET_http://ad.doubleclick.ne	-	-	-	-
GET_http://ad.doubleclick.ne	-	-	-	-
GET_http://ad.doubleclick.ne	-	-	-	-
GET_http://clients1.google.c	-	-	-	-
GET_http://googleads.g.dou	-	-	-	-
GET_http://googleads.g.dou	-	-	-	-
GET_http://googleads.g.dou	-	-	-	-
GET_http://googleads.g.dou	-	-	-	-

Double click on a column or row header or any table cell to view statistical details. Load Size: 1 Time: 1080

Figure 28: Statistics Report

Dashboard

The Dashboard displays real-time statistical information about the test session including the number of Virtual Clients running, hits per day, pages per day and throughput.

Measurement	Current Value	Average Value
Time	1340.00	
Running Virtual Clients	1	
Rounds Executed	3	
Failed Rounds	0	
Failed Hits	0	
Hits Per Second	1	1
Calculated Hits Per Day	86400	86400
Pages Per Second	1	1
Calculated Pages Per Day	86400	86400
Throughput	446	2248
Calculated Throughput Per Day	37 MB	186 MB

Figure 29: Dashboard

Transactions Dashboard

The Transactions Dashboard displays real-time statistical information about the transactions in your test, in graphical format.

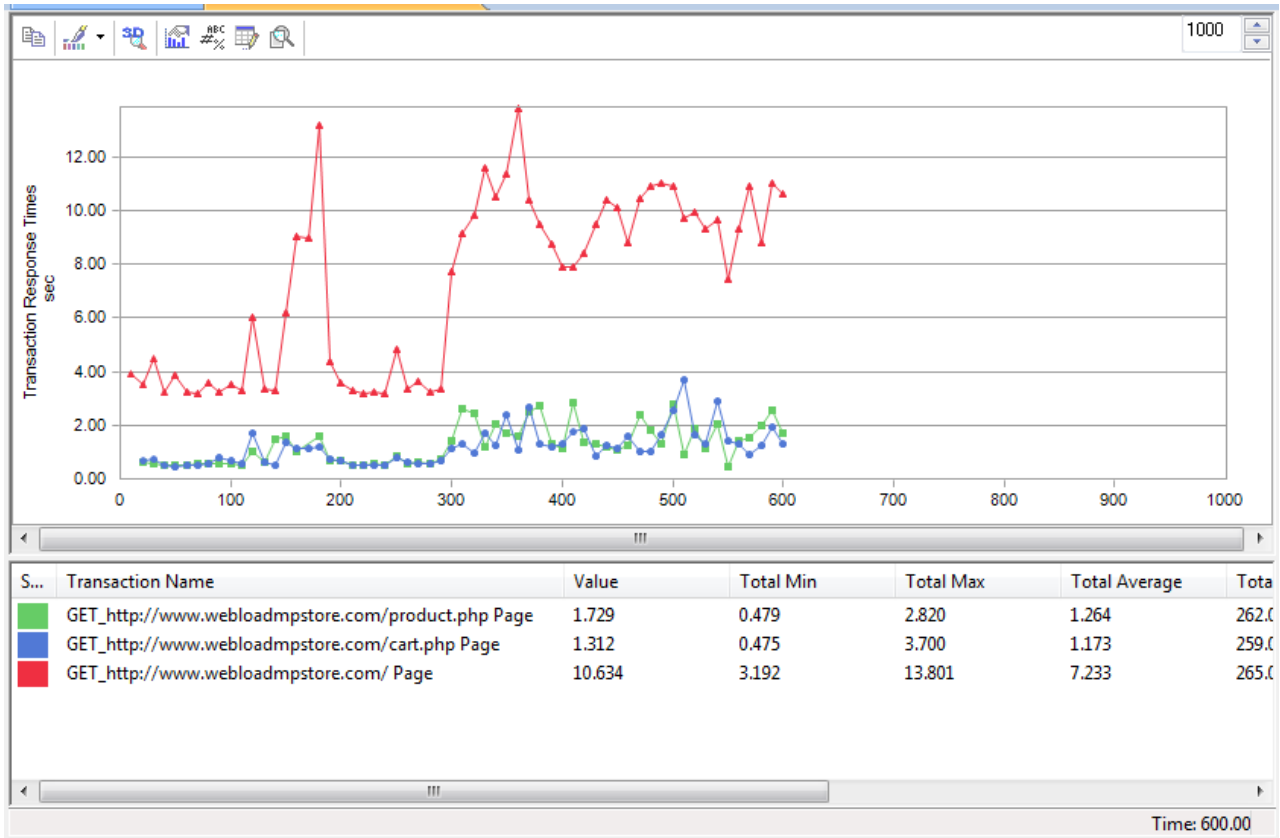


Figure 30: Transactions Dashboard

Data Drilling Reports

Data Drilling provides both a global and detailed account of hit successes and failures allowing you to verify the functional integrity of your Web application at the per-client, per-transactions, and per-instance level. The Data Drilling reports provide an extremely detailed yet easily accessible summary of all the statistical, timing, and performance information collected over the course of the test session.

Transaction Name	Total Count	Successful Count	Failed Count	Marked Count
GET_http://www.webloadmpstore.com/category.php F	12	12	-	-
GET_http://ad.doubleclick.net/adj/imeem.playlists/mus	4	4	-	-
GET_http://www.youtube.com/embed_api_rest Page	4	4	-	-
GET_http://www.webloadmpstore.com/cart.php Page	4	4	-	-
GET_http://www.youtube.com/crossdomain.xml Page	4	4	-	-
GET_http://www.google.com/ Page	5	5	-	-
GET_http://pagead2.google syndication.com/pagead/i	4	4	-	-
GET_http://www.imeem.com/ads/bannerad.ashx Page	16	16	-	-
POST_http://ads.imeem.com/ads/amf/ Page	4	4	-	-
GET_http://ad.doubleclick.net/adj/imeem.playlists/mus	4	4	-	-
GET_http://i2.yimg.com/crossdomain.xml Page	4	4	-	-
GET_http://i4.yimg.com/crossdomain.xml Page	4	4	-	-
GET_http://i3.yimg.com/crossdomain.xml Page	4	4	-	-
GET_http://www.webloadmpstore.com/index.php Page	4	4	-	-
GET_http://googleads.g.doubleclick.net/pagead/imga	12	12	-	-
POST_https://urs.microsoft.com/urs.asmx Page	8	8	-	-
GET_http://ad.doubleclick.net/adj/imeem.playlists/mus	4	4	-	-
POST_http://www.imeem.com/api/GetRenderedRelat	8	8	-	-
GET_http://googleads.g.doubleclick.net/pagead/goog	12	12	-	-
GET_https://www.webloadmpstore.com/checkout.php	4	3	1	-
GET_http://www.youtube.com/v/GvhJht7vZ7U Page	4	4	-	-
GET_http://www.webloadmpstore.com/login.php Page	4	4	-	-
GET_http://www.webloadmpstore.com/product.php P:	4	4	-	-
GET_http://s.yimg.com/yt/swf/cps-vfl81855.swf Page	4	4	-	-
POST_http://www.imeem.com/api/amf/ Page	56	56	-	-
GET_http://ad.doubleclick.net/adj/imeem.playlists/mus	4	4	-	-
GET_http://googleads.g.doubleclick.net/pagead/ads f	16	16	-	-
GET_http://www.youtube.com/get_video_info Page	4	4	-	-
GET_http://clients1.google.com/generate_204 Page	5	5	-	-
GET_http://www.webloadmpstore.com/ Page	4	4	-	-

Figure 31: Data Drilling Report

Event Viewer

The Event Viewer displays an illustration of each event triggered during the test.

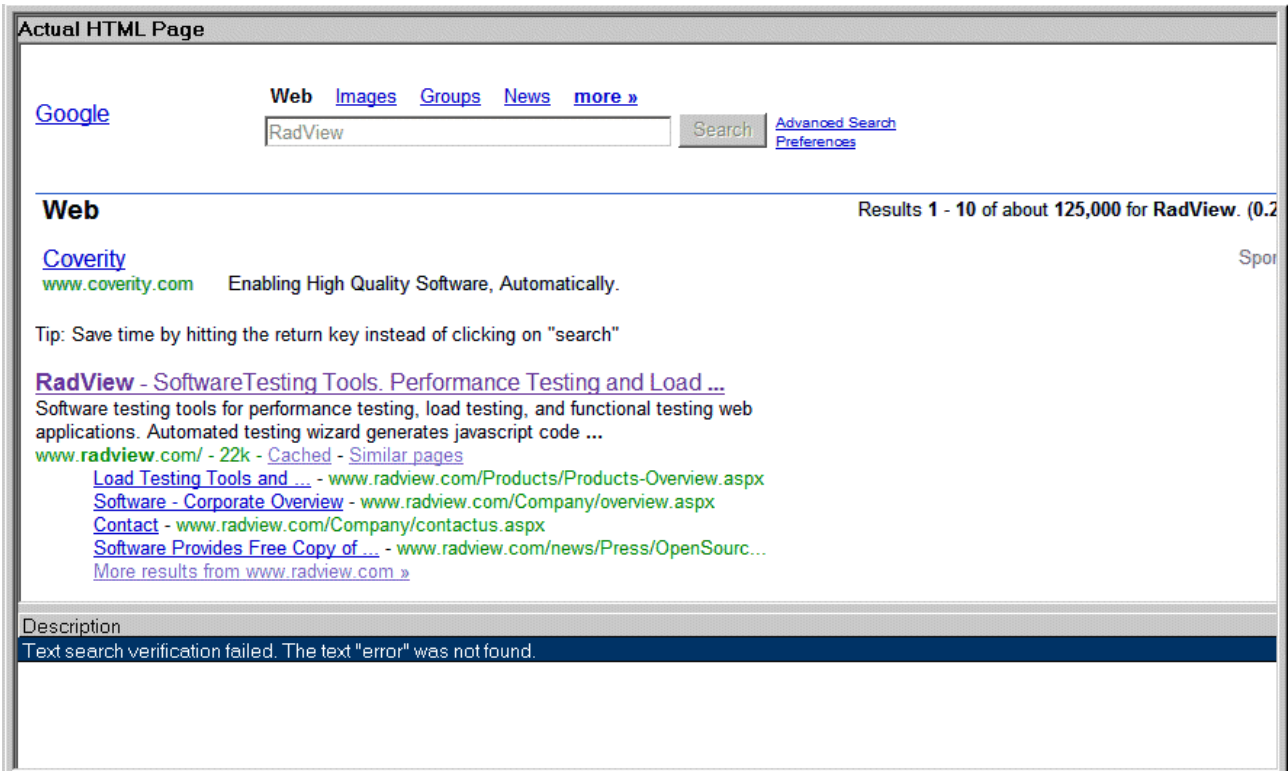


Figure 32: Event Viewer

For further information regarding use of the Event Viewer, see *Viewing Error Information in the Event Viewer* (on page 266).

Log Window

The Log Window displays a summary of the test including all log messages detected by WebLOAD Console in run time that are generated by the Console, the JavaScript compiler and any user messages programmed in the test script.

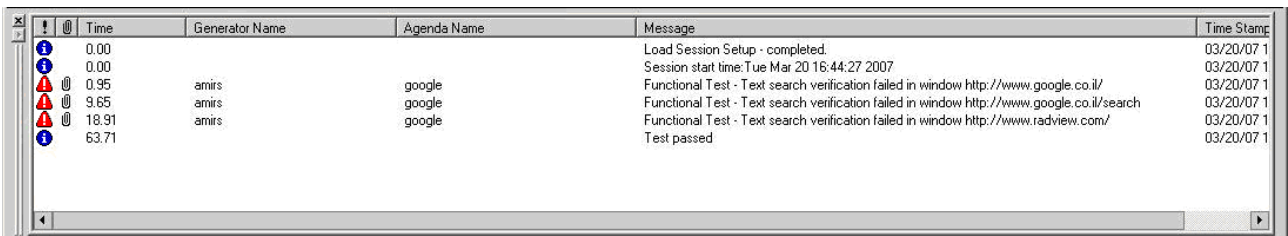


Figure 33: Log Window

For further information regarding use of the Log Window, see *Viewing Error Information in the Log Window* (on page 265).

The WebLOAD Console Startup Dialog Box

When you start WebLOAD Console, the WebLOAD Console Startup dialog box displays at the center of the screen, providing you with direct access to the WebLOAD Wizards, Load Templates and previously run Load Sessions.

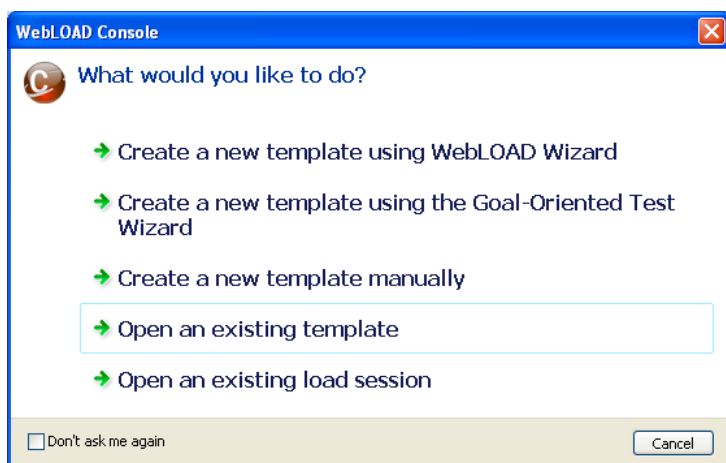


Figure 34: Console Startup Dialog Box

Table 18: Console Startup Dialog Box Items

Item	Description
WebLOAD Wizard	Opens the WebLOAD Wizard for creating a new Load Template.
Goal-Oriented Test Wizard	Opens the Goal-Oriented Test Wizard for creating a new Load Template.
Manual	Opens a new blank Session Tree for creating a new template.
Load Template	Opens a saved Load Template (*.tpl). Enter the name of the saved Load Template in the edit box.
Load Session	Opens a saved Load Session (*.ls) including template data from a completed test. Enter the name of the saved Load Session in the edit box.


Use either the Goal-Oriented Test Wizard to develop a goal-seeking performance test or the WebLOAD Wizard to develop a basic Load Template. Both of these wizards enable you to save your test parameters as a template that you can later edit manually using the Console ribbon.

After developing a test with either of the wizards, you can save the test parameters as a Load Template. After running the test you can save the test parameters and results as a Load Session. You can then edit the test parameters by editing the Load Template or view the Load Session test results. For more information on Load Sessions and Load Templates, see *Managing Load Templates and Load Sessions* (on page 91).


Edit Mode and Run Modes of Operation

When you open a new test, WebLOAD Console is in Edit mode. In edit mode, you can edit the parameters of your test.

When you run your test, WebLOAD Console switches to Run mode. When in run mode, you cannot edit the parameters of the test.

If you are in Run mode and you wish to edit the test parameters, click the **Edit Template**  button in the Session tab of the ribbon to switch to edit mode.

The Console in Edit Mode

In edit mode you can edit the Session Tree parameters. In run mode WebLOAD Console displays the configuration dialog boxes. Change from Run mode to Edit mode using the **Edit Template**  button in the Session tab of the ribbon.

To edit test session parameters:

- Place the cursor on a node in the Session Tree and click the right mouse button,
-Or-
Double-click an item in the Session Tree.

The Console in Run Mode

When you start running your Load Session, the Results window appears.

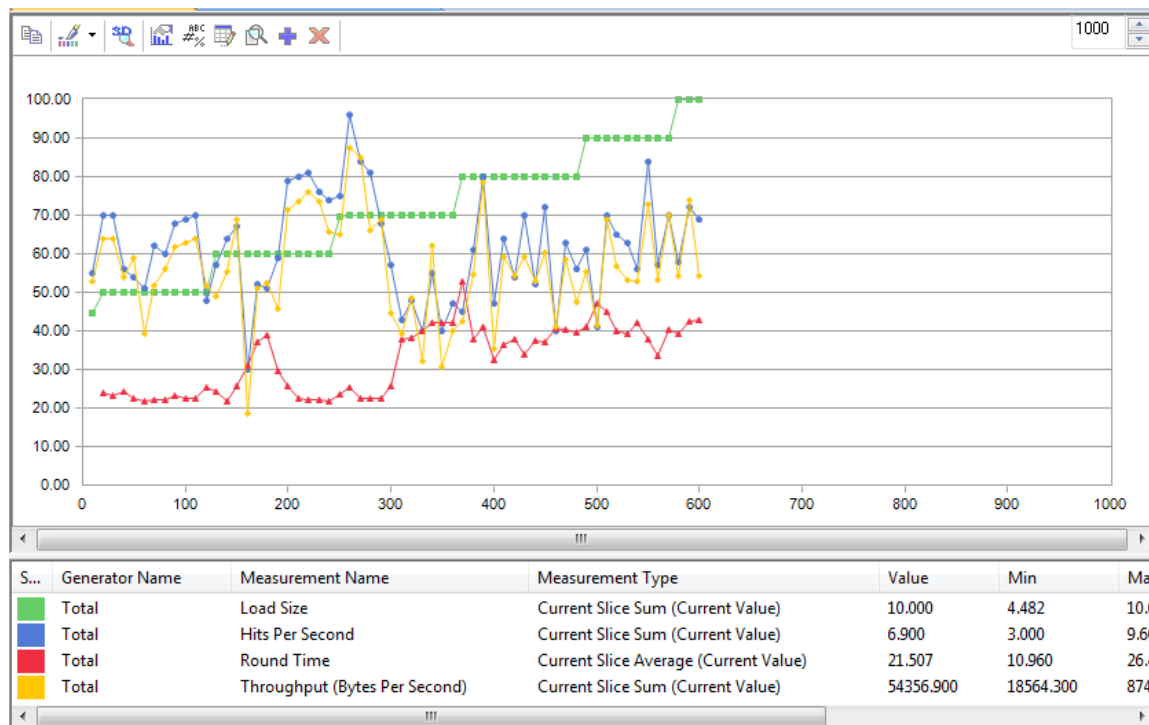


Figure 35: Results Window

The following table describes the segments of the Console in Run mode:

Table 19: Run Mode Console Segments

Segment	Function
Session Run Toolbar	Displays only during a test run. It contains buttons for controlling the test session.
Session Tree	Presents a graphic display of your test session.
Results Window	Displays all of the reports opened during a test session. Use the tabs located at the top and the bottom of the window to view different reports.
Log Window	Displays all of the error messages recorded during a test session. You can toggle the Log Window display on/off through the Session tab of the Console ribbon.
Status Bar	Indicates the program status.

The Console Ribbon Options

The Console ribbon is located at the top of the screen.

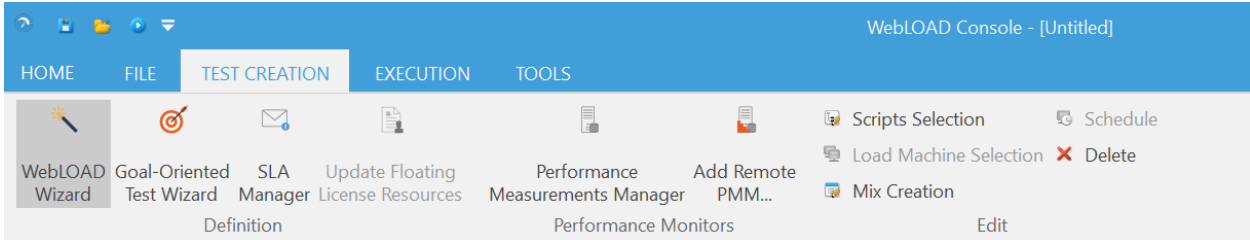


Figure 36: Console Ribbon

This section discusses each of the WebLOAD Console ribbon tabs and their options .

File Tab

The File tab includes the following functions for WebLOAD Console file management.



Figure 37: File Tab in Console Ribbon

Table 20: File Tab Options

Tab Option	Description
<i>File Group</i>	
New	Starts a new test session, creating a new WebLOAD Load Template (*.tpl) file. When you start a new Load Template an empty Session Tree is displayed on the screen.
Open	Enables opening any of the following: <ul style="list-style-type: none"> A saved Load Session. A Load Session is saved as a file of type *.ls. The Console displays a Session Tree window initialized to the settings saved in the file. A Load Template file of the type *.tpl. A Load Template file contains test configuration, the assigned scripts, the testing schedule, and the report configuration, without the test session results.

Tab Option	Description
Save	<p>Enables performing a Save or Save As operation</p> <p>If the session tree is currently displaying a Load Session, the Save operation saves the current Load Session configuration and results to a file of type *.ls. The following information is saved:</p> <ul style="list-style-type: none"> • Selected hosts, both Probing Clients and Load Machines • Assigned scripts for each host • Test session schedule • Report configuration • Test session results (Performance Report data) <p>If the session tree is currently displaying a Load Template, the following information is saved:</p> <ul style="list-style-type: none"> • Selected hosts, both Probing Clients and Load Machines • Assigned scripts for each host • Test session schedule • Report configuration
Print	Prints the report displayed with all contents and formats.

Home Tab

The Home tab includes functions for defining a Load Template

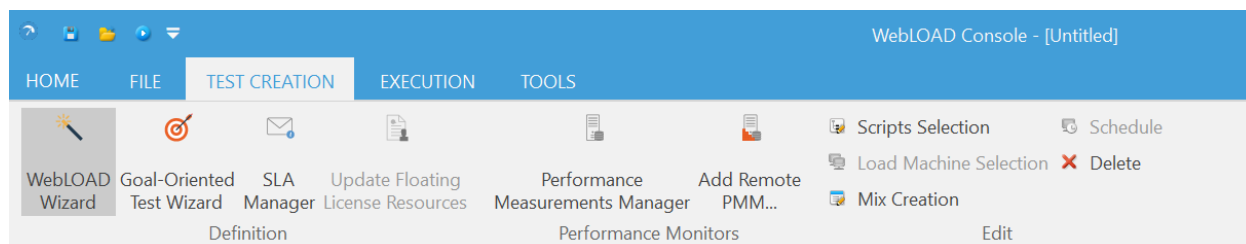


Figure 38: Home Tab in Console Ribbon

Table 21: Home Tab Options

Tab Option	Description
<i>Definition Group</i>	
WebLOAD Wizard	Opens the WebLOAD Wizard. This wizard walks you through the WebLOAD Console configuration process.
Goal-Oriented Test Wizard	Opens the Goal-Oriented Test wizard. The Goal-Oriented Test wizard enables you to run goal-seeking performance testing.

Tab Option	Description
SLA Manager	Opens the SLA Manager dialog box for defining the conditions under which you want to receive log messages and optionally also an email notification.
Update Floating License Resources	Opens the Update Resources dialog box.
<i>Performance Monitors Group</i>	
Performance Measurements Manager	Opens the Performance Measurements Manager dialog box for configuring the statistics parameters to be monitored.
Add Remote PMM	Open the Add PMM Probing Client dialog box for adding a Remote PMM script to the template.
<i>Edit Group</i>	
Scripts Selection	Opens the script / Mix Selection dialog box to modify the selections.
Load Machine Selection	This option is enabled only when a script is selected in the session tree. Opens the Host Selection dialog box to enable modifying the list of Load Machines and Probing Client Machines defined for the selected script or mix.
Mix Creation	Available for Mix only. Opens the Mix Creation dialog box, enabling you to select a script from the Mix for which to change the options.
Schedule	This option is enabled only when a load machine or script are selected in the session tree. <ul style="list-style-type: none"> • When a script is selected, opens the Schedule Manually dialog box, enabling you to modify the scheduling parameters for all the load machines running the script. • When a single load machine is selected, opens the Schedule Manually dialog box, enabling you to modify the scheduling parameters for the load machine.
Delete	Deletes the currently selected component. A message box pops up, asking for confirmation.

Session Tab

The Session tab is used to manage Load Sessions and view the results.

The Reports group provides options for creating real-time and summary reports showing the performance data gathered by WebLOAD Console. A full description of all WebLOAD reports can be found in *Running a Load Session* (on page 271).



The Session tab includes the following functions for Load Session management.

Table 22: Session Tab

Tab Option	Description
<i>Start Editing Group</i>	
Edit Template	Switches from Run mode to Edit mode.
<i>Execution Group</i>	
Start Session	Starts executing the current Load Session.
Stop All	Stops all Load Session activities. A message box pops up, asking for confirmation.
Throttle Control	Opens the Throttle Control dialog box. Use Throttle Control to dynamically change the load while the session is in progress.
<i>Reports Group</i>	
Integrated Report Manager	Opens an existing report from the reports list
Transactions Dashboard Window	Opens a Transactions Dashboard tab, which displays real-time updates of transaction statistics in a graphical display
Open Statistics	Opens a Statistics tab, which displays real-time updates of the statistics measurements for Load Machines, Probing Clients, and the performance measurements
Dashboard Window	Opens a WebLOAD Dashboard tab, which displays real-time updates of statistic measurements in a dashboard-type display.
Open Data Drilling	Opens a Transactions tab, which displays information on all user-defined and named transactions

Tab Option	Description
<i>Export Statistics Group</i>	
Export Report	Provides three options: <ul style="list-style-type: none"> • Export to Excel – Exports the displayed chart or grid to Microsoft Excel. • Export to Tab File – Exports the displayed chart or grid to a tab file • Export to HTML– Exports the displayed chart or grid to HTML.
Export All	Provides three options: <ul style="list-style-type: none"> • Export All to Excel – Exports all statistics to Microsoft Excel. • Export All to Tab File – Exports all statistics to a tab file • Export All to HTML– Exports all statistics to HTML.
Import	Opens the Import External Statistics dialog box to import statistics from an external file.
<i>Windows Group</i>	
Cascade Windows	Cascades the windows displayed in the WebLOAD Results window.
Tile Horizontally	Tiles the windows displayed in the WebLOAD Results window horizontally.
Tile Vertically	Tiles the windows displayed in the WebLOAD Results window vertically.
<i>View Group</i>	
Log Window	A toggle enabling hiding and displaying the Log Window. The Log window is both floating and dockable, and can be placed in any part of the screen.

Tools Tab

The Tools tab provides the following functions.

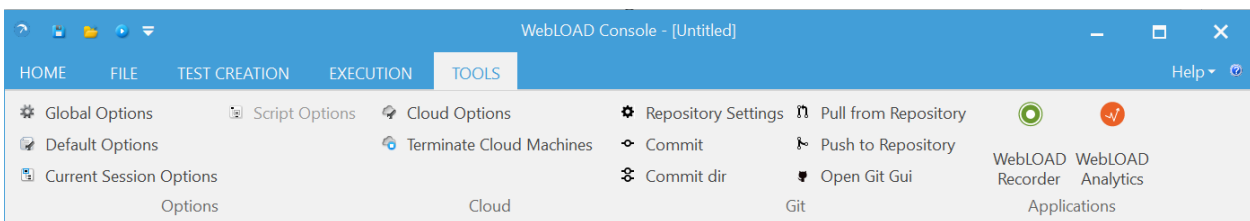


Table 23: Tools Tab

Tab Option	Description
<i>Options Group</i>	

Tab Option	Description
Global Options	Opens the Global Options dialog box. Use the global options to customize WebLOAD Console to your specifications, see <i>Setting Global Options</i> (on page 177).
Default Options	Opens the Default Options dialog box. Use the default settings to customize the script settings to your specifications. When building a new template or session, or editing an existing one, each new single script or script participating in a Mix is automatically assigned these defaults, see <i>Setting Script Options</i> (on page 211).
Current Session Options	Opens the Current Session Options dialog box. Use the session settings to customize the current session to your specifications. This option will update the options for all scripts in the current session, see <i>Setting Script Options</i> (on page 211).
Script Options	Opens the Script Options dialog box. Use the session settings to customize the current script to your specifications, see <i>Setting Script Options</i> (on page 211).
<i>Cloud Group</i>	
Cloud Options	Opens the Cloud Accounts dialog box, for creating a WebLOAD cloud account
Terminate Cloud Machines	Shuts down the cloud machines.
<i>Git Group</i>	
Repository Settings	Enables specifying basic Git repository settings, as prerequisites for WebLOAD support of basic Git operations.
Commit	Enables saving to the Git local repository any WebLOAD objects you are currently working on.
Commit dir	Enables saving to the Git local repository an entire folder with all its descendants.
Pull from Repository	Enables pulling data from the Git remote repository to the Git local repository.
Push to Repository	Enables pushing to the Git remote repository, all Commits you had made to the Git local repository.

Tab Option	Description
Open Git Gui	Enables launching your Git UI from within WebLOAD.
<i>Applications Group</i>	
WebLOAD Recorder	Opens WebLOAD Recorder. Use the WebLOAD Recorder to record, edit, and debug load test scripts. For more information, see the <i>WebLOAD Recorder User's Guide</i> .
WebLOAD Analytics	Opens WebLOAD Analytics. Use the WebLOAD Analytics to analyze data and create custom, informative reports after running a WebLOAD test session. For more information, see the <i>WebLOAD Analytics User's Guide</i> .

Help Drop Down

Table 24: Help Drop-down

Menu Entry	Description
Contents	Opens the WebLOAD Console Help screen, displaying the Contents tab.
Search	Opens the WebLOAD Console Help screen, displaying the Search tab.
JavaScript References	Opens the WebLOAD JavaScript Reference Manual Help screen, displaying the Contents tab.
Scripting Guide	Opens the WebLOAD Scripting Guide Help screen, displaying the Contents tab.
WebRM Help	Opens the WebRMUsers Help screen, displaying the Contents tab.
License Information	Opens the RadView License Information dialog box which displays information about your license.

About Button

About WebLOAD Console...	Displays the program information, version number, and copyright.
--------------------------	--

Creating Load Templates

Using WebLOAD Console, you create Load Templates to define the events to occur in the Load Session. This chapter describes how to quickly set up a Load Template. Load Templates define the participants (Clients and Servers) in the Load Session, the scripts to run, the number of Virtual Clients to emulate and the test schedule.

WebLOAD Console Options for Creating Load Templates

WebLOAD Console provides three options for creating Load Templates:

- Using the WebLOAD Wizard to build a basic Load Template. You define the scripts to run, the Load Machines on which to run the Virtual Clients and the test schedule. After creating a basic template with the wizard, you can change aspects of the configuration through the Console ribbon.
- Using the Goal-Oriented Test Wizard to create goal-seeking performance tests. Rather than running multiple tests to measure the performance at different loads, you can run the Goal-Oriented Test to measure the performance goal that you desire. You define the scripts to run and the goals to achieve and WebLOAD Console automatically schedules the test session and increases the number of Virtual Clients running until your goals are achieved.
- Manually using the Console ribbon.

The Load Template material is divided into the following chapters:

- *Planning a Test Session* (on page 79) describes the stages involved in planning and executing a test session.
- *Connecting to WebRM* (on page 87) describes the process of connecting to WebRM and requesting testing resources.
- *Managing Load Templates and Load Sessions* (on page 91) provides information about working with Load Templates and Load Sessions.
- *Creating Load Templates with the WebLOAD Wizard* (on page 99) guides you through a basic template configuration using the WebLOAD Wizard.

RADVIEW

- *Creating Load Templates with Goal-Oriented Test* (on page 135) teaches you to configure a goal-seeking performance test.
- *Configuring a Load Template Manually* (on page 165) provides instructions for configuring a test template manually.

After completing these chapters, you will have the basic skills necessary to create and run your own test.

Planning a Test Session

Planning your test is imperative for developing tests that accomplish your load testing objectives. Planning your test helps you:

- Plan the resources required for testing.
- Create templates that accurately emulate your user's typical working environment.
- Define test success criteria.

This chapter describes the stages involved in planning and executing a test session. It shows you where to find details in this guide about each concept so you can quickly find information of particular interest to your needs.

Load Session Workflow

A Load Session is performed in stages. This chapter explains the planning stage in detail. The rest of the stages are detailed in subsequent chapters.

- **Planning a Load Session**

Planning a Load Session involves considering:

- What you want to test – what application are you going to test?
- What functionality do you want to test – what actions will the users perform?
- How many Virtual Clients you want to simulate?
- How long your test will run?
- What are acceptable results? Acceptable results are defined by your test objective. For example, you can verify:
 - Acceptable user response times
 - Reliability by running stress tests
 - Performance degradation after updates
- What resources are required for performing the test?

- **Creating Visual Test Scripts**

Use WebLOAD Recorder to record the activity you want to test in a Web browser and save it as a script. If you plan to test users performing different activities, you need to record separate visual scripts for each activity. See the *WebLOAD Recorder User's Guide*.

- **Creating Load Templates**

Creating Load Templates involves defining the participants (Clients and Servers) in the Load Session. The definition includes selecting the scripts to run, the host computers and number of Virtual Clients participating in the Load Session, and scheduling the test. After you create a template it is saved in a *.tpl file. Use the commands in the **File** tab of the Console ribbon to create, open and save template files.

WebLOAD Console provides three ways for creating Load Templates:

- Using the WebLOAD Wizard to build a basic Load Template. You define the scripts to run, the Load Machines on which to generate the Virtual Clients and the test schedule. After creating a basic template with the wizard, you can change aspects of the configuration through the Console ribbon. For information on creating a template with the WebLOAD Wizard, see *Creating Load Templates with the WebLOAD Wizard* (on page 99).
- Using the Goal-Oriented Test Wizard to create goal-seeking performance tests. Rather than running multiple tests to measure the performance at different loads, you can run the Goal-Oriented Test to measure the performance goal that you desire. You define the scripts to run and the goals to achieve, and WebLOAD Console automatically schedules a test and increases the load until your goals are achieved. For information on creating a template with the Goal-Oriented Test Wizard, see *Creating Load Templates with Goal-Oriented Test* (on page 135).
- Manually using the Console ribbon. See *Configuring a Load Template Manually* (on page 165) for information on defining a template manually.

- **Running a Load Template**

After you create the Load Template, you run it. As your test runs, you receive various reports and messages from the various participating hosts. The activity in your test session is displayed on the Console screen. This screen is updated continuously in real time as you work. See *Running a Load Session* (on page 271).

- **Analyzing the Test Results**

WebLOAD Console displays the results of your Load Session in tabular format and in easy to read graphs.

Forming a Test Plan

Every test must begin with a test plan. When you plan a test session, the first step is to analyze your application to ensure that your test will accurately reflect the working environment of your users. You want to consider the goals of the test and your available resources. Consider the following:

1. What application are you testing?

When planning a test, the first thing you should consider is the target of your test. You must define what Web application you plan to test. Your test target is your SUT (System Under Test).

Before planning further, call up a browser and access the site to be tested, to make sure that the connection to the site on the Web is valid.

2. What criteria are you testing?

You must know what functionality of the Web application you want to test. Usually you want to test the amount of time required to perform a certain function on the website. You may want to test the time it takes to access a page, the time it takes to submit a form, or the number of clients that can simultaneously access your site.

Using the Goal-Oriented Test, you can test the number of clients that can perform a function, such as accessing a page or submitting a form, at the performance level that you have specified.

When defining the criteria for the test you should consider:

- How many users do you want to simulate? How many users are anticipated to connect to your application?
- How many different user activities do you want to simulate? If you plan to test users performing different activities, you need to create separate scripts for each activity.
- What type of users do you want to simulate? First time users respond differently to an application than returning users. Using the playback sleep time settings you can set WebLOAD Virtual Clients to run with the recorded sleep time delays, remove all delays to stress the application, or create random delays simulating different users.
- What type of response time is acceptable? Using the Goal-Oriented Test, you can create test sessions that increase the load and notify you at what load the response time exceeds acceptable levels.
- What is an acceptable failure rate? Should your test be stopped if a set number or percentage of errors is surpassed?

- What type of connections speeds do you want to simulate? Using the connection speed settings available with WebLOAD Console you can simulate users connecting through different connections.
3. What types of systems are testing these aspects?

Your test can include Load Machines and Probing Clients. A Load Machine bombards the SUT with a load of Virtual Clients. Load Machine testing gives you an indication of the way your program behaves under the stress of a heavy load.

Probing Client testing is usually performed simultaneously with Load Machine testing. Using a Probing Client, you can test the system performance with other functionality, while the system is undergoing stress testing. For example, you can measure the time required to submit a form, while the SUT is being bombarded by the Load Machine simulating multiple clients calling up a page.

WebLOAD Console generates exact values for Probing Client performance as opposed to averages for Load Machine performance.

Planning a Test Program

The test you devise must be coded in a script file. Scripts define the actions WebLOAD Virtual Clients perform. At this stage, you must plan the test program that you will develop. Your scripts should define the typical tasks you expect users to perform with your Web application.

For example, for a book store application you could create scripts that perform typical shopping activities such as searching for a book, adding books to the shopping cart and checking out. You should also consider your test objectives and define transactions to measure the response times for typical user activities such as searching for a book to ensure your Web application meets your requirements. You can also plan Synchronization Points to emulate peak server loads and ensure your requirements are met even under heavy load. Remember that if your plan indicates that your Load Machine and Probing Client perform different tasks, you need a script for each host.

WebLOAD scripts are created in WebLOAD Recorder. See the *WebLOAD Recorder User's Guide*, the *WebLOAD Scripting Guide*, and the *WebLOAD JavaScript Reference Guide* for further information on creating Scripts.

Creating the Test Scripts

The simplest way is to create a script is using WebLOAD Recorder that enables you to create a visual script by recording your activity in the browser.

For example, if you want to test the speed at which a certain Web application displays a page, and the speed required to submit a form, you can open WebLOAD Recorder and go to that application through a browser. With WebLOAD Recorder recording your activity, you can display the page desired and submit the form desired. WebLOAD Recorder saves your browser activity as a visual script. After recording the basic activity, you can add transaction statements to measure the time it takes to display the page.

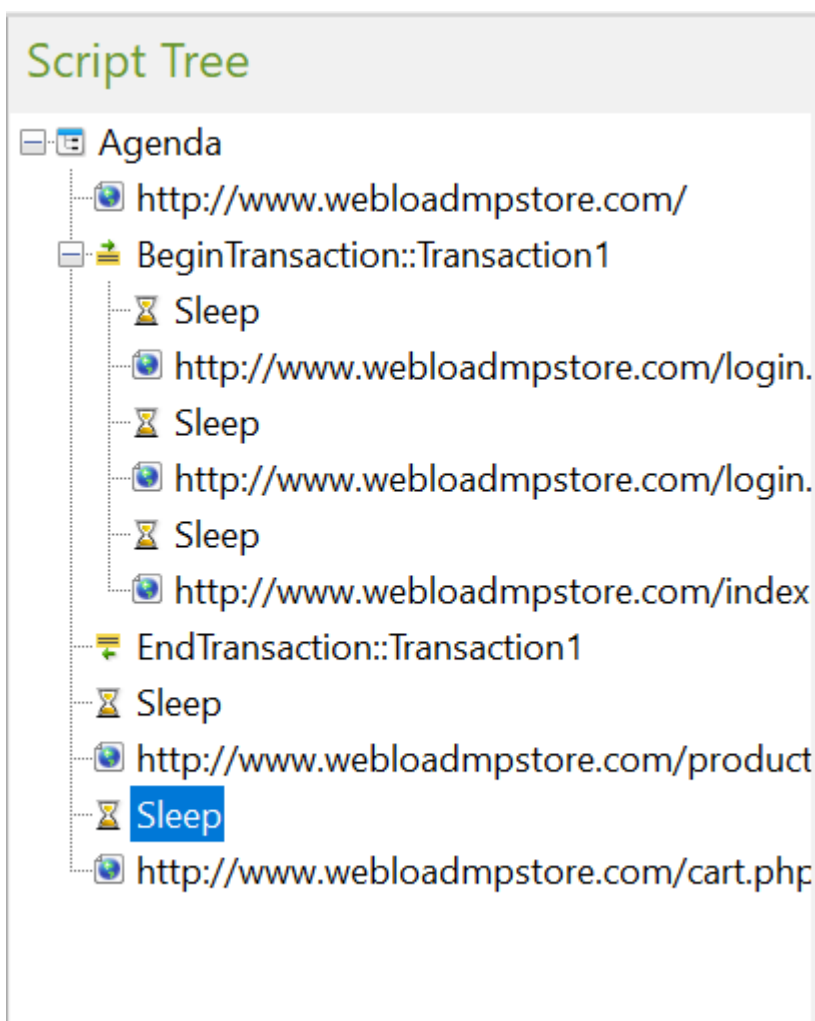


Figure 39: Visual Script

You can tailor the recorded script, by adding commands and timers available in WebLOAD Recorder. For more information on creating a script with WebLOAD Recorder, see the *WebLOAD Recorder User's Guide*.



Note: The script options you define in the Console for your current Session override the options defined in WebLOAD Recorder. For more information on script options, see *Setting Script Options* (on page 211).

Examining Your Test Resources

After you know what you are going to test, you must examine the resources that are available to you for running your test. This helps you allocate the resources for your test load. If the computers available to you cannot support the desired load, consider distributing the load over several computers.

Starting WebLOAD Console

When your test plan is complete and your scripts are ready, you can open WebLOAD Console and begin the test configuration process.

To open WebLOAD Console:

1. Verify that all the participating Host computers are properly connected.
2. Select **Start > All Programs > RadView > WebLOAD > WebLOAD Console**.

The Console calls up TestTalk automatically, on its system.

The Console opens with the Console Startup dialog box displaying a variety of shortcuts to the various WebLOAD Console components.

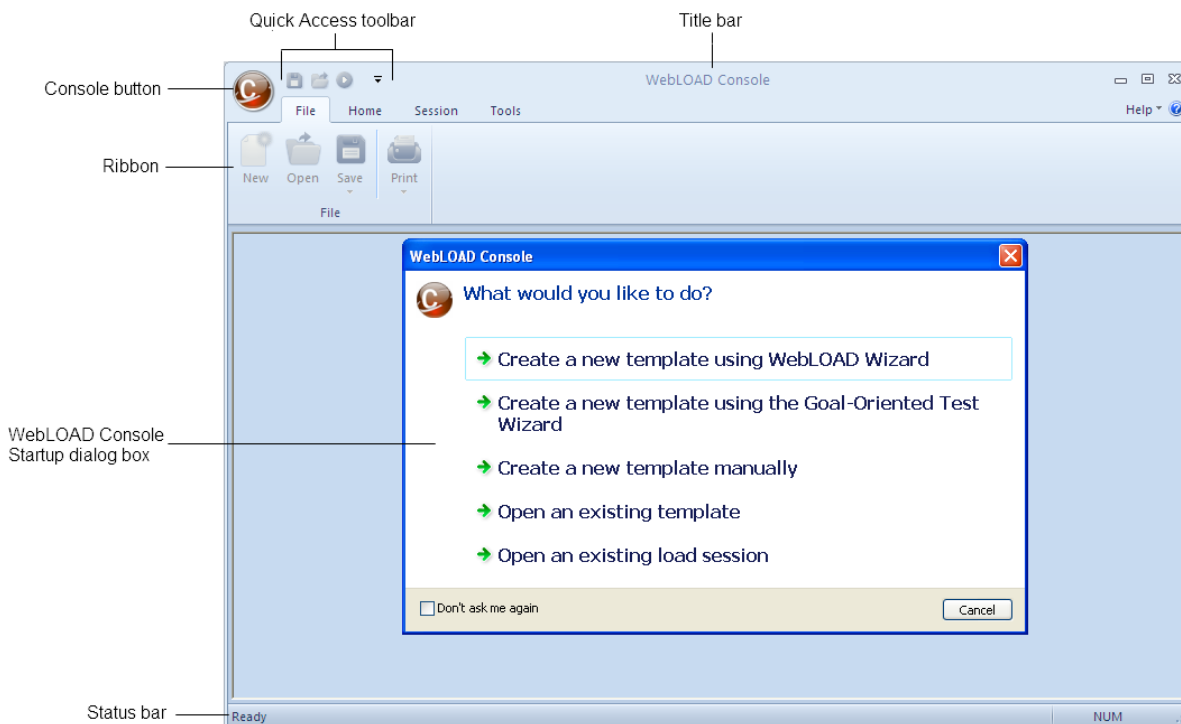


Figure 40: Console Startup Dialog Box


To create a new WebLOAD Console template, select one of the three methods available on the Startup dialog box.

- **Manual** – Create the Load Template by yourself, defining the scripts to run, the number of Virtual Clients to run, the Load Machines on which to generate the load and build a schedule for running the test.

- **WebLOAD Wizard** – The Wizard helps you through the process of configuring the Load Template.
- **Goal-Oriented Test wizard** – Define the scripts to run, the Load Machines on which to run the Virtual Clients, and the test goals to achieve, and WebLOAD Console automatically builds a schedule based on your goals.

The easiest way to configure a basic test is through the WebLOAD Wizard. Save your work in the Wizard as a Load Template. After you have closed the WebLOAD Wizard, you can use the Console ribbon to add functionality to your Load Template (*.tpl file).

Creating Load Templates with the WebLOAD Wizard (on page 99) discusses the WebLOAD Wizard. *Setting Global Options* (on page 177) provides instructions for using the Console features to add functionality to the Load Template created in the WebLOAD Wizard. *Creating Load Templates with Goal-Oriented Test* (on page 135) provides instructions for configuring a test session running the Goal-Oriented Test feature. *Connecting to WebRM* (on page 87) provides instructions for connecting the Console to WebRM using the WebRM Wizard. The Console can then request resources from WebRM so it can use the floating license, which is controlled by WebRM.

3. Click the TestTalk icon  on each host computer participating in the test. (TestTalk is automatically enabled only on the host running the Console.)

If you are using a Solaris or Linux system as a Load Machine, verify that the TestTalk software is running on the Solaris or Linux system. If you have closed TestTalk after the installation, run TestTalk again.

All of the systems participating in the test session, including the system running the Console, the Load Machines, and the Probing Clients must be running the network agent, TestTalk. This establishes communication between participating clients and servers. The SUT does not require TestTalk or any other WebLOAD-specific software.



Note: Do not close TestTalk in the middle of a test session. This can cause unexpected results. When shutting down the system, be sure to shut down the Console first. Then shut down TestTalk.

Connecting to WebRM

WebRM addresses corporate functional and performance testing efforts by managing and directing WebLOAD resources (Virtual Clients, Probing Clients and Consoles) for powerful and efficient sharing of testing resources and achieving load poling capability.

WebRM enables multiple users involved in various stages of application development and testing to share testing resources. By distributing WebLOAD testing resources each developer can run a test session to simulate, validate, and pinpoint where performance problems occur at any stage of the development life cycle, thus eliminating design flaws and ensuring product quality. Using WebRM, WebLOAD resources can be used optimally to fulfill an organizations goals and priorities.

WebRM can run on a separate machine from the other WebLOAD Consoles/Load Machines or can run on a machine that runs WebLOAD Consoles/Load Machines.

WebRM controls the following WebLOAD resources:

- The number of concurrent connected WebLOAD Consoles.
- The number of Virtual Clients controlled by all connected Consoles simultaneously.
- The number of Probing Clients controlled by all connected Consoles simultaneously.

For additional information about WebRM, see the *WebRM User's Guide*.

How Does WebRM Work

When you install WebRM, you apply your license file. The license file contains information regarding the total number of Virtual Clients, Probing Clients, and concurrent connected WebLOAD Consoles for which you are licensed.

When installing your WebLOAD Console, select the location where you installed your WebRM server. WebLOAD will communicate with your WebRM server in order to request and grant testing resources for you. If the WebRM server is not running or is not accessible by the WebLOAD Console, the Console will issue an error message and will not open.

When you open your WebLOAD Console you are prompted to request the number of Virtual Clients and Probing Clients you desire. These resources are granted to you as long as your WebLOAD Console is open, allowing you to complete your test session utilizing the allocated number of Virtual and Probing Clients.

WebRM is very straightforward and easy to use. Much of your interaction with WebRM is behind the scenes. Once you have selected the resources you wish to use, you can configure WebRM to automatically grant you these same resources each time you open your WebLOAD Console.

Requesting Resources from WebRM

Following the configuration of WebLOAD Console to work with WebRM, configure the connection settings by browsing to the machine where the WebRM server resides. (This is described in *WebRM Installation* in the *WebRM User's Guide*.) When opening your WebLOAD Console, the connection to WebRM occurs automatically. If the Console fails to connect to the WebRM server, the Console will not open.

To request resources from WebRM:

1. Open your Console by selecting **Start >Programs > RadView > WebLOAD > WebLOAD Console**.

The Request Resources dialog appears:

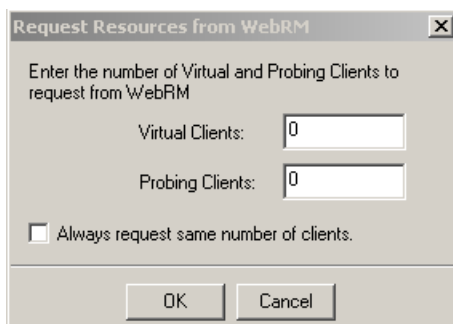


Figure 41: Request Resources Dialog Box

2. Enter the number of Virtual Clients and Probing Clients you will need for your test. If you request more than the number of resources allowed for your license, or more than are currently available, you will be prompted with a dialog stating that the resources you requested are unavailable. This dialog will also list what resources are currently available so that you can resubmit your request.
3. If you know that you will require the same number of resources each time you connect to WebRM, check **Always request same number of clients**.
4. Click OK.

Updating the Number of Requested Resources

You can easily change the number of resources granted to you.

To change the number of resources granted to you:

1. In the WebLOAD Console, select **Update Floating License Resources** in the **Home** tab of the ribbon.
The Request Resources dialog appears, displaying the resources currently granted to you.
2. Edit the Virtual Client and Probing Client settings to reflect the new total number of Virtual and Probing clients you need.

Managing Load Templates and Load Sessions

Load Templates define the events that occur in the test including:

- The scripts to run
- The hosts participating in a test session
- The test schedule
- The number of Virtual Clients to run

Load Templates save you the time of reconfiguring the test when you are repeating tests. Load Sessions include the test definitions in the Load Template along with test results.

Managing Load Templates

Load Templates define the events that occur in the test. You create test definitions manually or using one of the WebLOAD Wizards (WebLOAD or Goal-Oriented Test) available with WebLOAD. After creating a test definition, WebLOAD Console prompts you to save the test configuration as a Load Template. Load Templates contain complete test definitions including:

- Scripts
- Information on participating hosts
- Host schedules
- Virtual Clients to run
- Run time configuration options
- Report configurations

Load Templates include all test information except for the test results. Load Template files are saved as *.tpl files. Load Templates can be edited using the Console ribbon and button. You can then save the Load Template for future test iterations. For example, if you build a test in the WebLOAD Wizard you can call it up in the Console and edit the Load Template manually, changing perhaps the hosts, the number of Virtual Clients and even the script. Use the commands in the **File** tab of the ribbon to create, open and save.

Creating a New Load Template

When you create a new Load Template, you define the events to occur in the test. A blank Session Tree displays to enable you to begin configuring your test.

To create a new Load Template:

1. Click **New** from the **File** tab of the ribbon.

A blank template appears.

For information on creating templates with the WebLOAD Wizard, see *Creating Load Templates with the WebLOAD Wizard* (on page 99). For information on creating templates with the Goal-Oriented Test Wizard, see *Creating Load Templates with Goal-Oriented Test* (on page 135).

Saving a Load Template

To save a Load Template:

1. Click **Save** in the **File** tab of the ribbon and select **Save** or **Save As**.
2. In the **File Name** box, type a name for the Load Template.
3. Click **Save**.

The file is saved with the extension *.tpl to the specified location and the Save dialog box closes.

Opening a Saved Load Template

To open a saved Load Template:

1. Click **Open** in the **File** tab of the ribbon and browse your directory structure for the location of the saved Load Template (*.tpl) file,

-Or-

Select the **Open an Existing Template** option from the WebLOAD Console dialog box at system start-up and browse your directory structure for the location of the saved Load Template (*.tpl) file.

2. Double-click the template you want to open.

The file opens and the test configuration displays in the Session Tree.

Saving Additional Information with the Load Template

The Additional Information dialog box provides details about the Load Template that help identify it. For example:

- Descriptive title
- Author name
- Subject of the test
- System under test
- Other important information about the Load Template

Use the Additional Information dialog box to display information about the Load Template.

Setting Additional Information Properties

You can set additional information properties for the active Load Template.

To set additional information properties:

1. Select **Additional information** from the Console System button.

The Template Additional Information dialog box opens.

Figure 42: Template Additional Information Dialog Box

2. Complete the fields to save additional information, useful for later reference, with the Load Template.

The following fields are available:

Table 25: Load Template Additional Information Fields

Field	Description
Title	Provides a space for you to type a title for this Load Template. The title can be different then the template file name.
Subject	Provides a space for you to type a description of the subject of the Load Template. Use this property to group similar Load Templates together.
Created by	Provides a space for you to type the name of the person who authored this Load Template.
Test description	Provides a space for you to type a description of the test objectives and what the Load Template is designed to test.
Version and build of the System Under Test	Provides a space for you to type the name, version, and build number of the system under test.

Field	Description
Template comments	Provides a space for you to type any comments regarding the Load Template.
Custom	Provides a space for you to type any comments you want saved with this Load Template.

3. Click **OK**.

Creating a Mix

A Mix is a set of scripts, each performing different activity, to simulate groups of users performing different activities on the SUT at the same time. You can create a new Mix to use with your Load Template.

To create a new Mix:

- Click **Mix Creation** in the **Home** tab of the ribbon.

The Mix Creation dialog box appears.

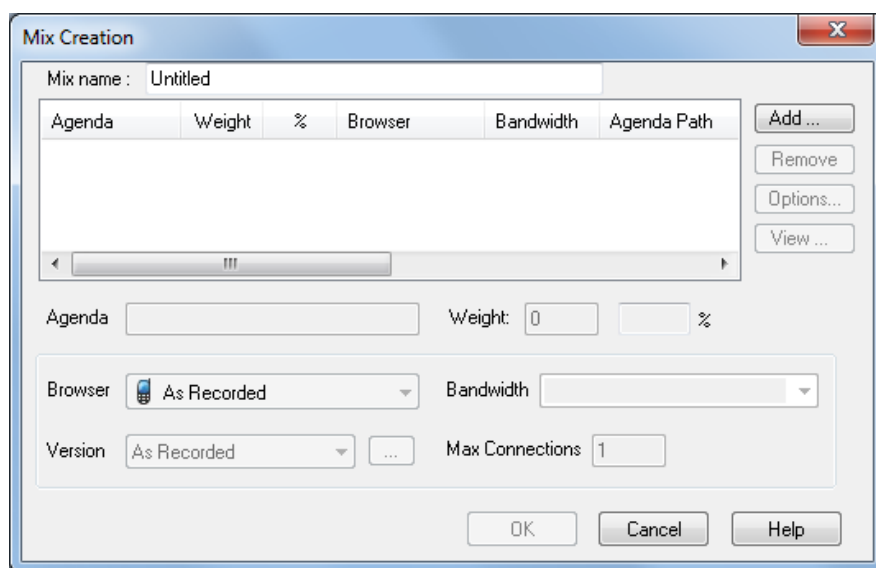


Figure 43: Mix Creation Dialog Box

Continue to one of the following sections for further instructions on creating a Mix. To create a mix with the WebLOAD Wizard, see *Creating a Mix* (on page 105). To create a mix with the Goal-Oriented Test Wizard, see *Running a Mix of Scripts* (on page 141).

Managing Load Sessions

A Load Session consists of the Load Template and the data accumulated during the test run. Load Sessions cannot be edited.

A saved Load Session includes:

- The test definition including configuration information on scripts, participating hosts, and their schedules – as illustrated in the Session Tree.
- The report configuration.
- All test session results.

Load Sessions are saved in `*.lS` (Load Session) files. Load Session results are saved in a `*.dat` (Data) file. Saved Load Sessions are useful for viewing results of previously run test sessions.

Saving a Load Session

To save a Load Session:

1. Click **Save** in the **File** tab of the ribbon.
2. Browse to the location to save the file and in the File Name dialog box, type a name for the Load Session.
3. Click **Save**.

The file is saved with the extension `*.lS`.

Saving Additional Information for the Load Session

The Additional Information dialog box provides details about the Load Session that help identify it. For example:

- Descriptive title
- Author name
- Subject of the test
- System under test
- Other important information about the Load Session

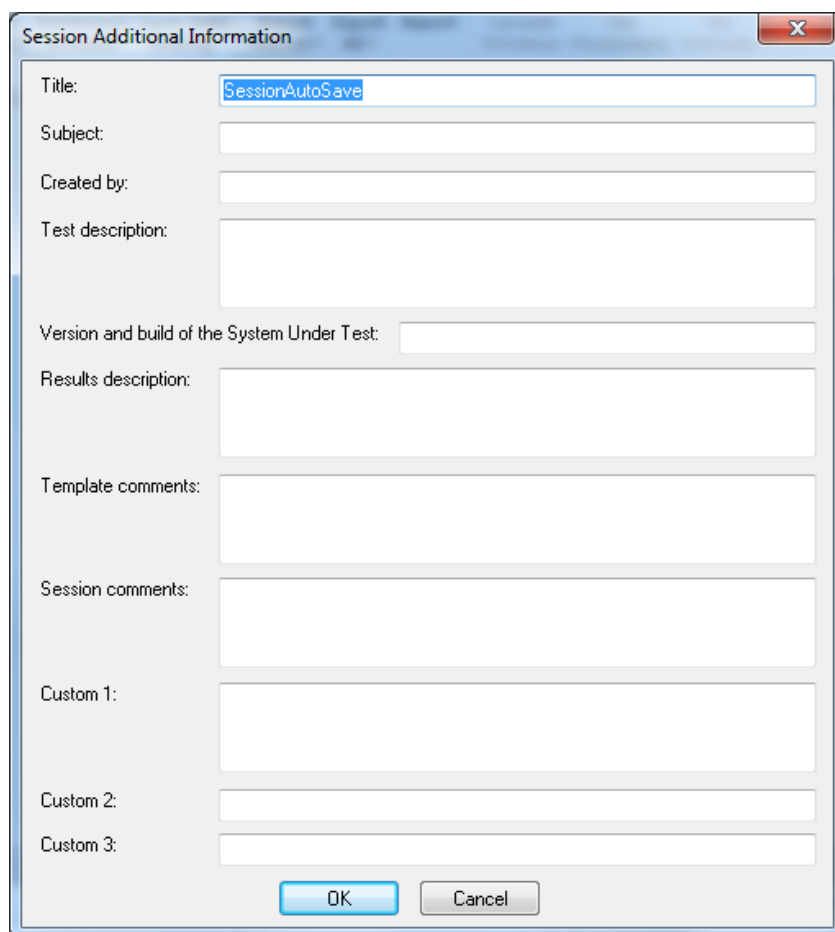
Use the Additional Information dialog box to display information about the Load Session.

Setting Additional Information Properties

You can set additional information properties for the active Load Session.

To set additional information properties:

1. Select **Additional information** from the Console System button.
The Session Additional Information dialog box opens.



The image shows a dialog box titled "Session Additional Information". It contains the following fields and controls:

- Title:
- Subject:
- Created by:
- Test description:
- Version and build of the System Under Test:
- Results description:
- Template comments:
- Session comments:
- Custom 1:
- Custom 2:
- Custom 3:
- Buttons: OK, Cancel

Figure 44: Session Additional Information Dialog Box

2. Complete the fields to save additional information, useful for later reference, with the Load Session. Some of the fields, such as title, subject, and system under test, may already be completed if the information was saved with the Load Template.

The following fields are available:

Table 26: Load Session Additional Information Fields

Field	Description
Title	Provides a space for you to type a title for this Load Session. The title can be different from the template file name.
Subject	Provides a space for you to type a description of the subject of the Load Session. Use this property to group similar Load Sessions together.
Created by	Provides a space for you to type the name of the person who authored this Load Session.
Test description	Provides a space for you to type a description of the test objectives and what the Load Session is designed to test.
version and build of the System Under Test	Provides a space for you to type the name, version and build number of the system under test.
Results description	Provides a space for you to type a description of the test results.
Template comments	Provides a space for you to type any comments regarding the Load Session.
Session comments	Provides a space for you to type any comments regarding the Load Session.
Custom	Provides a space for you to type any comments you want saved with this Load Session.

3. Click **OK**.

Opening Saved Load Sessions Files

To open a saved Load Session file:

1. Click **Open** in the **File** tab of the ribbon and browse your directory structure for the location of the saved (*.ls file),

-Or-

Select the **Open an Existing Load Session** option from the WebLOAD Console dialog box at system start-up and browse your directory structure for the location of the saved (*.ls file).

2. Double-click the session you want to open.

The file is opened, the test configuration is displayed in the Session Tree and the WebLOAD Default Report is displayed in the Results window.

Creating Load Templates with the WebLOAD Wizard

The easiest way to configure a Load Template is using the WebLOAD Wizard.

Using the WebLOAD Wizard to Create Load Templates

Using the wizard to create a Load Template, you define the following:

- Script(s) to run
- Load Machines on which the load is generated
- Number of Virtual Clients to run
- Load schedule

You can also specify script options including the type of browser to emulate connection speed, and playback sleep time options.

The WebLOAD Wizard walks you through the configuration process step-by-step. Each screen of the WebLOAD Wizard contains text explaining the configuration process.

After creating a Load Template with the WebLOAD Wizard, you can add functionality, not available in the wizard, through the Console ribbon. The Console allows you access to any part of the configuration process at any time. For details about the Console ribbon, see *The Console Ribbon Options* (on page 70).



Note: You cannot use the WebLOAD Wizard if you have not yet recorded (or otherwise created) any scripts. If you do not have any scripts to work with, you must exit the wizard and create a script using one of the authoring tools, such as WebLOAD Recorder.

The WebLOAD Wizard Workflow

The following diagram illustrates the WebLOAD Wizard workflow:

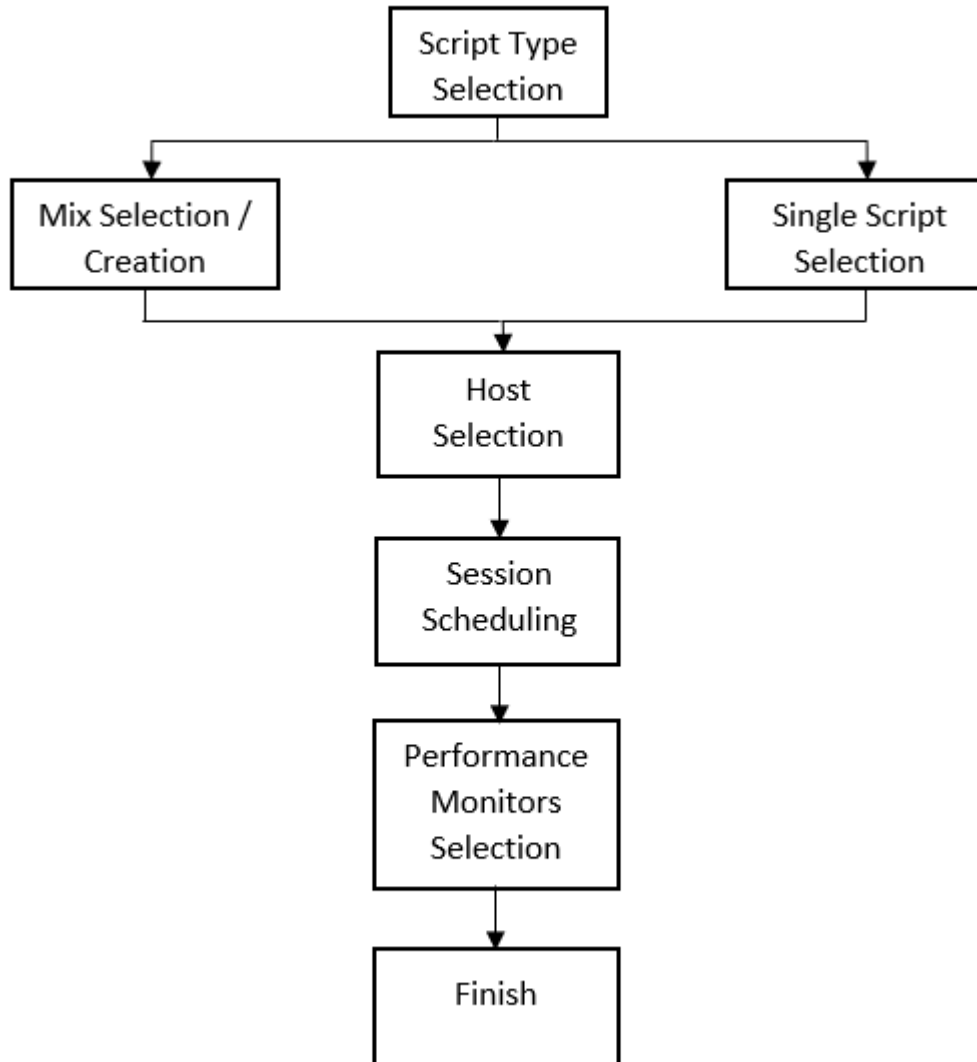


Figure 45: WebLOAD Wizard Workflow Diagram

Opening the WebLOAD Wizard

To open the WebLOAD Wizard:

1. Select **WebLOAD Wizard** from the WebLOAD Startup dialog box,
-Or-

Click **WebLOAD Wizard** in the **Home** tab of the ribbon.

The WebLOAD Wizard Welcome dialog box opens.

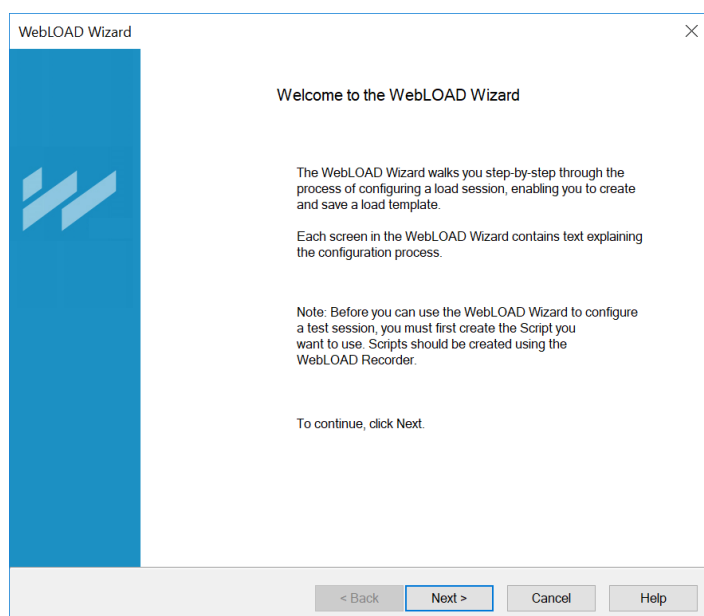


Figure 46: WebLOAD Wizard Welcome Dialog Box

2. Click **Next**.

The WebLOAD Wizard progresses to the script/Mix Type dialog box.

Selecting a script or Mix

The next step in a WebLOAD session is to select the type of script you want to run during your test session. A script is the test script used to test your application.

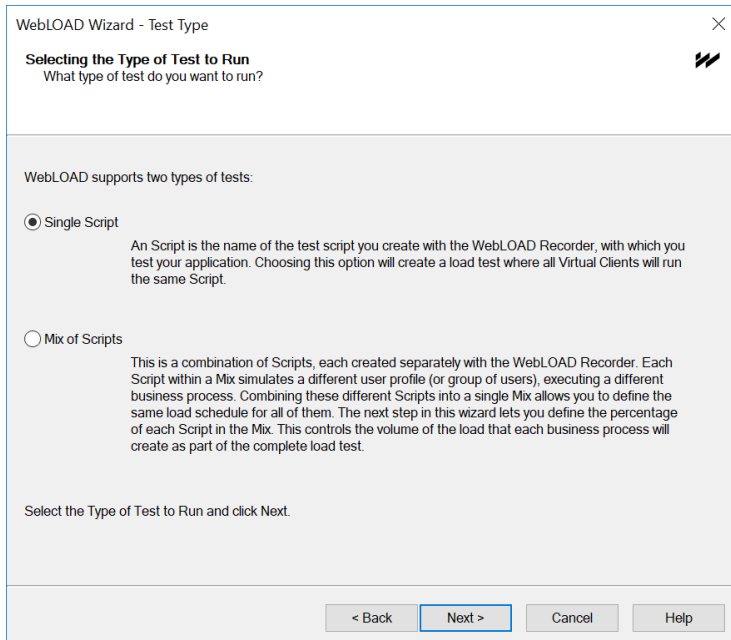


Figure 47: Test Type Selection Dialog Box

WebLOAD Console supports two types of scripts:

- **Single scripts** – lone test-scripts.
- **Mix of scripts** – a combination of scripts. Use a Mix to simulate different groups of users performing different activities on the SUT at the same time.

To run a single script:

1. Select **Single Script**.
2. Click **Next**.

WebLOAD Console progresses to the Script Selection dialog box. See *Selecting a* (on page 103).

To run a Mix of scripts:

1. Select **Mix**.
2. Click **Next**.

WebLOAD Console progresses to the Mix Selection dialog box. See *Selecting a Mix* (on page 104).

Selecting a Script

The WebLOAD Wizard enables you to create a test using previously recorded scripts. If you do not have any scripts to work with, you must exit the wizard, and create a script using one of the authoring tools, such as WebLOAD Recorder.

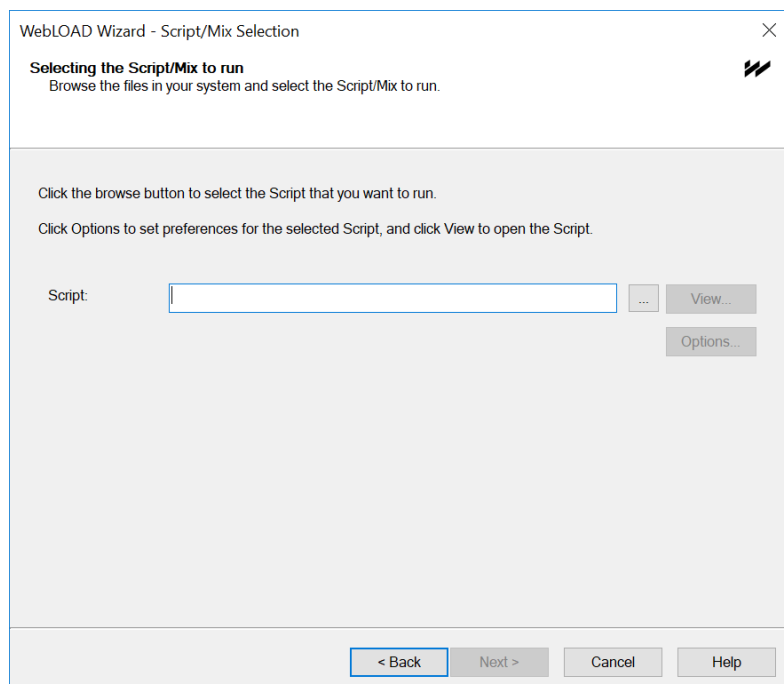



Figure 48: Script Selection Dialog Box

To select a script:

1. Click the  button to select the source of the script to be used for the test.
2. To optionally open and view or edit the selected Script, click **View**.
WebLOAD Recorder opens, displaying the selected script.
3. To optionally configure runtime options exclusive to this script, click **Options**.
The Script Options dialog box opens enabling you to define the runtime options for the selected script. For more information on script options, see *Setting Script Options* (on page 211).
4. Click **Next**.
The WebLOAD Wizard progresses to the Host Selection dialog box. See *Selecting Host Computers* (on page 107).

Selecting a Mix

The WebLOAD Wizard enables you to run your test using a saved Mix or to create a new Mix of scripts through the wizard. If you do not have any scripts to work with, to create a Mix, you must exit the wizard, and create a script using one of the authoring tools, such as WebLOAD Recorder.

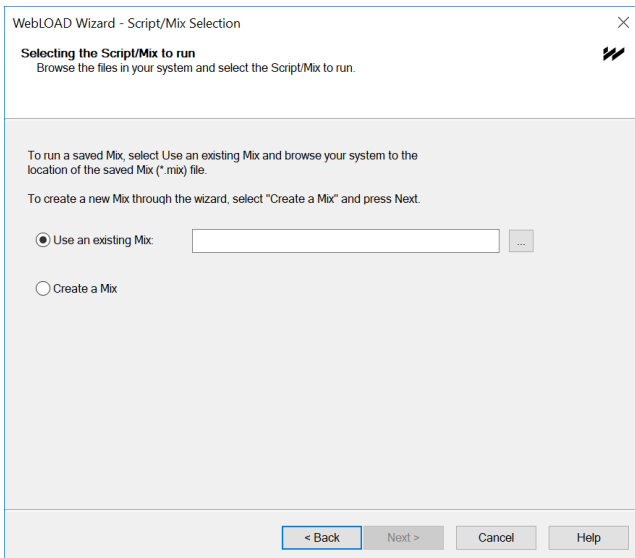



Figure 49: Mix Selection Dialog Box

To select a Mix:

1. Select the source of the Mix to be used for the test on the Mix Selection screen:
 - Select **Use an Existing Mix** and click the  button to select an existing Mix (*.mix) from your system.
 - Select **Create a Mix** to build a new Mix.
2. Click **Next**.

WebLOAD Console progresses to the Mix Creation dialog box. See *Creating a Mix* (on page 105).

Creating a Mix

A Mix is a set of scripts, each performing different activity, to simulate groups of users performing different activities on the SUT at the same time. If you selected **Use an Existing Mix** on the script / Mix Selection screen, the selected Mix appears on the screen, enabling you to modify the Mix. You can also create a new Mix.

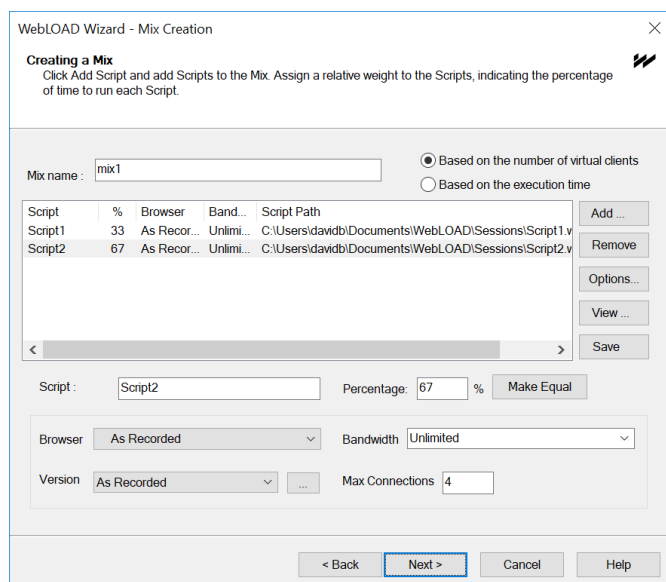


Figure 50: Mix Creation Dialog Box

To create a new Mix:

1. Enter a descriptive name in the **Mix name** field.
2. Specify whether the relative weights of the script refer to execution time or to number of virtual clients:
 - **Based on the number of virtual clients** – The mix interprets the relative weights in terms of number of virtual users. For example, if the weight of Script A is 25 and of Script B is 75, the mix execution runs script A with 25% of the virtual users and script B with 75% of the users.
 - **Based on the execution time** – The mix interprets the relative weights in terms of time. For example, if the weight of Script A is 25 and of Script B is 75, the mix execution run script A for 25% of the time and script B for 75% of the time.
3. Click **Add**.
The Open dialog box appears, enabling you to select the Script to add to the Mix.
4. Select a script from your file system and click **Open**.
The script is added to the Mix.
5. Optionally, select the script and set any of the following parameters:

- a. **Script** – Specify a name for the script and its particular settings.
Keep in mind that in a mix, in addition to running different scripts, you can also run the same script under different settings. For example, a mix can contain two items: Script A running on Chrome, and the same Script A running on Mozilla Firefox. Give a different descriptive name to each of the two mix items.
 - b. **Weight** – Specify the relative weight of this script, as a positive whole number. The application will run each script based on its weight in relation to the total of weights. When the total of weights is 100, the weight is equivalent to a percentage; however, this is not required. For example, if you specified a weight of 1 for Script A and a weight of 3 for Script B, then the system assigns script A a relative weight of 1/4, and to Script B the relative weight of 3/4. This is equivalent to a weight of 25 for Script A and 75 for Script B; in both cases, Script A will run 25% of the rounds, and Script B will run 75% of the rounds.
 - c. **Browser and Version** – Select the browser type and browser version. Alternatively, click the Change button to edit the user agent definition. See *Adding a Browser Version Definitions* (on page 223).
 - d. **Bandwidth** – Specify a bandwidth. You can do this in one of two ways:
 - Select a bandwidth from the drop-down list.
 - Enter a number, to specify that number of bits per second.
 - e. **Max Connections** – Specify the number of max connections for the load generator.
6. Optionally, click **Options** to set any of the settings available from the Script Options dialog box (such as the browser to emulate, connection settings and playback sleep options). This will change the settings of the currently selected item only.
- The Script Options dialog box opens. For information on setting the Script Options, see *Setting Script Options* (on page 211).
7. Repeat steps 2 through 5 to add additional scripts to the Mix.
 8. Click **Save**.
The Mix configuration is saved with the extension *.mix.
 9. Click **Next**.
The WebLOAD Wizard progresses to the Host Selection dialog box. See *Selecting Host Computers* (on page 107).

Deleting a script from a Mix

To delete a script from a Mix:

1. Select the Script you want to delete from the Mix.

2. Click **Remove**.

Viewing a script from a Mix

To view a script from a Mix in the WebLOAD Recorder:

1. Select the script you want to view.
2. Click **View**.

The WebLOAD Recorder is launched, and displays the selected script.

Selecting Host Computers

On the Host Selection dialog box you define the host computers participating in the test session. Host computers can be dedicated, local computers, or they can be computers which are rented on a temporary basis from the Amazon™ Elastic Compute Cloud™ (EC2). Host computers are assigned to be either Load Machines or Probing Client Machines.

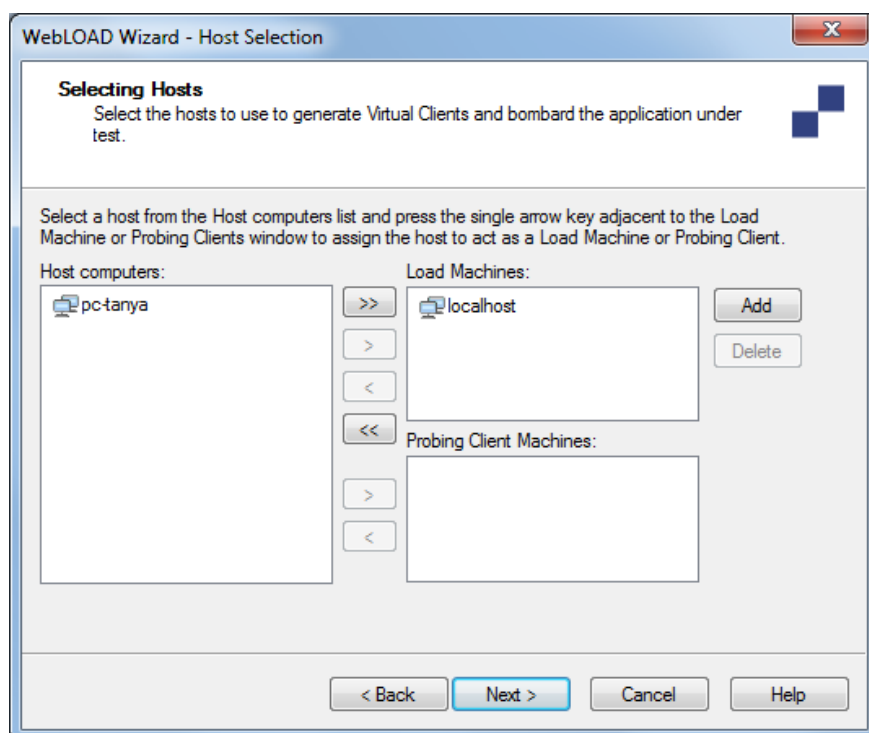


Figure 51: Host Selection Dialog Box

Load Machines

By selecting a group of Load Machines, you define the pool of computers WebLOAD Console uses to generate the Virtual Clients. The number of Virtual Clients that can be generated on each host is dependent on the power of each host machine.

Probing Client Machines

Probing Clients are single instances of Virtual Clients running WebLOAD Console while the Load Machines bombard your Web application. WebLOAD Console reports exact measurements for Probing Clients.

This dialog box has three windows. The large window on the left, entitled Host Computers, displays all the host computers defined for testing. The two windows to the right contain the systems defined as *Load Machines* and *Probing Client Machines*. Use the arrow buttons between the windows to change the roles of the host computers.

Your local system is automatically defined as the default Load Machine.



Notes:

At least one system must be configured as a Load Machine to continue working with the wizard.

After configuring the hosts participating in the test session, the WebLOAD Wizard progresses to the Schedule dialog box. See *Scheduling the Load Session* (on page 114).

If you are running a script with Perfecto Mobile script, define a single Load Machine or a single Probing Client in order to simulate a single user.

Setting Up Cloud Computers

WebLOAD supports using Amazon EC2 Cloud servers as an alternative to acquiring and maintaining physical load-generators.

The workflow for setting up EC2 computers for WebLOAD is as follows:

1. Creating an Amazon EC2 Account. For instructions, refer to <http://aws.amazon.com/ec2/>.
2. *Creating WebLOAD Cloud Accounts.*

Creating WebLOAD Cloud Accounts

After creating an Amazon cloud account, create a WebLOAD cloud account in which you specify your Amazon security credentials, the specific Amazon region where the machines should be located, and the WebLOAD image to install on the Amazon machines.

Note that you can use the same Amazon cloud account to create multiple WebLOAD cloud accounts. For example, to simulate a scenario where some users are located in US East and some in EU West, create two WebLOAD cloud accounts. Enter the same Amazon cloud account credentials in both, but specify a different region for each.

To create WebLOAD cloud accounts in WebLOAD Console:

1. Click **Cloud Options** in the **Tools** tab of the ribbon. The Cloud Accounts dialog box appears.

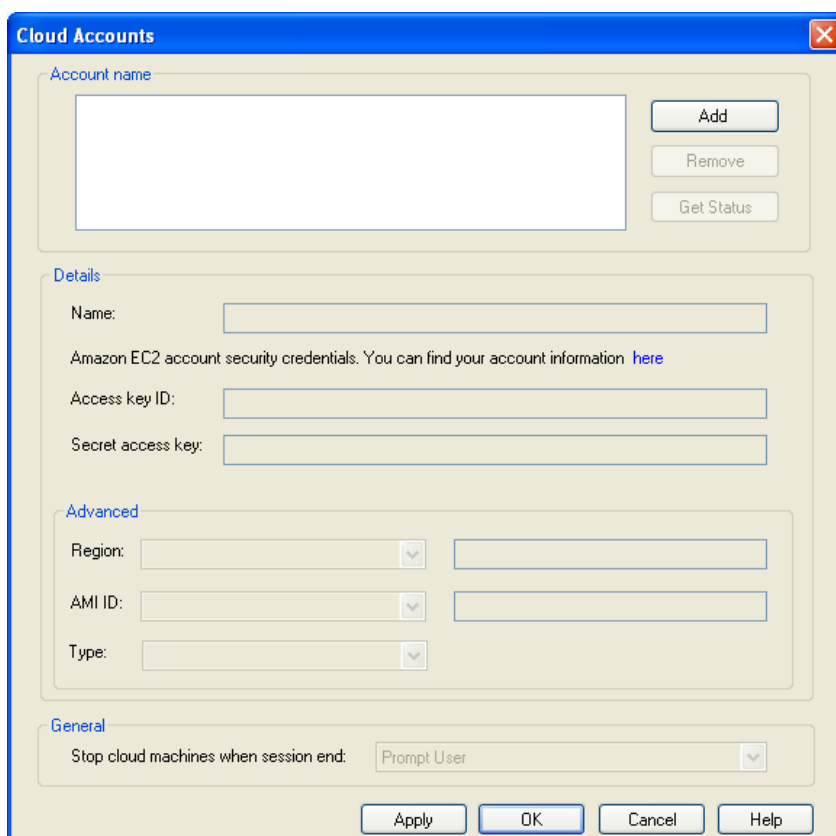


Figure 52: Cloud Accounts Dialog Box

2. Click **Add** to define a WebLOAD cloud account.
3. Enter a name describing the account in the **Name** field. For example, Cloud US.
4. Enter the Amazon cloud security credentials in the **Access Key ID** and **Secret Access Key** fields. To determine the security credentials, perform the following:

- a. Log into your EC2 Amazon account and click **Your Account > Security Credentials**. A dialog box appears displaying your Access Key ID. Your Secret Access Key is hidden by default.
- b. Click **Show** to display the Secret Access Key.

Alternatively, in the Cloud Accounts window, click **here** to open the Amazon AWS Accounts screen. In this screen you can create an Amazon cloud account, or enter your Amazon credentials if you had already created an Amazon cloud account.

5. In the **Region** drop-down box, you can select a specific Amazon data-center for use, by region. In the adjacent field you can optionally enter a specific URL for the cloud end-point. Any end-point which supports the EC2 API is a valid option for this field.
6. In the **AMI ID** drop-down box, select a specific Amazon Machine Image for use. If you have unique requirements, you can select **Other** and contact RadView support for advice.
7. In the **Type** drop-down box, specify the type of Amazon machine. We recommend the C1 High-CPU Medium (this is the default option).
8. Optionally, click **Get Status** to verify the details of the cloud you created.
9. For every additional WebLOAD cloud account you wish to create, repeat steps 2-8.
10. In the **General** section, specify the global policy for shutting-down Cloud Machines. The policy applies to all the WebLOAD cloud accounts you create.

In the **Stop cloud machines when session end** drop-down box, define whether the Cloud Machines being used during a session are to be automatically shut down upon stopping the session (thereby stopping the timer for payment). The possible options are:

- **Prompt User**
- **Stop Always**
- **Don't Stop**

Keep in mind the following:

- It takes Amazon approximately 15 minutes to prepare the Cloud Machines. If you select **Stop Always**, this process will need to be repeated after every session stop, even if you want to run multiple sessions or if you stopped a session prematurely with the intention of restarting the session immediately.
 - If you select **Don't Stop**, the Cloud Machines will not be automatically stopped when you end a session. This means that you will continue to be charged for those machines even after a session ends successfully. Click **Terminate Cloud Machines** in the **Tools** tab of the ribbon to manually shut down the Cloud Machines and stop the payment timer.
11. Click **OK**. The credentials are saved and the use of Cloud Machines as host computers is now supported.

Adding Host Computers

To add a host computer:

1. In the Host Selection dialog box (Figure 51), click **Add**.

The Add Host Computer dialog box appears.

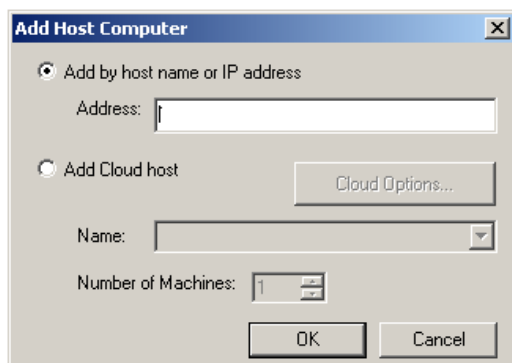


Figure 53: Add Host Computer Dialog Box

2. To add a local host computer, select **Add by host name or IP address** and enter the host computer name or IP address in the **Address** field.
3. To add cloud host machines:

- a. Select **Add Cloud host**.

If you have not yet created a WebLOAD cloud account, a dialog box appears. Click **OK** and the Cloud Accounts dialog box appears (Figure 52). Create one or more WebLOAD cloud accounts as described in *Creating WebLOAD Cloud Accounts* (on page 109).

- b. Select a WebLOAD cloud account from the **Name** drop-down box.
- c. In the **Number of Machines** field, enter the number of machines you wish to add from the selected WebLOAD cloud account.
- d. Repeat steps b-c to specify machines from another WebLOAD cloud account.


Optionally, click **Cloud Options** to add or modify WebLOAD cloud accounts. The Cloud Accounts dialog box appears (Figure 52). For information about the available actions, refer to the explanations following Figure 52.

4. Click **OK**.
 - If you added a local host machine, the machine is added to the Load Machines list in the Host Selection dialog box (Figure 51), and is listed by the host name or IP address you entered.
 - If you added cloud host machines, a list of machines is added to the Load Machines list in the Host Selection dialog box with the names “<WebLOAD Cloud Account Name><X>” where X is a number that is incremented for each added cloud host machine.

For defining host computers as Load Machines and Probing Clients, see *Designating a Host Computer as a Load Machine* (on page 112), and *Designating a Host Computer as a Probing Client Machine* (on page 112).

Designating a Host Computer as a Load Machine

To designate a host computer as a Load Machine:

1. Select the host from the Host Computers window and click the single arrow key  adjacent to the Load Machine window.

-Or-

Double-click the host.


The selected host moves to the Load Machines window.

2. To designate additional Load Machines repeat steps 1 and 2.

After assigning all Load Machines and Probing Client Machines, click **Next**. The wizard progresses to the Schedule dialog box. See *Scheduling the Load Session* (on page 114).

Designating all Host Computers as Load Machines

To designate all host computers as Load Machines:

- Click the double arrow key  adjacent to the Load Machines window.
All of the host computers move to the Load Machines window.


Designating a Host Computer as a Probing Client Machine

Probing Clients act as single Virtual Clients to measure the performance of targeted activities and provide individual performance statistics of the SUT.



Note: Probing Client Machines cannot run Mix scripts.


To designate a host computer as a Probing Client Machine:

1. Select the host from the Host Computers window.
2. Click the single arrow key  adjacent to the Probing Client Machines window.
The selected host moves to the Probing Client Machine window.
3. After assigning all Load Machines and Probing Client Machines, click **Next**.

The wizard progresses to the Schedule screen. See *Scheduling the Load Session* (on page 114).

Replacing a Load Machine or Probing Client Machine


To replace a Load Machine or Probing Client Machine:

1. From the Host Selection dialog box, select the host computer in the Load Machines or Probing Client Machine window and click the left arrow button .

-Or-

In the Load Machines window, double-click the host computer.

The host computer is moved from the Load Machines or Probing Client Machine window to the Host Computers window.

2. From the Host Computers window, select a host computer.
3. Click the right arrow button .

The host computer is moved from the Host Computers window to the Load Machines or Probing Client Machine window.

Deleting a Host Computer

To delete a host computer:

1. Select the host name from the Host Computers list.
2. Click **Delete**.

The host is deleted from the Host Computers list.

Scheduling the Load Session

When you schedule a test session, you define the load to be generated throughout the Load Session.

Use the Schedule dialog box to plan your test session.

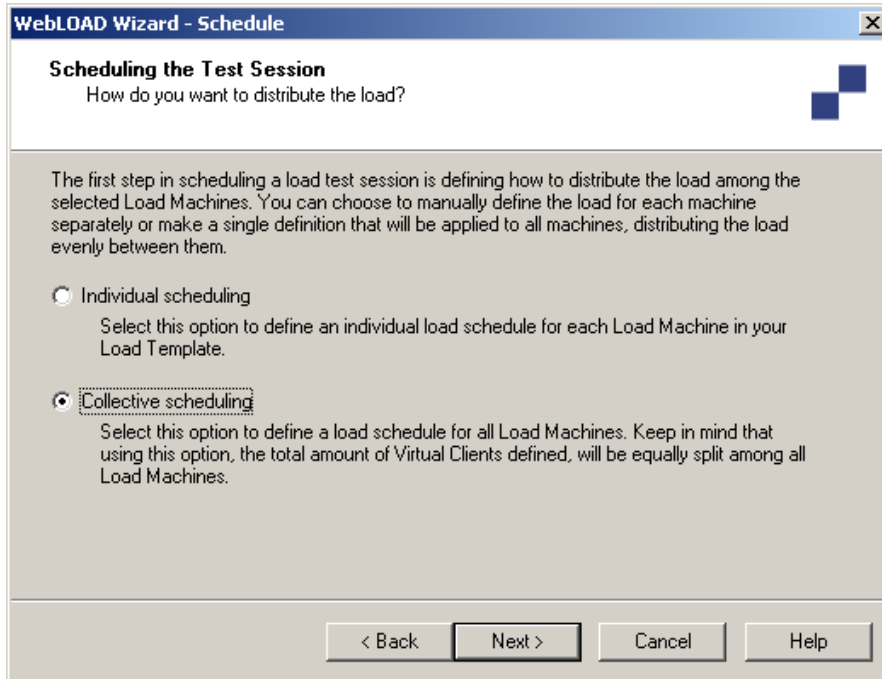


Figure 54: Schedule Dialog Box

To schedule the Load Session:

1. Define how you want to distribute the load between the Load Machines running the script.
 - Select **Individual scheduling** to instruct WebLOAD Console how to distribute the load over the Load Machines throughout the test.
 - Select **Collective scheduling** to distribute the load evenly between the Load Machines.
2. Click **Next**.
 - If you selected **Collective scheduling**, the WebLOAD Wizard progresses to the Collective Scheduling screen. See *Collectively Distributing the Load* (on page 115).
 - If you selected **Individual scheduling**, the Wizard progresses to the Individual Scheduling screen. See *Individually Scheduling the Load* (on page 118).

Collectively Distributing the Load

WebLOAD simplifies the process of scheduling the test session by enabling you to automatically distribute the load between the Load Machines, saving you the time of creating a load schedule for each Load Machine individually.

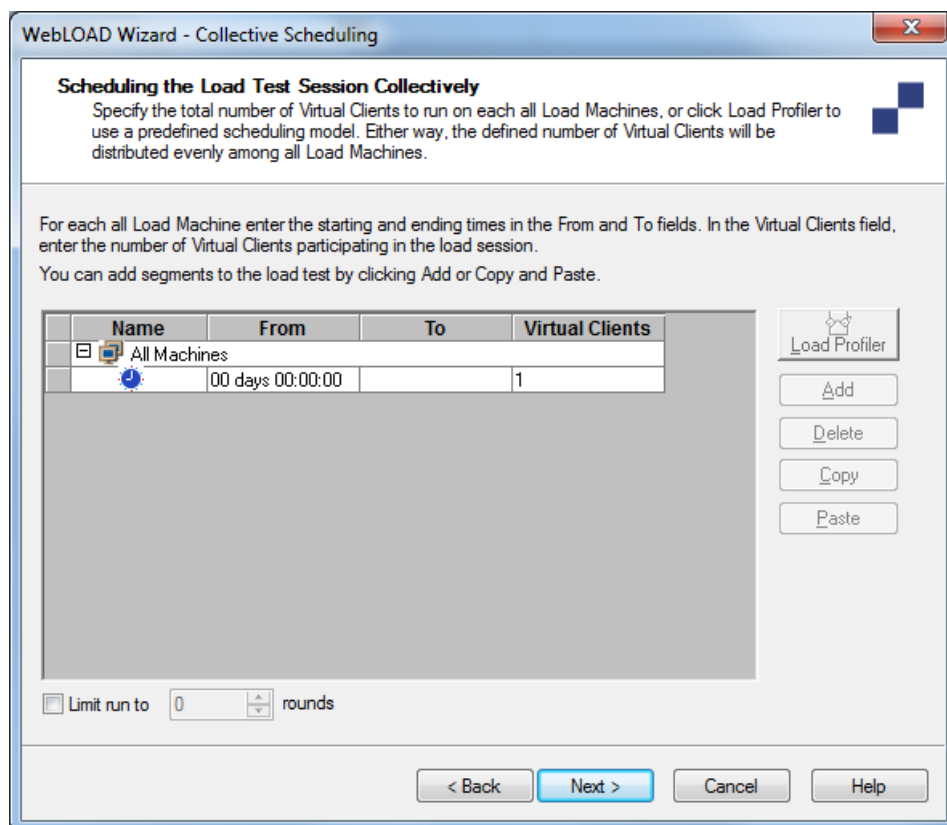


Figure 55: Collective Scheduling Dialog Box

You can create a load schedule by any combination of the following methods:

- Manually defining time frames and the load to generate over each frame
- Manually limiting the run by number of rounds
- Using the Load Profiler

Creating a Collective Load Schedule

To create a collective load schedule manually:

1. Enter the number of days and the starting time to generate the load in the From field.

The starting time is defined for each particular script relative to the beginning of the test session.



Note: Days are numbered 00 through to 99. Following them are hours:minutes:seconds. Seconds are numbered 1 through 59. To enter one minute, use the 1:00 syntax not 00:60.

2. Enter the number of days and the ending time to generate the load in the To field.
The ending time is defined for each particular script relative to the beginning of the test session.
If the To field is left blank, WebLOAD Console continues generating the load until the From time in the next entry for this script, or, if this is the last entry for this script, until the test session is stopped manually.
3. Under Load Size enter the number of Virtual Clients to generate during the time frame defined in steps 1 and 2. The load is distributed equally among the Load Machines defined for the script.
4. Optionally, schedule additional Virtual Client loads for this script:
 - a. Click **Add** to add a blank line to the schedule grid.
 - b. Repeat steps 1 through 3.
5. Click **Next**.

WebLOAD Console displays the Finish dialog box.

The WebLOAD Wizard continues the configuration process with the Finish dialog box. For further instructions, see *Completing the WebLOAD Wizard* (on page 130).

Limiting a Run by Number of Rounds

To limit a run by the number of rounds

- Enter the number of rounds in the **Limit Run to** field.

Creating a Collective Load Schedule with the Load Profiler

To create a collective load schedule with the Load Profiler:

1. Select an entry in the schedule grid by clicking it to enable the Load Profiler button for that entry.
2. Click the Load Profiler button.
The Load Profiler opens.

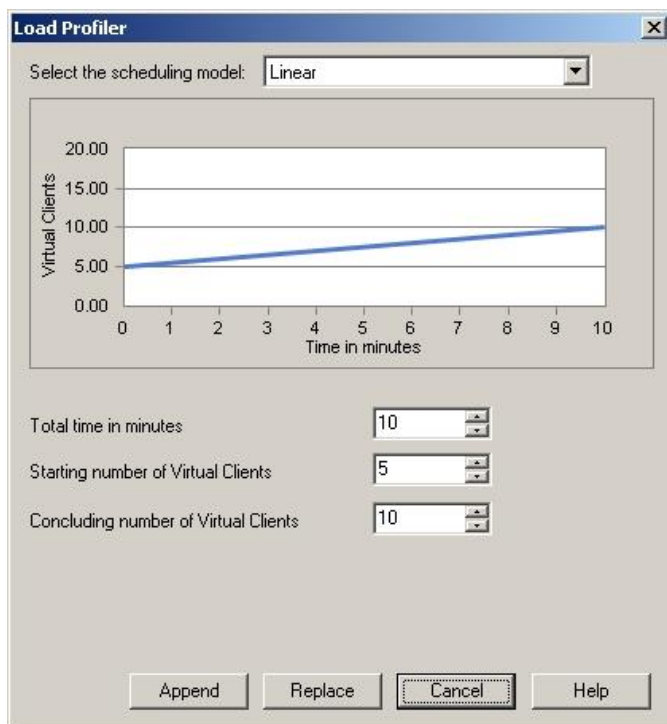


Figure 56: Load Profiler

3. Select a schedule pattern from the Scheduling model field at the top of the dialog box. For an explanation of each model, see *Scheduling Models* (on page 121).
4. Fill in the fields at the bottom of the dialog box.
5. Click **Append** to include the load definition in the schedule,
-Or-
Click **Replace** to substitute the load definition for the one appearing in the Schedule dialog box.

Individually Scheduling the Load

You can schedule the load to be generated on each Load Machine participating in the test session individually, enabling you to distribute the load based on the strengths of the Load Machines running in the test session.

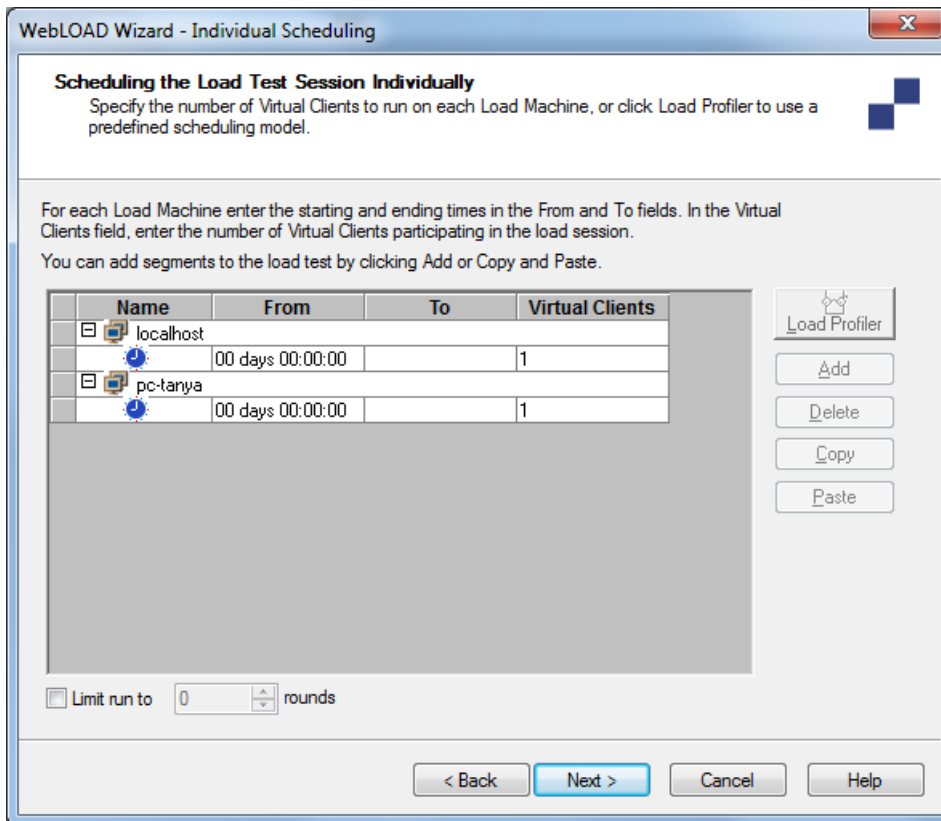


Figure 57: Individual Scheduling Dialog Box

Scheduling the Session Individually

To schedule the session individually:

1. Expand the Load Machine tree using the + / - button adjacent to the Load Machine you want to schedule.
2. Limit the run by time, by number of rounds, or both. If you limit the run by setting both a time condition and a number of rounds condition, then the run will stop as soon as one of the conditions is fulfilled.

3. To limit the run by time, perform the following for each Load Machine or Probing Client Machine defined in the Session Tree:

- a. Enter the number of days and the starting time, to generate the load in the **From** field.

The starting time is defined for each particular script relative to the beginning of the test session.



Note: Days are numbered 00 through to 99. Following the days are the hours, minutes, and seconds in the following format: hours:minutes:seconds. Seconds are numbered 1 through 59. To enter one minute, use the 1:00 syntax not 00:60.

- b. Enter the number of days and the ending time, to generate the load in the **To** field.

The ending time is defined for each particular script relative to the beginning of the test session. If the **To** field is left blank, WebLOAD Console continues generating the load until the From time in the next entry for this Load Machine, or if this is the last entry, until the test session is stopped manually.

- c. In the **Virtual Clients** field enter the number of Virtual Clients to generate during the time frame defined in steps (a) and (b).
- d. You can optionally schedule additional Virtual Client loads for this Load Machine:
 - Click **Add** to add a blank line to the schedule grid.
 - Repeat steps (a) through (c).

4. To limit the run by number of rounds, enter the number of rounds in the **Limit Run to** field.

5. Click **Next**.

The WebLOAD Wizard continues the configuration process with the Finish screen. For further instructions, go to *Completing the WebLOAD Wizard* (on page 130).

Scheduling a Session Individually with the Load Profiler

To schedule a session individually with the Load Profiler:

1. Click the **Load Profiler** button.

The Load Profiler opens.

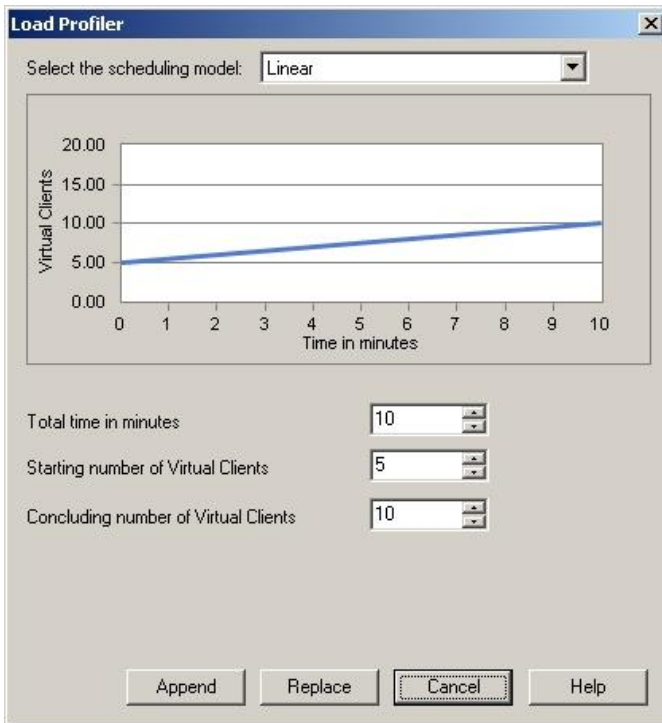


Figure 58: Load Profiler

2. Select a pattern from the Scheduling model field at the top of the dialog box.
For an explanation of each model, see *Scheduling Models* (on page 121).
3. Fill in the fields at the bottom of the dialog box.
4. Click **Append** to include the load definition in the schedule,
-Or-
Click **Replace** to substitute the load definition for the setting in the Schedule dialog box.

Scheduling Models

WebLOAD Console includes numerous scheduling models to provide a wide variety of options and aid you in quickly achieving a schedule that meets your testing goals.

Linear Scheduling Model

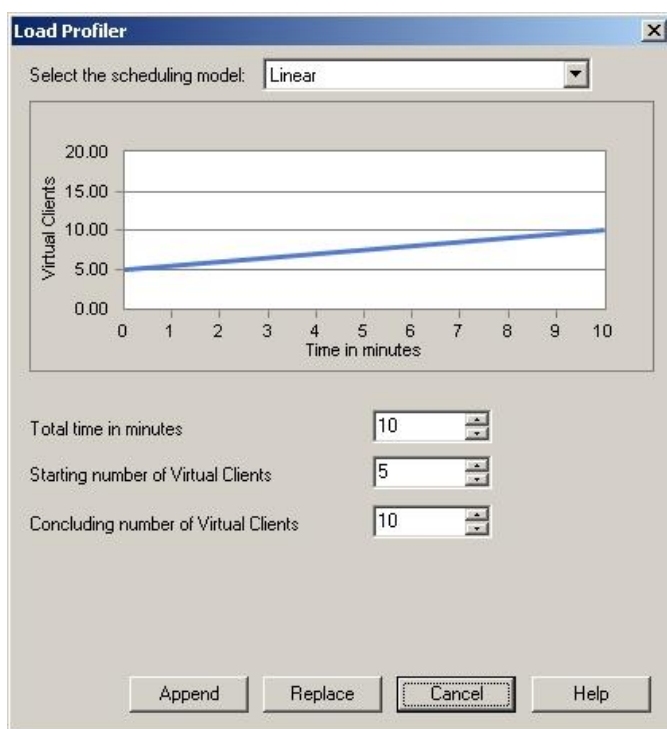


Figure 59: Linear Scheduling Model

The linear model increases the load with linearly increasing values.

Using this model you need to define:

- **Total time in minutes** – the number of minutes to generate the load using this model.
- **Starting number of Virtual Clients** – the number of Virtual Clients from which to begin.
- **Concluding number of Virtual Clients** – the number of Virtual Clients with which to end.

For example, using this model the load will start at the value defined in the Starting number of Virtual Clients field and increase (automatically calculating the intervals when the load should increase) over the time specified in the Total time in minutes field to the number of Virtual Clients defined in the Concluding number of Virtual Clients field.

Random

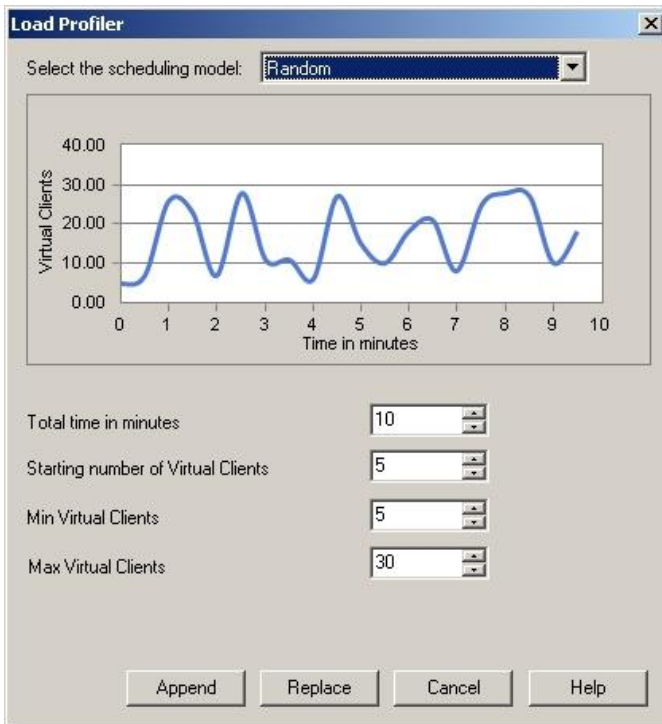


Figure 60: Random Scheduling Model

The random model scatters the load with no discernible pattern.

Using this model, you need to define:

- **Total time in minutes** – the number of minutes to generate the load using this model.
- **Starting number of Virtual Clients** – the number of Virtual Clients from which to begin.
- **Min Virtual Clients** – the absolute minimum number of Virtual Clients to generate.
- **Max Virtual Clients** – the absolute maximum number of Virtual Clients to generate.

For example, using this model the load will start at the value defined in the Starting number of Virtual Clients field and randomly increase and decrease the load between the min and max load values with no relationship or trend between each interval until the time defined in the Total time in minutes field has expired.

Incrementing Intervals

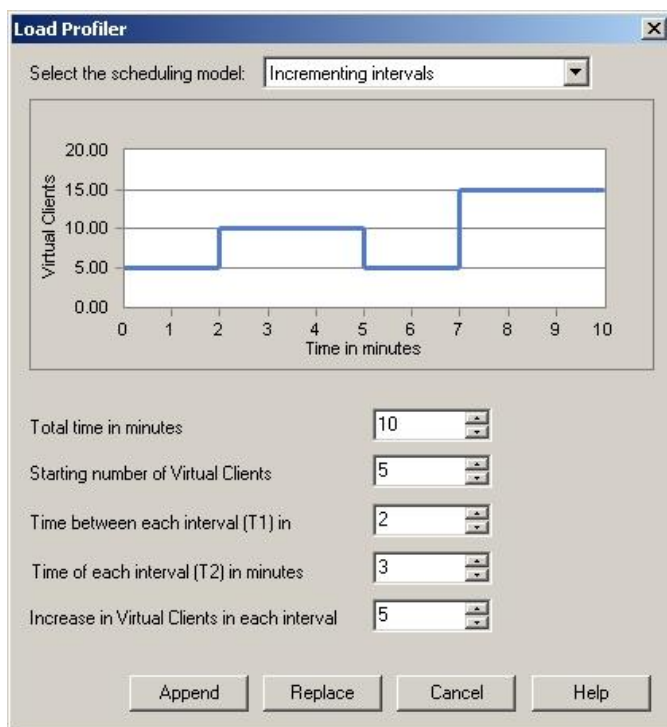


Figure 61: Incrementing Intervals Scheduling Model

The incrementing intervals model generates an increasing load in intervals while returning to the base load between intervals.

Using this model, you need to define:

- **Total time in minutes** – the number of minutes to generate the load using this model.
- **Starting number of Virtual Clients** – the number of Virtual Clients to begin from, and to return to between each interval.
- **Time between each interval (T1)** – the amount of time to wait before increasing the load for each interval.
- **Time of each interval (T2)** – the length of time for each interval.
- **Increase in Virtual Clients in each interval** – the number of Virtual Clients to add each interval (over the load generated in the last interval).

For example, using this model the load will start at the base number of Virtual Clients, increase for the time defined as T2, return to the base load and then increase again. Each subsequent increase will be the number of Virtual Clients from the previous interval plus the increment.

Incrementing Intervals (Time Calculate)

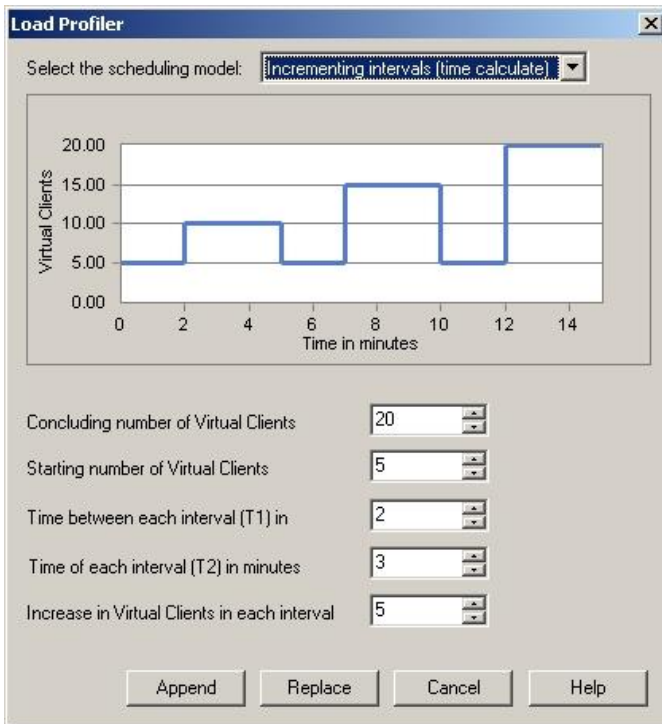


Figure 62: *Incrementing Intervals (time calculate) Scheduling Model*

The incrementing intervals (time calculate) model generates an increasing load in intervals while returning to the base load between intervals. The schedule stops when the concluding number of Virtual Clients is achieved.

Using this model you need to define:

- **Concluding number of Virtual Clients** – the number of Virtual Clients with which to end.
- **Starting number of Virtual Clients** – the number of Virtual Clients to begin from, and to return to between each interval.
- **Time between each interval (T1)** – the amount of time to wait before increasing the load for each interval.
- **Time of each interval (T2)** – the length of time for each interval.
- **Increase in Virtual Clients in each interval** – the number of Virtual Clients to add each interval (over the load generated in the last interval).

For example, using this model the load will start at the base number of Virtual Clients, increase for the time defined as T2, return to the base load and then increase again. Each subsequent increase will be the number of Virtual Clients from the previous interval plus the increment until the concluding number of Virtual Clients is achieved. The time it takes to complete the task is calculated automatically by WebLOAD Console based on the parameters defined.

Step Increments

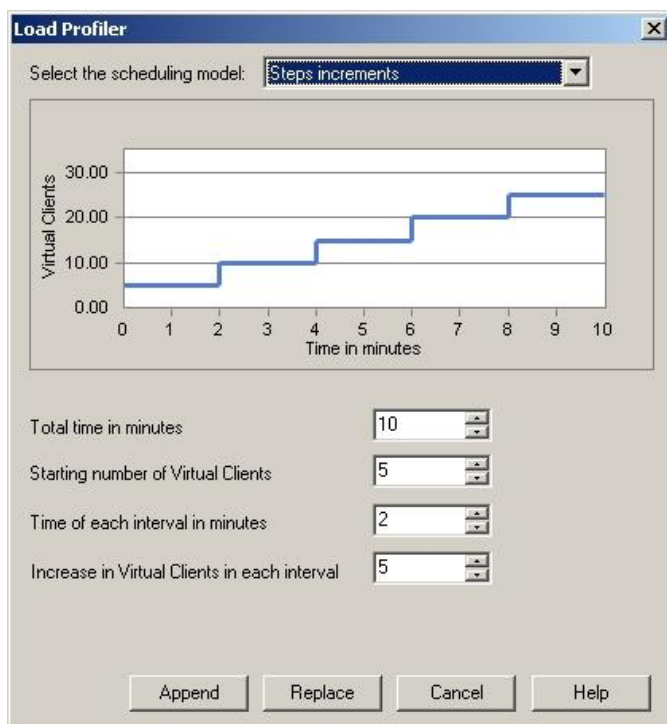


Figure 63: Step Increments Scheduling Model

The step interval model generates an increasing load in intervals.

Using this model you need to define:

- **Total time in minutes** – the number of minutes to generate the load using this model.
- **Starting number of Virtual Clients** – the number of Virtual Clients from which to begin.
- **Time of each interval** – the length of time for each interval.
- **Increase in Virtual Clients each interval** – the number of Virtual Clients to increase each interval.

For example, using this model the load will start at the value defined in the Start number of Virtual Clients field, and increase gradually the amount of user defined in the Increase in Virtual Clients each interval field.

Step Increments (Time Calculate)

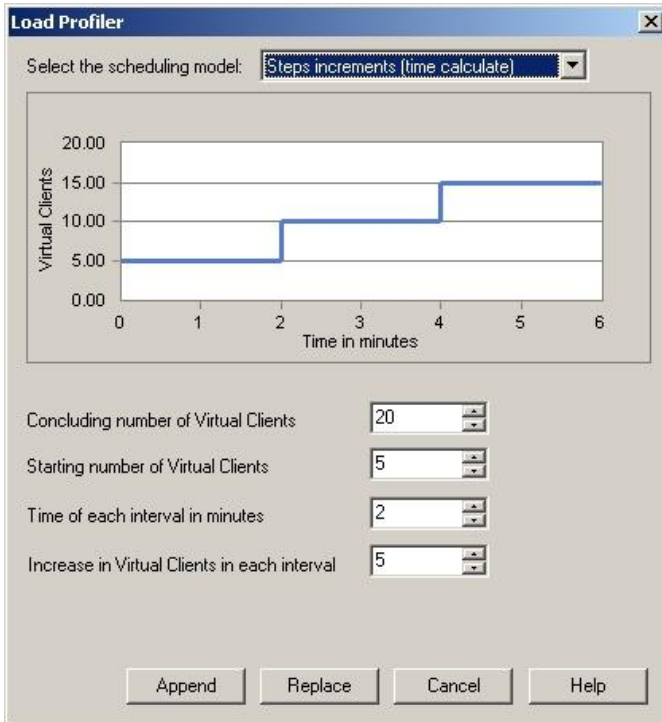


Figure 64: Step Increments (time calculate) Scheduling Model

The step interval model generates an increasing load in intervals until a defined load size is reached. The time to complete the scheduled model is calculated automatically by WebLOAD Console based on the parameters defined.

Using this model you need to define:

- **Concluding number of Virtual Clients** – the number of Virtual Clients with which to end.
- **Starting number of Virtual Clients** – the number of Virtual Clients from which to begin.
- **Time of each Interval** – the length of time for each interval.
- **Increase in Virtual Clients in each interval** – the number of Virtual Clients to increase each interval.

For example, using this model the load will start at the value defined in the Starting number of Virtual Clients field, and increase gradually the amount of user defined in the Increase in Virtual Clients each interval field until the concluding number of Virtual Clients is achieved.

Ramp Up

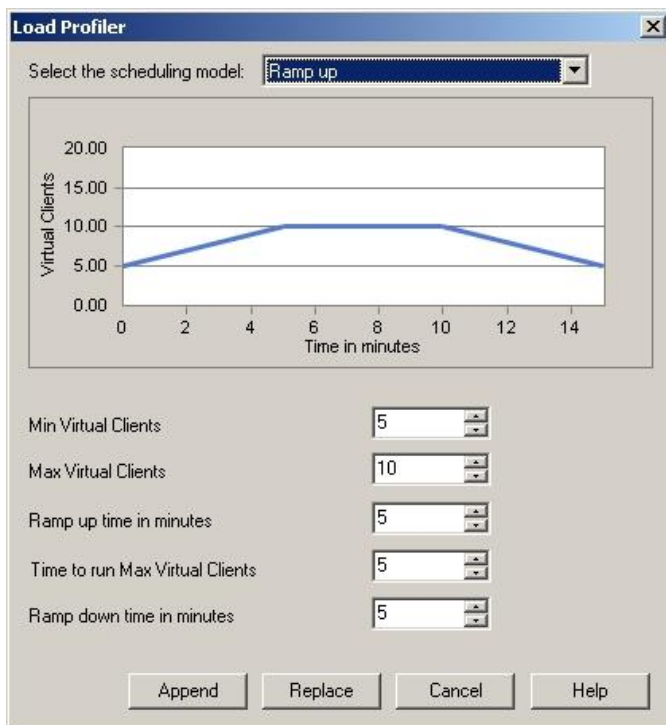


Figure 65: Ramp Up Scheduling Model

The ramp up model generates an increasing load over a set time frame until reaching a defined maximum load size. The maximum load size is then run for a set period and then the load is reduced back to the base load again over a time frame defined by the user. The load increments are calculated automatically by WebLOAD Console based on the parameters defined.

Using this model you need to define:

- **Min Virtual Clients** – the number of Virtual Clients from which to begin.
- **Max Virtual Clients** – the maximum number of Virtual Clients to generate using this model.
- **Ramp up time** – the time interval over which to increase the load in order to reach the max load size.
- **Time to run Max Virtual Clients** – the time frame to run the load at the max number of Virtual Clients.
- **Ramp down time** – the time interval over which to decrease the load in order to return to the minimum number of Virtual Clients.

For example, using this model the load will start at the value defined in the Min Virtual Clients field, and increase over the time frame defined in the Ramp up time field to reach the number of Virtual Clients defined in the Max Virtual Clients field. The test will then continue running with the maximum number of Virtual Clients for the period of time defined in the Time to run Max Virtual Clients field after which, load will return to the base number of Virtual Clients value over the time frame set in the Ramp down time field.

User Defined

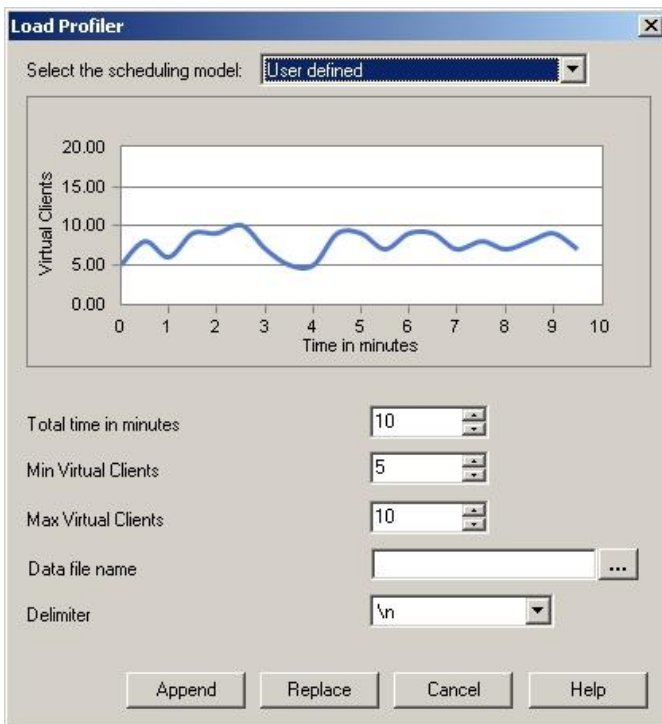


Figure 66: User Defined Scheduling Model

The user-defined model generates the load based on values extracted from a data file. The time to complete the scheduled model is defined by the user as is the minimum and maximum number of Virtual Clients.

Using this model you need to define:

- **Total time in minutes** – the number of minutes to generate the load using this model.
- **Min Virtual Clients** – the absolute minimum number of Virtual Clients to generate over the schedule. At no point in the schedule should the number of Virtual Clients dip below this value.

- **Max Virtual Clients** – the absolute maximum number of Virtual Clients to generate over the schedule. At no point in the schedule should the number of Virtual Clients rise above this value.
- **Data file name** – the name and location on the data file to read values from for generating the load.
- **Delimiter** – the delimiter used in the data file from which values are read to generate the load size.

Using this model the WebLOAD Scheduler will read values from the assigned data file and generate the load based on the values in the file. The number of Virtual Clients will fluctuate based on the number of values in the file and the time frame for which the schedule is set to run. Although the load values are based on the values in the file, the values you define for the minimum and maximum number of Virtual Clients are used to make sure that, at no time during this schedule, the number of Virtual Clients falls below the minimum value or rises above the maximum values defined. That is, the number of Virtual Clients values from the file are proportionally modified to fit within the minimum and maximum number of Virtual Clients values. For example, if your data file contains the values 50 and 100 over 1 minute and you define the minimum and maximum number of Virtual Clients as 1 and 10, WebLOAD Console automatically adjusts the values in the data file to 5 and 10 over 1 minute.

You can define a text document containing the different numbers of Virtual Clients you want and WebLOAD Console spreads those numbers out evenly during the test. For example, if your test is defined to run for ten minutes and your text file defines five numbers, then the number of Virtual Clients changes every two minutes according to the values in the file.

The following figure displays a sample text document. To use the file for the test, the name of the file is entered in the Data file name field.

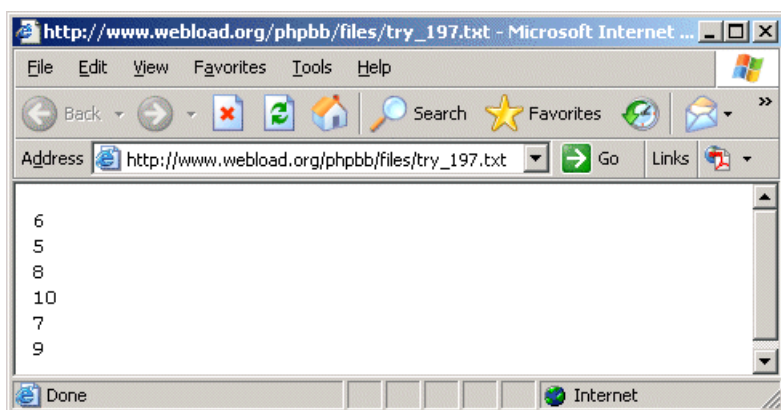


Figure 67: Sample Data File

Selecting the Performance Measurements

The WebLOAD Wizard enables you to open the Performance Measurements Manager and configure WebLOAD Console to collect performance statistics from your application server, database, and Web server while the test is running.

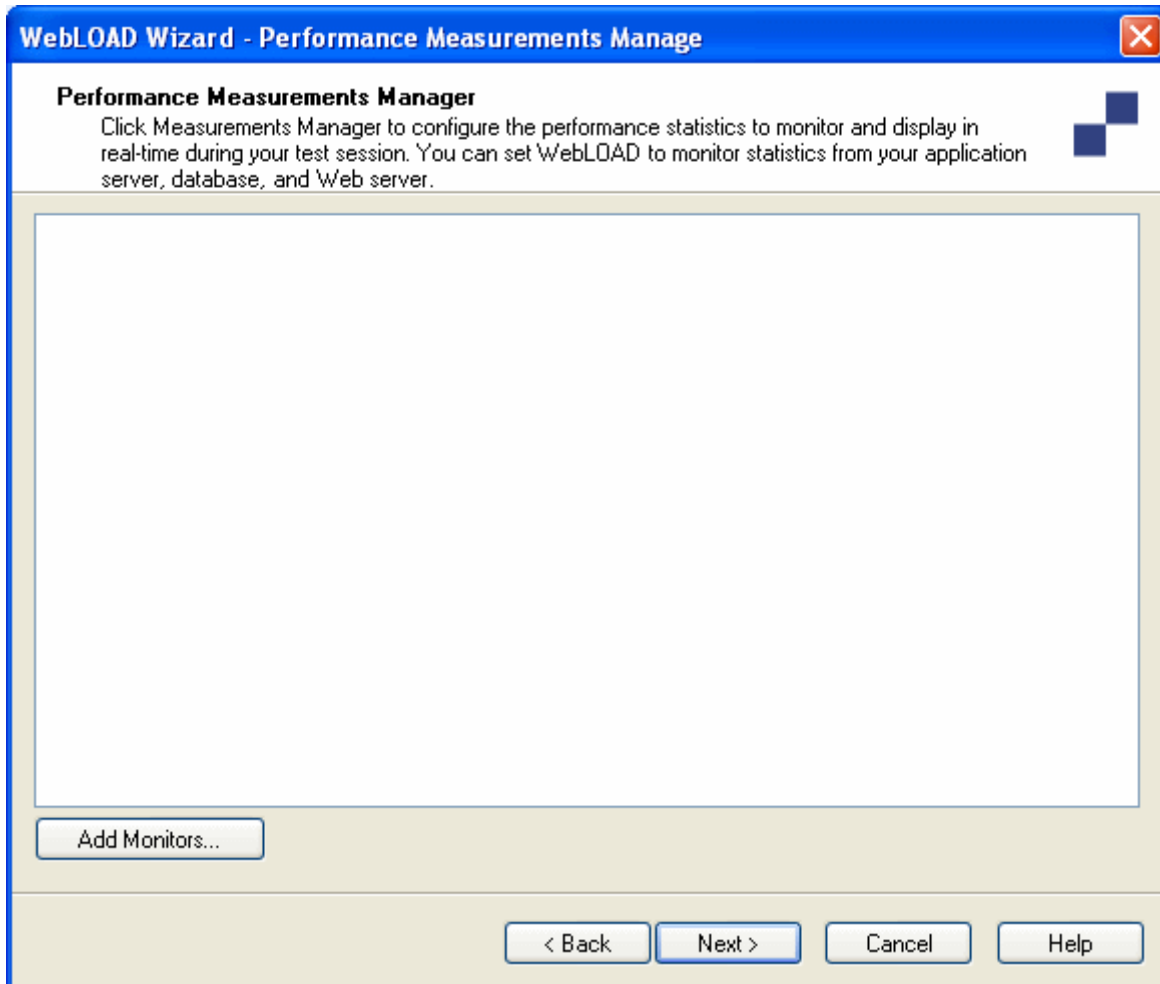


Figure 68: Performance Measurements Manage Dialog Box

- Click **Add Monitors** to configure the performance statistics to monitor and display in real-time during the test session.

The Performance Measurements Manager opens. For more information on configuring the statistics to monitor, see *Performance Measurements Manager* (on page 363).

Completing the WebLOAD Wizard

The product of your performance testing is the WebLOAD report. Here you can see how your Web application handles the load that you have tested. The WebLOAD Wizard automatically generates a predefined report that measures:

- Hits per Second
- Round Time (the amount of time required to run one round of the script)
- Number of Virtual Clients
- Throughput (the number of bytes per second transmitted from the SUT)

WebLOAD Console saves all the values for the complete set of measurements. You can decide after the test has run to display the results never before requested.



Figure 69: Completing the WebLOAD Wizard Dialog Box

To complete the WebLOAD Wizard:

1. Check **To run this Load Template immediately.**
2. Click **Finish.**

A message box appears asking if you want to save the current Load Template.

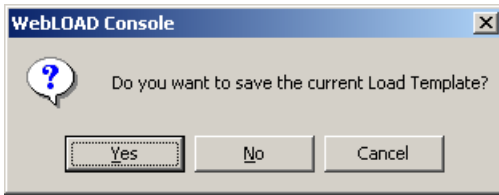


Figure 70: Save Current Template Message Box

3. Click **Yes** to save the current Load Template as a *.tpl file.

Load Templates contain the parameters that you defined for the test session, such as scripts to run, participating hosts, and the test plan. Using Load Templates saves you the time of reconfiguring the Load Session when repeating tests. For more information on Load Templates see *Managing Load Templates* (on page 91).

The WebLOAD Wizard closes and the load test begins.

After Completing the WebLOAD Wizard

After you have completed the WebLOAD Wizard you can do the following:

- Run the Wizard to add a new script/mix to the template
- Edit the Load Template
- View the test results
- Modify the host selection
- Modify the load schedule

Adding a New Script/Mix to the Template

If you want to add an additional combination of script/mix, host selection and load schedule, right click the Load Session and select **Open Wizard**. Complete the wizard.

Editing the Load Template

After completing the WebLOAD Wizard, a Load Template with one Load Machine is created. You can save this Load Template for future Load Sessions. The Load Template can then be edited through the Console ribbon and button.

Viewing the WebLOAD Wizard Default Report

The WebLOAD Wizard automatically generates a predefined report that measures:

- Hits per Second
- Round Time (the amount of time required to run one round of the script)

- Load Size (the number of Virtual Clients being generated)
- Throughput (the number of bytes per second transmitted from the SUT)

You can configure your own report through the Console Reports menu. WebLOAD Console saves all of the values for the complete set of measurements. You can display the results never before requested. For information on viewing reports, see *Running Tests and Analyzing Test Results* (on page 269).

Modifying the Host Selection

After closing the WebLOAD Wizard you can edit the parameters in the Host Selection dialog box.

To modify the host selection:

1. Click **Load Machine Selection** in the **Home** tab of the ribbon.
The Host Selection dialog box opens.
2. Make the changes as needed.
3. Click **OK**.
The host selection parameters are changed.



Note: If after closing the WebLOAD Wizard you want to edit parameters in the Host Selection dialog box, click **Load Machine Selection** in the **Home** tab of the ribbon.

Modifying the Load Schedule

After closing the WebLOAD Wizard you can edit the parameters in the load schedule.

To modify the load schedule:

1. Click **Schedule** in the **Home** tab of the ribbon.
The Schedule Manually screen appears.
2. Make the changes.
3. Click **OK**.
The load schedule is changed.

Creating Load Templates with Goal-Oriented Test

WebLOAD Console offers the Goal-Oriented Test Wizard, a unique feature that enables goal-seeking performance testing.

Using the Goal-Oriented Test

The Goal-Oriented Test is a predictive feature within WebLOAD Console that identifies the performance threshold of a Web application. You define one or multiple performance parameters and the Goal-Oriented Test automatically creates a schedule and increases the load until your Web application performance falls below the user-defined profile. The Goal-Oriented Test incorporates intelligent, distributed load generation capabilities for virtually unlimited load generation.

Using the Goal-Oriented Test wizard to create a Load Template, you define:

- The script(s) to run
- The Load Machines on which the load is generated
- The performance goals to achieve

You can also specify script options including the type of browser to emulate, connection speed, and playback sleep time options.



Note: You cannot use the Goal-Oriented Test wizard if you have not yet recorded (or otherwise created) any scripts. If you do not have any scripts to work with, you must exit the wizard and create a script using one of the authoring tools, such as WebLOAD Recorder.

How does the Goal-Oriented Test Work?

When operating WebLOAD Console manually, you specify the number of Virtual Clients that WebLOAD Console should simulate. The evaluation of the SUT's performance is based on this number. For example, you might configure WebLOAD Console to test the performance of your Web application when 200 Virtual Clients perform a certain function simultaneously. WebLOAD Console runs the test and may report that when 200 Virtual Clients are running, the application Response Time is "x".

The Goal-Oriented Test is different. The Goal-Oriented Test enables you to specify the performance goals that you want to achieve, and WebLOAD Console generates a Goal-Oriented Test test that automatically schedules a test to achieve the specified goals. For example, if your performance goal is for your application server's response time not to exceed three seconds, the Goal-Oriented Test devises the test for you. The Goal-Oriented Test tells you how many users your application server is able to accommodate when your performance goals are met.

The Goal-Oriented Test is a unique WebLOAD Console feature. It enables you to devise your test to reflect the real way that you want to see your application performance. Many times, you do not know the number of clients that will access your site, but you do know the quality of service you want to achieve. Rather than running multiple tests to measure the performance at different loads, you can run the Goal-Oriented Test once, measuring the performance level you desire.

The Goal-Oriented Test Wizard Workflow

The following diagram illustrates the Goal-Oriented Test Wizard workflow.

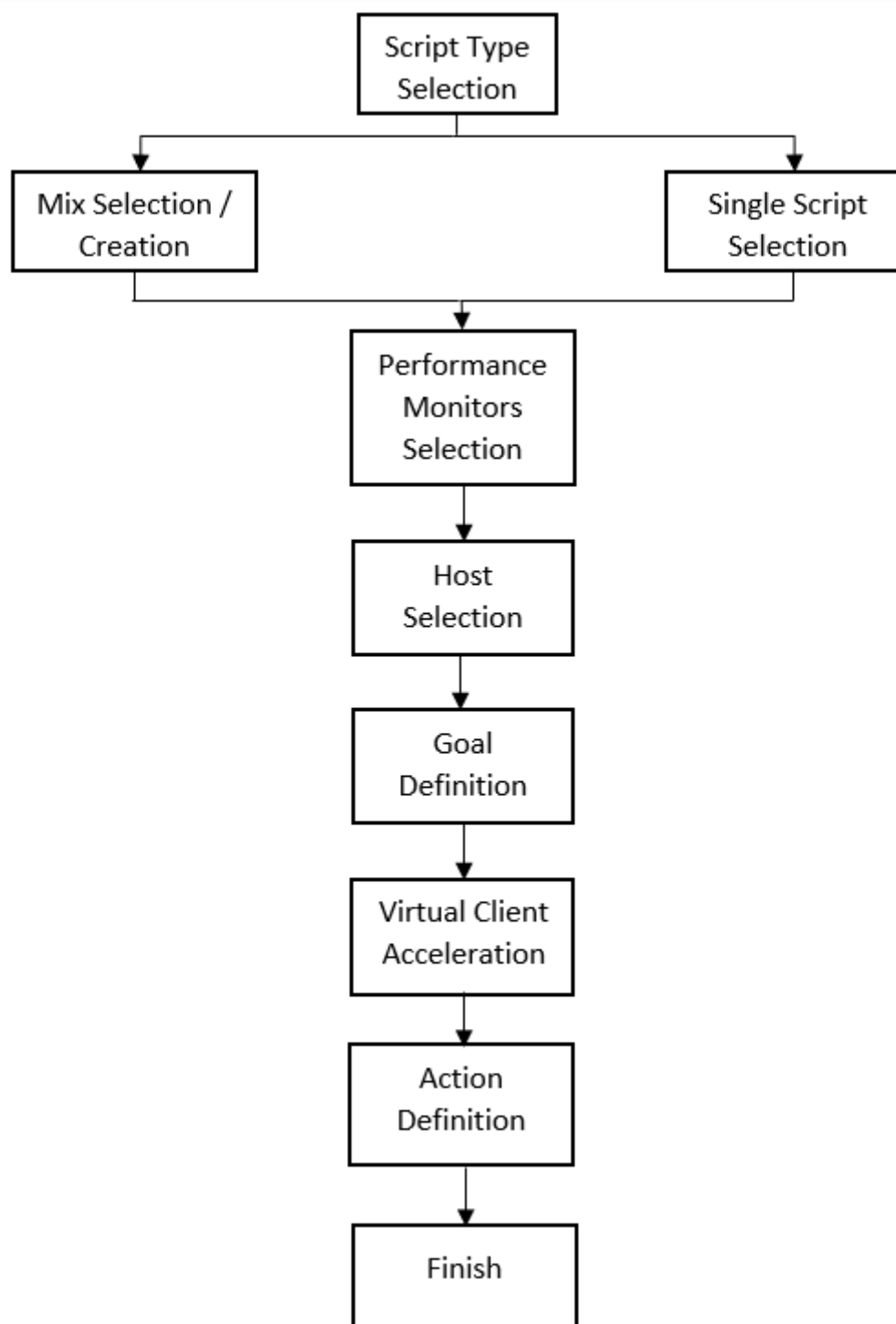


Figure 71: Goal-Oriented Test Wizard Workflow Diagram

Opening the Goal-Oriented Test Wizard

To open the Goal-Oriented Test Wizard:

1. Select **Goal-Oriented Test Wizard** from the Console shortcut dialog box at the system startup,

-Or-

Click **Goal-Oriented Test Wizard** in the Home tab of the ribbon.

The Goal-Oriented Test Wizard Welcome dialog box opens.

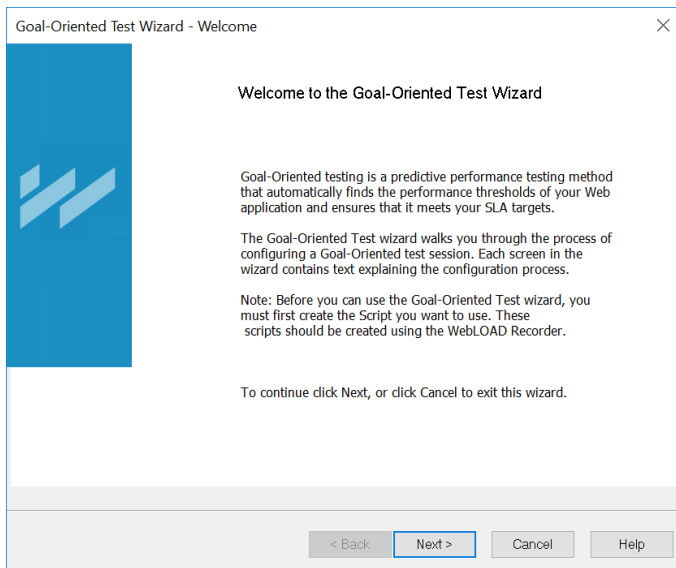


Figure 72: Goal-Oriented Test Wizard Welcome Dialog Box

2. Click **Next**.

The Goal-Oriented Test Wizard progresses to the Script Type screen.

Selecting a Script or Mix

The next step in configuring a Goal-Oriented test is to define the type of script you want to run during your test session. Scripts are test scripts (scenarios) that describe user activity, and are used to test your Web application.

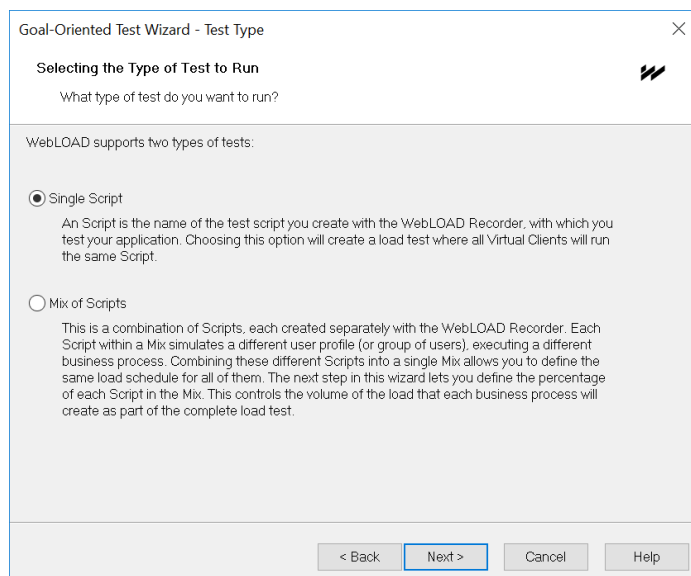


Figure 73: Script Type Selection Dialog Box

WebLOAD Console supports two types of scripts:

- **Single Scripts** – which are lone test-scripts.
- **Mix of Scripts** – which are a combination of scripts used to simulate groups of users performing different activities on the system under test (SUT) at the same time.

To run a single script:

1. Select **Single Script**.
2. Click **Next**.

WebLOAD Console progresses to the Script Selection screen. See *Running Single Scripts* (on page 140).

To run a Mix of Scripts:

1. Select **Mix**.
2. Click **Next**.

WebLOAD Console progresses to the Mix Selection screen. See *Running a Mix of Scripts* (on page 141).

Running Single Scripts

You must have a script to continue configuring a test with the Goal-Oriented Test Wizard. If you do not have any scripts to work with, you must exit the wizard and go into one of the authoring tools, for example WebLOAD Recorder, and create a script.

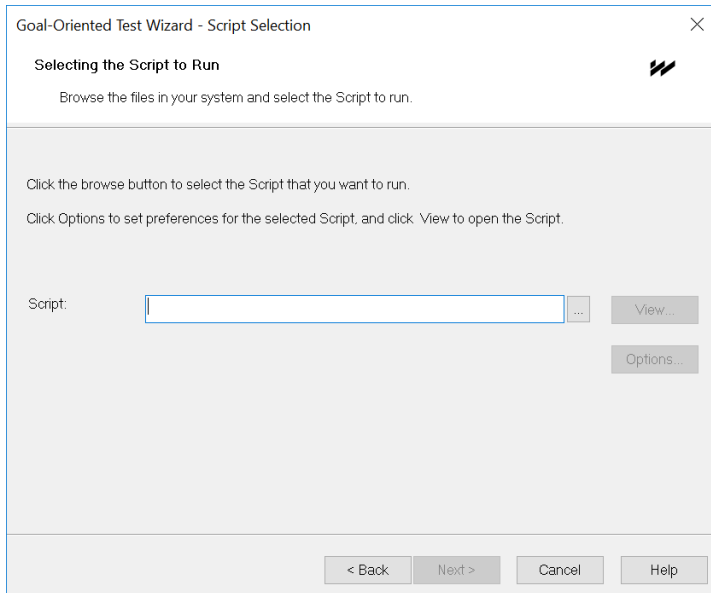



Figure 74: Script Selection Dialog Box

To run a script:

1. Select the source of the script to be used for this test. Click the  button and select an existing script from your system.
2. To optionally open and view or edit the selected script, click **View**.
WebLOAD Recorder opens displaying the selected script.
3. To optionally configure runtime options exclusive to this script, click **Options**.

The Script Options dialog box opens enabling you to define the runtime options for the selected script. For more information on script options, see *Setting Script Options* (on page 211).

4. Click **Next**.

The Goal-Oriented Test Wizard progresses to the PMM Settings screen. See *Selecting Performance Measurements* (on page 145).

Running a Mix of Scripts

Using the Goal-Oriented Test Wizard you can run your test using a saved Mix or create a new Mix of scripts through the wizard. The following illustration shows the process for selecting a Mix of scripts to be used in a Goal-Oriented Test test.

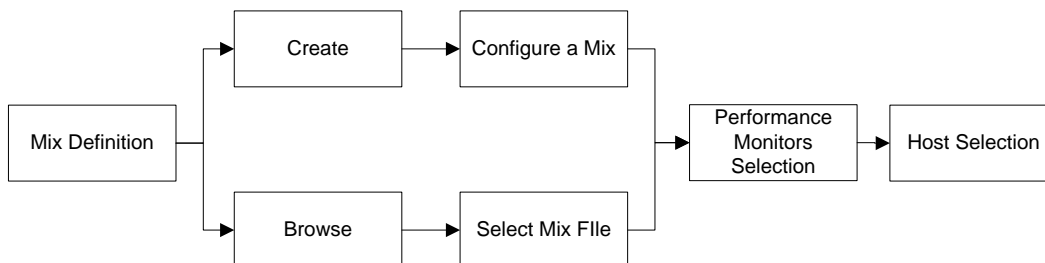


Figure 75: Mix of Scripts Selection Process Diagram


Using Mix you can prepare a combination of scripts, each performing a specific activity. Then you can tune your testing by changing the combination of scripts, and changing the weight of each script in the Mix.

Combining scripts through a Mix can simulate different groups of users performing different activities on the SUT at the same time. Using a Mix is another way you can create test scenarios emulating real-life behavior.



Figure 76: Mix Selection Dialog Box

To run a Mix of scripts:

1. Select the source for the Script Mix to be used for this test from the **Mix Selection** dialog box. You can either:
 - Select **Use an Existing Mix** and click the  button to select a *.mix file from your system.
 - Select **Create a Mix** to configure a new Mix.
2. Click **Next**.

WebLOAD Console progresses to the Mix Definition dialog box. See *Defining a Mix* (on page 142).

Defining a Mix

A Mix is a set of scripts, each performing different activity, to simulate groups of users performing different activities on the SUT at the same time. If you selected **Use an Existing Mix** on the script / Mix Selection screen, the selected Mix appears on the screen, enabling you to modify the Mix. You can also create a new Mix.

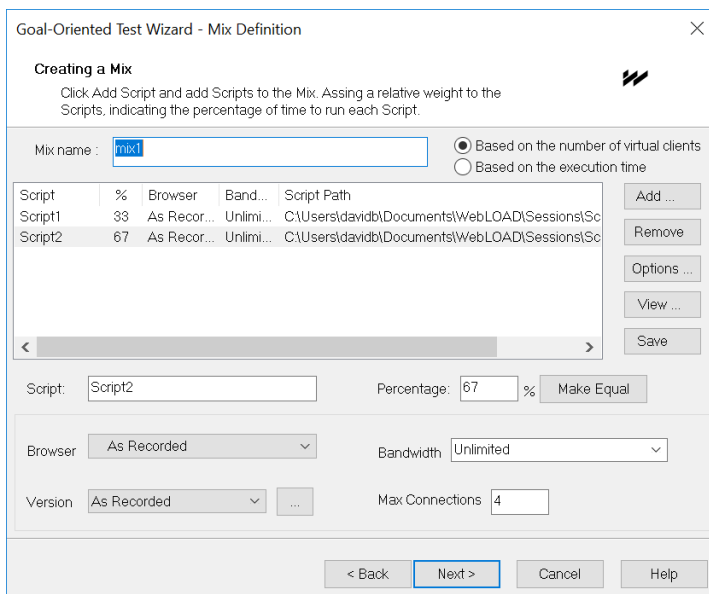


Figure 77: Mix Definition Dialog Box

Creating a Mix

To create a Mix:

1. Enter a descriptive name in the **Mix name** field.
2. Specify whether the relative weights of the script refer to execution time or to number of virtual clients:

- **Based on the number of virtual clients** – The mix interprets the relative weights in terms of number of virtual users. For example, if the weight of Script A is 25 and of Script B is 75, the mix execution runs script A with 25% of the virtual users and script B with 75% of the users.
 - **Based on the execution time** – The mix interprets the relative weights in terms of time. For example, if the weight of Script A is 25 and of Script B is 75, the mix execution run script A for 25% of the time and script B for 75% of the time.
3. Click **Add**.
The Open dialog box appears, enabling you to select the script to add to the Mix.
 4. Select a script from your file system and click **Open**.
The script is added to the Mix.
 5. Optionally, select the script and set any of the following parameters:
 - a. **Script name** – Specify a name for the script and its particular settings. Keep in mind that in a mix, in addition to running different scripts, you can also run the same script under different settings. For example, a mix can contain two items: Script A running on Chrome, and the same Script A running on Mozilla Firefox. Give a different descriptive name to each of the two mix items.
 - b. **Weight** – Specify the relative weight of this script, as a positive whole number. The application will run each script based on its weight in relation to the total of weights. When the total of weights is 100, the weight is equivalent to a percentage; however, this is not required. For example, if you specified a weight of 1 for Script A and a weight of 3 for Script B, then the system assigns script A a relative weight of 1/4, and to Script B the relative weight of 3/4. This is equivalent to a weight of 25 for Script A and 75 for Script B; in both cases, Script A will run 25% of the rounds, and Script B will run 75% of the rounds.
 - c. **Browser and Version** – Select the browser type and browser version. Alternatively, click the Change button to edit the user agent definition. See *Adding a Browser Version Definitions* (on page 223).
 - d. **Bandwidth** – Specify a bandwidth. You can do this in one of two ways:
 - Select a bandwidth from the drop-down list.
 - Enter a number, to specify that number of bits per second.

- e. **Max Connections** – Specify the number of max connections for the load generator.
6. Optionally, click **Options** to set any of the settings available from the Script Options dialog box (such as the browser to emulate, connection settings and playback sleep options). This will change the settings of the currently selected item only.

The Script Options dialog box opens. For information on setting the Script Options, see *Setting Script Options* (on page 211).

7. Repeat steps 3 through 6 to add additional scripts to the Mix.
8. In the Weight box, type or select the relative weight for the Script. Click **Make Equal** to assign equal weight to all of the Scripts in the Mix.
9. Click **Save**.

The Mix configuration is saved with the extension *.mix.

10. Click **Next**.

WebLOAD Console progresses to the PMM Settings screen. See *Selecting Performance Measurements* (on page 145).

Deleting a script from a Mix

To delete a script from a Mix:

1. Select the Script you want to delete from the Mix.
2. Click **Remove**.

Viewing a script from a Mix

To view a script from a Mix in the WebLOAD Recorder:

1. Select the script you want to view.
2. Click **View**.

The WebLOAD Recorder is launched, and displays the selected script.

Selecting Performance Measurements

You can optionally open the Performance Measurements Manager and configure WebLOAD Console to collect performance statistics from your application server, database, and Web server while the test is running.

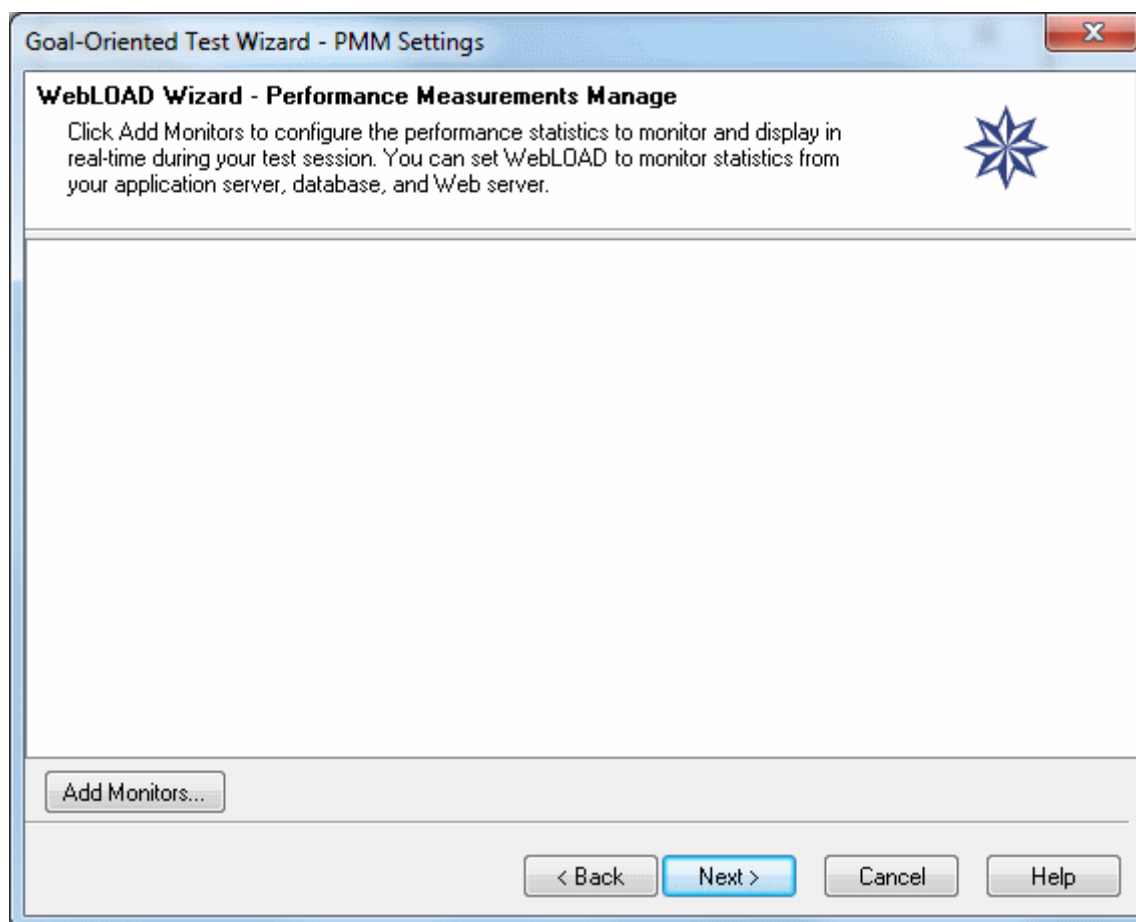


Figure 78: Performance Measurements Manage Dialog Box

1. Click **Add Monitors** to the Performance Measurements Manager and configure the performance statistics to monitor and display in real-time during the test.

The Performance Measurements Manager opens. For more information on configuring the statistics to monitor, see *Performance Measurements Manager* (on page 363).

2. Click **Next**.

The Goal-Oriented Test Wizard progresses to the Host Selection screen. See *Selecting Host Computers* (on page 146).

Selecting Host Computers

After selecting the script to use for the test, WebLOAD Console displays the Host Selection dialog box.

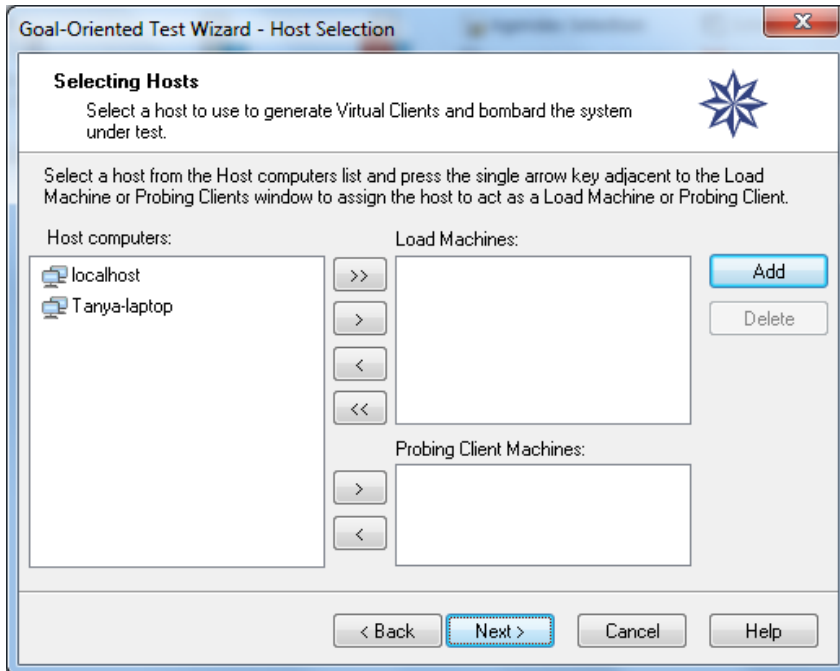


Figure 79: Host Selection Dialog Box

Hosts are the computer systems participating in the test session. In this dialog box you define the hosts to participate in the Goal-Oriented test. Host computers can be dedicated, local computers, or they can be computers which are rented on a temporary basis from the Amazon™ Elastic Compute Cloud™ (EC2). Hosts can act as Load Machines or Probing Client Machines.

For more information about setting up EC2 computers as host computers, refer to *Setting Up Cloud Computers* (on page 108).

Load Machines:

By selecting a list of Load Machines, you define a pool of Hosts that WebLOAD Console uses to generate a load until the Goal-Oriented Test goal is reached. To reach your defined goal, WebLOAD Console may need to generate a large number of users accessing the SUT. The number of users that can be generated on each host depends on the power of each host machine. Defining many machines as Load Machines allows for the simulation of more users throughout the Goal-Oriented test.

Probing Client Machines:

WebLOAD Console reports exact measurements for single Probing Clients. By defining Probing Client Machines on different hosts, you will receive exact performance measurements for that client, rather than averages calculated for users generated on Load Generators. By defining a list of Probing Client Machines, you can receive accurate measurements at different locations (depending on the location of the host). Therefore, it is recommended that you define at least one Probing Client Machine in a Goal-Oriented Test Session, see *Defining Performance Goals* (on page 151).



Note:

If you are running a script with Perfecto Mobile script, define a single Load Machine or a single Probing Client in order to simulate a single user.

Adding Host Computers to the Host Computers List

To add a host computer to the host computers list:

1. In the Host Selection dialog box (Figure 79), click **Add**.

The Add Host Computer dialog box appears.

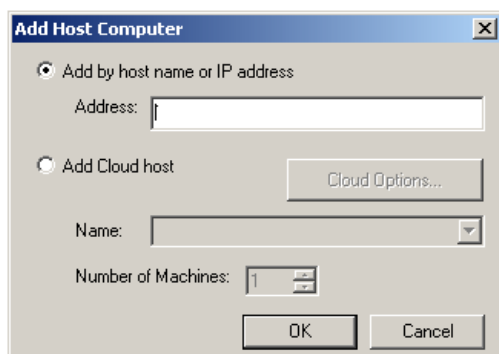


Figure 80: Add Host Computer Dialog Box

2. To add a local host computer, select **Add by host name or IP address** and enter the host computer name or IP address in the **Address** field.
3. To add cloud host machines:
 - a. Select **Add Cloud host**.
If you have not yet created a WebLOAD cloud account, a dialog box appears. Click **OK** and the Cloud Accounts dialog box appears (Figure 52). Create one or more WebLOAD cloud accounts as described in *Creating WebLOAD Cloud Accounts* (on page 109).
 - b. Select a WebLOAD cloud account from the **Name** drop-down box.
 - c. In the **Number of Machines** field, enter the number of machines you wish to add from the selected WebLOAD cloud account.
 - d. Repeat steps b-c to specify machines from another WebLOAD cloud account.


Optionally, click **Cloud Options** to add or modify WebLOAD cloud accounts. The Cloud Accounts dialog box appears (Figure 52). For information about the available actions, refer to the explanations following Figure 52.

4. Click **OK**.
 - If you added a local host machine, the machine is added to the Load Machines list in the Host Selection dialog box (Figure 79), and is listed by the host name or IP address you entered.
 - If you added cloud host machines, a list of machines is added to the Load Machines list in the Host Selection dialog box with the names “<WebLOAD Cloud Account Name><X>” where X is a number that is incremented for each added cloud host machine.

For defining Host Computers as Load Machines and Probing Clients, see *Designating a Host Computer as a Load Machine* (on page 148) and *Designating a Host Computer as a Probing Client Machine* (on page 149).

Designating a Host Computer as a Load Machine

To designate a host computer as a Load Machine:

1. Select a host from the Host Computer window.
To add hosts not yet defined to the Host Computer window, see *Adding Host Computers to the Host Computers List* (on page 147).
2. Click the single arrow key  adjacent to the Load Machine or Probing Clients window.
The selected host moves to the Load Machine or Probing Clients window.
3. Repeat steps 1 and 2 to define all of the hosts participating in the test.

4. Click **Next**.

After configuring the hosts participating in the Goal-Oriented test, the Goal-Oriented Test Wizard prompts you for the goals you want to achieve.

The Goal Definition dialog box appears, see *Defining Performance Goals* (on page 151).



Note: If you define a Probing Client host, the Probing Client Script Selection dialog box appears

Designating all Host Computers as Load Machines

To designate all Host Computers as Load Machines:

- Click the double arrow key  adjacent to the Load Machine window.


All of the host computers move to the Load Machine window.



Note: At least one system must be configured as a Load Machine to continue working with the wizard.

Designating a Host Computer as a Probing Client Machine

To designate a host computer as a Probing Client Machine:

- Select the host from the Host Computer window.
- Click the single arrow key  adjacent to the Probing Client Machines window.

The selected host moves to the Probing Client Machine window.

Deleting a Host Computer

To delete a host computer:

- Select the host name from the Host Computer list.
- Click **Delete**.

The host is deleted from the Host Computer list.

Configuring Probing Clients

If you designate a host computer as a Probing Client Machine, you are prompted to assign a script for each Probing Client participating in the test session. The following screen is displayed:

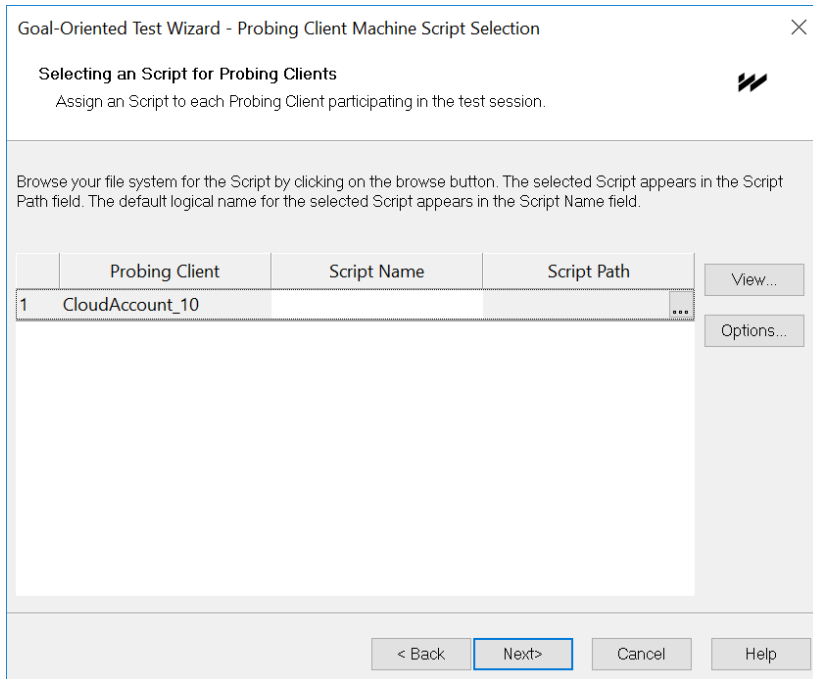


Figure 81: Probing Client Machine Script Selection Dialog Box

To assign a script for a Probing Client machine:

1. Browse your file system for the script by clicking the button next to the Script Path field. The path and the name of the script you select are displayed in the fields.
2. Click **Next**.

Defining Performance Goals

After defining the host computers participating in your test, the Goal-Oriented Test Wizard prompts you for the performance goals you want to achieve in your test session.

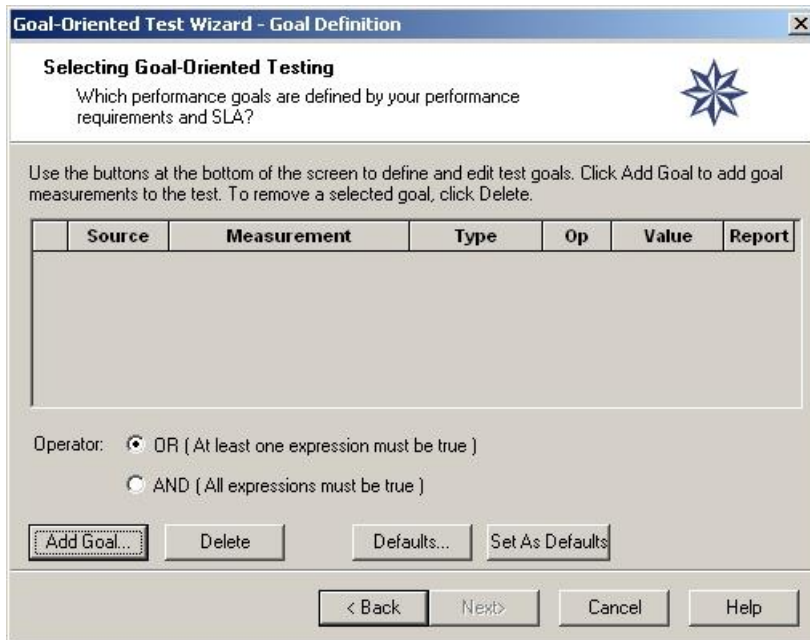


Figure 82: Selecting Goal-Oriented Test Performance Goals Dialog Box

About the Goal Definition Dialog Box

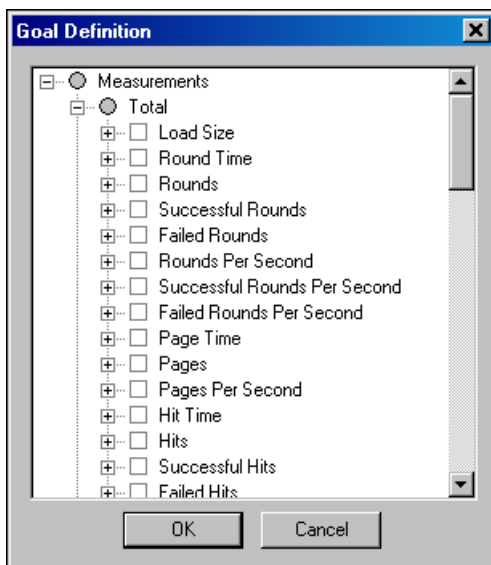


Figure 83: Goal Definition Dialog Box

The Goal Definition dialog box enables you to define the performance goals for the Goal-Oriented test. The defined goals are displayed at the top of the dialog box where you can see at a glance all of the participants in the test, the measurements being tested and the goals to be achieved.

On the bottom of the dialog box define the (OR/AND) conditions of the goals where you specify whether at least one or all of the goals must be met. The buttons on the lower half of the dialog box are used for defining and editing these goals.

The Goal Table

The goal table in the center of the dialog box contains the following fields:

Table 27: Goal Table Fields

Field Name	Description
Source	<p>The source you are tracking to determine when your goal is reached. The source can be either:</p> <ul style="list-style-type: none"> • A Probing Client Machine • A Load Machine <p>The source for Load Machines is "Total" because your results are the average of all Virtual Clients. The Source for Probing Client Machines is the name of the Probing Client.</p>
Measurement	<p>The measurement or activity that you are testing such as:</p> <ul style="list-style-type: none"> • Response Time • Hits Per Second • Round Time
Type	<p>The type of measurement or activity being tested. This is the current slice average or current slice sum.</p>
Op	<p>The operator is the condition of the test. This may be either >= or <= depending on the measurement being tracked. Here you indicate whether the results should be greater than, or less than the desired value for the goal to be reached. For example, if the measurement being tracked were Response Time, you would select the operator >= to achieve the goal when the time needed to receive a response from the server exceeds the limit for this goal.</p>
Value	<p>The limit that determines when this goal is achieved.</p>

Field Name	Description
Report	When this field is checked, the default Integrated Report includes the measurement for which the performance goal is defined. The Report field is selected by default.

Operator

Below the Goal Table are the Operator radio buttons. When more than one goal is defined, the Operator enables you to establish the conditions under which the goal will be achieved.

- Select **OR** to require only one expression to be true for the goal to be achieved.
- Select **AND** to require all expressions to be true for the goal to be achieved.

Control Buttons

Below the Operator radio buttons, are the following control buttons:

Table 28: Control Buttons

Button	Description
Add Goal	Open the Add Goal dialog box to add a new measurement to the test goals.
Delete	Delete a test goal.
Defaults	Open the Default Goals dialog box. For more information on using Defaults, see <i>Setting and Using Defaults</i> (on page 157).
Set As Defaults	Set the current goals as default measurements and save them for future use.

Adding Performance Goals

Goals are configured by defining boundary values for selected WebLOAD Console measurements. Goals define the performance level you expect from the SUT. WebLOAD Console increases the load and runs the test until the defined goals are reached.

To add performance goals:

1. Click the **Add Goal** button to add measurements on which to base your goals.

The Goal Definition dialog box opens.

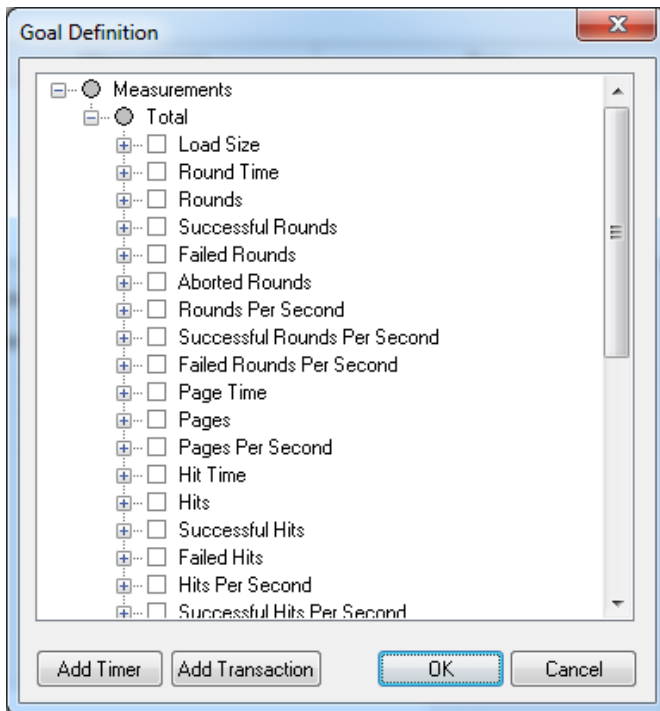


Figure 84: Goal Definition Dialog Box

All of the WebLOAD Console built-in measurements are displayed in a tree structure.

2. Optionally add a previously-defined custom measurement of type Timer to the tree, as follows:
 - a. Click **Add Timer**. A new node appears in the Goal Definition dialog box, with an empty name field.
 - b. Enter the timer's name in the name field.
3. Optionally add to the tree, measurements based on a previously-defined transaction (which you created when editing the script in the WebLOAD Recorder), as follows:
 - a. Click **Add Transaction**.
 - b. In the Add Transaction dialog box that appears, enter the transaction name.

Seven measurements, based on the specified transaction, are automatically added to the tree. For example, if you add the previously-defined "Login" transaction, the following seven measurements are added.

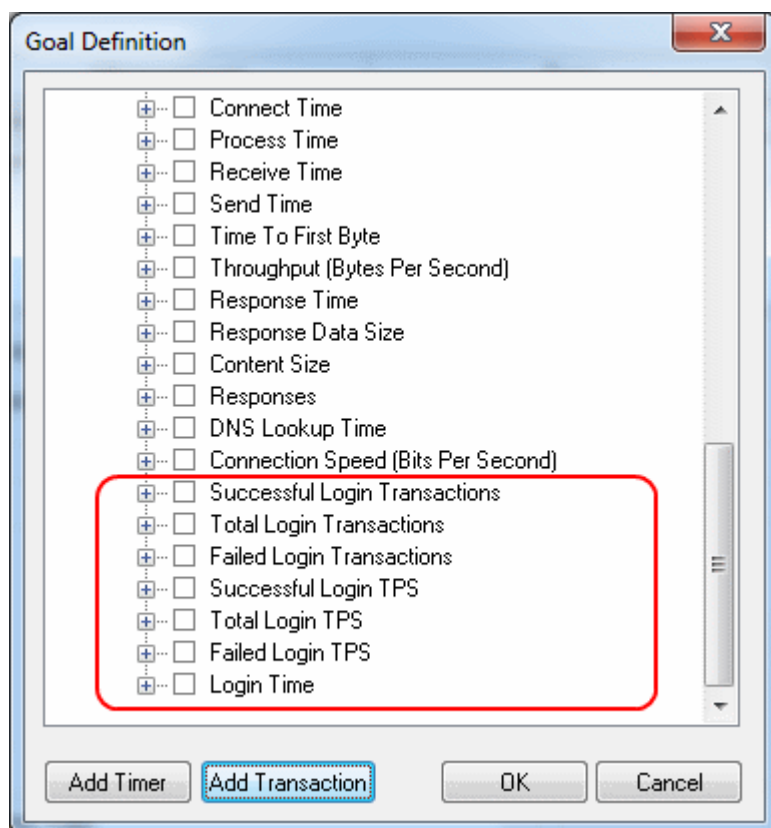


Figure 85: Goal Definition Dialog Box – Result of Add Transaction Action

4. To create rule conditions, select the checkbox adjacent to the measurements to include in the goal definition.

For example, to define a test goal for the response time not to exceed 5 seconds, select the checkbox adjacent to Response Time.



Note: A gray circle next to a measurement indicates that no default subcomponents are defined (the upper level component cannot be selected). To select subcomponents, click the + to expand the tree.

5. Click **OK**.

WebLOAD Console closes the list and opens a row for each of the selected measurements in the Goal Definition dialog box.

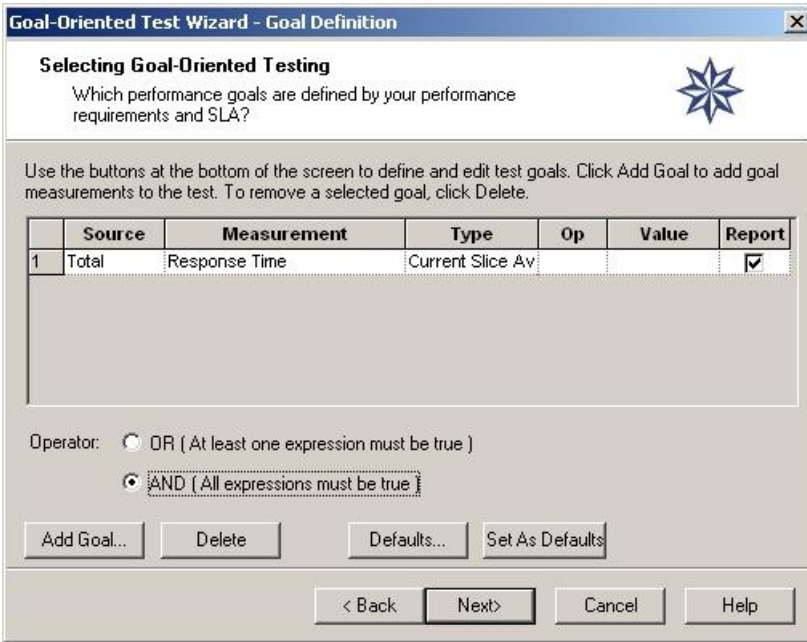


Figure 86: Selecting Goal-Oriented Test Performance Goals Dialog Box

6. Click in the **Op** field and select an operator from the drop-down list.
7. Click in the **Value** field and type a boundary value for the goal.
8. Ensure that the Report field is checked to include the goal measurement in the Goal-Oriented Test default Integrated Report.
9. Click **Next**.

WebLOAD Console displays the Acceleration dialog box. For additional information, see *Defining the Acceleration Rate* (on page 158).

Configuring a Goal - An Example

Suppose your goal is for your Web application server Response Time not to exceed five seconds.

To configure the goal:

1. Click **Add Goal**.
The Goal Definition window is displayed.
2. Select **Response Time** from the list of measurements and click OK.
This measurement is displayed in the table on the Goal Definition dialog box.
3. Because the goal is for the Response Time not to exceed five seconds, select the operator **>=** and the Goal Value **5** by modifying the Op and Goal Value cells.

When you run the test, WebLOAD Console increases the load on the SUT until the Response Time for each simulated user reaches or exceeds five seconds and then it stops the test.

Deleting a Goal

To delete a goal:

1. Select the goal to be deleted on the Goal Definition screen by clicking the number to the left of the goal.
2. Click **Delete**.

The goal is removed from the Goal Definition dialog box.

Setting and Using Defaults

The Goal-Oriented Test Wizard enables you to save goals that you have defined, and reuse them in subsequent tests. To do this, set your defined goals as the default, and then use the defaults in later tests.

Setting Defaults

The Set As Default button sets all of the Goals located on the Goal Definition dialog box as defaults. This operation overrides any previous default configurations.

To set defaults:

1. Configure the test goals that you want to save as the default.
2. Click **Set As Defaults**.

Using Defaults

To use defaults:

1. On the Goal Definition dialog box, click **Default**.

The Goal Defaults dialog box appears.

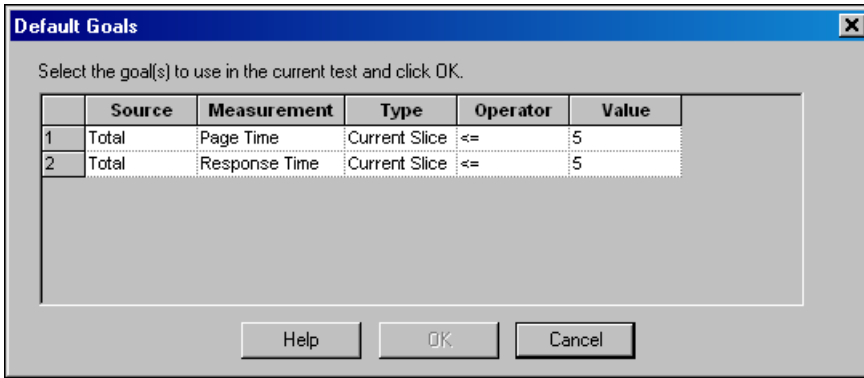


Figure 87: Goal Defaults Dialog Box

2. Select the goal to be included in the Goal-Oriented Test test session by selecting the number to the left of the goal (use the Ctrl or Shift keys to select more than one goal).
3. Click **OK**.

The goal(s) are added to the test definition.

Defining the Acceleration Rate

The Goal-Oriented Test increases the number of Virtual Clients accessing the application until the performance goals are met. The acceleration rate is the rate at which WebLOAD Console increases the number of Virtual Clients accessing the SUT at each acceleration step. Use a large number of clients for a fast acceleration to reach your goals quickly. Set WebLOAD Console to increment the number of Virtual Clients more slowly for fine-tuned results.

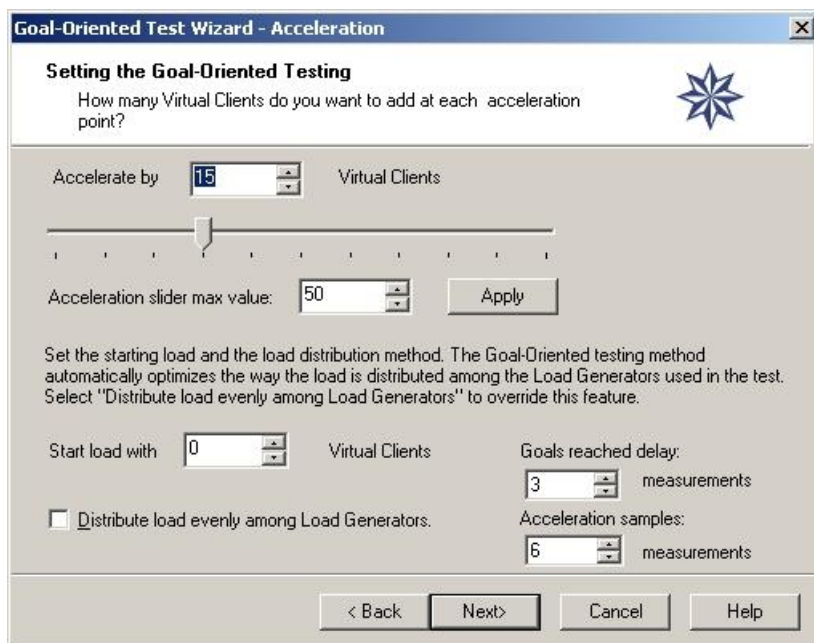


Figure 88: Acceleration Dialog Box

To define the acceleration rate:

1. Click the slider and move up and down the scale to select an acceleration rate. RadView recommends:
 - Accelerating at a rate of 15 Virtual Clients to reach your goals quickly.
 - Accelerating at a rate of 5 Virtual Clients for fine-tuned results.
2. Type or select the initial load to begin the Goal-Oriented Test Session in the Start load with Virtual Clients box.
3. The Goal-Oriented Test automatically optimizes the load distribution of Virtual Clients among the Load Generators participating in the test. To override this feature and distribute the load evenly between the Load Generators select the Distribute load evenly amongst Load Generators checkbox.
4. Click **Next**.

The wizard progresses to the Goals Achieved dialog box.



Note: The number of users that WebLOAD Console can generate on any given Load Machine in an efficient manner (so that the addition of users will not reduce the performance on that machine) is dependent on the power of the host machine. When WebLOAD Console estimates that it has reached the maximum output on a specific host, a log message is sent. WebLOAD Console continues to accelerate the load until the end of the Goal-Oriented test session.

Defining WebLOAD Console's Action Once the Goals are Reached

After configuring your Goal-Oriented test you must define WebLOAD Console's response once the test goals are reached.

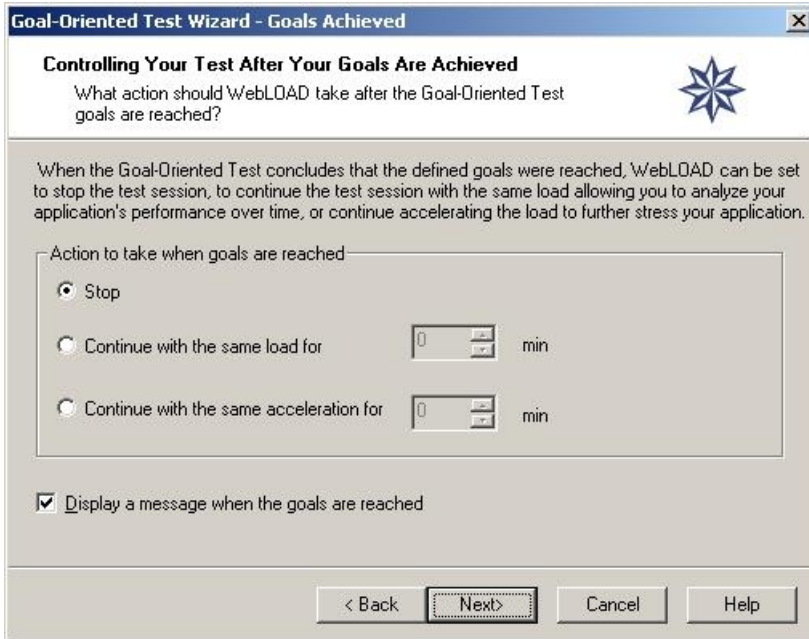


Figure 89: Goals Achieved Dialog Box

The Goal-Oriented Test can be set to stop, to continue testing under the same load, or to continue accelerating the number of Virtual Clients.

To set WebLOAD Console to stop:

1. Select **Stop**.
2. Click **Next**.

The wizard progresses to the Finish dialog box.

To set WebLOAD Console to continue running with the same load:

1. Select **Continue with the same load**.
2. Type or select the amount of time (in minutes) that WebLOAD Console should continue running with the same load after the test goals are reached.
3. Click **Next**.

The wizard progresses to the Finish dialog box.

To set WebLOAD Console to continue accelerating the number of Virtual Clients:

1. Select **Continue with the same acceleration**.
2. Type or select the amount of time (in minutes) that WebLOAD Console should continue accelerating the load after the test goals are reached.
3. Click **Next**.

The wizard progresses to the Finish dialog box.

Finishing the Goal-Oriented Test Wizard

The product of your testing is the WebLOAD report, where you can see how your Web application handles the load that you tested. The Goal-Oriented Test Wizard automatically generates a predefined report that measures:

- Load Size (the number of Virtual Clients being generated)
- The WebLOAD Console measurements for which goals are defined

WebLOAD Console saves all of the values for the complete set of measurements. You can decide after the test has run to display the results never before requested.



Figure 90: Goal-Oriented Test Wizard – Finish Dialog Box

To finish the Goal-Oriented Test Wizard:

1. From the Finish dialog box, click **Finish**.

A message box appears asking if you want to save the current Load Template.

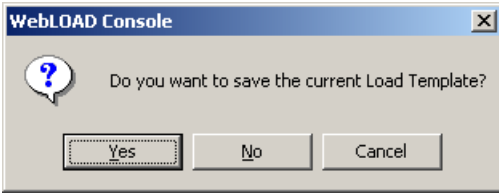


Figure 91: Save Current Load Template Message Box

2. Click **Yes** to save the current Load Template as a *.tpl file.

Load Templates contain the parameters that you defined for the test, such as the participating hosts in a test and the test plan. Using Load Templates saves you the time of reconfiguring the Load Session when repeating tests.

The Goal-Oriented Test Wizard is finished and the test begins. WebLOAD Console displays the Goal-Oriented Test Default Report.

Viewing the Goal-Oriented Test Default Report

When running a Goal-Oriented test session, the WebLOAD Goal-Oriented Test Default Report opens automatically in Chart View.

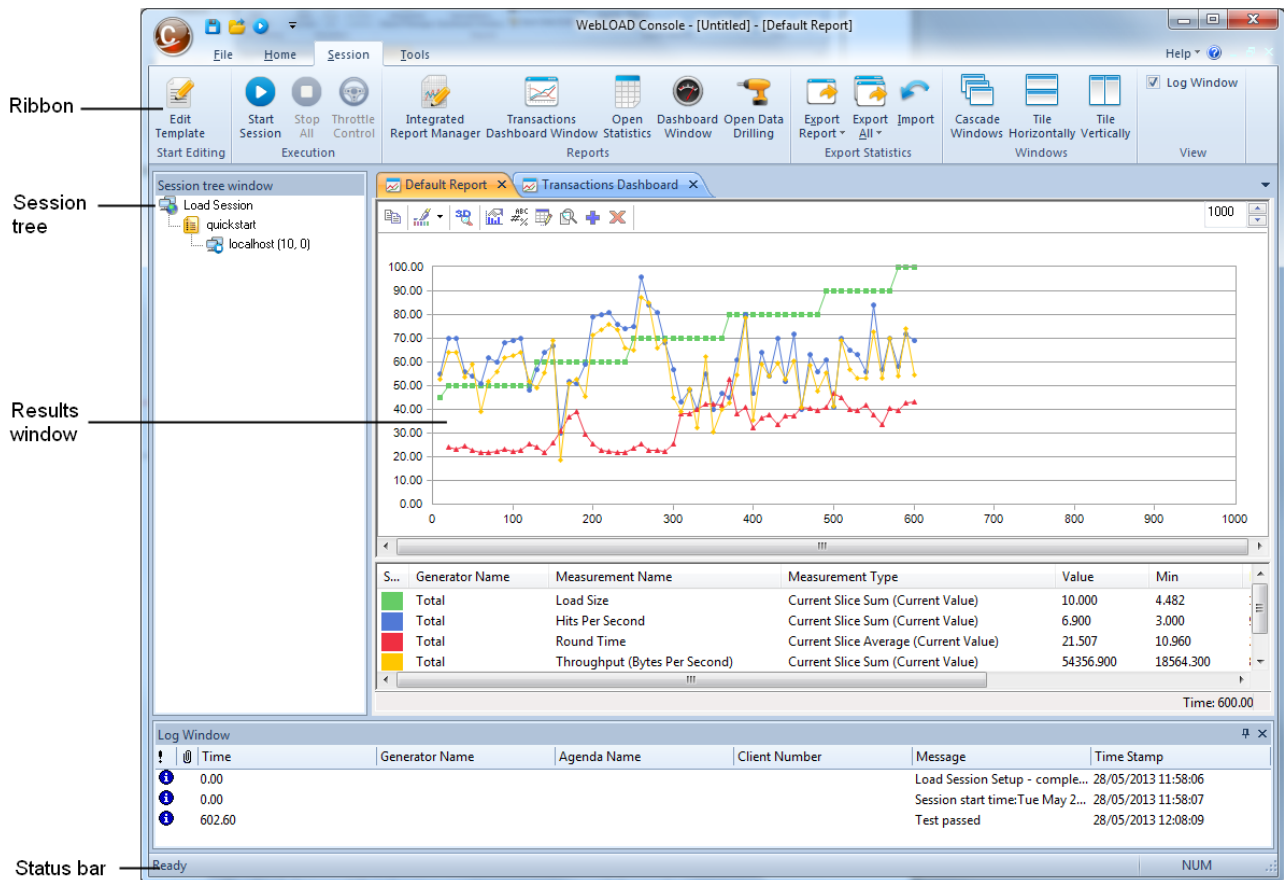


Figure 92: Default Report

Real-time results are displayed in graphs on the right side of the Console screen.

The following elements are displayed on the screen:

Table 29: Goal-Oriented Test Default Report Screen

Elements	Description
Ribbon	Contains options for configuring and controlling the Goal-Oriented test session.
Session Tree	Presents a graphical display of your Goal-Oriented test session.
Results Window	All of the reports opened during the Goal-Oriented test session will appear in this window. Use the tabs located at the top and bottom of the window to view different reports.
Status Bar	Indicates the program status.

The Goal-Oriented Test Session continues running until the performance goals are achieved. When the performance goals are reached, WebLOAD Console displays a message box with the test results.

The following is an example of a Test Results message box:

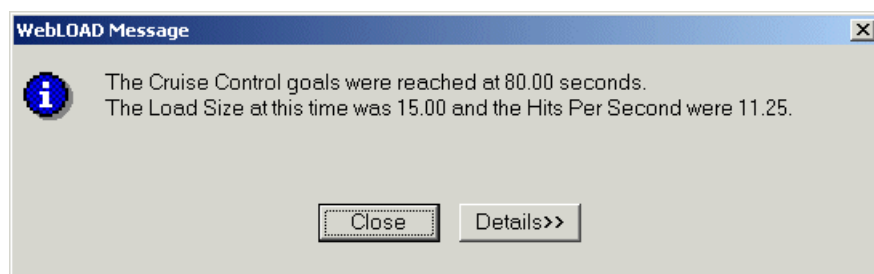


Figure 93: Test Results Message Box

Click **Details** to examine the original goal settings and statistics for them. A complete report displayed as follows:

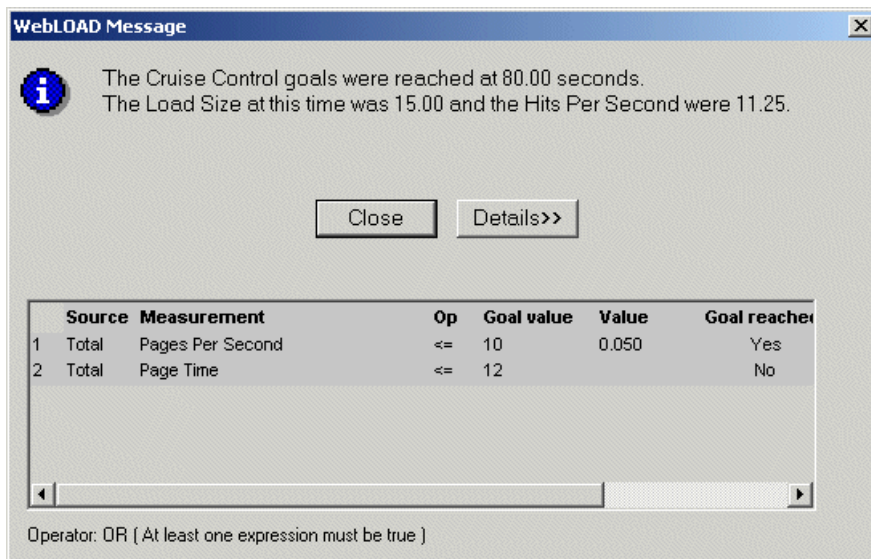


Figure 94: Details Report Window



Note: You can view the Details Report window while the test is running by right-clicking the **Load Session** icon in the Session Tree.

Configuring a Load Template Manually

This chapter describes the process of manually configuring a Load Template. Load Templates are configured in the Console through a sequential series of dialog boxes. Through these dialog boxes, you can select and schedule the script or Mix files to run during the test, define the number of Load Machines and Probing Clients to run each script and the number of Virtual Clients to run.



Note: You cannot create Load Templates if you have not yet recorded (or otherwise created) any scripts. If you do not have any scripts to work with, you must create them with WebLOAD Recorder.

The Manual Test Configuration Workflow

The following diagram illustrates the workflow for manually configuring a load test.

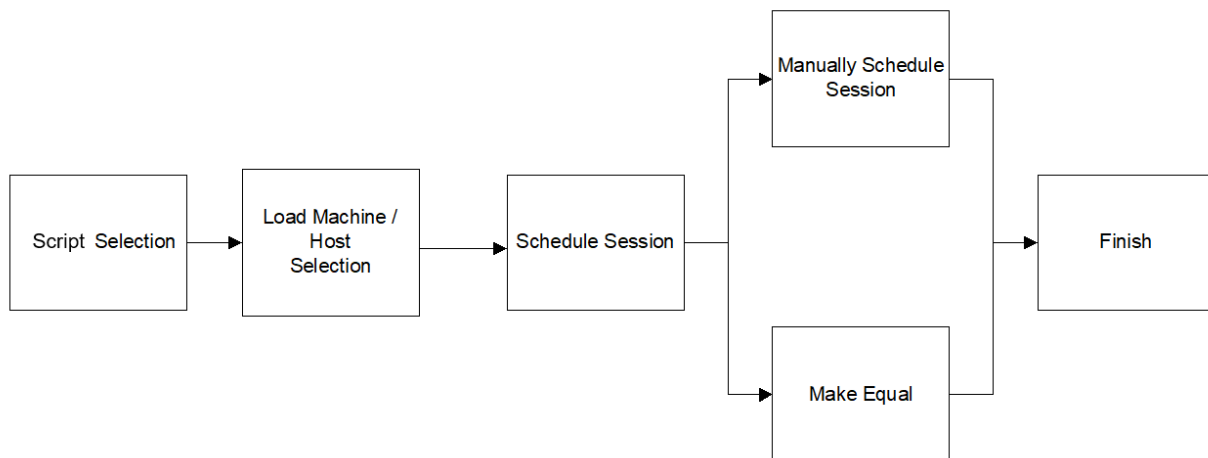


Figure 95: Load Test Manual Configuration Workflow Diagram

Starting a New Template Manually

To create a new Load Template manually:

1. Select **Create a new template manually** on the WebLOAD Startup dialog box.

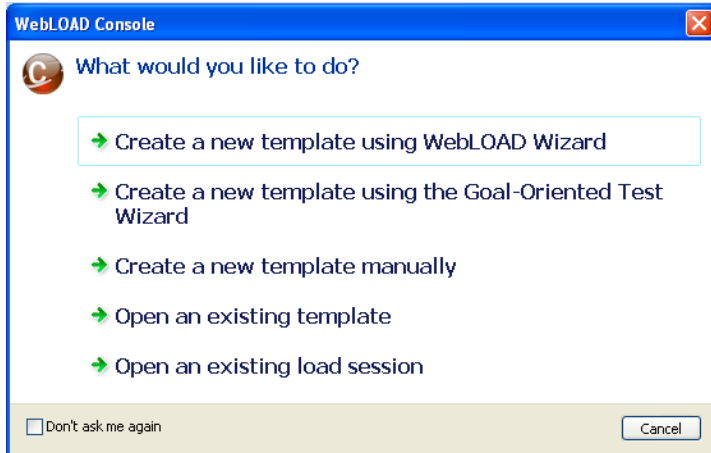


Figure 96: WebLOAD Console Startup Dialog Box

2. Click **OK**.

The WebLOAD Console appears.

3. Click **Scripts Selection** in the **Home** tab of the ribbon,

-Or-

Right-click the Load Session item in the Script Tree and select **Modify Script Selection**.

The script/Mix Selection dialog box opens. See *Selecting a* (on page 166).

Selecting a Script

Scripts are test scripts that describe user activity and are run during WebLOAD Console test sessions.

WebLOAD Console supports two types of scripts:

- **Single Scripts** – which are lone test-scripts.
- **Mix of Scripts** – which are a combination of existing single scripts. Using a Mix you can simulate different groups of users performing different activities on the system under test (SUT) at the same time.

Select the scripts to run through the script / Mix Selection dialog box.

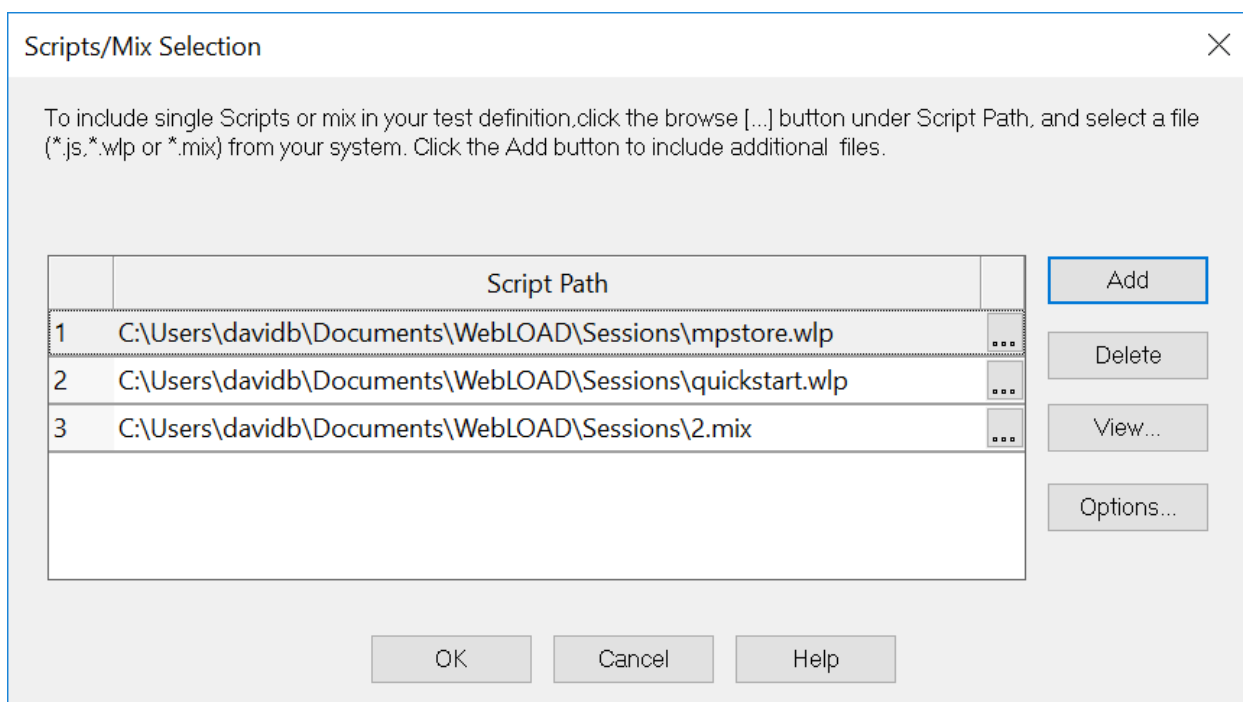


Figure 97: Script/Mix Selection Dialog Box

Creating Load Templates with One or More Single Scripts

To create Load Templates with one or more singlescripts:

1. Click in the **Script Path** field and enter the name and the full path for the script or mix of scripts.

The path name can be entered manually or through browsing the directory structure.

To include additional scripts/mix of scripts, click **Add** and repeat step 1.

2. To optionally open and view or edit a selected script, click **View**.

WebLOAD Recorder opens displaying the selected script.

3. To optionally open and view or edit a selected mix of scripts, click **View**.

The Edit Mix dialog box opens (it is essentially identical to the *Mix Creation Dialog Box*).

4. To optionally configure runtime options exclusive to a selected Script, click **Options**.

The Script Options dialog box opens enabling you to define the runtime options for the selected script. For more information on Script options, see *Setting Script Options* (on page 211).

5. To optionally configure runtime options exclusive to a script in a selected mix, click **View**. In the Edit Mix dialog box that appears, select the script and click **Options**. The Script Options dialog box opens enabling you to define the runtime options for the selected script. For more information on script options, see *Setting Script Options* (on page 211).
6. Click **OK**.

The scripts selected to run in the test are displayed in the Session Tree.

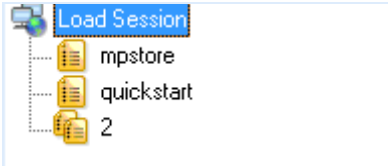


Figure 98: Session Tree

Selecting Hosts

Hosts are the computer systems participating in the test session. Each host has an assigned task, acting as either a Load Machine or a Probing Client. Both Load Machines and Probing Clients are configured in the same Host Selection window. Hosts must be defined for each script or Mix file running in the test session.

Opening the Host Selection Dialog Box

To open the host selection dialog box:

- Right-click the script or Mix icon in the Session Tree and select **View Load Machines Selection**,

-Or-

Select **Load Machine Selection** in the **Home** tab of the ribbon.

The Host Selection dialog box opens.

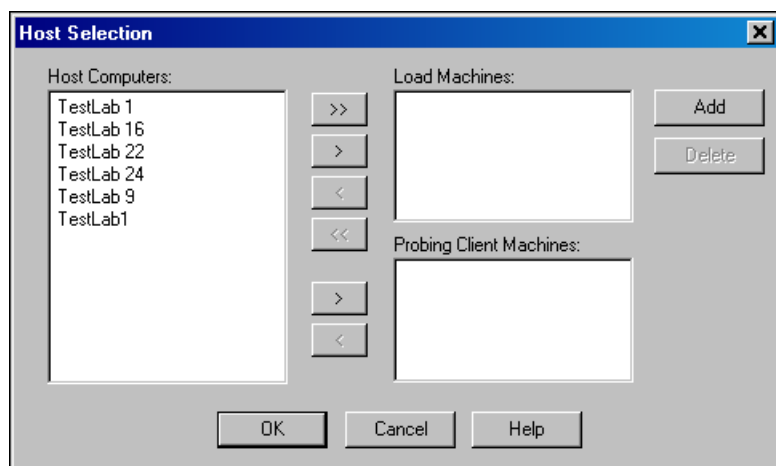


Figure 99: Host Selection Dialog Box

In this dialog box, you define the host computers participating in the test. There are three fields in this dialog box. The Host Computers field, lists all of the host computers available for testing. The two fields to the right contain the systems defined as Load Machines and Probing Client Machines. Use the arrow buttons between the fields to change the roles of the host computers, or double-click systems in the Host Computers field to move them to the Load Machines field.



Note: If you are running a script with Perfecto Mobile script, define a single Load Machine or a single Probing Client in order to simulate a single user.

Defining Host Computers

To define a host computer:

1. Click **Add**.

The Add Host Computer dialog box opens.

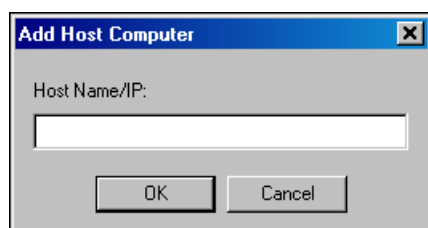


Figure 100: Add Host Computer Dialog Box

2. Enter the name or IP address of the host computer you want to add.
3. Click **OK**.


WebLOAD Console adds the machine to the Host Computer list.

Defining Load Machines

Load Machines are the machines that generate the Virtual Clients and bombard the system under test.


Designating a Host Computer as a Load Machine

To designate a host computer as a Load Machine:

1. Select the Host from the Host Computer field.
2. Click the single arrow key  adjacent to the Load Machine field.
-Or-
Double-click the host.
The selected Host moves to the Load Machine field.
3. Click **OK**.

Designating all Host Computers as Load Machines

To designate all host computers as Load Machines:


1. Click the double arrow key  adjacent to the Load Machine field.
All of the Host Computers move to the Load Machine field.
2. Click **OK**.

Defining Probing Client Machines

Probing Clients act as single Virtual Clients to measure the performance of targeted activities and provide individual performance statistics of the SUT.

Designating a Host Computer as a Probing Client Machine

To designate a host computer as a Probing Client Machine:

1. Select the **Host** from the Host Computer field.
2. Click the single arrow key  adjacent to the Probing Client Machines field.
The selected Host moves to the Probing Client Machine field.
3. Click **OK**.

Deleting a Host Computer

To delete a host computer:

1. Select the host name from the Host Computer list.
2. Click **Delete**.

The host is deleted from the Host Computer list.

Scheduling the Test Session

The Schedule dialog box is used to specify the test parameters. Here you specify the size of the load to be generated on each Load Machine throughout the test.

You can create a load schedule by:

- Manually defining time frames and the load to generate over each frame
- Using the Load Profiler
- A combination of both

Creating a Load Schedule Manually

To create a load schedule manually:

1. Double-click a host in the Session Tree,

-Or-

Right-click the script the Load Machines are running and select **Modify Load Machine Schedule**.

The Schedule Manually dialog box opens.

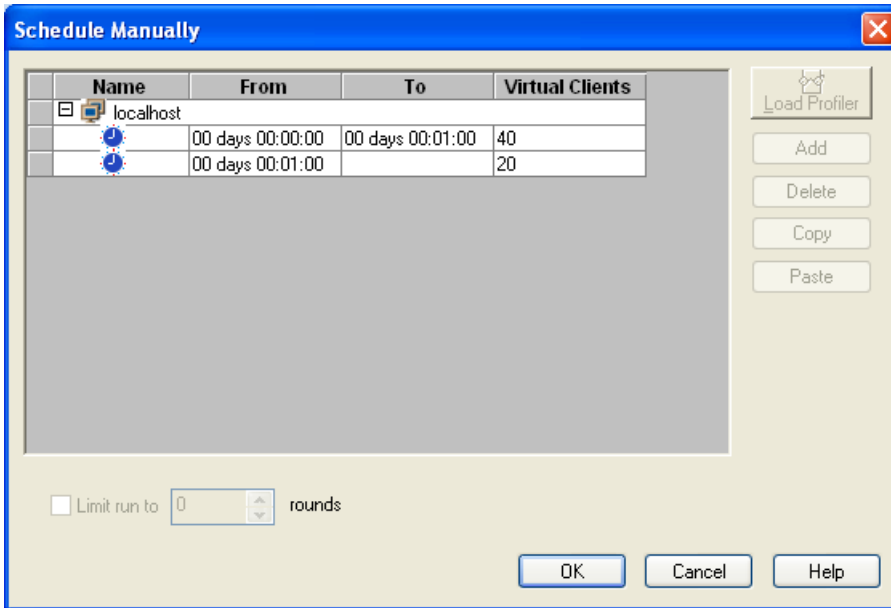


Figure 101: Schedule Manually Dialog Box

2. Expand the Load Machine tree using the + / - button adjacent to the Load Machine you want to schedule.
3. Enter the number of days and the starting time to generate the load in the From field.

The starting time is defined for each particular script relative to the beginning of the test.



Note: Days are numbered 00 through to 99. Seconds are numbered 1 through 59. To enter one minute, use the 1:00 syntax, not 00:60.

4. Enter the number of days and the ending time to generate the load in the To field.

The ending time is defined for each particular Script relative to the beginning of the test. If the To field is left blank, WebLOAD Console continues generating the load until the From time in the next entry for this Load Machine, or if this is the last entry, until the test is stopped manually.
5. Enter the number of Virtual Clients to generate during the time frame defined in steps 3 and 4.
6. Schedule additional Virtual Client loads for this Load Machine:
 - a. Click **Add** to add a blank line to the schedule grid.
 - b. Repeat steps 3 through 5.
7. Repeat steps 3 through 6 for each Load Machine or Probing Client defined in the Session Tree.
8. Click **OK**.



Note: The **Limit run to X rounds** option is disabled because you cannot set or change it for an individual load machine. This option can only be set at the script level.

Creating a Load Schedule with the Load Profiler

To create a load schedule with the load profile:

1. Double-click a host in the Session Tree,

-Or-

Right-click the script the hosts are running and select **Modify Load Machine Schedule**.

The Schedule Manually dialog box opens.

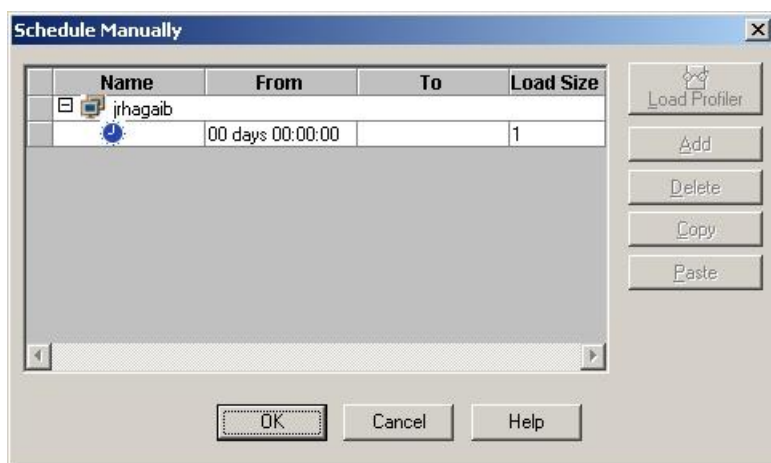


Figure 102: Schedule Manually Dialog Box

2. Click the Load Machine you want to schedule.
3. Click the **Load Profiler** button.

The Load Profiler opens.

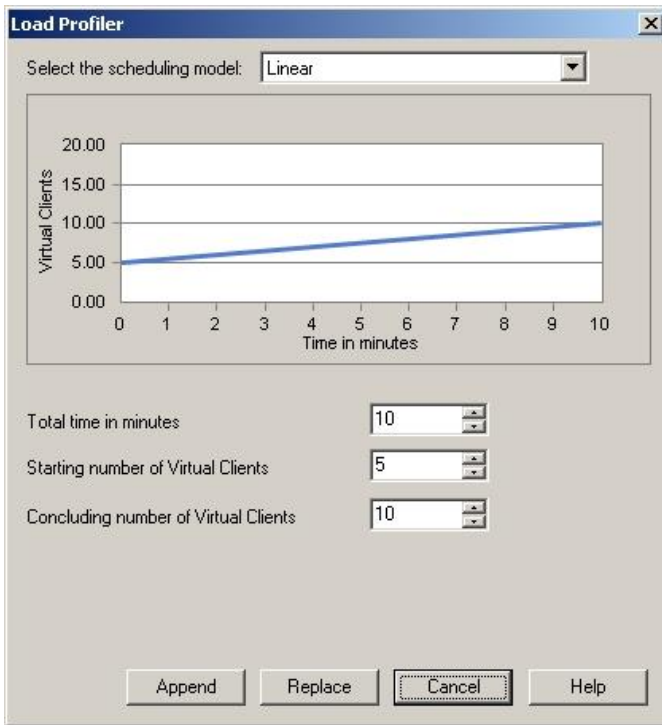


Figure 103: Load Profiler

4. Select a pattern from the Select scheduling model field at the top of the dialog box.
For an explanation of each model, see *Scheduling Models* (on page 121).
5. Fill in the fields at the bottom of the dialog box.
6. Click **Append** to include the load definition in the schedule,
-Or-
Click **Replace** to substitute the load definition for the one appearing in the Schedule Manually dialog box.

Completing Your Manual Load Template Configuration

After configuring the scripts to run, the Load Machines to generate the load, and the test schedule, you are ready to run your Load Session.

To complete your manual Load Template configuration:

1. Click **Start Session** in the **Session** tab of the ribbon.

A message box appears asking if you want to save the current Load Template.

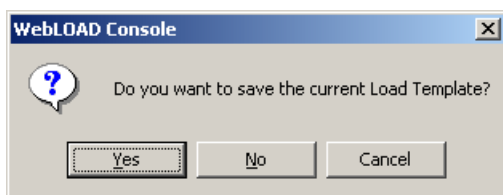


Figure 104: Save Current Load Template Message Box

2. Click **Yes** to save the current Load Template as a `*.tpl` file.

Load Templates contain the parameters that you defined for the test, such as the scripts to run, participating hosts, and the test plan. Saving Load Templates saves you the time of reconfiguring the Load Session when repeating tests.

For more information on running tests, see *Running a Load Session* (on page 271).

Setting Global Options

Global Options enable you to modify WebLOAD Console defaults to your specifications. All WebLOAD Console tests use these defaults.

How to Set Global Options

This section guides you through the process of setting WebLOAD Console global options and introduces you to some of WebLOAD Console's more advanced features. Global options can be modified at any time from the WebLOAD Console.

To set global options:

1. Click **Global Options** in the **Tools** tab of the ribbon,
-Or-
Select **Global Options** from the Console System button.

The Global Options dialog box opens.

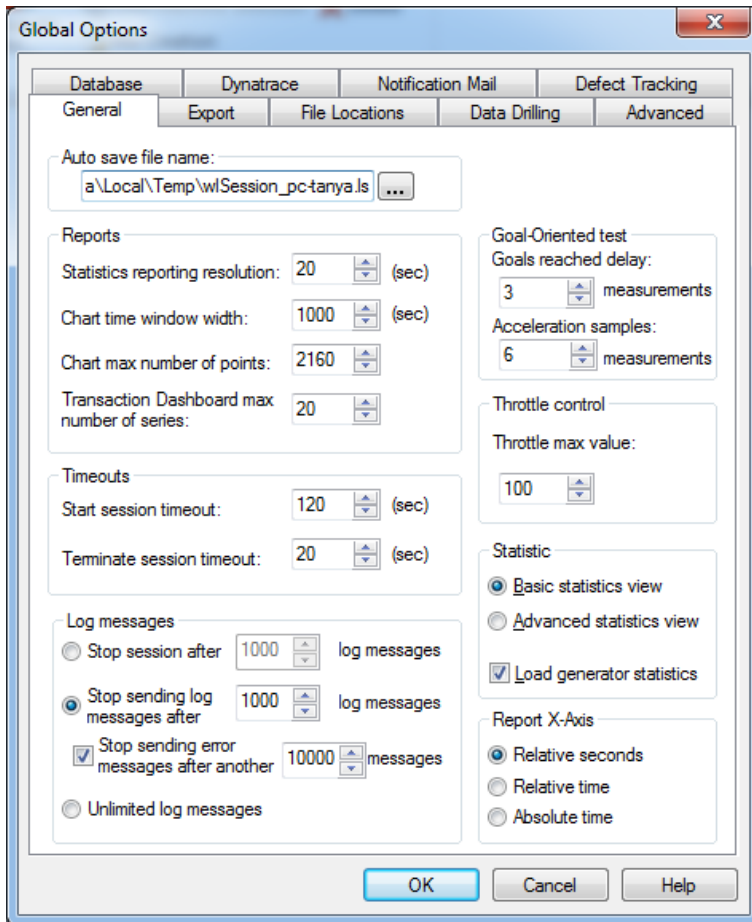


Figure 105: Global Options Dialog Box

The Global Options dialog box contains the following tabs:

- General
 - Export
 - File Locations
 - Data Drilling
 - Advanced
 - Database
 - Notification Mail
 - Dynatrace
 - Defect Tracking
2. Select the desired tab, and modify as many of the options as you need.
 3. Select another tab and modify its options, if necessary.
 4. Click **OK** to accept the modifications, or **Cancel** to reject them, and exit the Global Options dialog box.

The General Tab

The General tab lists a number of options that you can modify to facilitate your work in test sessions.

To access the General tab options:

1. Click **Global Options** in the **Tools** tab of the ribbon,
-Or-
Select **Global Options** from the Console System button.
2. Select the **General** (default) tab.

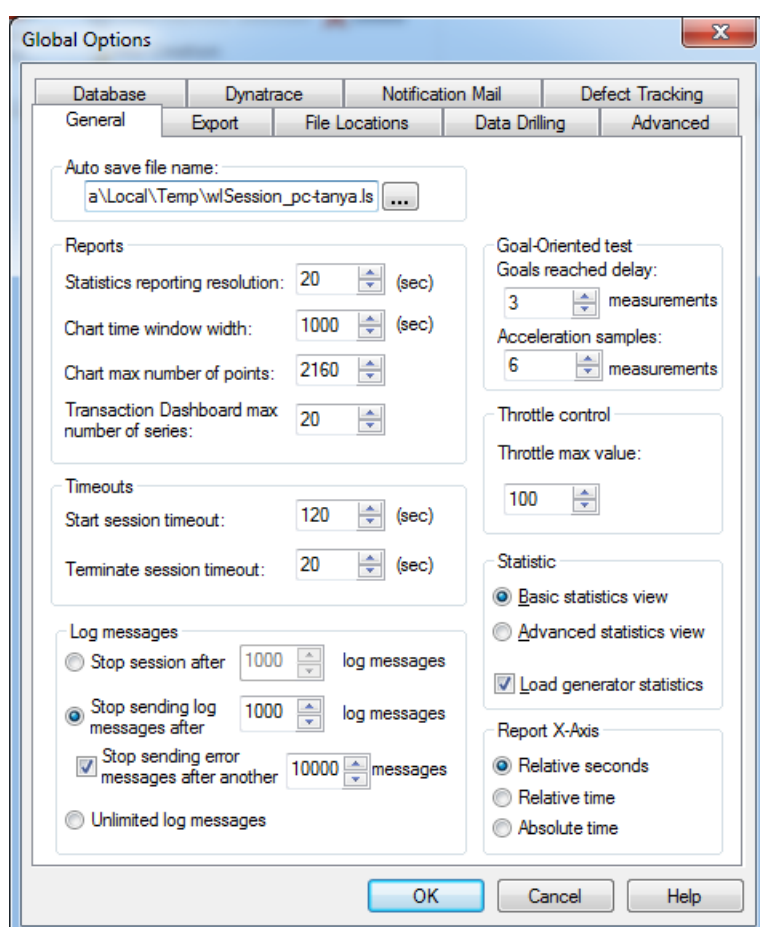


Figure 106: Global Options General Tab

Setting the Autosave File Name Default Location

WebLOAD Console performs fast saves while running a test. This option enables you to specify the filename and location of the autosave (backup) file. It is recommended to save files to a central location where they are available to all users.

To set the Autosave file name default location:

1. Click **Global Options** in the **Tools** tab of the ribbon,
-Or-
Select **Global Options** from the Console System button.
2. Select the **General** (default) tab.
3. Type or select the desired file name and location in the Auto Save File Name field.
4. Click **OK**.

Setting the Reports Options

The Reports options enable you to define the:

- Interval between each reported sample (referred to as the Statistics reporting resolution).
- Default time range to be displayed as the x-axis of the Integrated Report chart (out of the total time span of the Load Session).
- The maximum number of points in a chart. This is intended for improving charts performance during execution. By default, the maximum number is limited, but you can change it to unlimited. Note that even when the number is limited, all statistics are saved in the WebLOAD statistics data base; following execution, you can change the maximum number of points to unlimited, and re-open the chart to display all points.

To set the Reports options:

1. Click **Global Options** in the **Tools** tab of the ribbon,
-Or-
Select **Global Options** from the Console System button.
2. Select the **General** (default) tab.
3. In the **Statistics reporting resolution** field, type or select the desired statistics reporting resolution (in increments of 10 seconds). The minimum allowed value is one second.



Note: The default reporting interval is 20 seconds. When running long test sessions, it is a good idea to increase this interval. Reducing this interval to lower than 10 seconds might result in high consumption of your machine's resources.

4. In the **chart time window width** field, type or select the desired time width of the chart window. The default time width is 1000 seconds.

5. In the chart max number of points field, type or select the maximum number of points displayed in a chart during execution. To specify an unlimited number, enter the value 0. The default number of points is 2160.
6. In the **Transaction Dashboard max number of series** field, type or select the maximum number of series in the Transactions Dashboard report. The default number of series is 20.

Setting Goal-Oriented Test Options

The Goal-Oriented Test increases the load generation until your Web application performance falls below the profile you have defined.

The performance parameters that you can define are:

- **Goals reached delay** – the number of samples required to declare the Goal-Oriented Test goals as achieved. The Goal-Oriented Test declares the defined goals reached only after receiving the desired goal results in several samples.
- **Acceleration samples** – the number of sample measurements to read before automatically accelerating the load if the Goal-Oriented Test goal has not been reached by then.

To set Goal-Oriented Test options:

1. Click **Global Options** in the **Tools** tab of the ribbon,
-Or-
Select **Global Options** from the Console System button.
2. Select the **General** (default) tab.
In the Goals reached delay field, type or select the number of samples to read before proclaiming goals reached.
3. In the Acceleration point measurements field, type or select the number of samples to read before accelerating the load.

Setting WebLOAD Console Timeout Values

To set WebLOAD Console Timeout values:

1. Click **Global Options** in the **Tools** tab of the ribbon,
-Or-
Select **Global Options** from the Console System button.
2. Select the **General** (default) tab.
3. In the Start Session Timeout field, type or select the amount of time the Console should attempt to contact a host at the start of a test session (default: 30 seconds).

4. In the Terminate Session Timeout field, type or select the amount of time the Console should attempt to contact a host at the close of a test session (default: 20 seconds).

Setting the Throttle Control Max Value

The throttle control utility enables you to dynamically modify the number of Virtual Clients participating in a test during runtime.

To set the Throttle Control max value:

1. Click **Global Options** in the **Tools** tab of the ribbon,
-Or-
Select **Global Options** from the Console System button.
2. Select the **General** (default) tab.
3. In the Throttle max value field, type or select the maximum number of Virtual Clients that can participate in the test.

Setting the Log Message Options

Log messages generated by the Console, the JavaScript compiler, and any user messages programmed in the script and detected by WebLOAD Console during runtime are displayed in the Log Window. For further information on Log Messages, see *Viewing the Log Window* (on page 283).

To set the log message options:

- In the Stop Session after __ log messages field, type or select the maximum number of log messages to read before stopping WebLOAD Console,

-Or-

In the Stop sending log messages after __ log messages field, type or select the maximum number of log messages to record. WebLOAD Console will continue the Load Session but stop recording log messages. The maximum number of log messages is 32767.

Note that log messages can be of type Info, Error, or Warning. Even if you restrict the number of total log messages, you may still wish to see Error messages. Therefore you can separately set the maximum number of error messages to record by typing or selecting a number in the Stop sending error messages after another __ messages field. When that number is reached, WebLOAD Console will continue the Load Session but stop recording error messages.

-Or-

Set WebLOAD Console to send an unlimited number of log messages.

Setting the Statistic View Settings

To set the Statistic View settings:

1. Click **Global Options** in the **Tools** tab of the ribbon,
-Or-
Select **Global Options** from the Console System button.
2. Select the **General** (default) tab.
3. Select **Basic Statistics** view to:
 - Access only the Current Value for each Current Slice measurement in the Integrated Report.
 - Display the Min, Max, Average, and Count values for in the Test Summary.

The Basic Statistics view is the default setting.

-Or-

Select **Advanced Statistics view** to display all values available for the current slice and test summary in the WebLOAD Integrated Reports. These values are: Min, Max, Average, Count, Sum, and STDev.

4. Check the **Load generator statistics** checkbox (checked by default) to display the statistics for each Load Generator separately in Integrated Reports.

The Export Tab

WebLOAD reports can be exported to other programs as Tab files (tab-delimited text files). Export options, such as the default Tab file viewer, can be set through the Export dialog box.

To access the Export tab options:

1. Click **Global Options** in the **Tools** tab of the ribbon,
-Or-
Select **Global Options** from the Console System button.
2. Select the **Export** tab.

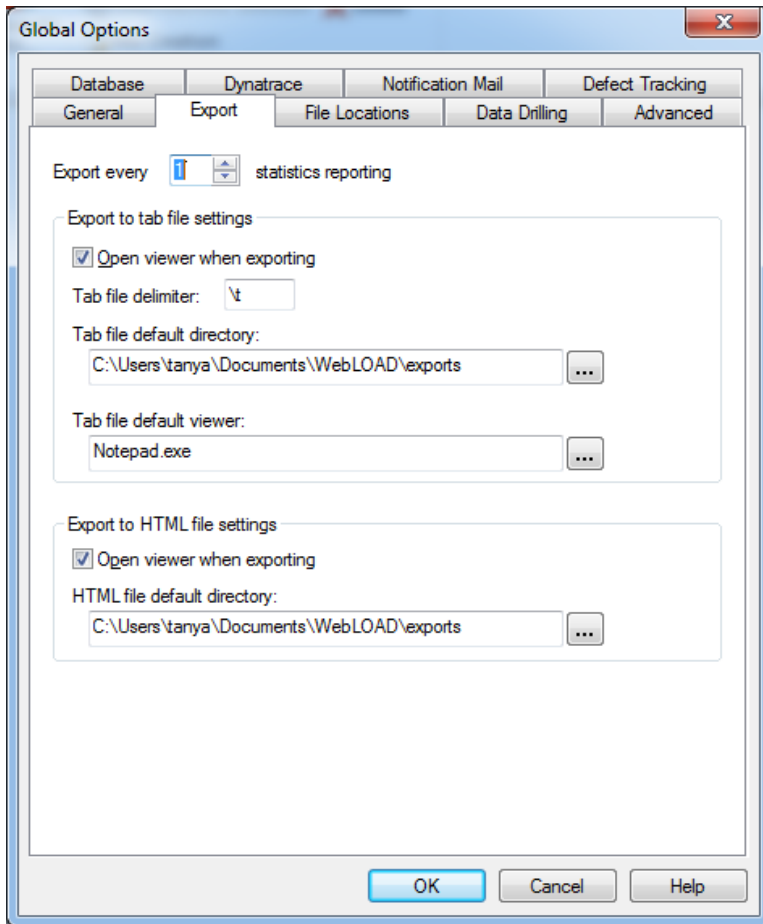


Figure 107: Global Options Export Tab

Setting the Statistics Resolution to Export

The statistics exporting resolution is the frequency of the number of samples to export when exporting WebLOAD reports. For example, if 1 is entered for this value, WebLOAD Console exports every sample. If 2 is entered for this value, WebLOAD Console exports every second sample.

To set the Statistics resolution to export:

1. Click **Global Options** in the **Tools** tab of the ribbon,
-Or-
Select **Global Options** from the Console System button.
2. Select the **Export** tab.
3. Type or select the desired exporting resolution in the Export every statistics reporting field.

When deciding the value for this option you must consider the setting selected for the sampling. If, for example, the Statistics reporting resolution (set through **Global Options > General**) is set to 20 seconds and Export every statistics reporting is set to 2, the measurements exported reflect a sampling every 40 seconds. They are the measurements gathered at the intervals 20, 60, 100, 140...

This option is useful in cases where a large amount of data results from a long test session.

Setting the Export to Tab File Settings

WebLOAD reports can be exported as Tab files (tab-delimited text files).

Open Viewer When Exporting Tab Files

When the Open viewer when exporting option is enabled (default), WebLOAD Console launches the program which serves as the default viewer (e.g., Notepad, Excel) for an exported Tab file. The default viewer is defined in the Tab File default viewer field (see below).

Tab File Export Settings

These options define the Tab file that can be exported to other programs.

To define the Tab File for export:

1. Click **Global Options** in the **Tools** tab of the ribbon,
-Or-
Select **Global Options** from the Console System button.
2. Select the **Export** tab.
3. In the Tab File delimiter field, enter the delimiter to be used in the Tab files that contain the WebLOAD Console statistical data for export.
Any string can be used as a delimiter.
 - To use a tab character as a delimiter, enter `\t`.
 - To use a new line as a delimiter, enter `\r` or `\n`.
 - To use a space as a delimiter, enter `\s`.
4. In the Tab File default directory field, set the default directory where the Tab files should be saved.

5. In the Tab file default viewer field, specify which program should serve as the default viewer (e.g., Notepad, Excel) for an exported Tab file.

Export to HTML Settings

WebLOAD reports can be exported as HTML files.

Open Viewer When Exporting HTML Files

When the Open viewer when exporting option is enabled (default), WebLOAD Console launches the program, which serves as the default viewer.

HTML Export Settings

These options define the HTML files that can be exported to other programs.

To define the HTML files for export:

1. Click **Global Options** in the **Tools** tab of the ribbon,
-Or-
Select **Global Options** from the Console System button.
2. Select the **Export** tab.
3. In the HTML File default directory field, set the default directory where the HTML files should be saved.

The File Locations Tab

This dialog box lists the default locations of various WebLOAD Console files, such as script files. You can modify these locations if you wish.

To access and configure the File Location tab options:

1. Click **Global Options** in the **Tools** tab of the ribbon,
-Or-
Select **Global Options** from the Console System button.
2. Select the **File Locations** tab.

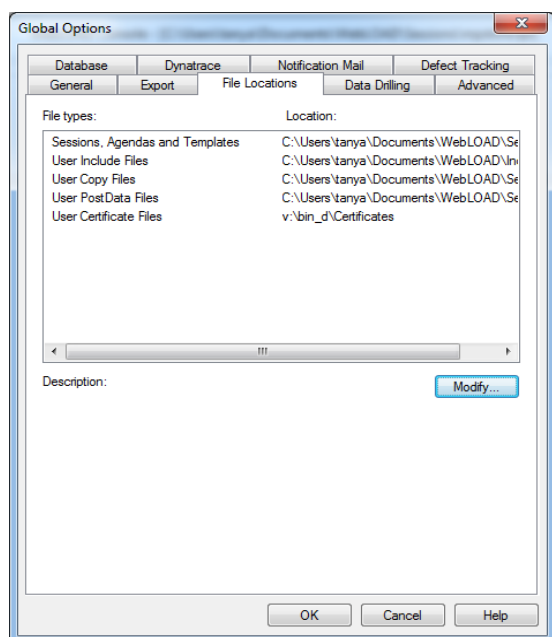



Figure 108: Global Options File Locations Tab

WebLOAD Console’s various file types are described in the following table:

Table 30: WebLOAD Console File Types

File Type	Description
Session, Scripts and Templates	Session, Script, and Template files created by WebLOAD Console. These files have the following extensions: .wls, .wlp, and .tlp.
User Include Files	External text files included in the script by using the IncludeFile function.

File Type	Description
<p>User Copy files</p>	<p>Copied files loaded into the script by using the <code>CopyFile</code> function. These files can be all file types. The most commonly used files are <code>.txt</code> files used with the <code>GetLine</code> function and <code>.dat</code> files used with the <code>wlHttp.Post</code> and <code>Get</code> commands. The <code>CopyFile</code> function searches in various system directories in addition to the location specified in the tab.</p> <p>The load engine first looks for the file to be copied in the default User Copy Files directory. If the file is not there, the file request is handed over to WebLOAD Console, which searches for the file using the following search path order:</p> <ol style="list-style-type: none"> 1. If a full path name has been hardcoded into the <code>CopyFile</code> command, the system searches the specified location. If the file is not found in an explicitly coded directory, the system returns an error code of File Not Found and will not search in any other locations. <p> Note: It is not recommended to hardcode a full path name, since the Sscript will then not be portable between different systems. This is especially important for networks that use both UNIX and Windows systems.</p> <ol style="list-style-type: none"> 2. Assuming no hardcoded full path name in the script code, the system looks for the file in the current working directory, the directory from which WebLOAD Console was originally executed. 3. Finally, if the file is still not found, the system searches for the file sequentially through all the directories listed in the File Locations tab.
<p>User PostData Files</p>	<p>Data files created when the PostData recording options specify to record the data in a data file. The system first searches for the data files in the directory in which the script is located, then, if not found, in the current working directory, and lastly, in the folder defined in the default User PostData Files directory specified in this dialog box.</p>

File Type	Description
User Certificate Files	Certificate files include files used to emulate a server certificate for the user client application and list the trusted certificate authorities. By default this is the certificate supplied with the WebLOAD Console installation. The file extension is .pem.

3. Select the item whose default location you want to change, and click **Modify**.
The Modify Location dialog box opens.
4. Select a new default location and click **OK**.
The dialog box closes and the new location is listed.

The Data Drilling Tab

Data Drilling enables you to display a detailed description of all user-defined and named transactions to the instance level. An instance is a single occurrence of a user-defined transaction in a given round. Data Drilling is only enabled if the Collect instances checkbox is selected.

To access and configure the Data Drilling tab options:

1. Click **Global Options** in the **Tools** tab of the ribbon,
-Or-
Select **Global Options** from the Console System button.
2. Select the **Data Drilling** tab.

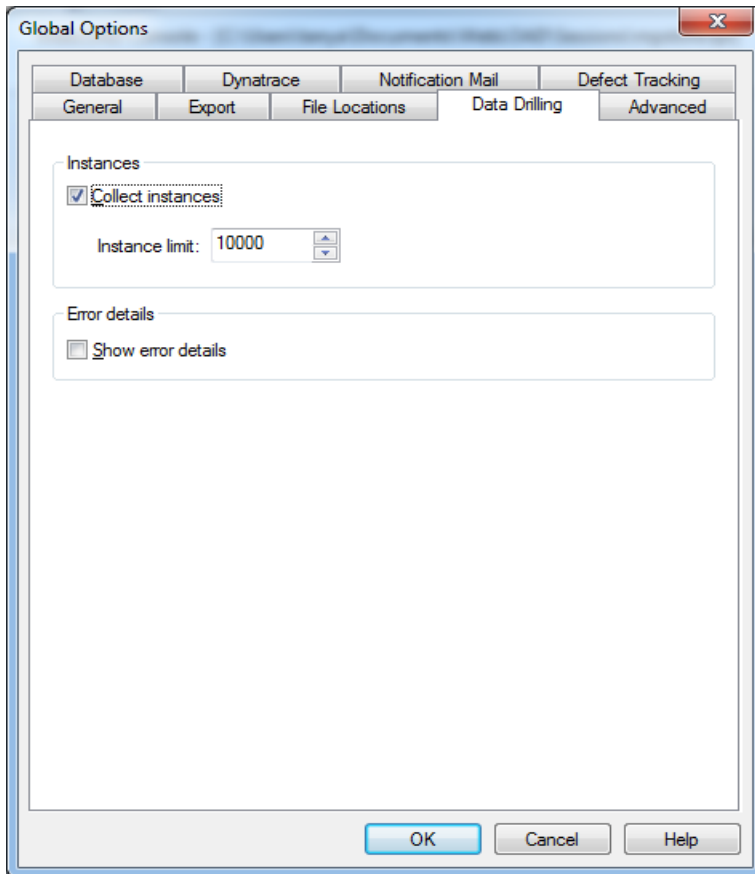


Figure 109: Global Options Data Drilling Tab

3. Set the Data Drilling options:
 - a. Click the **Collect instances** checkbox (default) to enable Data Drilling.
 - b. Specify the limit of collected instances in the Instance limit field. The default limit is 1000.



Note: If you disable the **Collect instances** checkbox, the Data Drilling options of the Functional Testing tab are also disabled.

4. Click the **Show error details** checkbox to enable viewing full details of HTTP errors. For more information, see *Full View of HTTP Errors* (on page 191).

Full View of HTTP Errors

This feature enables viewing the full details of HTTP errors that occurred. The feature is enabled by selecting the **Show error details** checkbox in the **Data Drilling** tab of the **Global Options** window (Figure 109).

When the feature is enabled, you can view full details of HTTP errors as follows.

1. In the Log Window, double click an error message.

The error is displayed in page view.

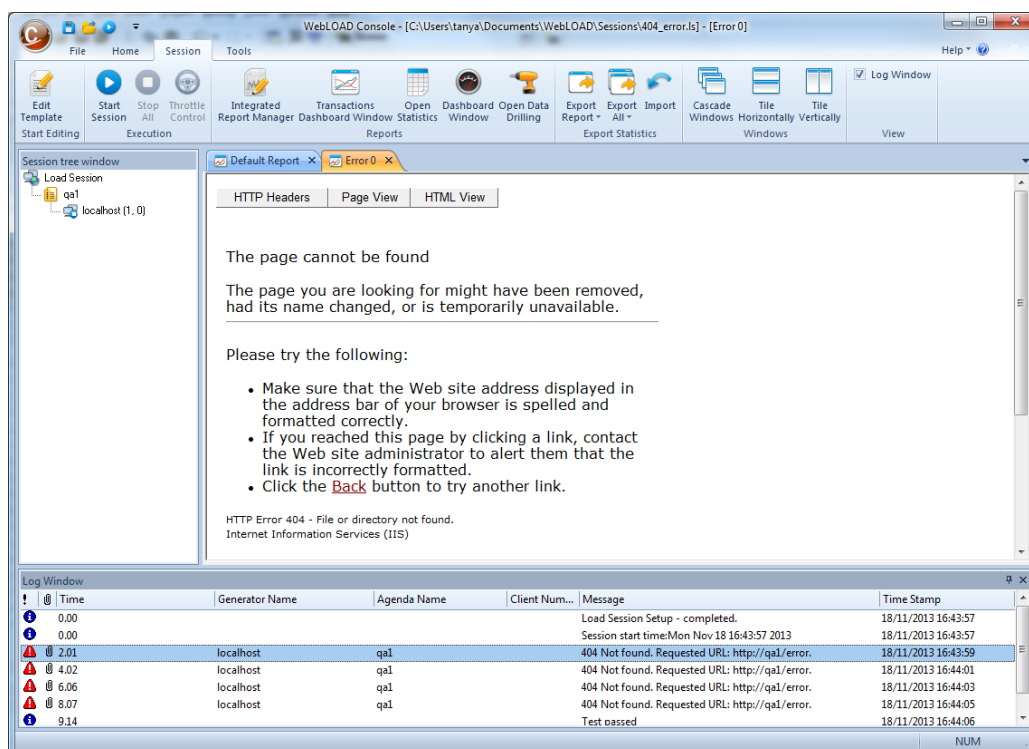


Figure 110: Full View of Error – Page View

2. To view the HTTP Headers (Request and Response) relating to the error, click **HTTP Headers**.

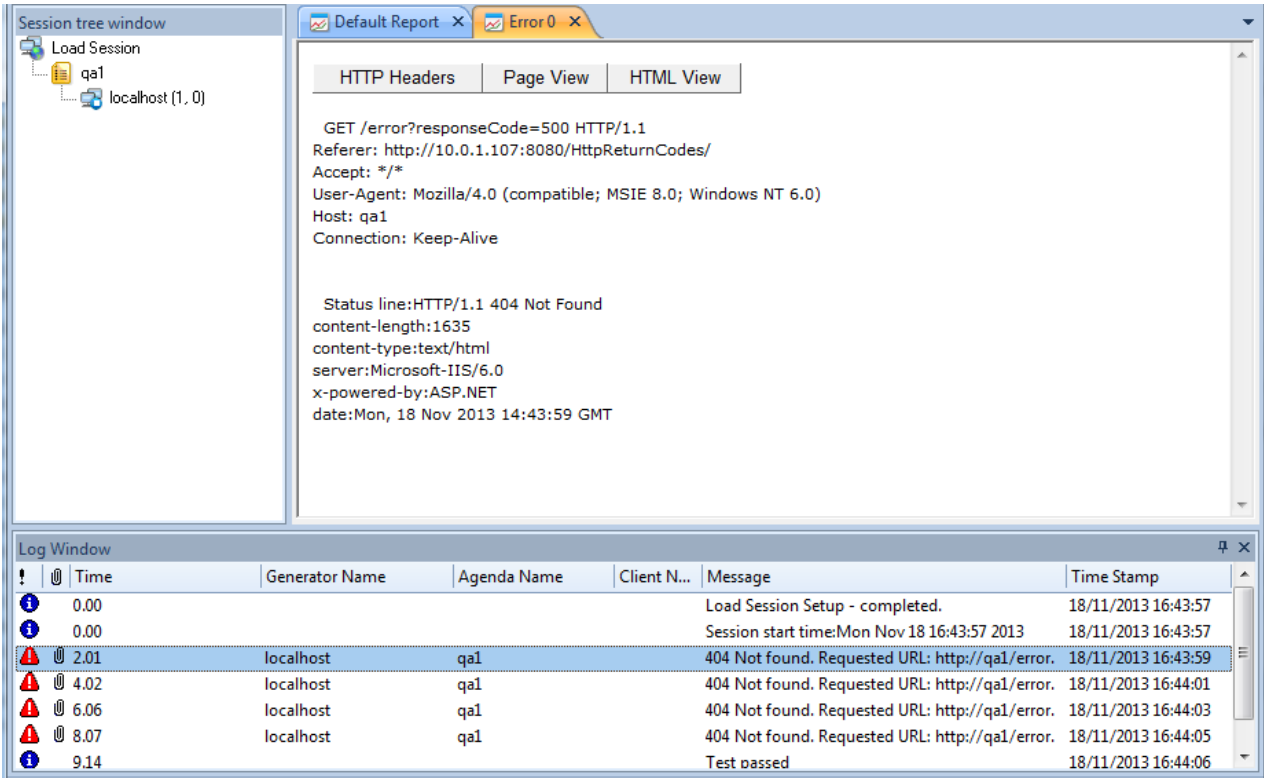


Figure 111: Full View of Error – HTTP Headers View

- To view the HTML source of the error message, click **HTML View**.

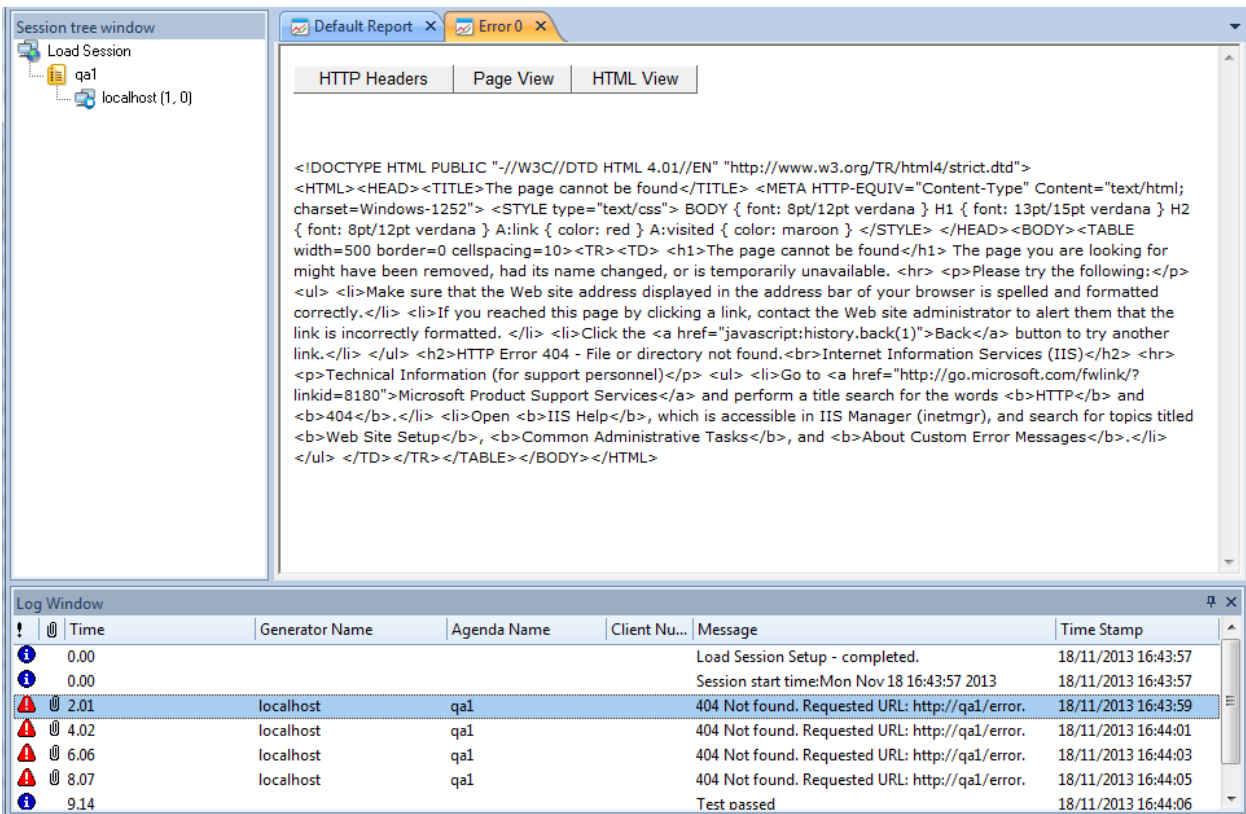


Figure 112: Full View of Error – HTML View

The Advanced Tab

Load Generator spawning sets the number of threads at which WebLOAD Console should automatically spawn Load Generators. Spawning Load Generators improves performance.

To access and configure the Advanced tab options:

1. Click **Global Options** in the **Tools** tab of the ribbon,
-Or-
Select **Global Options** from the Console System button.
2. Select the **Advanced** tab.

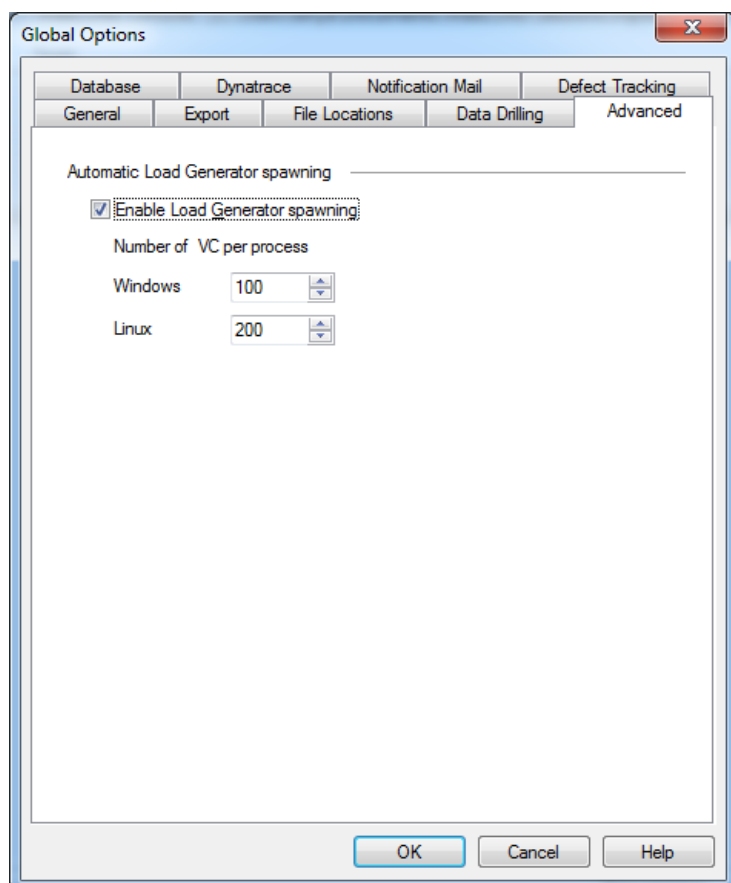


Figure 113: Global Options Advanced Tab

3. Under Automatic Load Generator Spawning, click **Enable Load Generator Spawning** (default).
4. Type or select the number of threads at which WebLOAD Console should automatically spawn Load Generators for the appropriate operating systems.
5. Click **OK**.

**Notes:**

Only client threads running within a single spawned process, on the same Load Generator, are able to share global variables. So if, for example, you have spawning set to 100 and you are running a total of 300 threads, you are actually running three spawned processes on three separate Load Generators.

Only client threads running within a single spawned process, on the same Load Generator, can be synchronized.

The Database Tab

The Database option enables you to instruct the Console to write all session data to a database during session execution. This is useful if you want to be able to view and analyze the session in the Web Dashboard (refer to the *WebLOAD™ Web Dashboard User Guide*).

To access and configure the Database tab options:

1. Click **Global Options** in the **Tools** tab of the ribbon,
-Or-
Select **Global Options** from the Console System button.
2. Select the **Database** tab.

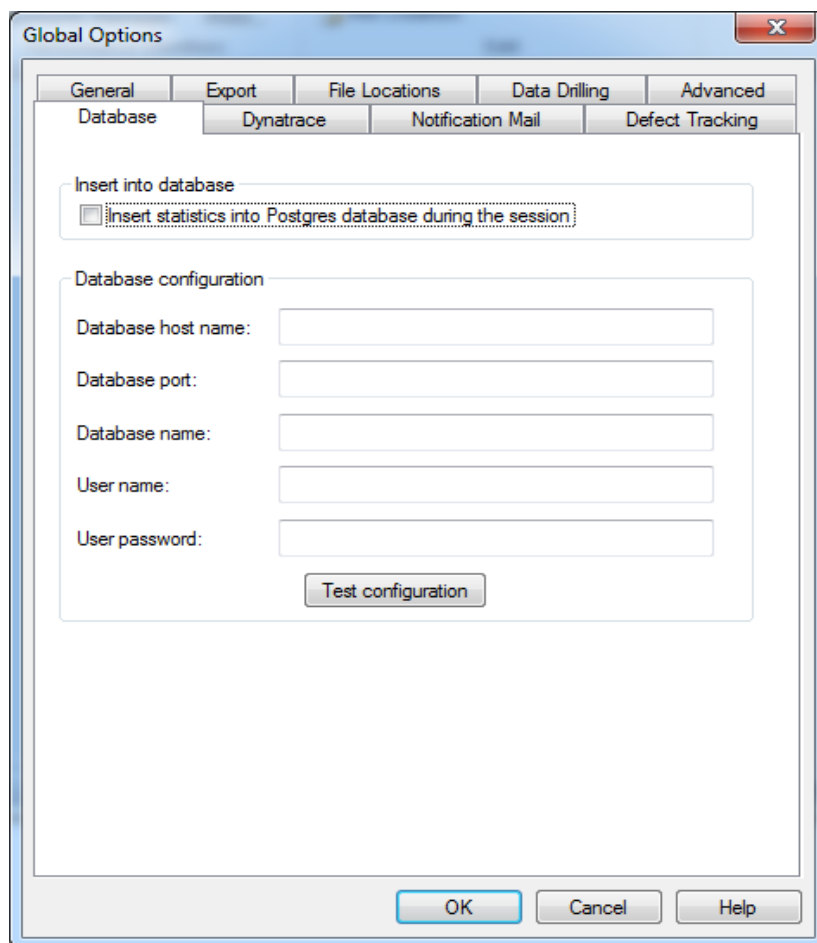


Figure 114: Global Options Database Tab

3. To instruct the Console to write to the database all session data during session execution:
 - a. Check the **Insert statistics into Postgres database during the session** checkbox.
 - b. In the **Database host name** field, enter the name of the server hosting the PostgreSQL database.
 - c. In the **Database port** field, enter the port number through which to connect to the server.
 - d. In the **Database name** field, enter the name of the database.
 - e. In the **User name** and **User password** fields, enter the credentials for accessing the database
4. Click **Test configuration** to test whether the Console can connect to the database based on the information you entered. A message is returned, indicating whether the connection attempt succeeded.
5. Click **OK**.

The Notification Mail Tab

The Notification Mail options enable you to configure the settings by which SLA notification emails are sent. An SLA notification email is optionally sent when a certain condition is met that requires special attention. For more information on defining SLA notification email conditions, see *The SLA Manager* (on page 204).

To access and configure the Notification Mail tab options:

1. Click **Global Options** in the **Tools** tab of the ribbon,
-Or-
Select **Global Options** from the Console System button.
2. Select the **Notification Mail** tab.

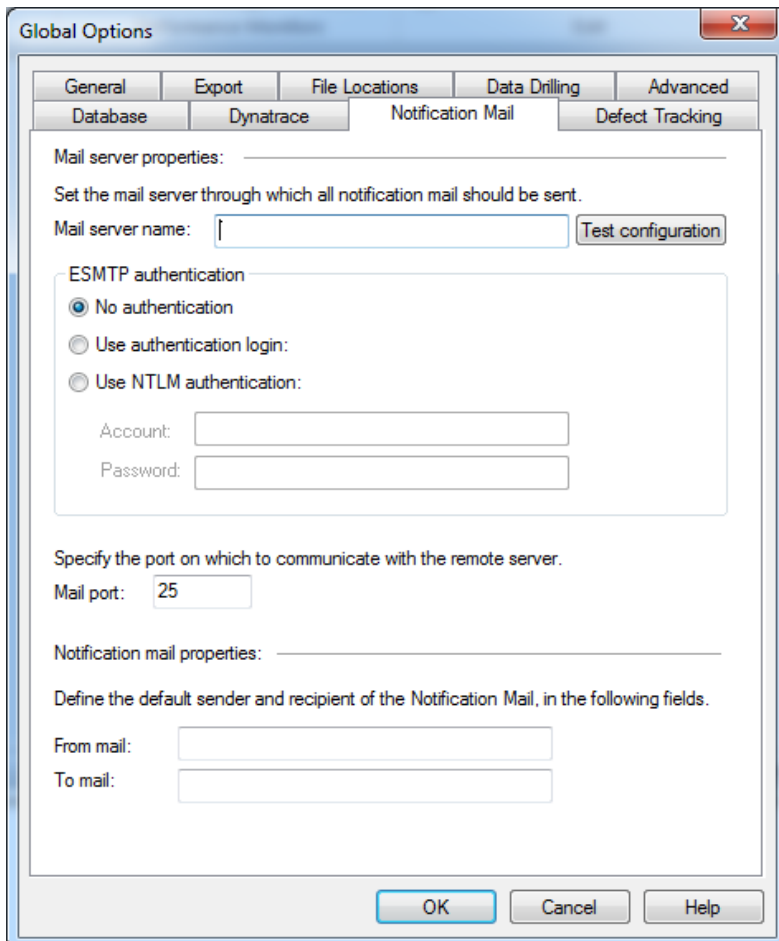


Figure 115: Global Options Notification Mail Tab

3. Under Mail Server Properties, enter a value in the Mail Server Name field. You can click **Test configuration** to test whether the Console can connect to the mail server based on the information you entered.

4. Under ESMTP Authentication, select **No Authentication, Use Authentication Login** or **Use NTLM authentication**. If you select **Use Authentication Login** or **Use NTLM authentication**, enter an Account and Password.
5. Under Notification Mail Properties, enter a value for the From Mail and To Mail fields.
6. Click **OK**.

The Dynatrace Tab

The Dynatrace options enable you to integrate WebLOAD with Dynatrace.

For information on using Dynatrace in conjunction with WebLOAD, refer to the *Integrating with Dynatrace* appendix.

To access and configure the Dynatrace tab options:

1. Click **Global Options** in the **Tools** tab of the ribbon,
-Or-
Select **Global Options** from the Console System button.
2. Select the **Dynatrace** tab.

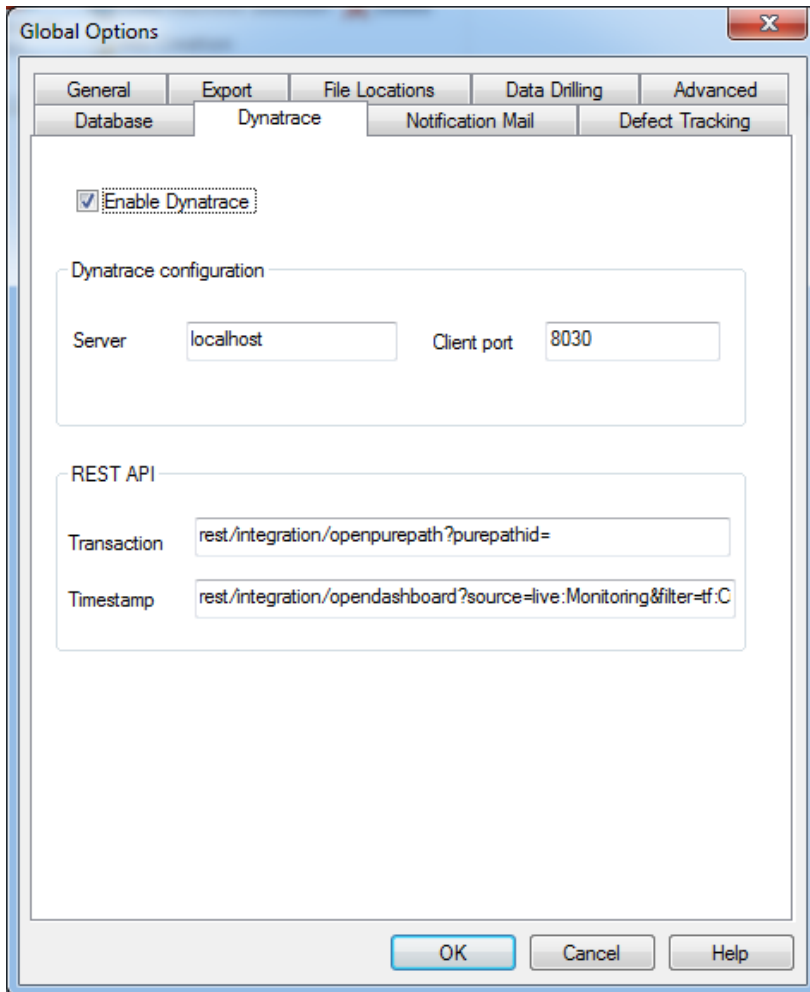


Figure 116: Global Options Dynatrace Tab

3. Select the **Enable Dynatrace** checkbox if you wish to integrate WebLOAD with Dynatrace.
4. If you wish to change the default system profile in which to view points of interest in Dynatrace, refer to *Specifying the Dynatrace System Profile* (on page 422).

The Defect Tracking Tab

The Defect Tracking options enable you to set the tracking system to use for tracking defaults.

To access and configure the Defect Tracking tab options:

1. Click **Global Options** in the **Tools** tab of the ribbon,
-Or-
Select **Global Options** from the Console System button.
2. Select the **Defect Tracking** tab.

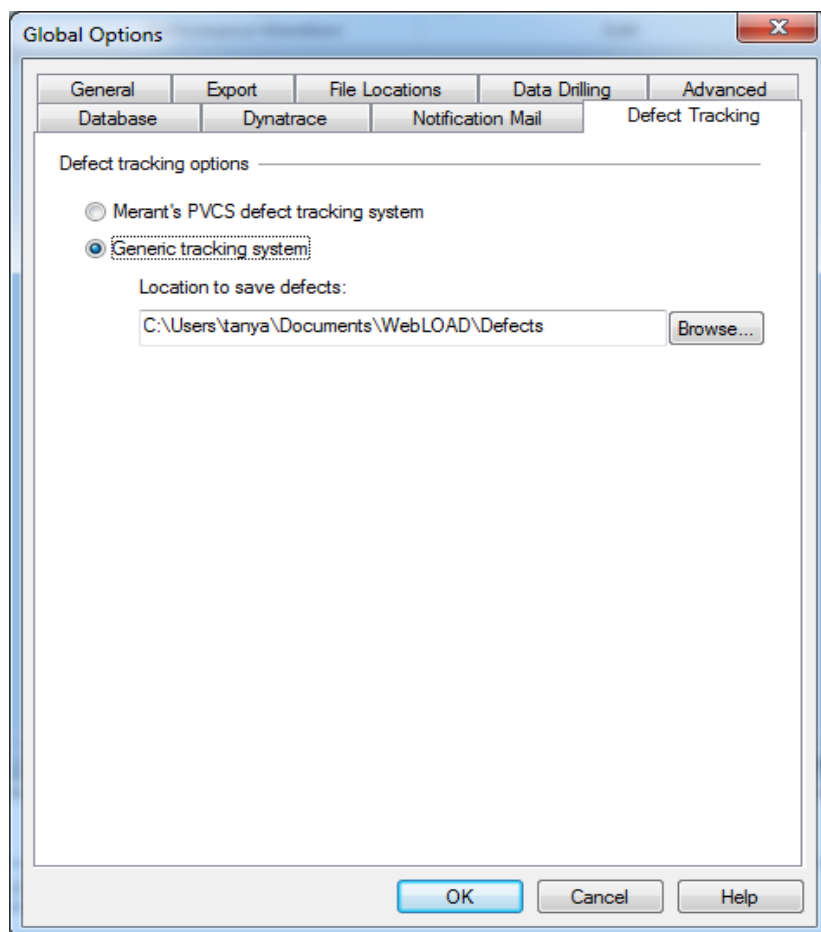


Figure 117: Global Options Defect Tracking Tab

3. Select the tracking system to use. If you select a generic tracking system, enter the location to save the defects.

Tracking Defects

After selecting the system to use when tracking defects, you can add an Issue that records problems that need to be tracked in the database. When you submit an Issue, you can add notes or files to it.

Accessing the Report Defect Dialog Box

To access the Report Defect dialog box:

1. Right click inside the Log Window, and select **Report Defect**.
2. If **Merant's PVCS defect tracking system** was selected in the **Defect Tracking** tab (Figure 117), the PVCS Tracker Project Login dialog box is displayed.



Figure 118: PVCS Tracker Project Login Dialog Box

- a. Enter your username and password.
- b. Click **OK**.

The Report Defect dialog box opens with three tabs:

- General tab
- Notes tab
- Attached Files tab

Entering a New Record

Use the General tab to enter a new record into the defect tracking project database.

To enter a new record into the defect tracking project:

1. Select the **General** tab. The General tab is displayed.

Figure 119: Report Defect – General Tab

2. Enter a title for the record in the Title field and a description of the record in the Description field.



Note: The Title field is required; all other fields are optional.

3. Enter the information in the remaining fields.
4. If you want to clear the form and reset the form fields to their default values, click **Clear**.

Adding, Modifying, and Deleting a Note

You can add, modify, or delete a note that indicates the status of a record.

To add, modify, or delete a note:

- Select the **Notes** tab. The Notes tab is displayed.

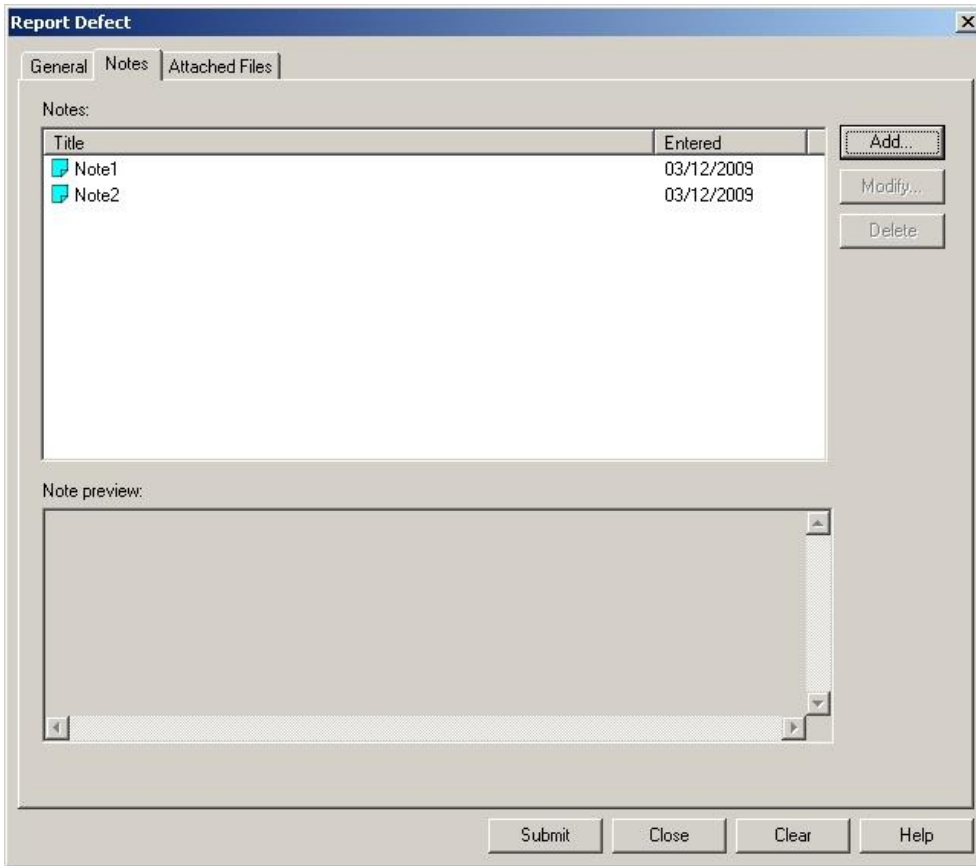


Figure 120: Report Defect – Notes Tab

To add a note:

1. Click **Add**. The Add Note dialog box is displayed.
2. Enter a Note Title and Note Text.
3. Click **OK**. The note title is added to the list in the Title field of the Notes tab.

To modify a note:

1. Select the note in the Notes section of the Notes tab.
2. Click **Modify**. The Modify dialog box is displayed.
3. Make the desired changes and click **OK**.

To delete a note:

1. Select the note in the Notes section of the Notes tab.
2. Click **Delete**. A confirmation dialog box is displayed.
3. Click **OK**. The note title is removed from the list in the Title field of the Notes tab.

Use the Add, Modify, and Delete buttons to add, edit, or delete Note entries, as illustrated in the preceding figure.

Attaching Files

You can attach files relevant to the record.

To attach or delete files:

1. Select the **Attached Files** tab. The Attached Files tab is displayed.

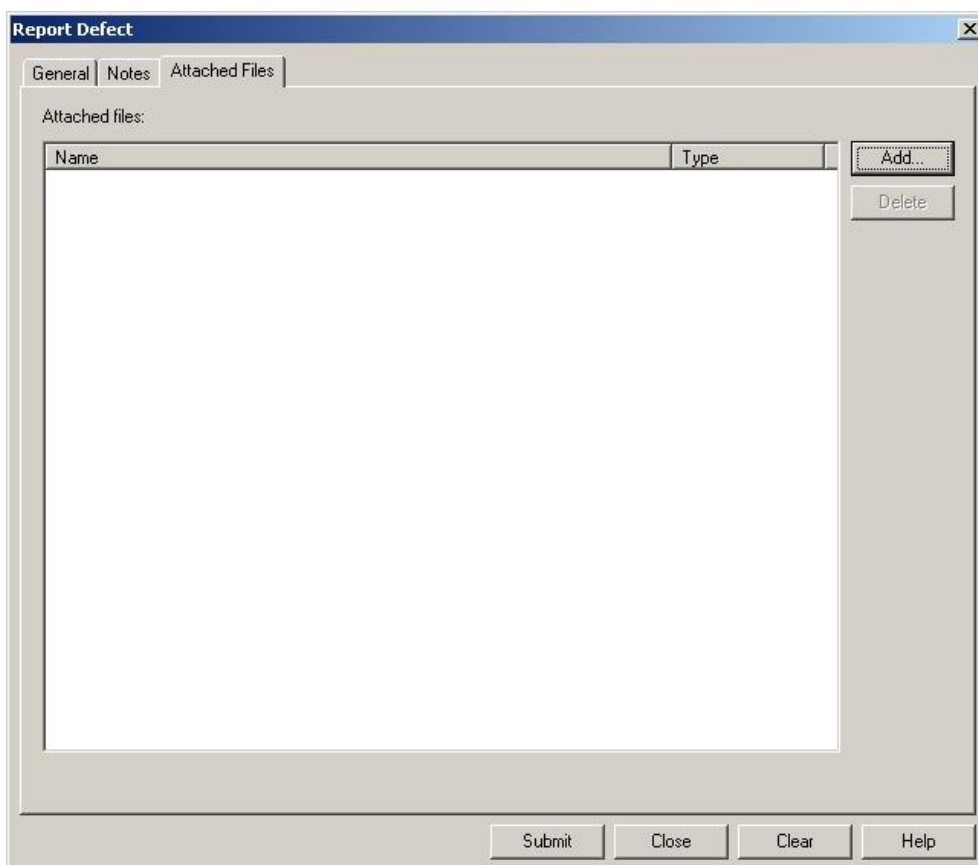


Figure 121: Report Defect – Attached Files

Attach a file:

- a. Click **Add**. The Add Attached Files dialog box is displayed.
- b. Select the desired file.
- c. Click **Open**. The selected file is displayed on the Attached Files tab.

Delete a file:

- a. On the Attached Files tab, select the file you want to delete.
- b. Click **Delete**. A confirmation dialog box is displayed.
- c. Click **OK**. The selected file is removed from the Attached Files tab.

2. Click **Submit** to submit the record.

The SLA Manager

The Service Level Agreement (SLA) Manager allows you to define conditions (SLA breaches) under which error messages should appear in the WebLOAD Console, and optionally an email notification sent as well. Conditions include cases where a specified Measurement value exceeds, equals, or drops down to less than a specified threshold value. For more information about configuring the email settings by which notification mail is sent, see *The Notification Mail Tab* (on page 196).

Accessing the SLA Manager

To access the SLA Manager:

- Select **SLA Manager** from the **Home** tab of the ribbon.

The SLA Manager dialog box appears. The currently defined SLA rules are listed in the dialog box. If no SLA rules are defined, the list is empty.

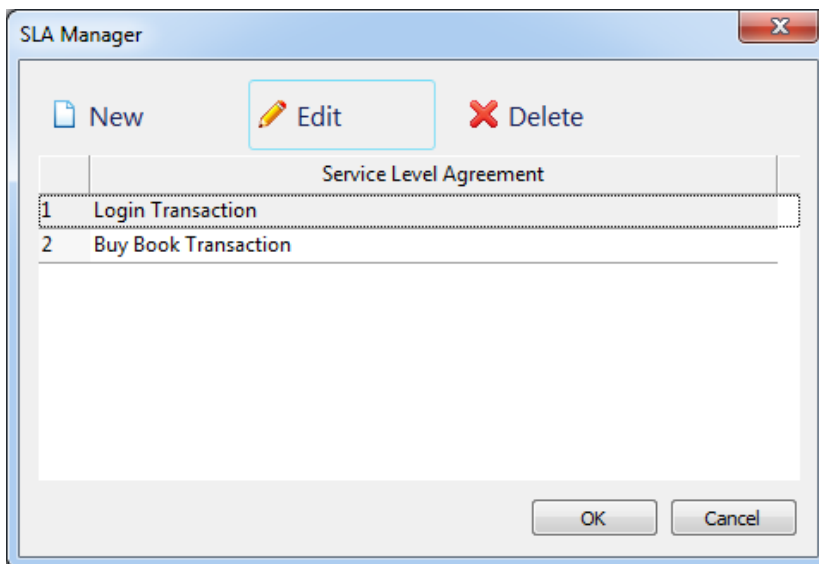


Figure 122: SLA Manager Dialog Box

Configuring the SLA Manager Rules and Actions

To configure the SLA Manager options:

1. In the SLA Manager dialog box, click **New** to define a new SLA rule, or select an existing SLA rule and click **Edit** to modify the existing rule.

The Rules and Actions dialog box appears for defining the SLA rule.

An SLA rule consists of one or more conditions, logically ANDed or ORed, as well as the actions to be taken (= error message and notifications to be sent) when the rule conditions are met.

Each line in the Rule Definition table defines one rule condition.

Figure 123: Rules and Actions Dialog Box – Adding a New Rule

2. If you are creating a new SLA rule, enter the rule name in the **Name** field.
3. To add a rule condition, click **Add** in the Add Notification Rules and Actions dialog box.

The Goal Definition dialog box appears, listing all the WebLOAD Console built-in measurements in a tree structure. The measurements are the building blocks of SLA rule conditions.

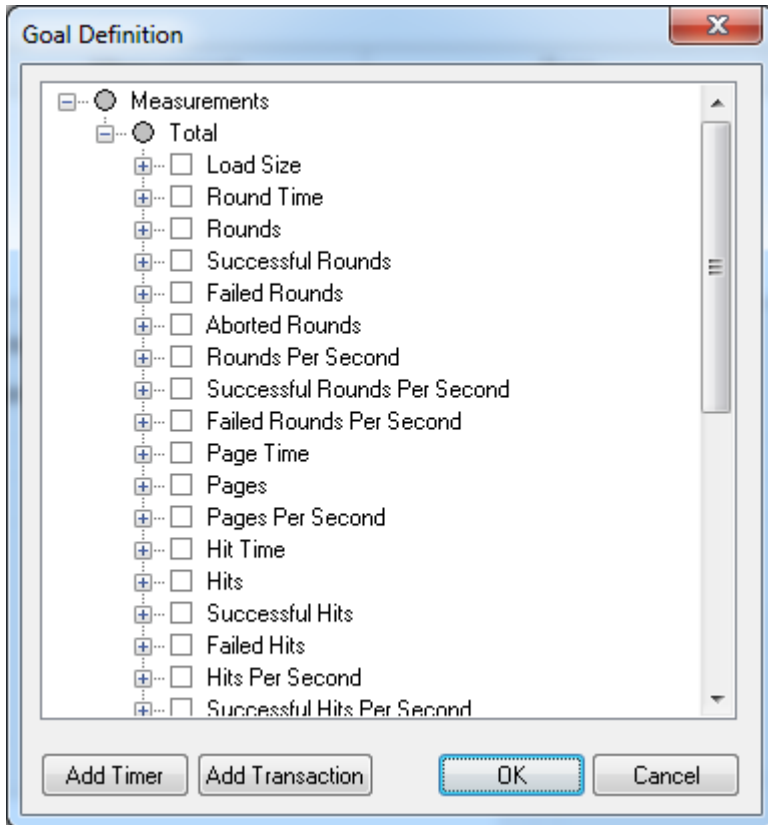


Figure 124: Goal Definition Dialog Box

- a. Optionally add a previously-defined custom measurement of type Timer to the tree, as follows:
 - ♦ Click **Add Timer**. A new node appears in the Goal Definition dialog box, with an empty name field.
 - ♦ Enter the timer's name in the name field.
- b. Optionally add to the tree, measurements based on a previously-defined transaction (which you created when editing the script in the WebLOAD Recorder), as follows:
 - ♦ Click **Add Transaction**.
 - ♦ In the Add Transaction dialog box that appears, enter the transaction name.

Seven measurements, based on the specified transaction, are automatically added to the tree. For example, if you add the previously-defined "Login" transaction, the following seven measurements are added.

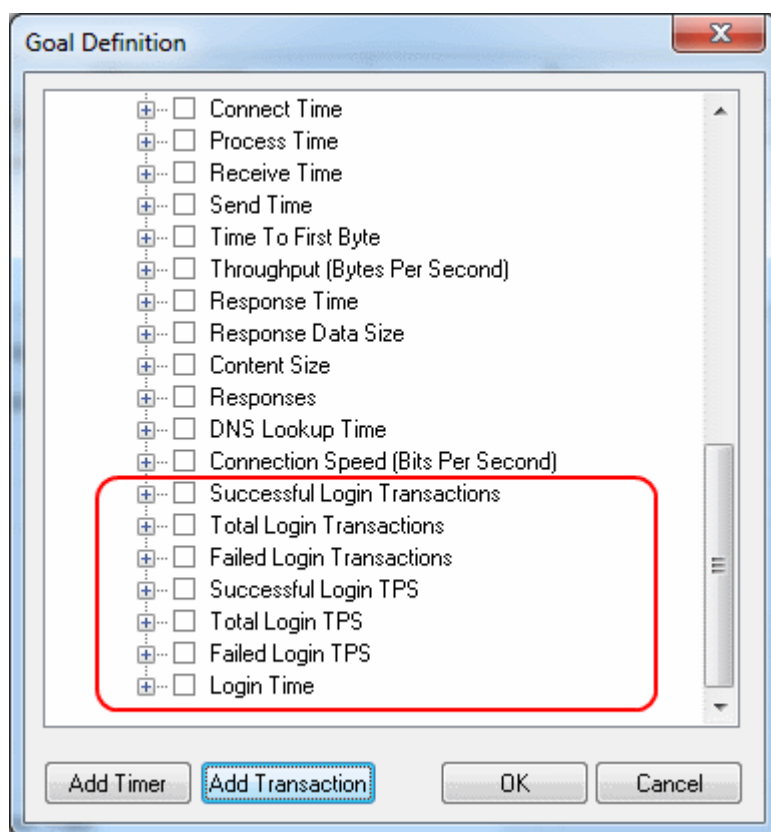


Figure 125: Goal Definition Dialog Box – Result of Add Transaction Action

- c. To create rule conditions, select the checkboxes adjacent to each of the measurements you wish to include in the SLA rule.

For example, to define an SLA rule condition that response time not exceed 5 seconds, select the checkbox adjacent to Response Time.



Note: A gray circle next to a measurement indicates that no default subcomponents are defined (the upper level component cannot be selected). To select subcomponents, click the + to expand the tree.

- d. Click **OK**.

WebLOAD Console closes the Goal Definition dialog box, and in the Rule and Actions dialog box opens a “condition” row for each of the selected measurements.

Rules and Actions

Name: Login Transaction

Rule Definition

	Source	Measurement	Type	Op	Value
1	Total	Failed Login Transactions	Current Slice Sum	>=	3.000000
2	Total	Login Time	Current Slice Average (Current Value)	>=	1.000000

Operator: OR (At least one expression must be true) AND (All expressions must be true)

Action Definition

Message

Display info message
 Display warning
 Display error
 Display fatal error and stop test execution

Message (Optional):

Mail

Send email

From mail:
 To mail:
 Subject:
 Message body:

OK Cancel

Figure 126: Rules and Actions Dialog Box – Showing Rule Conditions

4. For each condition, click in the **Op** field and select an operator from the drop-down list.
5. For each condition, click in the **Value** field and type a threshold value for the condition.
6. When more than one condition is defined, you can the logical connection between the conditions:
 - Select **OR** to require only one condition to be true for the rule to be met.
 - Select **AND** to require all conditions to be true for the rule to be met.

7. In the **Message** section under the **Action Definition** section of the Rules and Actions dialog box, define the log message to be displayed in the log if the SLA rule conditions are met:
 - Specify the type of log message to display: info, warning, error or fatal error. Note that a fatal error also causes test execution to stop.
 - Optionally specify the message text.
If you specify the message text, the log will display the specified text.
If you do not specify the message text, the log message will display the default SLA message format which begins with "SLA <SLA rule name> failed".

Note that a log message will appear in the log every time the SLA rule conditions are met, even if they are met multiple times in the course of load test execution.

8. In the **Mail** section under the **Action Definition** section of the Rules and Actions dialog box, optionally define the email notification to be sent out if the SLA rule conditions are met:
 - Check the **Send email** checkbox if you want an email notification to be sent.
 - Specify the email sender in the **From Mail** field.
 - Specify the email recipient in the **To Mail** field.
 - Specify the email subject line in the **Subject** field.
 - Specify the message text body in the **Message Body** field.

Note that an email notification is sent only once if the SLA rule conditions are met, even if they are met multiple times in the course of load test execution.

9. Click **OK** to save your settings and return to the main SLA Manager dialog box.

Setting Script Options

This chapter guides you through the process of setting WebLOAD Console default script options, individual script options, and Session script options, using some of WebLOAD Console's more advanced features. Script Options enable you to modify scripts to your specifications, and define the way WebLOAD Console behaves during runtime. Setting script options allows you to emulate different types of user activity. For example, you can set WebLOAD Console to emulate users connecting to your application at different modem speeds.



Notes:

Script options, set through the Console, are not uploaded when editing the script in WebLOAD Recorder. Debugger options are available (in WebLOAD Recorder) for debugging the script in WebLOAD Recorder.

Changes you make to the script through the Console override the settings for the script created through WebLOAD Recorder.

About Script Options

Script options allow you to configure WebLOAD Console behavior to emulate different types of user activity and ensure that the Virtual Clients running the test accurately emulate real users accessing your Web application.

There are three modes for modifying script settings. Most of the Options tabs described in this section can be edited in any of these three modes. Option tabs that are only available through a specific mode are noted in the tab description.



Note: To modify script options, you must be in Edit mode (click **Edit Template** in the **Session** tab of the ribbon or select **Edit Template** from the Console System button).


The three option modification modes are:

- **Default Options** – When building a new template or session, or editing an existing one, each new single script or script participating in a Mix is automatically assigned these defaults.

To activate the Default Options from the WebLOAD Console:

- Click **Default Options** in the **Tools** tab of the ribbon,

-Or-

Right-click the **root node** (Load Session node ) and select **Default Options**,

-Or-

Select **Default Options** from the Console System button.

- **Current Session Options** – You can modify script options for all scripts participating in a specific Load Session or part of a specific Load Template without changing the default settings for all Sessions.

To activate this mode:

- Click **Current Session Options** in the **Tools** tab of the ribbon,

-Or-

Right-click the **root node** (Load Session node ) and select **Current Session Options**,

-Or-

Select **Current Session Options** from the Console System button.

Modify the defaults as desired. All participating scripts currently listed in the Session Tree are automatically assigned the modified default values.

- **Script Options** – You can modify script options for an individual script within a new or existing Load Session or Load Template.

To activate this mode:

- Click **Script Options** in the **Tools** tab of the ribbon,

-Or-

Right-click the desired script in the Session Tree and select **Script Options**,

-Or-

Select **Script Options** from the Console System button.

The modifications you make here only affect that specific script.

Modifying Script Options

The Options dialog boxes are identical for all three modes, although they are activated differently, as described in *About Script Options* (on page 211).

The Options dialog boxes include the following tabs:

- SSL tab (see *Setting the SSL Options* on page 213)
- Sleep Time Control tab (see *Setting Sleep Time Playback Settings* on page 215)
- Terminate Round (see *Configuring Round Termination Settings* on page 217)
- Browser Cache tab (see *Setting the Browser Cache* on page 219)
- Browser Parameters tab (see *Configuring Browser Parameter Settings* on page 221)
- HTTP Parameters tab (see *Setting the HTTP Parameters* on page 229)
- Authentication tab (see *Setting Authentication* on page 232)
- Connection tab (see *Connection Speed Settings* on page 234)
- Diagnostic tab (see *Diagnostic Settings* on page 240)
- Functional Testing tab (see *Functional Testing Options* on page 245)
- Pass / Fail Definition tab (see *Pass / Fail Definition Settings* on page 249) (only edited through the Default and Current Session Options dialog boxes)
- Reset Mechanism tab (see *Configuring Reset Mechanism Settings* on page 250) (only edited through the Default and Current Session Options dialog boxes)
- Java Options tab (see *Configuring the Java Options* on page 252) (only edited through the Default and Current Session Options dialog boxes)

To modify script or session configuration options:

1. Select the desired tab.
2. Modify the options as needed.
3. Select another tab to modify additional options.
4. Click **OK** to accept the modifications, or **Cancel** to reject them, and exit the Default / Current Session Options dialog box.

Setting the SSL Options

To access and configure the SSL tab options:

1. Click **Default/Current Session Options** or **Script Options** in the **Tools** tab of the ribbon,
-Or-

Select **Default/Current Session Options** or **Script Options** from the Console System button.

2. Select the **SSL** tab.

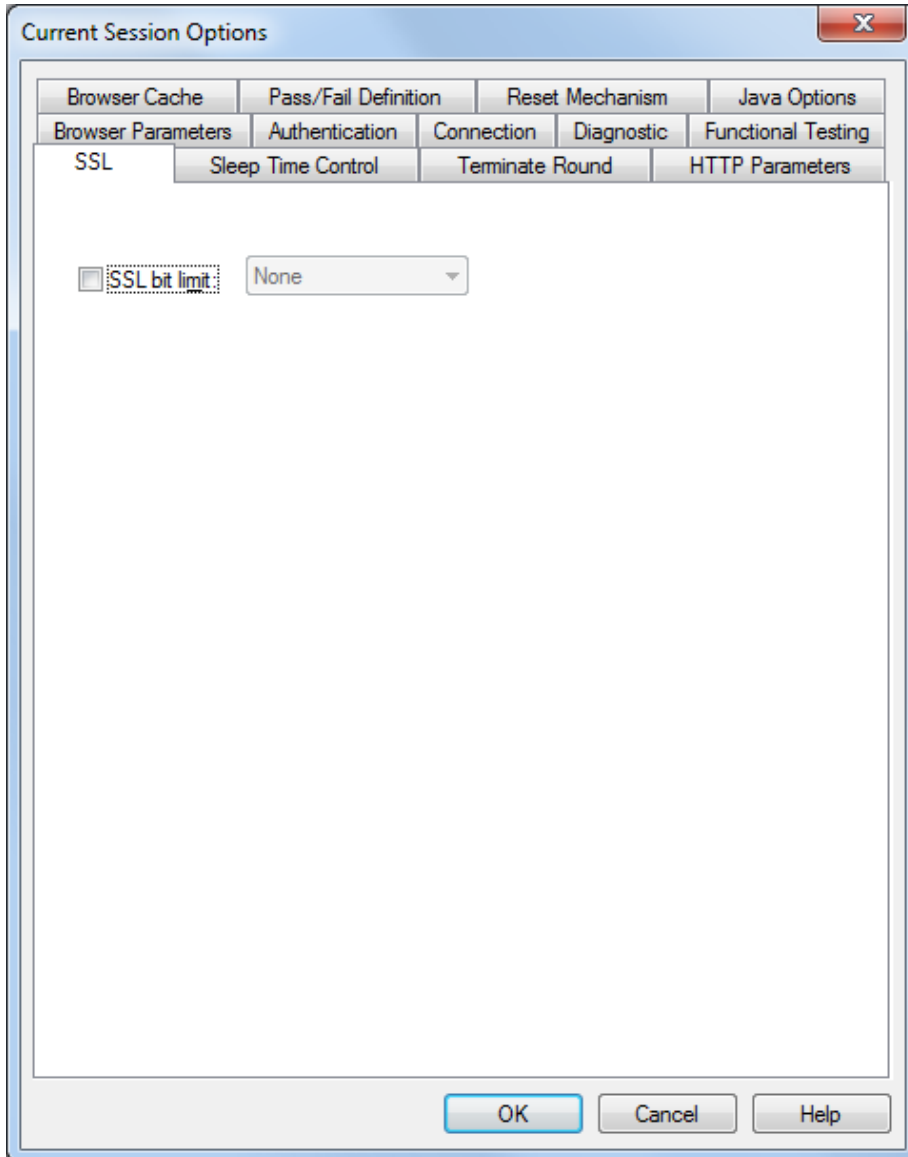


Figure 127: Current Session Options SSL Tab

3. Check **SSL Bit Limit** and select the highest SSL Bit Limit available between the Virtual Clients and the server.
4. Click **OK** to accept the modifications, or **Cancel** to reject them, and exit the Default/Current Session Options dialog box.

Setting Sleep Time Playback Settings

Sleep time is a pause to simulate the intermittent activity of real users. WebLOAD Console enables you to set a script to execute with the sleep times recorded in the script, random sleep times or no sleep times at all.

To set sleep time playback settings:

1. Click **Default/Current Session Options** or **Script Options** in the **Tools** tab of the ribbon,

-Or-

Select **Default/Current Session Options** or **Script Options** from the Console System button.
2. Select the **Sleep Time Control** tab.

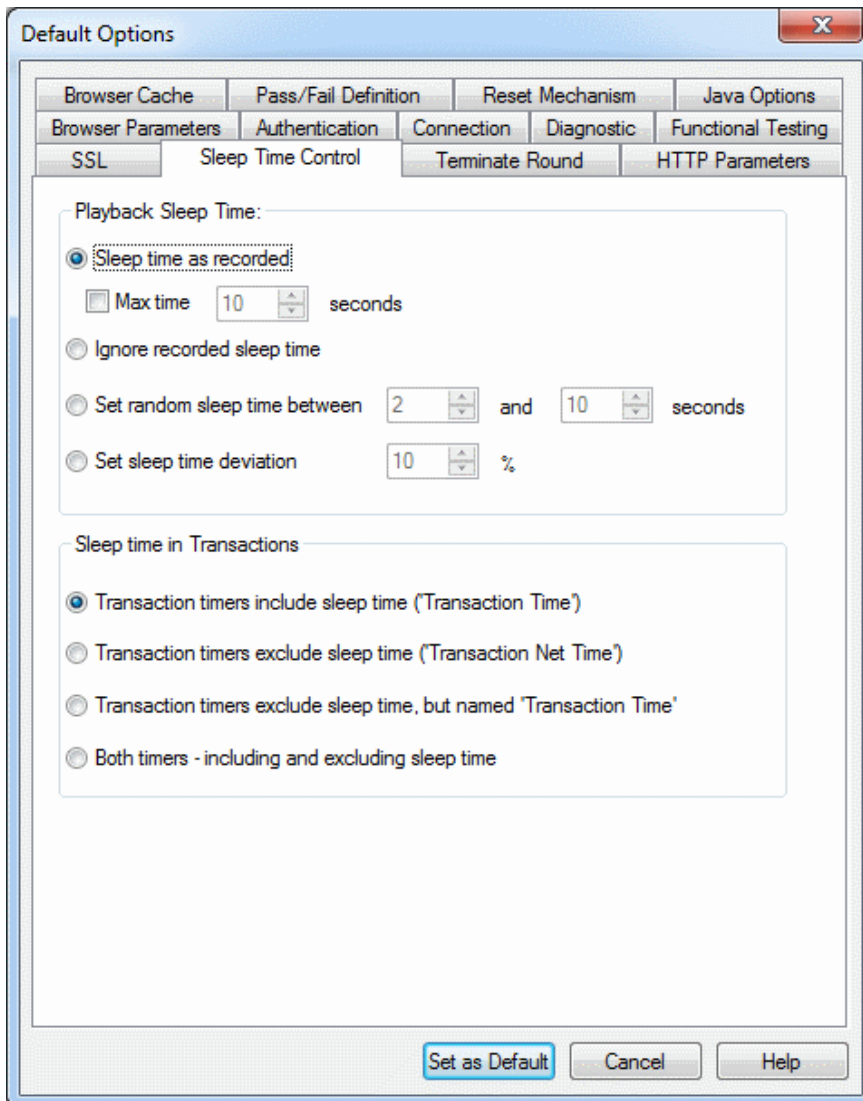


Figure 128: Current Session Options Sleep Time Control Tab

3. In **Playback Sleep Time**, specify the type of sleep to use when playing back the script:
 - Select **Sleep time as recorded** (default) to run the script with the delays corresponding to the natural pauses that occurred when recording the script. **Max time** provides an option of restricting playback of length sleep times, to no more than a specified number of seconds.
 - Select **Ignore recorded sleep time** to eliminate any pauses when running the script and run a worst-case stress test.
 - Select **Set random sleep time** and set the range (minimum and maximum) of delays to represent a range of users.
 - Select **Set sleep time deviation** and set which percentage of the recorded sleep time to use when running the script.

4. In **Sleep Time in Transactions**, specify how to calculate and display transaction timers:
 - **Transaction timers include sleep time ('Transaction Time')** – If you select this option, then if a transaction includes sleep commands, by default the sleep times will be included in the transaction time. The transaction time measurement will be called "Transaction <transaction name> Time".
 - **Transaction timers exclude sleep time ('Transaction Net Time')** – If you select this option, then if a transaction includes a sleep command, by default the sleep time will be excluded from the transaction time. The transaction time measurement will be called "Transaction <transaction name> Net Time".
 - **Transaction timers exclude sleep time, but named 'Transaction Time'** – If you select this option, then if a transaction includes a sleep command, by default the sleep time will be excluded from the transaction time. However, the transaction time measurement will be called "Transaction <transaction name> Time" (without the word "Net" in the parameter name).
 - **Both timers – including and excluding sleep time** – If you select this option, then if a transaction includes a sleep command, by default, two timers will be displayed. One timer, called "Transaction <transaction name> Time", will show the transaction time if sleep time is included in the transaction time. The second timer, called "Transaction <transaction name> Net Time", will show the transaction time if sleep time is excluded from the transaction time.
5. Click **OK** to accept the modifications, or **Cancel** to reject them, and exit the Default / Current Session Options dialog box.

Configuring Round Termination Settings

The WebLOAD Recorder enables you to define round termination settings.

To define the round termination behavior:

1. Click **Default/Current Session Options** or **Script Options** in the **Tools** tab of the ribbon,
 -Or-
 Select **Default/Current Session Options** or **Script Options** from the Console System button.
2. Select the **Terminate Round** tab.
 The Terminate Round tab moves to the front of the dialog box.

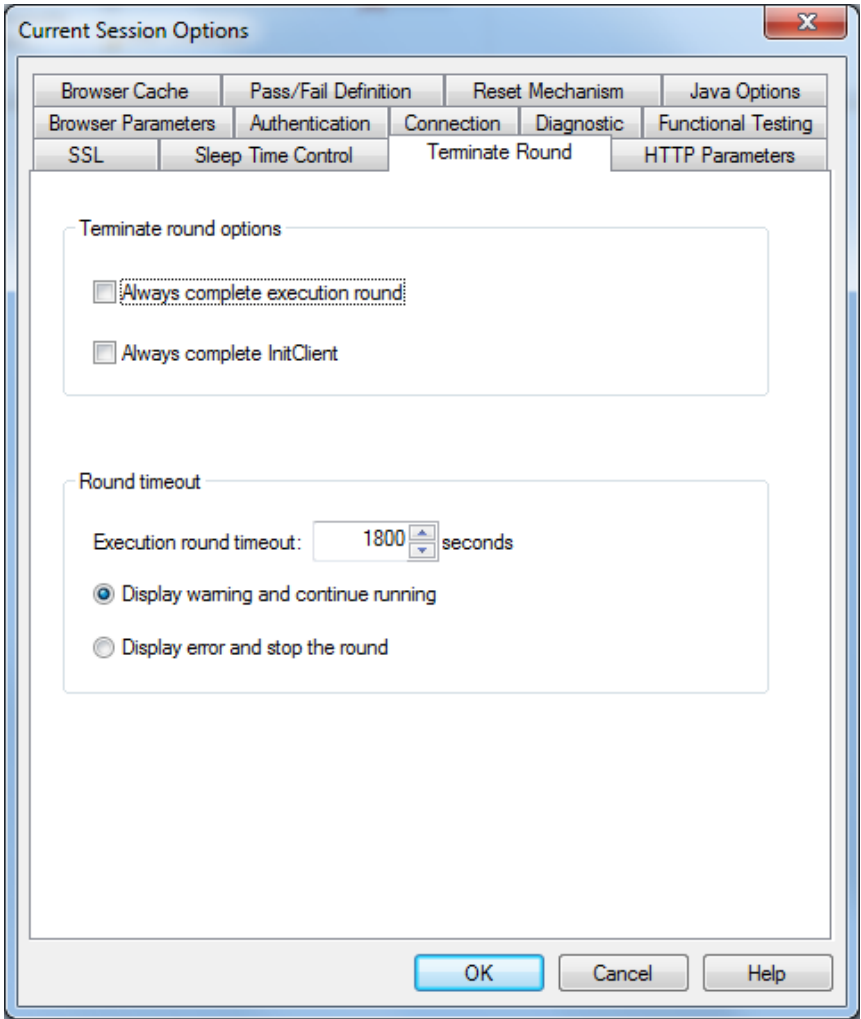


Figure 129: Current Session Options Terminate Round Tab

3. In the **Terminate round options** section, you can specify whether to complete a round even if a Stop command is received, as described in the following table.
4. In the **Round timeout** section, you can set the timeout period of a single round and the behavior of the system in cases where the timeout is reached, as described in the following table.

Table 31: Terminate Round Dialog Box Fields

Field	Description
<i>Terminate round options</i>	
Always complete execution round	If a Stop command is received, complete the current round before stopping.
Always complete InitClient	If a Stop command is received, complete the InitClient phase of the current round before stopping.

Field	Description
<i>Round timeout</i>	
Execution round timeout	The timeout period for a single round. Default value: 1800 seconds.
Display warning and continue running	Specifies that if the timeout has been reached and the round has not yet ended, the system will display a warning but will not stop the round. This is the default setting.
Display error and stop the round	Specifies that if the timeout has been reached and the round has not yet ended, the system will display an error, stop the round, and start a new round.

Setting the Browser Cache

The WebLOAD Recorder enables you to define the behavior of the cache that WebLOAD Console uses in order to simulate the behavior of a browser's cache. WebLOAD can cache JavaScript files, style sheets, images, applets, and XML files.

To define the browser cache behavior:

1. Click **Default/Current Session Options** or **Script Options** in the **Tools** tab of the ribbon,
-Or-
Select **Default/Current Session Options** or **Script Options** from the Console System button.
2. Select the **Browser Cache** tab.
The Browser Cache tab moves to the front of the dialog box.

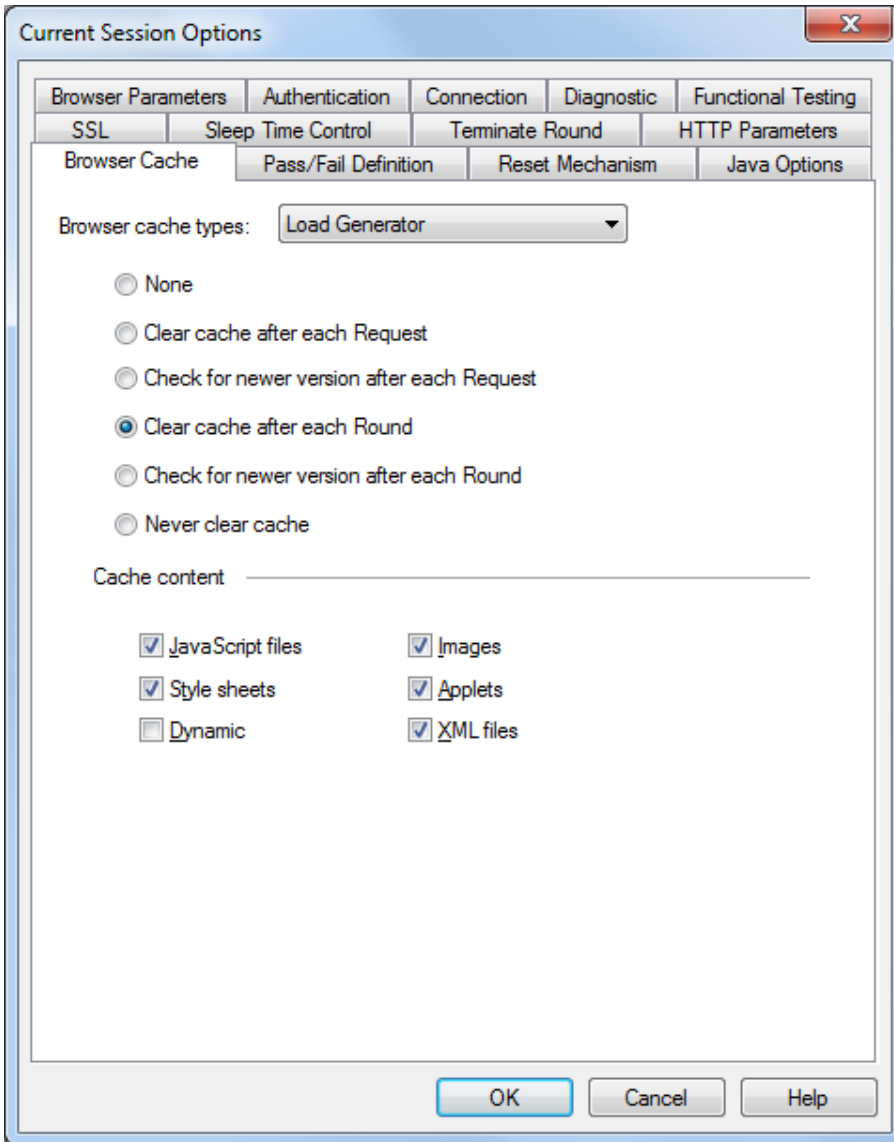


Figure 130: Current Session Options Brower Cache Tab

3. From the **Browser Cache Types** drop-down list, select **Load Generator** or **Probing Client**.
4. Select the type of cache and the content to filter according to the following table and click **OK**.

Table 32: Browser Cache Dialog Box Fields

Field	Description
<i>Browser cache types</i>	
None	All Virtual Clients simulate a browser with no available cache.
Clear cache after each Request	Defines that the cache will be cleared after each request.

Field	Description
Check for newer version after each Request	<p>Defines that WebLOAD will check for a newer version of the cached item with every request.</p> <p>Whenever the engine has a request for a cached resource, the engine sends the request with an “if-modified-since” header. If the server responds with a 200 status, the engine will refresh the cache.</p>
Clear cache after each Round	<p>Defines that the cache will be cleared after each script execution round. This is the default setting.</p>
Check for newer version after each Round	<p>Defines that WebLOAD will check for a newer version of the cached item after each round.</p> <p>Whenever the engine has a request for a cached resource, the engine sends the request with an “if-modified-since” header. If the server responds with a 200 status, the engine will refresh the cache.</p>
Never clear cache	<p>Defines that the cache will never be cleared. Each client maintains its own cache.</p>
<i>Cache content</i>	<p>You can select a filter, enabling you to indicate specific content types to be cached in the script. The available filters are:</p> <ul style="list-style-type: none"> • JavaScript files • Style sheets • Images • Applets • XML files • Dynamic

Configuring Browser Parameter Settings

The Browser Parameters option enables you to define Virtual Client behavior, such as:

- Browser type
- SSL encryption strength
- HTTP version
- Keep-Alive status
- Character encoding

Each of these configuration options is described in this section.

To configure the Browser Parameters options:

1. Click **Default/Current Session Options** or **Script Options** in the **Tools** tab of the ribbon,
-Or-
Select **Default/Current Session Options** or **Script Options** from the Console System button.
2. Select the **Browser Parameters** (default) tab.

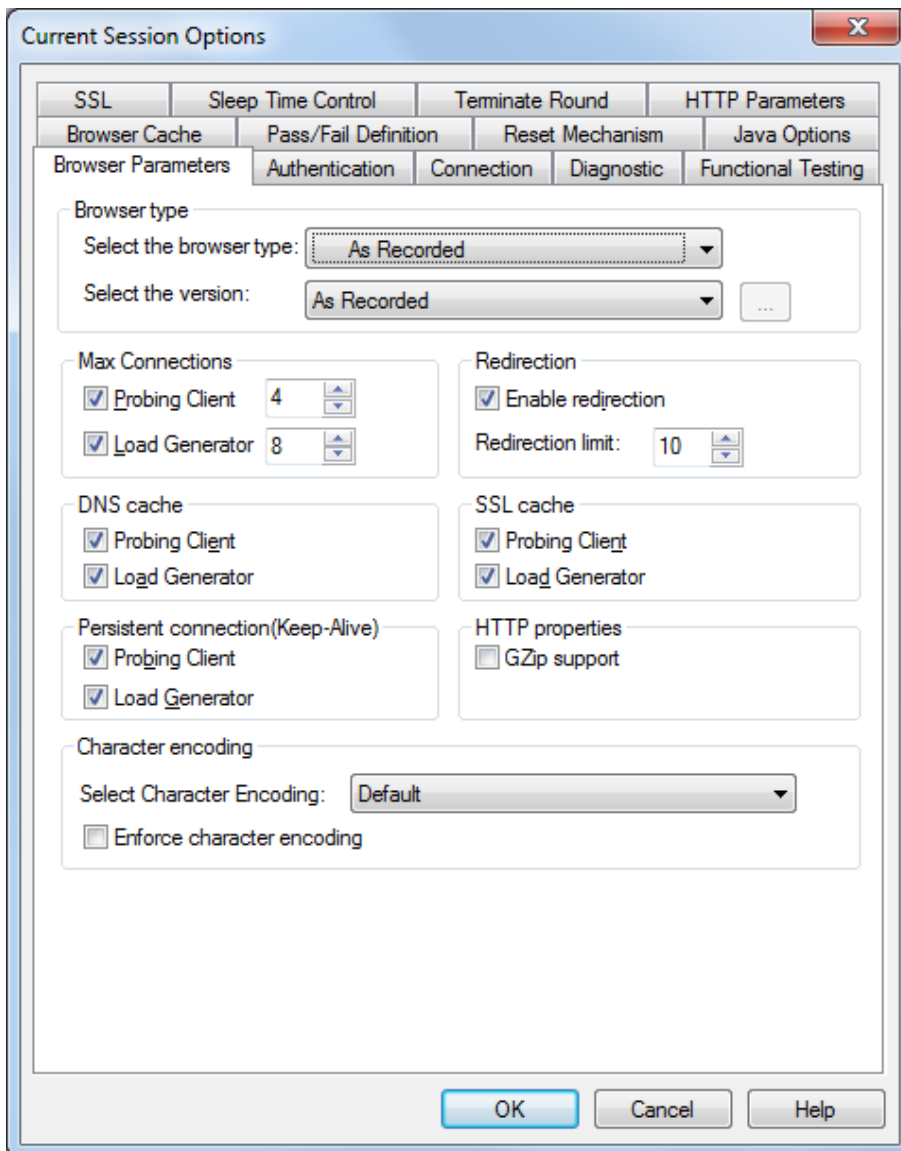


Figure 131: Current Session Options Browser Parameters Tab

3. Check the browser features you wish to use.
Each option is described in this section.

- Click **OK** to accept the modifications, or **Cancel** to reject them, and exit the Default / Current Session Options dialog box.



Note: The script (coded to account for a particular option) always overrides the Browser Parameters options. WebLOAD recommends always setting these options through the Console user interface.

Setting the Browser Type and Version

The browser type and version settings specify the type of browser the WebLOAD Virtual Clients should emulate. You can set WebLOAD Console to emulate any of the standard browsers, or you can specify any specific application by supplying a custom user-agent that is included in all HTTP headers. By default, all Virtual Clients use the WebLOAD Default browser agent.

To set the browser type and user-agent:

- Click **Default/Current Session Options** or **Script Options** in the **Tools** tab of the ribbon,
-Or-
Select **Default/Current Session Options** or **Script Options** from the Console System button.
- Select the **Browser Parameters** (default) tab.
- Select a browser type from the **Select the browser type** drop-down list.

Note that by default, the browser type and version are **As recorded**, which means: as recorded in the WebLOAD Recorder.



Figure 132: Browser Type Drop-Down List

- You can select an alternative version from the drop-down list, or click the Browse button to add a new user agent definition. See *Adding a Browser Version Definitions* (on page 223).

Adding a Browser Version Definitions

The available browser version list is appropriate for the browser type you select. You can add to the browser version list.

To add a browser version:

1. Select **Custom** in the **Select the browser type** drop-down list within the **Browser Type** area of the Browser Parameters tab (see Figure 132).
2. Click the browse button adjacent to **Select the version**.

The User Agent dialog box appears.

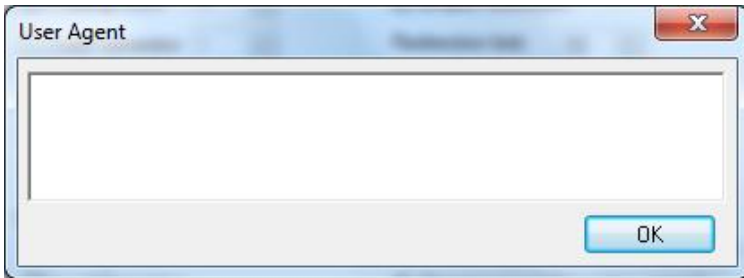


Figure 133: User Agent Dialog Box

3. Enter the new browser version definition.
4. Click **OK**.

Setting Load Generator and Probing Client Max Connections

To set the Load Generator and Probing Client multiple connection settings:

1. Click **Default/Current Session Options** or **Script Options** in the **Tools** tab of the ribbon,
-Or-
Select **Default/Current Session Options** or **Script Options** from the Console System button.
2. Select the **Browser Parameters** (default) tab.
3. Set the max connection properties as follows in the **Max Connections** section:
 - To enable multiple connections for Probing Clients, check the **Probing Client** checkbox. If you enable multiple connections, select the number of connections from the drop down list.
 - To enable multiple connections for Load Generators, check the **Load Generator** checkbox. If you enable multiple connections, select the number of connections from the drop down list.

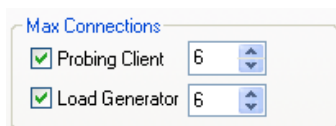


Figure 134: Max Connections

Setting Redirection Limits

To set a redirection limit:

1. Click **Default/Current Session Options** or **Script Options** in the **Tools** tab of the ribbon,
-Or-
Select **Default/Current Session Options** or **Script Options** from the Console System button.
2. Select the **Browser Parameters** (default) tab.
3. Select **Enable redirection**.



Figure 135: Redirection Properties

4. In the Redirection limit field, type or select the desired redirection limit. The default limit is 10.

Simulating DNS Cache

When DNS Caching is enabled (default), WebLOAD Console caches the IP addresses received from the domain name server, reducing the domain name resolution time. Disable DNS caching if you want to include the time for domain name resolution in the WebLOAD Console performance statistics.

To simulate DNS cache:

1. Click **Default/Current Session Options** or **Script Options** in the **Tools** tab of the ribbon,
-Or-
Select **Default/Current Session Options** or **Script Options** from the Console System button.
2. Select the **Browser Parameters** (default) tab.
Under DNS Cache, select **Probing Client** and **Load Generator**.

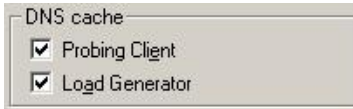


Figure 136: DNS Cache Properties

Simulating SSL Cache

When SSL Caching is enabled (default), WebLOAD Console caches the SSL decoding keys received from an SSL server, and WebLOAD Console only receives the key on the first SSL connection in each round. In subsequent connections, WebLOAD Console retrieves the key from cache. Enabling SSL Cache also reduces transmission time during SSL communication. Disable SSL caching if you want to measure the transmission time of the decoding key in the WebLOAD Console performance statistics for each SSL connection.

To simulate SSL cache:

1. Click **Default/Current Session Options** or **Script Options** in the **Tools** tab of the ribbon,

-Or-

Select **Default/Current Session Options** or **Script Options** from the Console System button.

2. Select the **Browser Parameters** (default) tab.

Under SSL Cache, select **Probing Client** and **Load Generator**.

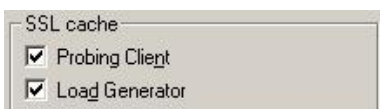


Figure 137: SSL Cache Properties

Enabling Persistent Connection (Keep-Alive)

When enabled, WebLOAD Console keeps an HTTP connection alive between successive accesses in the same round of the main script. Keeping a connection alive saves time between accesses. WebLOAD Console attempts to keep the connection alive unless you switch to a different server. However, some HTTP servers may refuse to keep a connection alive. You should not keep a connection alive if establishing the connection is part of the performance test.

To enable persistent connection:

1. Click **Default/Current Session Options** or **Script Options** in the **Tools** tab of the ribbon,

-Or-

Select **Default/Current Session Options** or **Script Options** from the Console System button.

2. Select the **Browser Parameters** (default) tab.

Under Persistent Connection, select **Probing Client** and **Load Generator**.

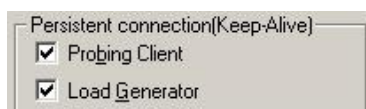


Figure 138: Persistent Connection Properties

Enabling GZip Support

Enabling GZip support sets the `wlGlobals.AcceptEncodingGzip` flag. For each request, WebLOAD Console sends the header "Accept-Encoding: gzip, deflate". This tells the server that the client can accept zipped content. When this header is turned on, the server MAY send a response with the header "content-encoding: gzip" or "content-encoding: deflate". If either of these headers is sent, it means that the response is zipped/deflated and WebLOAD Console will unzip/inflate the content.



Note: Most servers will work correctly even if the client does not send the "Accept-Encoding: gzip, deflate" header. Therefore, unless needed, it is recommended not to set the `wlGlobals.AcceptEncodingGzip` flag because it is performance heavy. However, some servers will fail if it is not sent. Microsoft Internet Explorer/Mozilla sends it by default.

You can manually code the script to send the "Accept-Encoding: gzip, deflate" header even if the `wlGlobals.AcceptEncodingGzip` flag is not set. In this case, if the server returns zipped content, WebLOAD Console will not unzip it, so the content will not be available for WebLOAD Console to function properly.



Note: Enabling GZip support sets the Browser Emulation Settings to the maximum browser emulation accuracy. Disabling GZip support sets the Browser Emulation Settings to the maximum Load Generator capacity.

To enable GZip support:

1. Click **Default/Current Session Options** or **Script Options** in the **Tools** tab of the ribbon,

-Or-

Select **Default/Current Session Options** or **Script Options** from the Console System button.

2. Select the **Browser Parameters** (default) tab.
3. Under HTTP Properties, select **GZip support**.

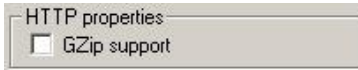


Figure 139: GZip Support

Setting Character Encoding

The character encoding settings enable you to select a specific character set. You can indicate whether the parser should use the character set it parses in the HTML pages or override it using the specified character set.

To specify character encoding:

1. Click **Default/Current Session Options** or **Script Options** in the **Tools** tab of the ribbon,
 -Or-
 Select **Default/Current Session Options** or **Script Options** from the Console System button.
2. Select the **Browser Parameters** (default) tab.
3. Under Character encoding, in the Select Character Encoding drop-down, select the value corresponding to the character set that you want to use. The default value is Default (0), the regional settings of the computer.
4. Check the **Enforce character encoding** checkbox to override the character set parsed from the HTML pages with the value specified in the Select Character Encoding drop-down list. Uncheck the **Enforce character encoding** checkbox to indicate that the parser should use the character set it parses in the HTML pages. The default value is `false` (use the encoding from the HTML pages).

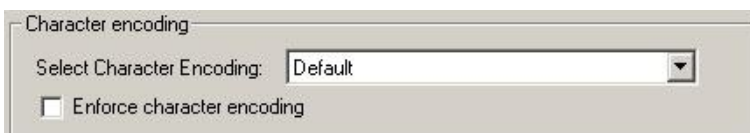


Figure 140: Character Encoding Settings

Setting the HTTP Parameters

The HTTP Parameters option enables you to define HTTP client behavior on the HTTP protocol level.

To set the HTTP Parameters options:

1. Click **Default/Current Session Options** or **Script Options** in the **Tools** tab of the ribbon,

-Or-

Select **Default/Current Session Options** or **Script Options** from the Console System button.

2. Select the **HTTP Parameters** tab.

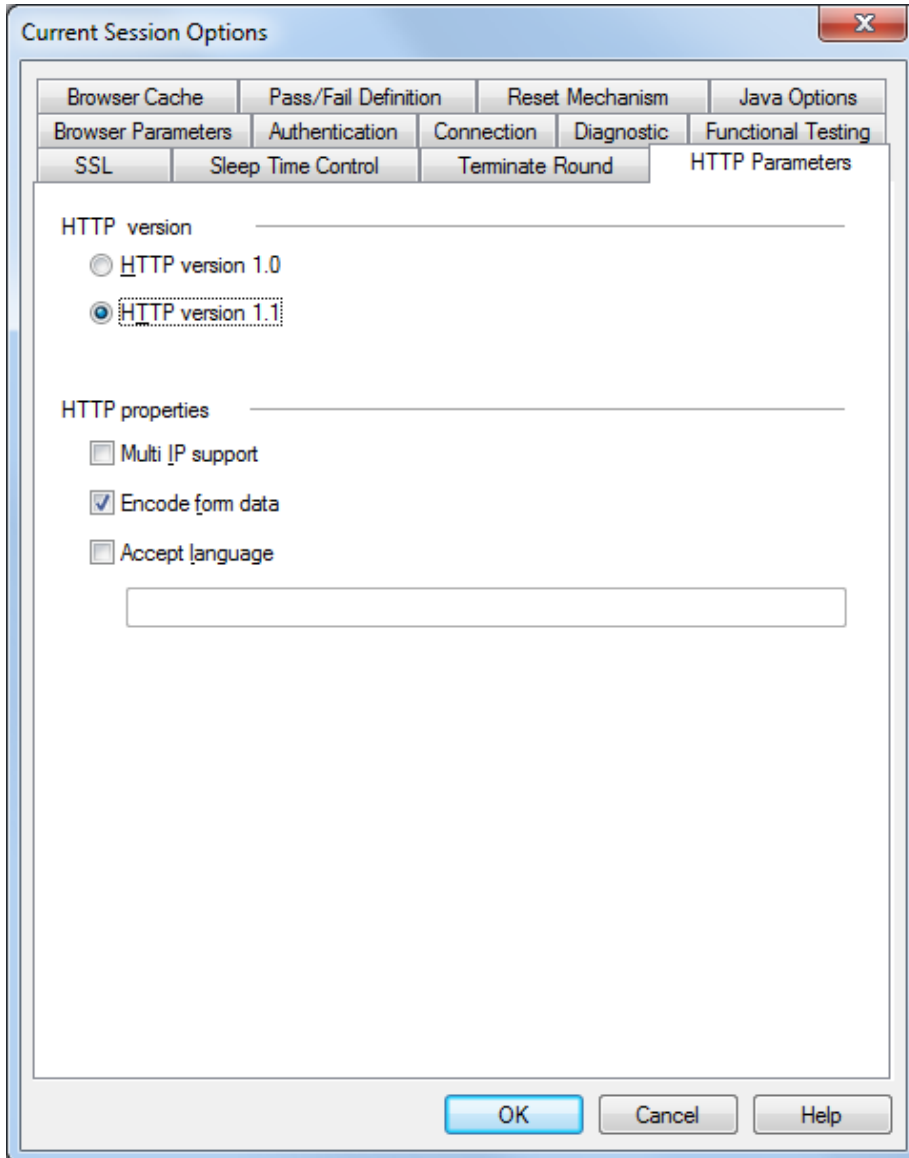


Figure 141: Current Session Options HTTP Parameters Tab

3. Set the HTTP version by clicking **HTTP version 1.0** or **HTTP version 1.1**.



Note: For the WebLOAD Console 7.x protocol engine, the default HTTP version is 1.1. For the WebLOAD Console 5.x protocol engine, the default HTTP version is 1.0.

4. Select one or more **HTTP Properties** checkboxes.
5. Click **OK**.

The following table describes the fields and buttons in the HTTP Parameters dialog box.

Table 33: HTTP Parameters Dialog Box

Field	Description
HTTP Version	The appropriate HTTP protocol version (for example "HTTP/1.1").
HTTP version 1.0	Sets the HTTP protocol version to 1.0.
HTTP version 1.1	Sets the HTTP protocol version to 1.1.
HTTP Properties	
Multi IP Support	Sets the <code>wlGlobals.MultiIPSupport</code> flag to indicate that the HTTP protocol version supports more than one IP address.
Encode Form Data	<p>Sets the <code>wlGlobals.EncodeFormdata</code> flag.</p> <p>In general, when an HTTP client (Microsoft Internet Explorer/Firefox or WebLOAD Console) sends a post request to the server, the data must be HTTP encoded. Special characters like blanks, ">" signs, and so on, are replaced by "%xx". For example, space is encoded as "%20".</p> <p>This encoding can be performance heavy for large data, so WebLOAD Console allows you to turn it off.</p> <p>This should ONLY be done if you are sure that the data does not contain special characters. If it does, and the customer wants to improve performance via this flag, then you should replace the special characters within the script or use <code>wlHttp.EncodeFormdata</code> to set the flag for specific requests.</p>
Accept Language	<p>Sets the <code>wlGlobals.AcceptLanguage</code> flag. This flag defines a global value for the "Accept-Language" header which will be sent with each request. Some applications/servers will behave differently depending on the setting. It is a simple string and WebLOAD Console does not enforce any checks on the values.</p> <p>It is similar to the <code>wlGlobals.UserAgent</code> property in the sense that it is a <code>wlGlobals/wlHttp</code> setting that affects the value of request headers.</p>

Setting Authentication

The Authentication option enables you to define the Global and Proxy authentication settings.

To set authentication:

1. Click **Default/Current Session Options** or **Script Options** in the **Tools** tab of the ribbon,


-Or-

Select **Default/Current Session Options** or **Script Options** from the Console System button.
2. Select the **Authentication** tab.

Figure 142: Current Session Options Authentication Tab

3. Type the appropriate user names and passwords into the corresponding Global Authentication Settings input text boxes:

Table 34: Global Authentication Settings Fields

Field	Description
User name and Password	<p>The user name and password the script should use to log onto restricted HTTP sites.</p> <p> Note: User names must be entered using one of the following formats:</p> <ul style="list-style-type: none"> • Domain\User • Domain\\User • Domain/User

Field	Description
NT user name and NT password	The user name the script should use for Windows NT Challenge response authentication.
Proxy user name and Proxy password	The user name and password the script should use for proxy servers that require user authorization.
SSL client certificate file and SSL client certificate password	The filename (optionally including a directory path) and password of the certificate WebLOAD Console makes available to SSL servers.
Authentication method	The authentication method supported by the server: <ul style="list-style-type: none"> • NTLM (default). • Kerberos.
Kerberos server	The Kerberos server Fully Qualified Domain Name (FQDN). For example, specify "qa4" rather than "qa4.radview.co.il". This field is only enabled when the authentication method is set to Kerberos.

4. Type the appropriate proxy host name, port number, user name, and password into the corresponding Proxy Authentication input text boxes when working with proxy authentication.
5. Click **OK** to accept the modifications, or **Cancel** to reject them, and exit the Default/Current Session Options dialog box.

Bandwidth Settings

WebLOAD Console enables you to simulate various systems and connection configurations, including setting a 'virtual limit' on the bandwidth available during a test. Setting a limit on bandwidth allows testers to test applications for users that may be limited in their workplace connections speeds.

To configure the connection speed:

1. Click **Default/Current Session Options** or **Script Options** in the **Tools** tab of the ribbon,
-Or-
Select **Default/Current Session Options** or **Script Options** from the Console System button.
2. Select the **Connection** tab.

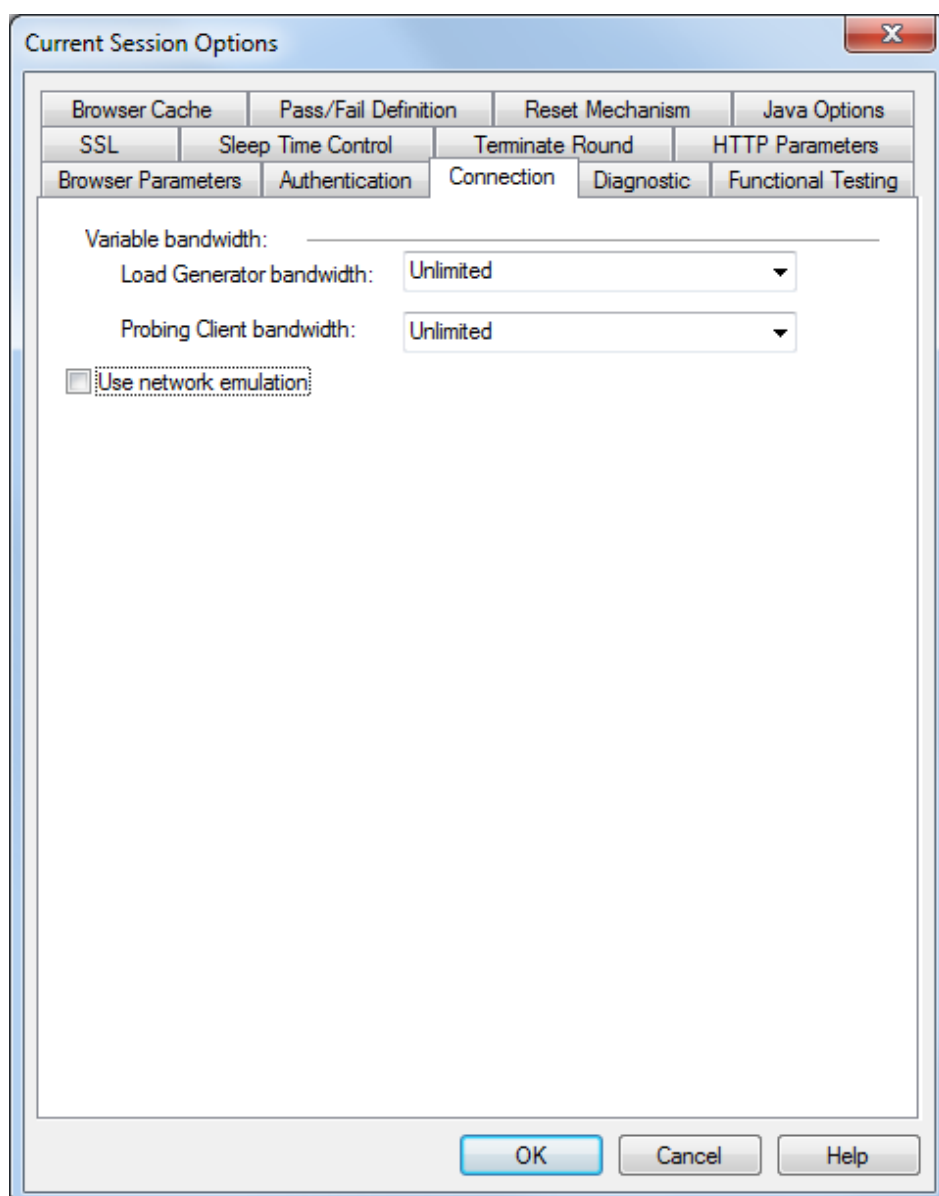


Figure 143: Current Session Options Connection Tab

The Variable Bandwidth feature enables you to set the bandwidth for the load and emulate users connecting through different modem connections (14400, 28800...).

3. Select the desired bandwidth from the **Load Generator Bandwidth** drop-down list.
4. Select the desired bandwidth from the **Probing Client Bandwidth** drop-down list.
5. Check the **Use network emulation** option if you wish to view and control the emulation settings (download/upload bandwidth, delay, and packet loss) that are used to emulate each of the selections you make in the **Load Generator Bandwidth** and **Probing Client Bandwidth** fields.

You can click any of the emulation settings to manually set it to a custom value, as shown in Figure 144.

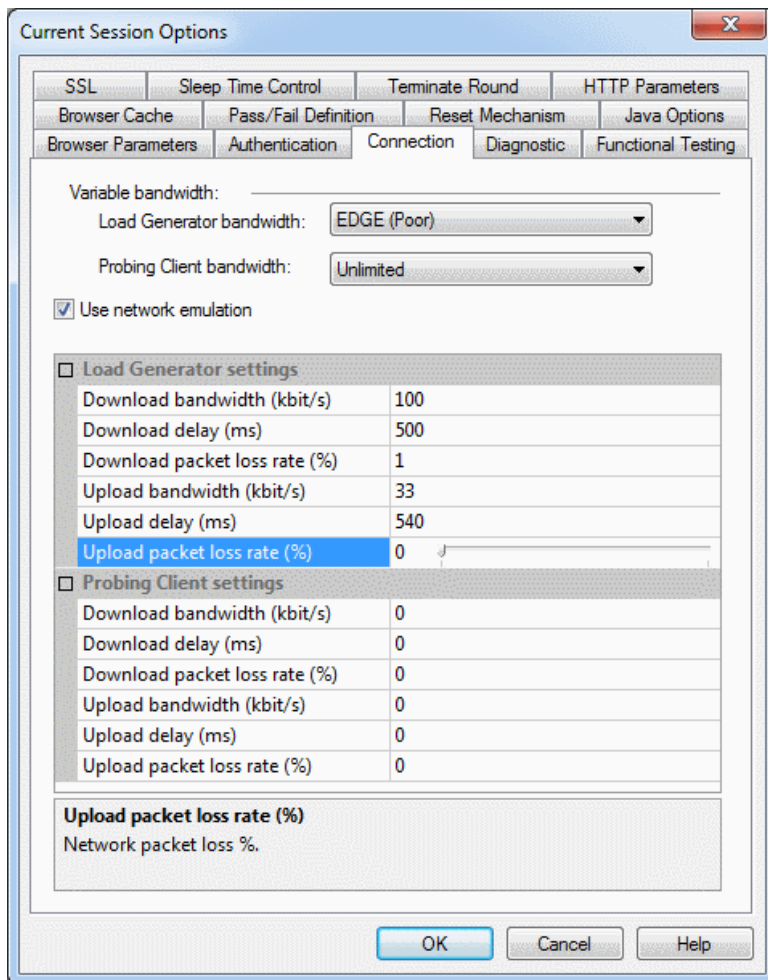


Figure 144: Current Session Options Connection Tab – Manual Editing

- Click **OK** to accept the modifications, or **Cancel** to reject them, and exit the Default/Current Session Options dialog box.

Enabling Network Emulation

Network Emulation services are supported by WebLOAD through the open source ipfirewall (ipfw) engine. To use network emulation, you need to install ipfw, as described below.

Network emulation is performed by the Load Generators. Ipfw needs to be installed on each Load Generator in which you want to run network emulation. Installation depends on the Load Generators type:

Cloud Load Generators – The default WebLOAD native Amazon cloud image includes the network emulation engine, therefore no action is required in order to use the network emulation in this option.

Physical Load Generators – the procedure depends on whether the Load Generator is installed on a Windows or Linux machine.

Installing ipfw on a Windows Load Generator

On each Load Generator.

To install ipfw on a Windows machine:

1. Download <https://s3.amazonaws.com/radview/dummynet.zip>. The zip file includes two folders: 32bit and 64bit. You need the folder that corresponds to the Windows version installed on the Load Generator – either 32-bit or 64-bit.
2. Change the name of the relevant folder to ipfw, and place it under the directory C:\Program Files (x86)\RadView\WebLOAD\bin.

Alternatively, copy the ipfw folder to any other location, and in the C:\Program Files (x86)\RadView\WebLOAD\bin\networkemulation.ini file, set IPFW_PATH="`<path>/ipfw.exe`".

3. Install the NDIS driver as follows:
 - a. Open the configuration panel for the network card in use by right-clicking the icon on the systray, or by selecting **Control Panel** -> **Network** and then selecting the card.
 - b. Select **Properties** > **Install** > **Service** > **Add**.
 - c. Click **Driver Disk** and select netipfw.inf in this folder.
 - d. Select **ipfw+dummynet** which is the only service you should see.
 - e. Click **Accept** when warnings appear regarding the installation of an unsigned driver (typically two warnings per installed network card).
4. Verify installation is successful, as follows:

- a. Open a command prompt window by running `cmd.exe` as an Administrator.
 - b. Run the `cd` command to navigate to the `ipfw` installation directory.
 - c. Run `ipfw.exe list`
 - d. Verify you get the following output: `65535 allow ip from any to any`
5. Make sure to run TestTalk as administrator on the Load Generator machine.

To remove ipfw from a Windows machine:

1. Open the configuration panel for the network card in use by right-clicking the icon on the systray, or by selecting **Control Panel -> Network** and then selecting the card.
2. Select **Properties**.
3. Select **ipfw+dummysnet**.
4. Click **Remove**.

Installing ipfw on a Linux CentOS 7 Load Generator

On each Load Generator.

To install ipfw on a Linux CentOS 7 machine:

1. Install the following necessary prerequisites for building a dummysnet kernel module:

```
$ sudo yum -y groupinstall "Development Tools"  
$ sudo yum -y install glibc-devel.i686 glibc-devel
```


2. Install a kernel source matching the specific kernel on the machine, as follows:

- a. If you are not sure which kernel version you have, run the following command:

```
$ sudo uname -r
```

- b. Download the kernel sources as follows (replace the version in the example with your version):

```
$ cd /tmp
$ wget http://vault.centos.org/7.1.1503/centosplus/Source/SPackages/kernel-plus-3.10.0-229.el7.centos.plus.src.rpm
$ mkdir -p /tmp/rpmbuild/{BUILD,RPMS,SOURCES,SPECS,SRPMS}
$ echo "%_topdir /tmp/rpmbuild" > ~/.rpmmacros
```

- c. Run the following commands, using the kernel version of the kernel you downloaded in the previous step:

```
$ rpm -i <downloaded file name> 2>&1 | grep -v mockb
$ sudo tar xvfJ /tmp/rpmbuild/SOURCES/linux-<kernel id/name>.tar.xz -C /usr/src/kernels/
$ cd /usr/src/kernels/linux-<kernel id/name>
$ sudo make oldconfig
$ sudo make prepare
$ sudo make scripts
$ sudo make
```

3. Build the ipfw, as follows:

- a. Compile the ipfw_source_package.zip, which is part of WebLOAD installation package, by running the following commands:

```
$ cd /tmp/
$ cp
/opt/share/radview/webload<version>/linux/ipfw_source_package.zip
$ unzip ipfw_source_package.zip
$ cd dummynet-master
$ make KERNELPATH=/usr/src/kernels/linux-<kernel id/name>/
```

- b. If the build is successful, a dummynet kernel module called ipfw_mod.ko will be created in the dummynet-master/kipfw-mod directory. Try loading the module into the kernel by running the following commands:

```
$ cd kipfw-mod
$ sudo insmod ipfw_mod.ko
```

- c. If successful, install the dummynet module in the global modules repository by running the following commands:

```
$ sudo cp ipfw_mod.ko /lib/modules/<kernel id/name>/kernel/net/netfilter
```

```
$ sudo depmod
```

- d. Optionally, have it automatically loaded upon boot by running the following commands:

```
$ sudo sh -c 'echo modprobe ipfw_mod >> /etc/rc.modules'
```

```
$ sudo chmod +x /etc/rc.modules
```

Diagnostic Settings

Diagnostic options can be enabled while developing a script or for tracking problems in existing scripts.

To configure the diagnostic options:

1. Click **Default/Current Session Options** or **Script Options** in the **Tools** tab of the ribbon,

-Or-

Select **Default/Current Session Options** or **Script Options** from the Console System button.

2. Select the **Diagnostic** tab.

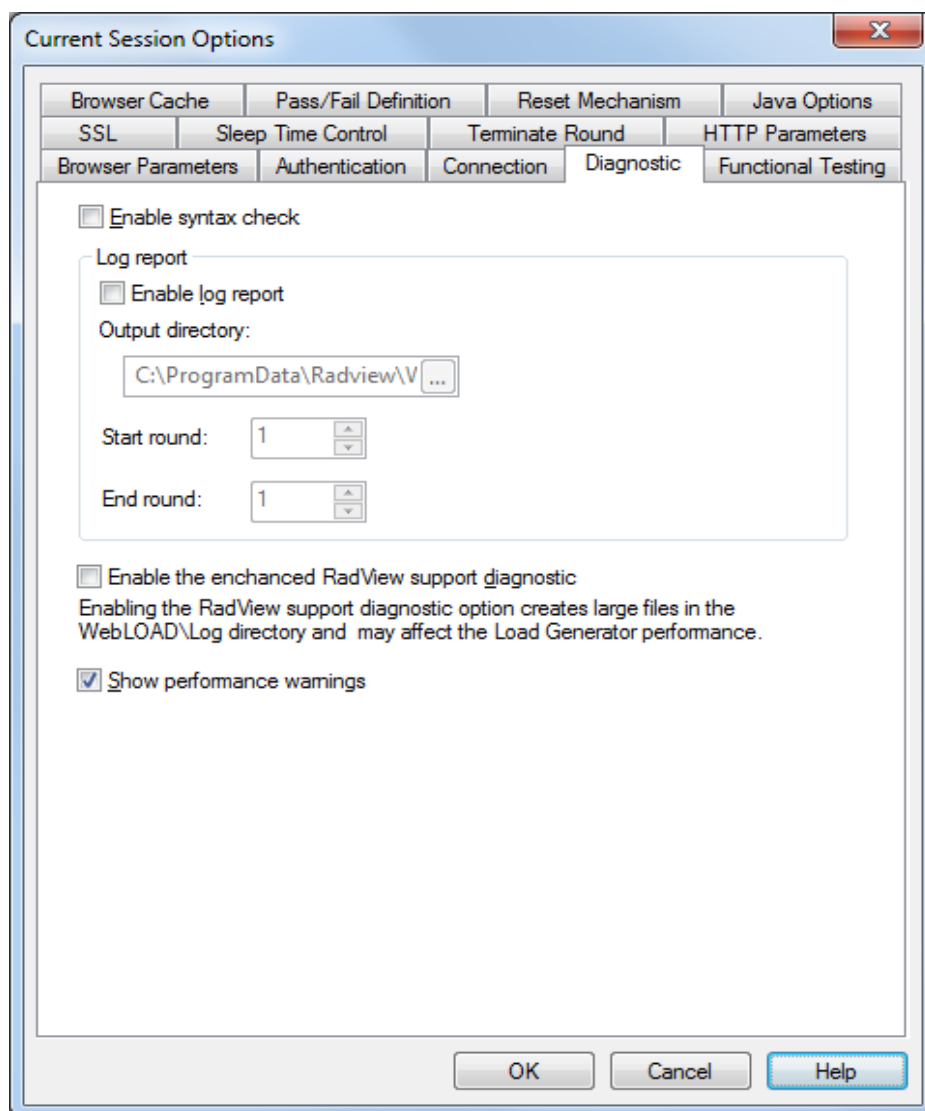


Figure 145: Current Session Options Diagnostic Tab



Note: Running a Load Session with the diagnostic options enabled may affect system performance. Therefore, we recommend that these options remain disabled while running a Load Session with multiple threads.

3. Check the diagnostic features you wish to use.

Each diagnostic option is described in this section.

4. Click **OK** to accept the modifications, or **Cancel** to reject them, and exit the Default/Current Session Options dialog box.

Enabling Syntax Checking

Enable Syntax Checking to perform the following tests on a script while it is running.

To enable syntax checking:

1. Click **Default/Current Session Options** or **Script Options** in the **Tools** tab of the ribbon,

-Or-

Select **Default/Current Session Options** or **Script Options** from the Console System button.

2. Select the **Diagnostic** tab.
3. Select **Enable syntax checking**.

The following diagnostics are performed:

- **Type inspections** – WebLOAD Console checks that each property receives the correct type.

For example, `wlLocals.ParseForms = 14` prompts the following log message:

```
"Wrong type for the property ParseForms. The correct type is Boolean. Legal values are: "Yes"/"No" or true/false."
```

- **Value inspections** – WebLOAD Console checks to ensure that each property is assigned a legal value.

For example, `wlHttp.Version = "2.1"` prompts the following log message:

```
"2.1 is an illegal value for the property Version. Legal values are: 1.0, 1.1."
```

- **Scope inspections** – WebLOAD Console checks that each property is assigned a permitted scope.

For example, `wlLocals.ConnectionSpeed = 28800` prompts the following log message:

```
"The property ConnectionSpeed is not valid for the object wlLocals."
```

- **Case inspections** – WebLOAD Console objects and properties are case sensitive. When Syntax Check is enabled, WebLOAD Console checks to ensure that all objects and properties are written correctly.

For example, `wlLocals.parse = "No"` prompts the following log message:

```
"The property parse should be written as Parse."
```

It is recommended that a syntax check be run at least once while developing your script.

Log Report Diagnostics

Log reports are used to see the script flow, and the response headers received while a script is running. When log report is enabled, an outfile is automatically created for each get and post transaction and for each frame received.

To enable log report diagnostics:

1. Click **Default/Current Session Options** or **Script Options** in the **Tools** tab of the ribbon,
-Or-
Select **Default/Current Session Options** or **Script Options** from the Console System button.
2. Select the **Diagnostic** tab.
3. Select **Enable log report** to configure WebLOAD Console to create a Log Report.

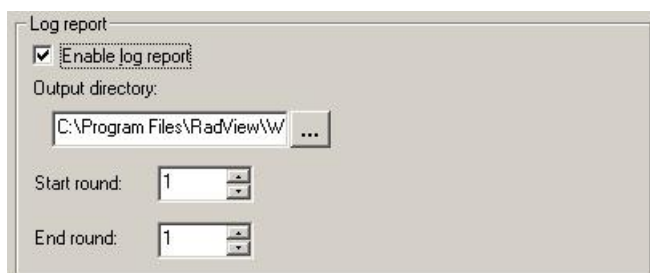


Figure 146: Enable Log Report Checkbox

4. Click in the Output directory field and select the directory for saving Log Report files.
The default directory is: C:\Program Files\RadView\WebLOAD\User\Log.
5. Select which rounds should generate the Log Report in the Start round and End round fields.

The log report displays the time a request was made, the round number, the request number, the URL received and the response headers. If `GetContents = "Yes"` the same information is displayed for each frame.

A message is sent to the Log Window for each request. The message appears in the following format:

```
### Thread:000, Round Num:001, Request Num:001 ### URL
```

Each request is numbered sequentially for each thread and round number.

Enabling enhanced RadView Support Diagnostic

Enabling the enhanced RadView Support Diagnostic option creates large files in the WebLOAD\User\Log directory and may affect Load Generator performance.

To enable enhanced RadView support diagnostic:

1. Click **Default/Current Session Options** or **Script Options** in the **Tools** tab of the ribbon,
-Or-
Select **Default/Current Session Options** or **Script Options** from the Console System button.
2. Select the **Diagnostic** tab.
3. Select **Enable the enhanced RadView Support Diagnostic**.

Enabling Show Performance Warnings

Enable the **Show Performance Warnings** option if you want WebLOAD to automatically send warning messages when Load Generators are getting overloaded. This can enable you to better utilize the Load Generators in your current and future tests.

To enable showing performance warnings:

1. Click **Default/Current Session Options** or **Script Options** in the **Tools** tab of the ribbon,
-Or-
Select **Default/Current Session Options** or **Script Options** from the Console System button.
2. Select the **Diagnostic** tab.
3. Select **Show Performance Warnings**.

Functional Testing Options

Functional testing enables you to insert verification tests into the script to verify that the links, images, forms, tables and frames in your Web application continue to act as expected.

To access and configure the functional testing options:

1. Click **Default/Current Session Options** or **Script Options** in the **Tools** tab of the ribbon,
-Or-
Select **Default/Current Session Options** or **Script Options** from the Console System button.
2. Select the **Functional Testing** tab.

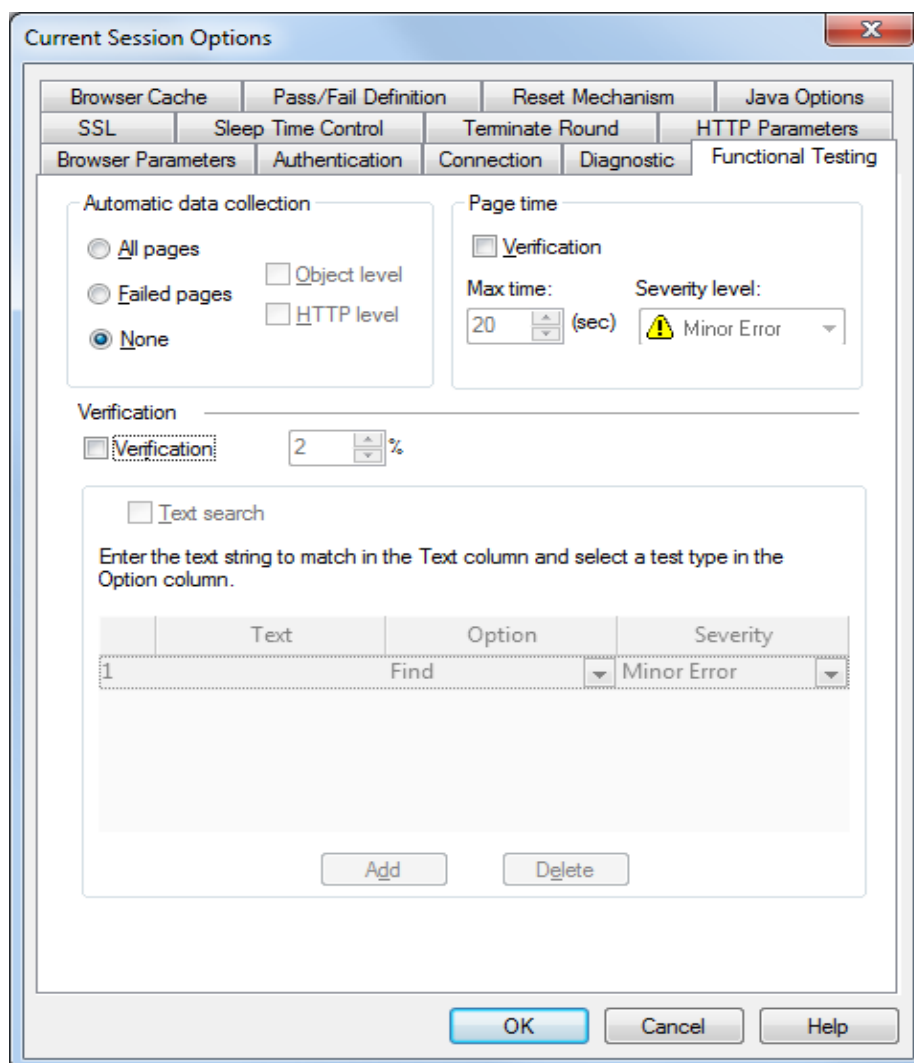


Figure 147: Current Session Options Functional Testing Tab

3. Select the functional testing tools you wish to use.

Each testing option is described in this section.

4. Click **OK** to accept the modifications, or **Cancel** to reject them, and exit the Default / Current Session Options dialog box.

Setting Automatic Data Collection

Setting automatic data collection instructs WebLOAD Console to handle every page, object and HTTP object in the Script as a transaction. The name of the transaction is the name of the page or object and data is gathered for each transaction.

To set automatic data collection:

1. Click **Default/Current Session Options** or **Script Options** in the **Tools** tab of the ribbon,

-Or-

Select **Default/Current Session Options** or **Script Options** from the Console System button.

2. Select the **Functional Testing** tab.
3. Select one of the following:
 - **All Pages** or **Failed Pages** to turn all page transactions in the script into Named Transactions. The transaction name will be the page name (the URL being accessed without the query string) with the HTTP method used to retrieve it (GET or POST). The following statistics will be gathered for all page transactions: total number of transactions (counter), number of failed and successful transactions (counter), and the page time for each transaction (timer). Transaction Data Drilling will be available for all transactions if you select All Pages or for failed transactions only if you select Failed Pages. Select one or more of the following to specify whether to collect data at object or HTTP level or both.
 - **Object level** – Creates sub transactions for all content (for example, gif, ico, and js).
 - **HTTP level** – Breaks down every transaction, including sub transactions, into HTTP elements (for example, open connection, send, time to first byte, and receive).
 - **None** to disable turning page transactions in the script into Named Transactions. Statistics are not gathered for page transactions.



Note: For the WebLOAD Console 7.x protocol engine, the default automatic data collection setting was None. For the WebLOAD Console 5.x protocol engine, the default automatic data collection setting was Failed Pages.

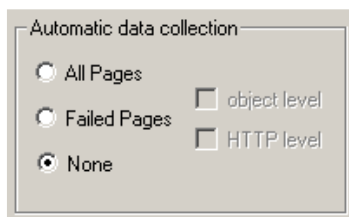


Figure 148: Pages Checkbox

Setting Page Time Verifications

Setting page time verifications enables you to define a limit to the maximum amount of time a page should take to load. If the maximum time is exceeded, a message or error entry appears in the log window, depending on the severity level set for page time verifications.



Note: You cannot set page time verifications unless automatic data collection is active. See *Setting Automatic Data Collection* on page 246.

To set page time verification:

1. Click **Default/Current Session Options** or **Script Options** in the **Tools** tab of the ribbon,
-Or-
Select **Default/Current Session Options** or **Script Options** from the Console System button.
2. Select the **Functional Testing** tab.
3. Check **Verification**.



Figure 149: Verification Checkbox

4. Specify the maximum amount of time that a page should take to load in the Max Time field.
5. Specify the severity level if the page exceeds the specified time in the Severity Level field.



Note: It is important to specify a meaningful time period. For example, if you specify 3 seconds for a page which is likely to take longer, numerous essentially meaningless failures will result. Conversely, if you specify 10 seconds for a page which should take less than 1 second, the test becomes meaningless.

Inserting a Text Verification Test

Use Text verifications to verify the absence or presence of selected text expressions within your Web page.

To insert a text verification test into your script:

1. Click **Default/Current Session Options** or **Script Options** in the **Tools** tab of the ribbon,
-Or-
Select **Default/Current Session Options** or **Script Options** from the Console System button.
2. Select the **Functional Testing** tab.
3. Under Verification, check the **Verification** checkbox.
4. Select the percentage of time the tests should be performed. This is the percentage of Virtual Clients that will perform the verification tests.



Note: For backward compatibility of scripts that were recorded with the AAT and support table comparisons, you can select only the Verification checkbox, without selecting any additional options.

5. Check the Text Search checkbox.
6. Enter a literal string with the text for which you are searching in the Text field.



Note: The text string is case sensitive.

7. Select the type of test to run from the drop-down list in the Option field:
 - **Find** – The verification test will pass if the selected text is found in the current Web page.
 - **Not to Find** – The verification test will fail if the selected text is found in the current Web page.

For example, to ensure that the word “error” does not appear on the page during runtime, enter the word “error” in the Text field, and select **Not to Find**. If the text error is found on the page, WebLOAD Console will report the error.

8. Select the severity level from the Severity level drop-down list. For further information, see *Verification Function Return Values* (on page 261).
9. Click **Add** to add another string to the list of Text Search strings,

-Or-

Click **Delete** to delete a selected string from the list of Text Search strings.

Pass / Fail Definition Settings

The Pass / Fail Definition options enable you to set test failure conditions.



Note: This option is not available for individual scripts. It can only be set for the entire session through the Default Options or the Current Session Options dialog boxes.

To configure pass/fail definition options:

1. Click **Default/Current Session Options** in the **Tools** tab of the ribbon,
-Or-
Select **Default/Current Session Options** from the Console System button.
2. Select the **Pass/Fail Definition** tab.

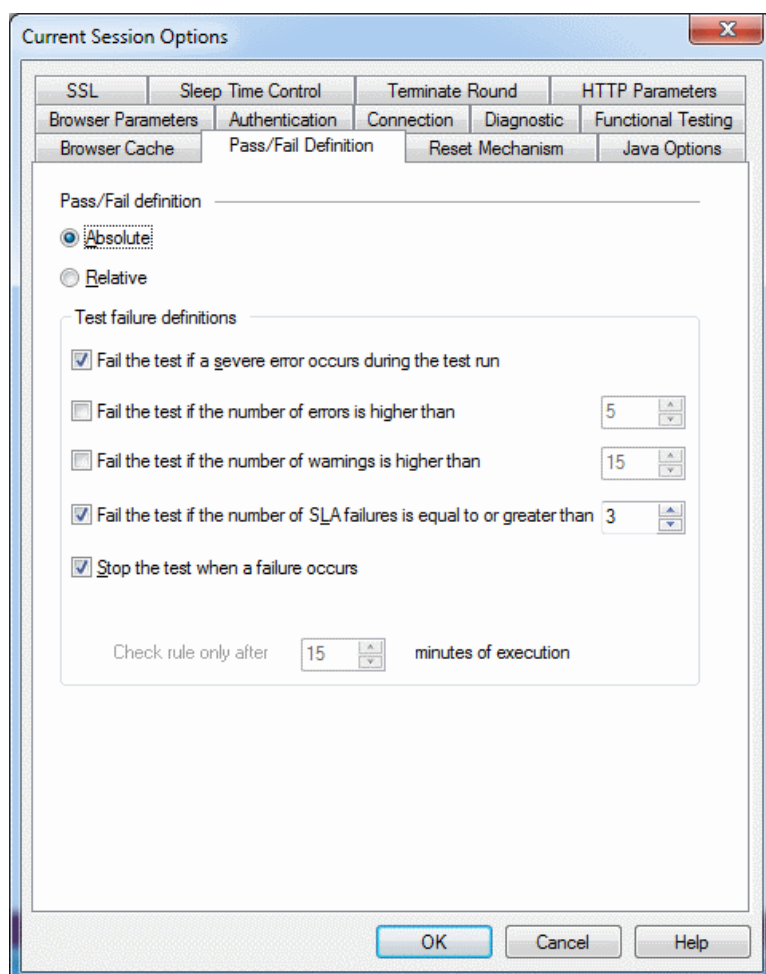


Figure 150: Current Session Options Pass/Fail Definition Tab

You can set WebLOAD Console to declare a test as failed based on a number of criteria. First, you must decide if you want the test to be professed a failure after an absolute number of failures or a relative number of failures. Then you set the conditions upon which a test is declared a failure.

3. Select **Absolute** (default) to set the total number of failures required for a test to be declared failed,
-Or-
Select **Relative** to set the percent of failed transactions need for a test to be declared failed.
4. Set the conditions upon which to declare a test failed and the action to take if a test fails.
 - **Fail the test if a severe error occurs during the test run** – Sends a failure message and declares the test failed in the event of a severe error.
 - **Fail the test if the number of errors is higher than** – Declares the test failed if the number of errors received is higher than the number stated. If the pass / fail definition is set to relative then the percentage of errors is considered rather than the absolute value.
 - **Fail the test if the number of warnings is higher than** – Declares the test failed if the number of warnings received is higher than the number stated. If the pass / fail definition is set to relative then the percentage of warnings is considered rather than the absolute value.
 - **Fail the test if the number of SLA failures is equal to or greater than** – Declares the test failed if the number of SLA breaches received is equal to or higher than the number stated. If the pass / fail definition is set to relative then the percentage of warnings is considered rather than the absolute value.
 - **Stop the test when a failure occurs** – Stops the test if the test is declared failed.
 - **Check rule only after __ minutes of execution** – This option is only enabled when the pass / fail definition is set to relative. Use this option to set a stabilizing period for the test when it starts. Errors or warning received during this time frame will not fail the test, and the rules set for failing the test will only be checked after this timeframe has expired.
5. Click **OK** to accept the modifications, or **Cancel** to reject them, and exit the Default/Current Session Options dialog box.

Configuring Reset Mechanism Settings

The Reset Mechanism options enable you to set the reset values.



Note: This option is not available for individual scripts. It can only be set for the entire session though the Default Options or the Current Session Options dialog boxes.

To configure reset mechanism options:

1. Click **Default/Current Session Options** in the **Tools** tab of the ribbon,
-Or-
Select **Default/Current Session Options** from the Console System button.
2. Select the **Reset Mechanism** tab.

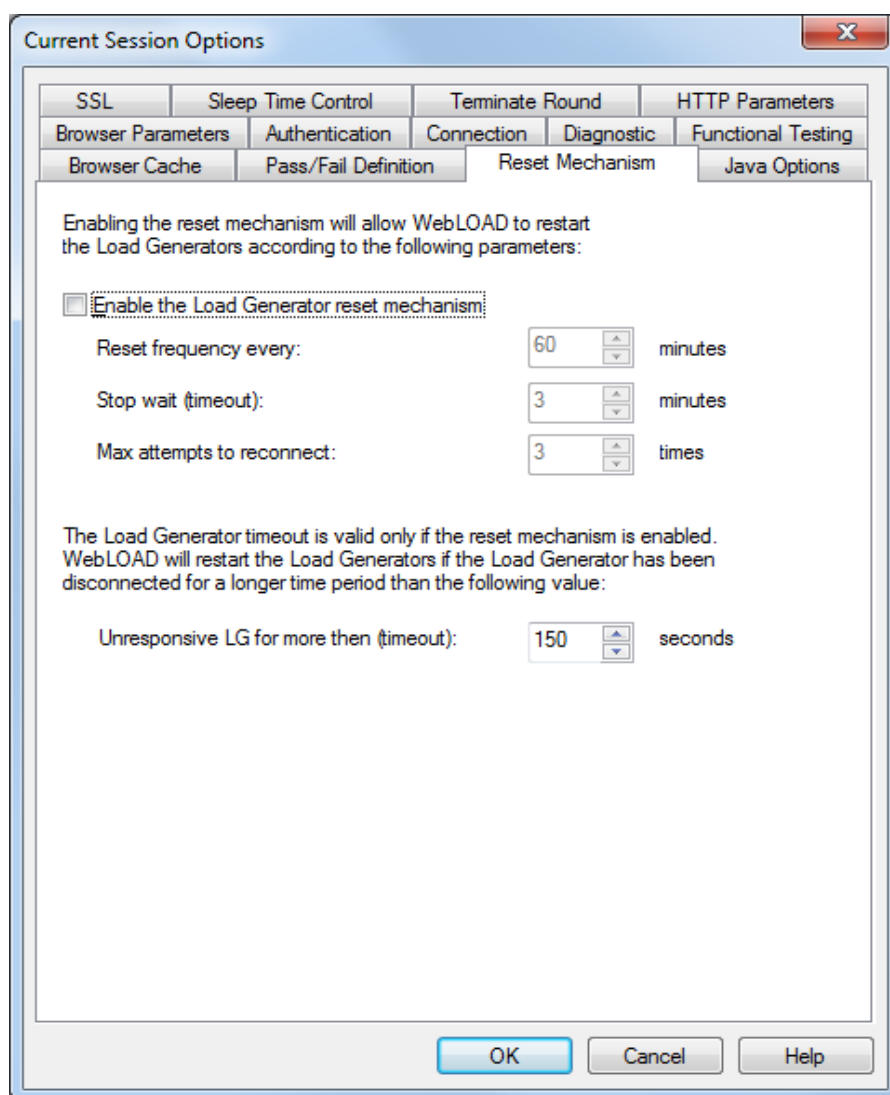


Figure 151: Current Session Options Reset Mechanism Tab

3. Check the **Enable the Load Generator reset mechanism** checkbox to allow WebLOAD Console to restart the Load Generators.
4. Set the number of minutes for Reset Frequency (Every) and Stop Wait (Timeout).
5. Set the number of times for Max attempts to reconnect.

6. Set the number of seconds for Unresponsive LG for more than (Timeout).



Note: The Load Generator Timeout is valid only if the Reset Mechanism is enabled.

7. Click **OK** to accept the modifications, or **Cancel** to reject them, and exit the Default/Current Session Options dialog box.

Configuring the Java Options

The Java options enable you to define the Java Virtual Machine to be used by WebLOAD Recorder, for executing Java classes.

To configure Java Option settings:

1. Click **Default/Current Session Options** in the **Tools** tab of the ribbon,
-Or-
Select **Default/Current Session Options** from the Console System button.
The Default/Current Session Options dialog box appears.
2. Select the **Java Options** tab.
The Java Options tab moves to the front of the dialog box.

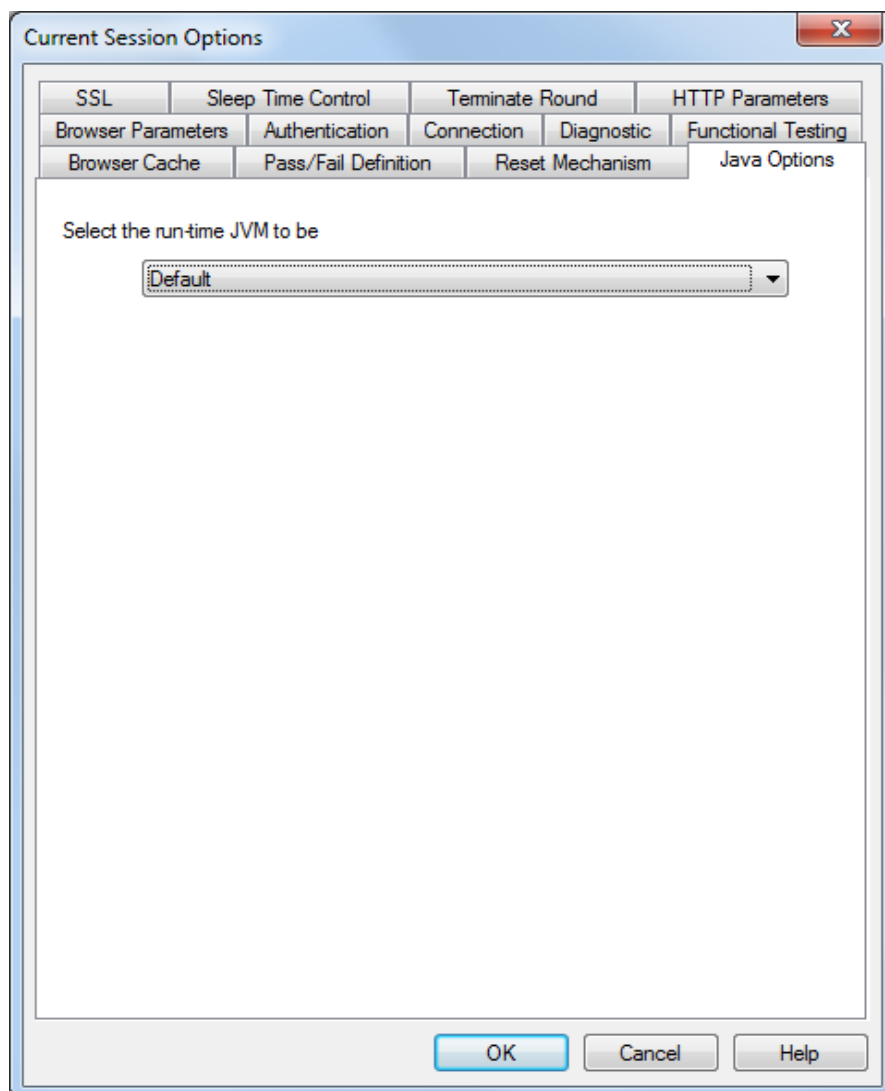


Figure 152: Current Session Options Java Options Tab

3. In the Select run-time JVM to be drop down, select one of the available Java Virtual Machines.

The default setting is the WebLOAD standard Java Virtual Machine. The selected value is passed to `wlGlobals.JVMtype` and is the key for `WLJVMs.xml`. This XML file (located on every WebLOAD Machine in the `<WebLOAD Installation Directory>\extensions\JVMs` directory) contains the following parameters for each JVM:

- Type (the value from the flag)
- Path (should be machine-agnostic)
- Options

When Type is “Default”, the RadView default (installed) JVM will be used. The default JVM’s path is defined in `webload.ini`, as it depends on the WebLOAD Console installation path.

4. Click **OK**.

The Java Options are saved.

Functional Verification Testing

WebLOAD Console combines unified load, performance, and functional testing into a single process, allowing you to verify the scalability and integrity of Web applications at the per-client, per-transaction, and per-instance level under user-defined load conditions. By creating and analyzing the Document Object Model (DOM) for every Virtual Client during a test session, WebLOAD Console is able to verify each success and failure and present detailed information to you about each transaction.

The WebLOAD Functional Verification Test Tool is an easy-to-use tool which automates the process of inserting verification tests into the script and verifies that the links, images, forms, tables and frames in your Web application continue to act as expected.

WebLOAD Console also enables you to insert application level transactions into the script and define verification criteria, which compare expected outcomes against actual ones. WebLOAD Console tracks the success and failure rates of the transactions for each Virtual Client and integrates the data into the WebLOAD Statistics Report. You can find additional information about functional testing in the *WebLOAD Recorder User's Guide*.

The Functional Testing Tab

The verification tests you specify within WebLOAD Console are performed on all Web pages in the script. To configure verification tests on all pages, set the global verification options through the **Default / Current Session Options > Functional Testing** dialog box. This section describes the options available through the Functional Testing tab.

Opening the Functional Testing Tab

To open the Functional Testing tab:

- Click **Default Options** in the **Tools** tab of the ribbon or select **Default Options** from the Console System button to set the verification tests to run for all scripts, -Or-

Click **Current Session Options** in the **Tools** tab of the ribbon or select **Current Session Options** from the Console System button to set the verification tests to run for the current session.

The Functional Testing tab opens.

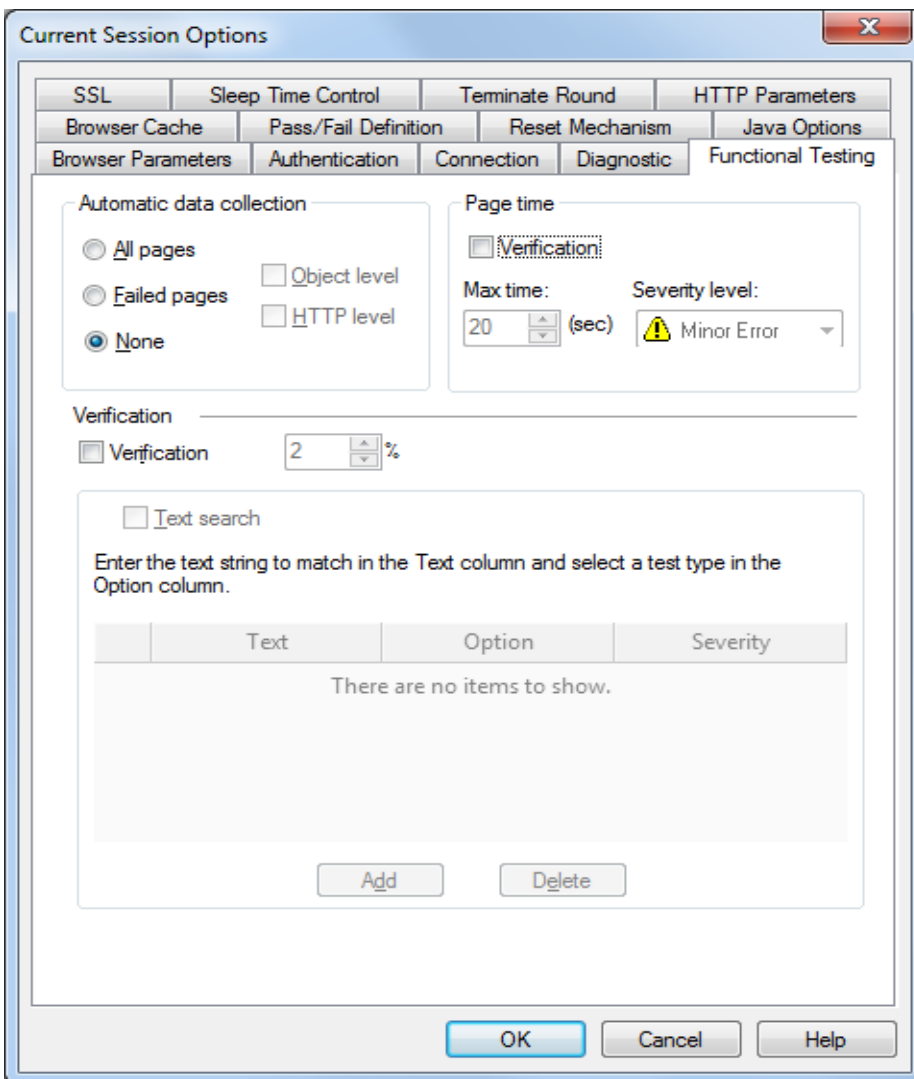


Figure 153: Current Session Options Functional Testing Tab

No global verification tests are selected by default.

The Functional Testing tab enables you to insert verification tests into the script that verify that the links, images, forms, tables, titles, and frames on your Web application continue to act as expected. You can also set tests to verify selected text and to report the default values presented on each form. Each test can be assigned a different severity level.



Note: Tests set through the Current Sessions Options dialog box override the Default Options dialog box settings.

You can choose to compare any or all of the following:

- Each Web page title.
- For each collection of forms, frames, images, and links, the URL and the number of each object within each collection, as well as the number of elements found on each form.
- The number of tables and/or structure.

You can also:

- Search for a specific user-defined string throughout the script.
- Report the default values for each element in all forms found in the script.
- Select the tests you want to run and specify the severity level of the test results, which determines how the script will continue to execute.

For information on severity levels, see *Verification Function Return Values* (on page 261).

Selecting the Tests to Perform

Once the Functional Testing tab is open, you must decide which tests you want to perform. The tests are performed on all pages in the script.

WebLOAD Console enables you to select specific tests to run.

In each case you can specify the severity level in case of test failure, which determines whether the script will continue to execute or not.

For full details, see *Verification Function Return Values* (on page 261).

Verifying Text

Use Text verifications to verify the absence or presence of selected text expressions within your Web page.

To insert a text verification test into your script:

1. Open the Functional Testing tab, as described in *Opening the Functional Testing Tab* (on page 256).
2. Select the **Verification** checkbox.
3. Select the percentage of time the tests should be performed. This is the percentage of Virtual Clients that will perform the verification tests.
4. Select the checkbox adjacent to the Text Search test.



Note: Selecting a lower level checkbox selects and highlights the upper level checkbox.

5. Select the severity level from the drop-down list next to the test. For further information, see *Verification Function Return Values* (on page 261).
6. Enter a literal string with the text for which you are searching in the Text field.



Notes:

The text string is case sensitive.

A search on the specified text is performed throughout the entire HTML, including the tags.

7. Select the type of test to run from the drop-down list in the Option field:
 - **Find** – The verification test will pass if the selected text is found in the current Web page.
 - **Not to Find** – The verification test will fail if the selected text is found in the current Web page.

For example, to ensure that the word `error` does not appear on the page during runtime, enter the word `error` in the Text field, and select **Not to Find**. If the text error is found on the page, WebLOAD Console will report the error.

8. Select the severity level from the Severity drop-down list. For further information, see *Verification Function Return Values* (on page 261).
9. Click **Add** to add another string to the list of Text Search strings.
10. Click **Delete** to delete a selected string from the list of Text Search strings.
11. Click **OK**.

User-Defined Transactions and Verification

Using WebLOAD Console, you can define application-level transaction and verification criteria to determine whether the transaction succeeded or failed. When a user-defined transaction executes, WebLOAD Console initiates a timer and a counter. The timer measures the time required to execute the transaction. The counter tracks the success and the failure rate of the transactions, based on user-defined criteria. For each transaction, the counter automatically increments either the “successful transaction count” or the “failed transaction count.” WebLOAD Console also lets you define the behavior of the program depending on the transaction results.

WebLOAD Console tracks all of the results from user-defined transactions and integrates the data into the WebLOAD Statistics Report, including the timer and counter statistics as well as statistics on specific failures. The results are then automatically displayed with the other WebLOAD Console statistical information.

Including a User-Defined Transaction in your script

Application-level transactions and verifications are defined through the WebLOAD Recorder script editing functions. See the *WebLOAD Recorder User's Guide* for more information.

User-Defined Transactions and Verification Syntax

```
BeginTransaction("<Transaction-Name>")
    <any valid JavaScript statements>
EndTransaction("<Transaction-Name>",
               <Verification-Expression>, <Save-Flag>)
```

Where:

- `Transaction-Name` is a user-supplied string.

- `Verification-Function` (optional) is the name of the function to be called that evaluates the transaction severity level (`MinorError`, `Error`, `SevereError` or `Custom Function`). If no such expression is present, the default value is `Success`.
- `save-flag` is an optional Boolean flag specifying whether WebLOAD Console should save only the results of problematic transaction instance that triggered an error (default), or save all transaction instances.

User-Defined Transaction Example

The following example illustrates implementing a user-defined transaction:

```
BeginTransaction("UpdateBankAccount")
    /* The user-defined transaction "UpdateBankAccount" starts*/
    try {
    wlHttp.ExpectNavigation("http://...")
    wlHttp.Navigate("www.....com")
    wlHttp.SyncDOM(1);
    }
    /* The body of the transaction*/
    <any valid JavaScript statement>

EndTransaction("UpdateBankAccount",
               UpdateBankAccount_VerificationFunction())
function UpdateBankAccount_VerificationFunction()
{
    return WLSuccess;
}
/* The user-defined transaction "UpdateBankAccount" ends*/
```



Note: The user-supplied `UpdateBankAccount_VerificationFunction()` is called to determine whether the transaction succeeded or failed. This function can be defined either in the script or in an included file.

If during the test the above transaction succeeded 15 times and failed 5 times, and it took an average of 1.2 seconds for the transaction to execute, the Statistics Report displays as follows:

Table 35: Statistics Report Sample Output

Total UpdateBankAccount transactions	20
Successful UpdateBankAccount transactions	15
Failed UpdateBankAccount transactions	5
UpdateBankAccount timer	1.2 sec



Note: The transaction timer measures the time it took to execute the code between the `BeginTransaction` statement and the `EndTransaction` statement.

Verification Function Return Values

You define the pass/fail criterion for user-defined transactions. For each verification test you must also specify the severity level of a failure. The severity level determines the execution path when the main script resumes control. Thus, depending on the severity level, you can program the test to ignore the failure, jump to the next navigation block, or stop the test. Extreme failures with high severity levels can be set to abort the test, whereas less severe failures can be ignored.

You set the severity level in the return statement of the verification function. The assigned severity levels and their results are handled in the same manner as other test failure results in WebLOAD Console. All failures are logged and are displayed in the WebLOAD Console log window. Refer to Appendix C: *WebLOAD Console Error Messages* (on page 443) for more details on severity levels.

Transactions may be assigned one of the following return values:

Table 36: Transaction Return Values

Severity Level	WebLOAD Behavior
Success	The transaction completed successfully.
MinorError	The specific transaction failed, but the script continues to execute from the following statement.
Error	The specific transaction failed. The script continues from the next navigation block.
SevereError	The specific transaction failed. The Load Generator on which the error occurred is stopped.



Note: If a return value (`return()`) is not specified, the default value is `Success`, indicating that the transaction completed successfully.

Reporting Events

WebLOAD Console enables you to record specific events and transaction instances as they occur. This information is very useful when analyzing website performance with Data Drilling, described in *Data Drilling* (on page 343).

Record transaction events with the following syntax:

```
ReportEvent ("<event-name>", ["<description>"])
```

Where:

- `event-name` is a string that identifies the specific event.
- `description` is an optional string providing more information about the specific event.

Tracking Reasons for Transaction Failure

WebLOAD Console enables you to specify possible reasons for the failure of a given transaction. WebLOAD Console tracks the reasons for failure individually, in the Statistics Reports. The reasons for the transaction failure must be specified in the verification function.

The default reason for failure of basic transactions is General-Failure. Unless another reason for failure is specified, this is set automatically whenever the transaction fails.

Set the reason for a transaction failure using the following syntax:

```
SetFailureReason ("<failure-reason>")
```


For example:

Continuing the `UpdateBankAccount` transaction example described in *User-Defined Transactions and Verification* (see *User-Defined Transactions and Verification Syntax* on page 259), the `UpdateBankAccount` verification function might appear as follows:

```
function UpdateBankAccount_VerificationFunction()
}
    VerifyUpdateBankAccount () {
        if <Condition-success> then
            return(WLSuccess) }
        else
            if (document.URL == "../no-login") then {
                SetFailureReason("User Not Logged")
                return(WLError) }
            else
                if (document.URL == "...wrong- password") then {
                    SetFailureReason("Wrong Password")
                    return(WLError); }
                else
                    return(WLSevereError)
            }
    }
```

Explanation:

- `Return(WLSuccess)` – indicates that the actual transaction succeeded and it is added to the `SuccessfulUpdateBankAccount` transactions counter.
- The `SetFailureReason()` function accepts a string as a parameter. This string is used to identify the cause of the transaction failure (“User Not Logged”, “Wrong Password”, etc.). This data enables tracking of the number of failures that occur for a specific reason, as well as the time that the failures occurred.
- When the reason for failure is “User Not Logged” or “Wrong Password”, the severity level is Error. Consequently, when the main script resumes control the current round aborts and a new round commences.
- If you mark the transaction as failed, but do not specify any failure reason, the system registers a “General Failure” which is the default failure severity.

If during the execution of a test the above transaction succeeded 15 times and failed 5, and the `SetFailureReason()` function was called 3 times with the string “Wrong Password” and 2 times with the string “User Not Logged”, the Statistics Report appears as follows:

Table 37: Test Result Example

Test	Time
Total UpdateBankAccount transactions	20
Successful UpdateBankAccount transactions	15
Failed UpdateBankAccount transactions	5
Failed UpdateBankAccount transactions: Wrong Password	3
Failed UpdateBankAccount transactions: User Not Logged	2
UpdateBankAccount timer	1.2 sec

All of the selected tests are performed on all pages in the script when you run a WebLOAD Console test session. Verification test failures are automatically added to the Data Drilling reports. All verification test failures are also displayed in the Event Viewer that highlights the “actual” results, described in *Viewing Verification Test Results* (on page 264).

Viewing Verification Test Results

After running your script, WebLOAD Console provides information on all major events that occurred during runtime such as verification failures and error messages.

When a verification test fails:

- The failure is logged in the Log Window, described in *Viewing Error Information in the Log Window* (on page 265).
- The failure is added to the Data Drilling report, described in *Data Drilling* (on page 267).

All verification failures can be viewed in the Event Viewer, which displays the “actual” results, and a description of the error. The Event View can be accessed through either the Log Window or the Data Drilling reports, as described here.

Viewing Error Information in the Log Window

If an error occurs at any time during the test session, an error message displays in the Log Window. A browser error or a Minor Error does not cause the test session to stop. An Error stops the current round. A Severe Error terminates the session upon completion of the current navigation block.

A description of each error is listed in the Log Window. You can select whether to display or hide the error log using the **Log Window** checkbox in the **Session** tab of the window.

!	@	TIME	GENERATOR NAME	SCRIPT NAME	CLIENT NUMBER	MESSAGE	TIME STAMP
!	@	193.19	10.0.1.186	Test	0	500 Internal server error. Requested URL: http://10.0.1.65/qasite/dynamic/imgOnly.asp.	10/29/2018 3:24:52 PM
!	@	194.21	10.0.1.186	Test	0	500 Internal server error. Requested URL: http://10.0.1.65/qasite/dynamic/imgOnly.asp.	10/29/2018 3:24:53 PM
!	@	195.23	10.0.1.186	Test	0	500 Internal server error. Requested URL: http://10.0.1.65/qasite/dynamic/imgOnly.asp.	10/29/2018 3:24:54 PM
!	@	196.25	10.0.1.186	Test	0	500 Internal server error. Requested URL: http://10.0.1.65/qasite/dynamic/imgOnly.asp.	10/29/2018 3:24:55 PM
!	@	197.26	10.0.1.186	Test	0	500 Internal server error. Requested URL: http://10.0.1.65/qasite/dynamic/imgOnly.asp.	10/29/2018 3:24:56 PM
!	@	198.29	10.0.1.186	Test	0	500 Internal server error. Requested URL: http://10.0.1.65/qasite/dynamic/imgOnly.asp.	10/29/2018 3:24:57 PM
!	@	199.30	10.0.1.186	Test	0	500 Internal server error. Requested URL: http://10.0.1.65/qasite/dynamic/imgOnly.asp.	10/29/2018 3:24:58 PM
!	@	200.32	10.0.1.186	Test	0	500 Internal server error. Requested URL: http://10.0.1.65/qasite/dynamic/imgOnly.asp.	10/29/2018 3:24:59 PM
!	@	201.34	10.0.1.186	Test	0	500 Internal server error. Requested URL: http://10.0.1.65/qasite/dynamic/imgOnly.asp.	10/29/2018 3:25:00 PM

DONE NUM ELAPSED TIME: 00:03:21 REMAINING TIME: N/A

Figure 154: Log Window

The Log Window displays the following additional information about your test session results:

- **Message Status** – The result and severity of each message.
Messages fall into four categories:
 - Success / Info Message
 - Minor Error
 - Error
 - Severe Error
- **Attachments** – Double-click the paperclip adjacent to the message to view additional information.
- **Time** – The duration of the action specified in the Message column.
- **Generator Name** – The generator that originated the message.
- **Script Name** – The script that originated the message.
- **Client Number** – The number of the client that originated the message.
- **Message** – The runtime action and information about failed actions.
- **Timestamp** – The date and time on which the action specified in the Message column is completed.

To view more information on an error:

- Double-click the paperclip adjacent to the verification failure in the Log Window. The Event Viewer opens.



Note: The Event Viewer is only available after the test session is complete.

Viewing Error Information in the Event Viewer

When you double-click a verification failure in the Log Window, the Event viewer opens. The Event Viewer displays the Actual views of the object that triggered the failure, with the object highlighted. A description of any errors that occurred during the test run appears below.

Time	Generator Name	Agenda Name	Client Number	Message
0.00				Load Session Setup - completed.
0.00				Session start time:Thu Jun 27 01:37:33 2013
3.83	localhost	facebook_login		Functional Test - Text search verification failed in window https://www.facebook.com/login.php?login_attempt=1
6.24	localhost	facebook_login		Functional Test - Text search verification failed in window https://www.facebook.com/login.php?login_attempt=1
8.97	localhost	facebook_login		Functional Test - Text search verification failed in window https://www.facebook.com/login.php?login_attempt=1
12.09				Test passed

Figure 155: Event Viewer

Double-click the description below the viewer to highlight any changes relating to the error.

Note that if you integrated WebLOAD with Dynatrace, a **Dynatrace** button appears in the Event Viewer window. Clicking the button opens a Dynatrace window displaying the relevant transaction. Refer to *Viewing in Dynatrace the Transactions Related to WebLOAD Errors* (on page 423).

Printing the Contents of the Log Window

To print the contents of the log window:

1. Right-click inside the log window.
2. Select **Print** from the right-click menu.
The Print Setup dialog appears.
3. Select a printer and click **OK**.

Saving the Contents of the Log Window

To save the contents of the log window:

1. Right-click inside the log window.
2. Select **Save** from the right-click menu.
The Save As dialog appears.
3. In the File Name field, type in the name for the file.
4. Click **Save**.

The file is saved with the extension * .mdb.

You can view the saved log file using Microsoft Access or Excel.

Data Drilling

When testing the integrity of your Web application, it is important to see not only if a given transaction succeeded or failed, but also the reasons for the failure to pinpoint the transaction's exact weak points and bottlenecks. Data Drilling enables you to display a detailed description of all user-defined and named transactions to the instance level.

With Data Drilling you can view all transaction failures and the reasons for them. Data Drilling reports provide a complete breakdown of all information about a selected event. Click an event entry to bring up detailed information about that event in the Event Viewer. The Event Viewer provides a graphic illustration of the verification failure event, displaying the "actual" results, and a description of the error.

Data Drilling is described in detail in *Data Drilling* (on page 343).



Note: Data Drilling is only available to you if:

- You initiated user-defined transactions in the scripting for functional testing.
- You configured Automatic data collection in the **Functional Testing** tab of the Script Options dialog box or Default/Current Session Options dialog boxes.

Running Tests and Analyzing Test Results

This chapter introduces you to WebLOAD Console's performance reports.

Explanation of Creation and Analysis of Performance Reports

Performance Reports are the product of your WebLOAD testing. WebLOAD Console collects all of the data for a wide variety of measurements. You can view and configure these reports through the Console while tests are running, or after the Load Session is completed.

This material is divided into the following chapters:

- *Running a Load Session* (on page 271) provides instructions for running a WebLOAD test session.
- *Viewing the Log Window* (on page 283) describes the WebLOAD Log window.
- *WebLOAD Console Performance Reports* (on page 295) provides instructions for creating and viewing WebLOAD Integrated Reports.
- *WebLOAD Statistics Reports* (on page 317) describes the statistics reports provided by WebLOAD.
- *Data Drilling* (on page 343) describes how to view all transactions failures and the reasons for them.
- *Exporting Performance Reports* (on page 351) describes how to export all of the data displayed in the Integrated Reports and Statistics to HTML, Microsoft Excel and to Tab files.
- *Performance Measurements Manager* (on page 363) describes how to configure the Performance Measurements Manager to collect data from different sources during the test.

After completing this material, you will have the skills necessary to create and analyze all of the performance reports provided by WebLOAD.

Running a Load Session

After configuring a Load Session, you can run your Load Session. This chapter describes the procedure for running your Load Session, and the way WebLOAD Console behaves while a Load Session is in progress.

WebLOAD Console collects a complete set of test data while a Load Session is running. You can configure reports that include only the data needed. Reports can be configured during or after a test.

WebLOAD Console run commands can start, pause, and stop your Load Session while it is in progress. These commands are detailed in this chapter.

Starting a Load Session

You can start a Load Session in one of the following ways:

- Click the **Start Session** icon in the Quick Access toolbar.
- Click **Start Session** in the **Session** tab of the ribbon.
- From the WebLOAD Wizard - Finish dialog box, select the checkbox to run the Load Session immediately and click **Finish**.
- From the Goal-Oriented Test - Finish dialog box, select the checkbox to run the Load Session immediately and click **Finish**.

Running a Load Session from the System Memory

You can run a test from the configuration parameters in your system memory, without saving the Load Session. If you start running a Load Session without saving the Load Session configuration, you are prompted to save the Load Template.

The following message box appears.

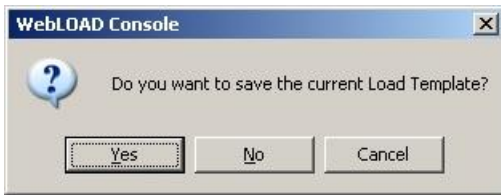


Figure 156: Save Current Load Template Message Box

Saving the Current Load Template

To save the current Load Template:

- Click **Yes**.

The test session definition is saved to a Load Template (*.tpl) file.

Launching WebLOAD Console through the Command Line Interface

You can perform load testing through a command line interface. You can enter the WebLOAD Console launch command into a batch file or into an external script and WebLOAD Console will run directly, without user intervention, using the parameters specified.



Note: You can also perform unattended WebLOAD Console testing at prescribed times by running WebLOAD in Jenkins. Jenkins CI is a Continuous Integration server that is used by Agile development teams to run their builds, tests and deployments as well as their performance tests if possible. Jenkins is extendable by means of plugins. As of WebLOAD 10.1, a WebLOAD plugin is available for Jenkins. For more information, refer to the *WebLOAD Automation User Guide*

To run WebLOAD Console through the command line interface:

Enter the `WebLOAD.exe` command together with a series of optional parameters (described below) into your external script to automatically launch a WebLOAD test. When your script runs, the executable file will invoke WebLOAD Console and run the specified test according to the specified parameters.



Note: Verify that the script used with the template specified, and any included files, are accessible to the Load Template or Load Session file that will be run.

Syntax

Use the following syntax to define the parameters for running a WebLOAD test through a command line interface:

```
WebLOAD.exe [<file name to open>][<file name to save>]
[<flags>][<time>][/ag <script name>]
```

Parameters

When running a test invoked by the executable, you can specify the following parameters:

Table 38: Test Parameters

Parameter	Description	Comments
File name to open	The name of the *.tpl or *.ls file (Load Template or Load Session file) to run.	Optional parameter
File name to save	The name of the *.ls file containing the test data. This file will be saved in the current directory unless otherwise specified.	Optional parameter
Flags	<ul style="list-style-type: none"> /ar – Automatically run the WebLOAD test without waiting for user input. If this flag is not specified, the Console is called up with the specified Load Template/Load Session but the test is not automatically run. The system waits for user input. /ar <time> – Automatically run the test for the length of time specified in <time>/ /vc – The number of Virtual Client licenses to allocate when using WebRM License Server. /pc – The number of Probing Client licenses to allocate when using WebRM License Server. /rc <results_file_name> – Place the results in the specified file (an XML file). 	Optional parameter
/ag <script name>	The name of an existing script (*.wlp) to open.	Optional parameter

The parameters are all optional. If no parameters are entered, the executable launches the Console and does not run a test. If the autorun flag </ar> flag is not set, the <file name to save> and the <time> parameters are ignored.



Note: If there is a conflict between the time defined in this command and the time defined in the WebLOAD Scheduler, the load test runs for the shorter of the two periods defined.

Examples

Example 1:

```
WebLOAD.exe test1.tpl
```

This command opens the Console and the `test1.tpl` template. The Console waits for user input.

Example 2:

```
WebLOAD.exe test1.tpl march9.ls /ar 30
```

This command opens the Console and automatically runs a test using the `test1.tpl` template file. The test results are saved in the Load Session file `march9.ls`, which includes all of the test data and results. This file is saved in the current directory, unless otherwise specified. The `autorun` flag is set, meaning that the test runs without user intervention. The test will run for 30 seconds.

Example 3:

```
WebLOAD.exe /ag c:\scripts\MyScript.wlp
```

This command opens the Console and the WebLOAD Wizard to the script/Mix Selection dialog box. The `MyScript.wlp` script is automatically selected and the WebLOAD Wizard waits for user input. For more information about the script/Mix Selection dialog box, see *Selecting a* (on page 103).

Example 4:

```
WebLOAD.exe test1.tpl march9.ls /ar 30 /rc result1.xml
```

This command performs all the actions described in *Example 2* above, and in addition the execution return code is saved in `result1.xml`.

Example 5:

```
WebLOAD.exe test1.tpl march9.ls /ar 30 /vc 100 /pc 3
```

This command performs all the actions described in *Example 2* above, and in addition it allocates 100 virtual clients and 3 probing clients from the WebRM server.

Establishing Communication

WebLOAD Console begins executing a test by verifying the test parameters and attempting to communicate with the hosts participating in the test session.

While WebLOAD Console is preparing to run the test, the following message appears:

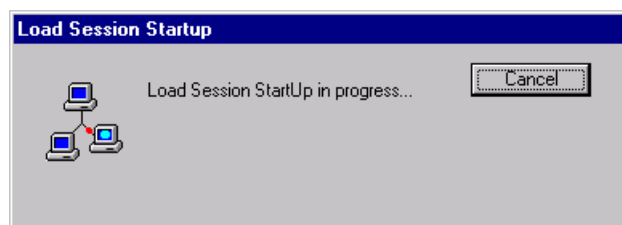


Figure 157: Load Session Startup Message Box

Errors in Communication

The following errors may occur during Load Session startup:

- All hosts are unreachable or stopped
- Load Session setup not completed
- Load Session setup timed out

Correcting the All Hosts are Unreachable or Stopped Error

When the All hosts are unreachable or stopped error occurs and the session cannot be started, the following error message appears:

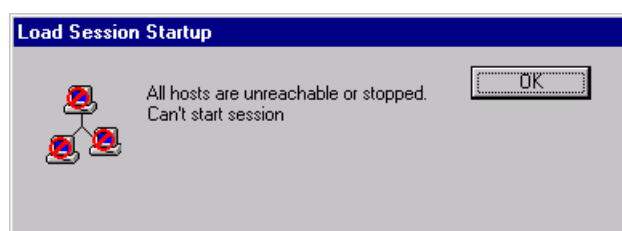


Figure 158: All Hosts Are Unreachable or Stopped Error Message Box

To correct the all hosts are unreachable or stopped error:

1. Click **OK**.
2. Test your connection. Verify that the host systems participating in the test are all up and running. If hosts are still unreachable, inform your system administrator.

Correcting the Load Session Setup Not Completed Error

When only some of the requested hosts are active, the “Load Session Setup not completed” error message appears as follows:

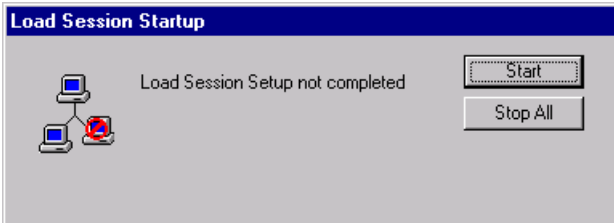


Figure 159: Load Session Setup Not Completed Error Message Box

To correct the Load Session Setup not completed error:

- Click **Start** to begin testing with Hosts that have been reached,
-Or-
Click **Stop All** to stop testing.

Correcting the Load Session Setup Timed Out Error

When the Load Session setup has timed out, the following error message appears:

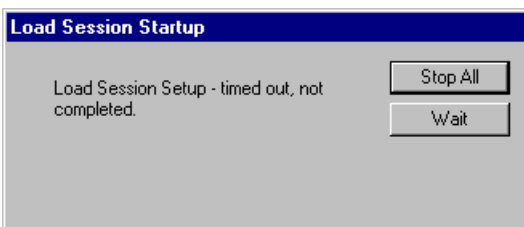


Figure 160: Load Session Setup Timed Out Error Message Box

To correct the Load Session Setup timed out error:

- Click **Stop All** to stop testing,
-Or-
Click **Wait** to give WebLOAD Console more time to connect to the unreachable hosts. WebLOAD Console allows another interval of time equal to the timeout interval, for the systems to connect.

WebLOAD Console has a built-in start session timeout value, which is the length of time WebLOAD Console waits while TestTalk attempts to make contact with the hosts. You can change the timeout values through the **Global Options** ► **General** tab, see *The General Tab* (on page 179) for details.

The Console Screen in Session Mode

When you start running your Load Session, the Results window appears.

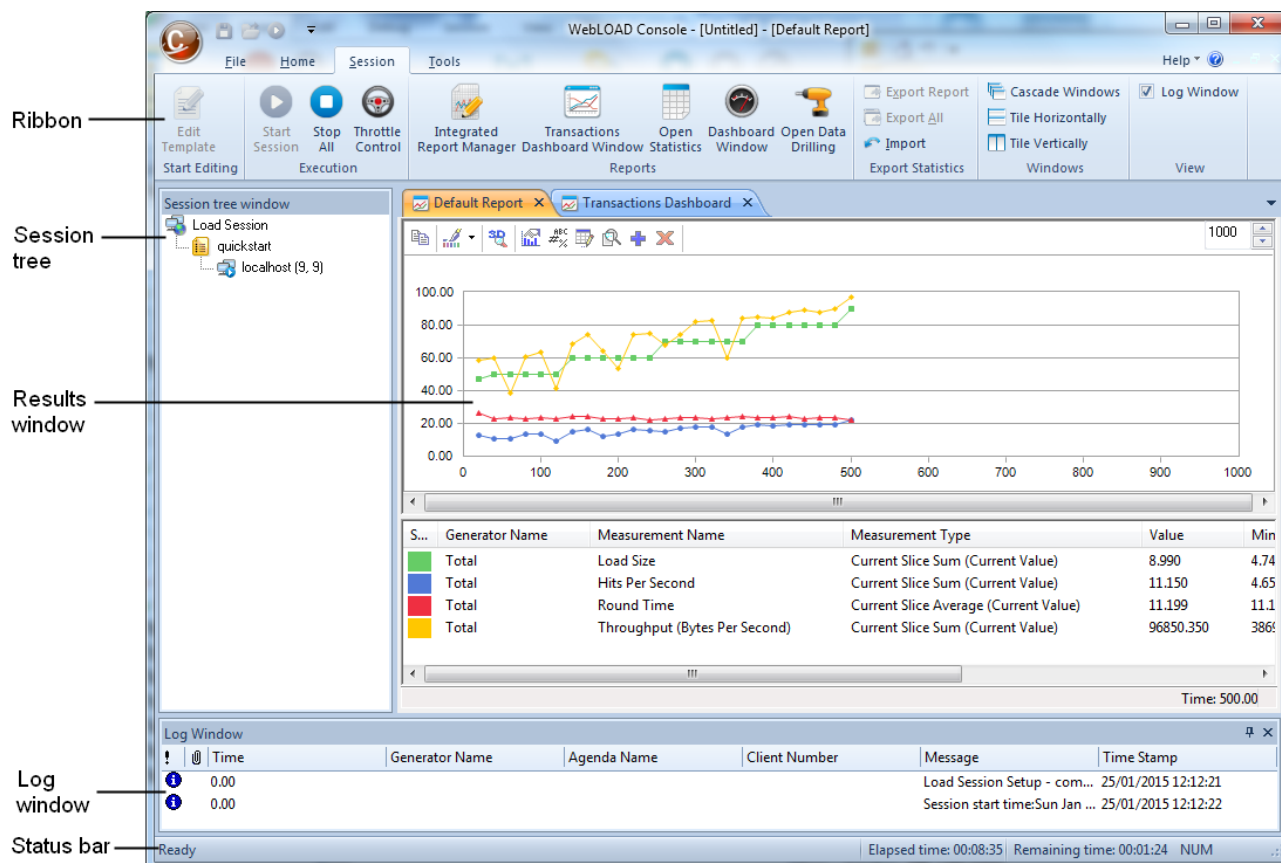


Figure 161: Results Window

The following table describes the segments of the Console in Session Mode:

Table 39: Session Mode Console Segments

Segment	Function
Session Tree	Presents a graphic display of your test session. See <i>Viewing the Session Tree in Session Mode</i> (on page 278) for a complete discussion of the Session Tree.
Results Window	Displays all of the reports opened during a test session. Use the tabs located at the top and the bottom of the window to view different reports.
Log Window	Displays all of the error messages recorded during a test session. You can toggle the Log Window display on/off through the Session tab of the Console ribbon.

Segment	Function
Status Bar	Indicates the program status, including continually updated information about the Elapsed time since the session started running, and the Remaining time till session end (if known).

Viewing the Session Tree in Session Mode

The Session Tree displays the complete configuration of the current Load Session. Status icons reflect the current status of your Load Session.

Through the Session Tree you can see:

- The scripts running
- The hosts running each script
- The current operating status of each host and script

Icons display on each line of the tree, making it easy to view your test activity.

While the Load Session is running, status icons reflect the current status for each script. The status icons are described in the above table.

To the right of each script two numbers are displayed in parentheses, for example Script1[45, 75]. These display the maximum Load Size (number of Virtual Clients) that script is scheduled to generate, followed by the Load Size currently being generated. The current Load Size number changes according to the Load Size schedule that is defined in the Load Generator Schedule dialog box. When a test session has not yet begun, the 'current' Load Size being generated is 0. In most modes of operation, the current Load Size never exceeds the maximum Load Size scheduled for that script listed to the left of the current value.



Note: Using Throttle Control, you can increase the current Load Size to exceed the maximum Load Size – but only if you are not using the Goal-Oriented Test.

WebLOAD Console Operating Commands


WebLOAD Console offers five commands for controlling a test session:

- Start
- Pause
- Resume
- Stop
- Stop All

You access them through the **Session** tab of the Console ribbon, or the Session Tree pop-up menus.

Starting a Test Session

To start a test session:

- Click the **Start Session**  button in the Quick Access toolbar,
-Or-
Select **Start Session** in the **Session** tab of the ribbon.

Pausing a Test Session

To pause a test session:

- Right-click a component in the test session Tree and select **Pause** from the pop-up menu.

When you pause a script it temporarily stops the execution of that script. The remainder of the components in the current test session (other scripts running simultaneously) will continue running as scheduled. If the root is selected, all components are paused.

When a paused script is continued (using the Go or Resume command), the script skips any actions scheduled for the duration of the pause. Script execution continues with whatever load was scheduled for the current time.

Pausing a Host that is generating a heavy load causes an immediate jump in system response time because the system load drops immediately.

Resuming a Test Session

Resuming resumes execution of a paused Load Session component. Keep in mind that pausing and continuing does not automatically reschedule Load Session Scripts. Scheduling continues in relation to the start of the session.

To resume a test session:

- Right-click a component in the test session Tree and select **Resume**.

Resuming a Host that is generating a heavy load causes an immediate drop in system response time.

Stopping a Load Session Component

To stop a Load Session component:

- Right-click a component in the test session Tree and select **Stop**,

The selected component will stop. Once an individual component is stopped, it cannot be restarted unless the whole Load Session is restarted.

You may stop some of the Load Session components without affecting the remaining ones.

Stopping a Load Session component affects system performance by causing a sudden drop in system load, which leads to a corresponding jump in system performance.

Stopping a Test Session

To stop a test session:

- Select **Stop All** in the **Session** tab of the ribbon,

-Or-

Right-click a component in the Load Session Tree and select **Stop All**.

Stopping a complete Load Session affects system performance because the system load immediately drops to zero.

Manually Shutting Down Cloud Machines

You can manually shut-down Amazon Cloud Machines still running. This is relevant if WebLOAD is configured to leave Cloud Machines running, or if you chose to leave them running at the end of a Load Session.



Note: The data residing on an Amazon Cloud Machine is not accessible once the machine is shut down.

For more information about working with Cloud Machines, refer to *Selecting Host Computers* on page 107.

For information about changing the shut-down policy of Cloud Machines, refer to *Creating WebLOAD Cloud Accounts* on page 109.

To shut down Cloud Machines:

- Click **Terminate Cloud Machines** in the **Tools** tab of the ribbon. The Cloud Machines are shut down.



Note: After manually shutting down Cloud Machines, verify on the Amazon Management Console that no machines are left running.

Throttle Control – Changing the Load Size Dynamically During Runtime

Using Throttle Control utility, you can override the test schedule and dynamically control the Load Size during runtime. You can select a component (from the Session Tree) and modify Load Size – the number of Virtual Clients participating in the test.

The selected component can be:

- The root of the Session Tree
- A Load Machine

When using Throttle Control consider the following:

- You *cannot* modify a script using Throttle Control.
- You *cannot* use Throttle Control with the Goal-Oriented Test.
- You *cannot* use Throttle Control to create a load of more than 20 users in Standalone Workstation mode.
- If you select the root from the Session Tree and you are testing with more than one Load Machine, WebLOAD Console distributes the load among the Load Machines. During this process, the original schedule is “suspended” for the duration of Throttle Control.

Opening Throttle Control

To open Throttle Control:

- Select **Throttle Control** from the **Session** tab of the ribbon,

-Or-

Right-click a Session component and select **Throttle Control** from the pop-up menu.

The Throttle Control dialog box appears.

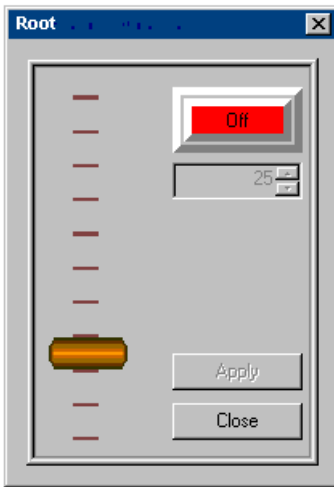


Figure 162: Throttle Control Dialog Box

The Throttle Control dialog box displays:

- The current Load Size.
- The currently selected component. (In the figure above, the selected component is the Root of the Session Tree.)

Activating Throttle Control

To activate Throttle Control:

1. Toggle the **Off** button to **On**.
2. Adjust the Throttle by moving the lever up or down and set the desired Load Size.
3. Click **Apply**.

The Load Size changes dynamically while the test is running.

Deactivating Throttle Control

To deactivate Throttle Control:

1. Toggle the **On** button to **Off**.
2. Click **Close**.

The Throttle Control dialog box is closed. Once Throttle Control is turned off, the Load Session resumes the manual schedule and continues running from the point at which Throttle Control was deactivated.

Working with Git

Git is a free and open source distributed version control system for tracking changes in computer files and coordinating work on those files among multiple people.

WebLOAD supports basic Git operations, with easy access to Git operations from within WebLOAD Recorder and WebLOAD Console. WebLOAD assumes that you have downloaded and installed Git, and are familiar with the way it works.

Prerequisites to working with Git

As prerequisites to working with Git, you need to:

1. Specify once in WebLOAD how to access the Git Local and Remote Repositories. Refer to *Setting up Access to Git Local and Remote Repositories* (on page 284).
2. Make sure your WebLOAD files are saved in the directory defined as the Git local repository. This can be done in one of two ways:
 - Define the Git local repository directory to be the WebLOAD default directory.
 - Change the WebLOAD default directory to be the directory of the Git local repository.

To do either of the above, you need to access the window where you can view or change the location of the WebLOAD default directory:

- In the WebLOAD Console, navigate to **Global Options > File Locations**.
- In the WebLOAD Recorder, navigate to **Settings > File Locations**.

Note that changing the WebLOAD default directory in either the Console UI or the WebLOAD Recorder UI, changes it for both.

Setting up Access to Git Local and Remote Repositories

As a prerequisite to using Git operation from within WebLOAD, you need to once specify in WebLOAD how to access the Git Local Repository and Remote Repository.



Note: You can perform this prerequisite task either from the WebLOAD WebLOAD Recorder, or from the WebLOAD Console. Once you perform it in one of them, Git operations will be accessible from both.

To specify the Git local and remote repositories:

1. Click **Repository Settings** in the **Tools** tab of the ribbon.

The WebLOAD Repository Settings window appears.

WebLOAD Repository Settings

Local Repository

Repository Browse...

Current branch

Remote Repository

Repository URL

Remote branch

User Name

Password

OK Cancel

Figure 163: WebLOAD Repository Settings

2. In the **Local Repository** section:
 - a. Specify the location of the local repository.
 - b. Specify the **current branch**.
3. In the **Remote Repository** section:
 - a. Specify the location of the remote repository (either a URL or a directory).
 - b. Specify the **remote branch**.
 - c. Specify the **User Name** and **Password** for accessing the remote repository.
4. Click **OK**.

Performing a Commit action

If you setup WebLOAD to support Git, you can perform a Commit, that is, save to the Git local repository, any WebLOAD objects you are currently working on. This functionality is available in both the WebLOAD Recorder and the WebLOAD Console.

- When you select Commit in the WebLOAD Recorder, the script you are currently working on as well as all its additional information, are saved to the Git local directory.
- When you select Commit in the WebLOAD Console, the test definition of the test you are currently working on, as well as the session created by test execution, are saved to the Git local directory

To Commit a WebLOAD object:

1. Click **Commit** in the **Tools** tab of the ribbon.

A window appears in which you can enter a Commit comment.

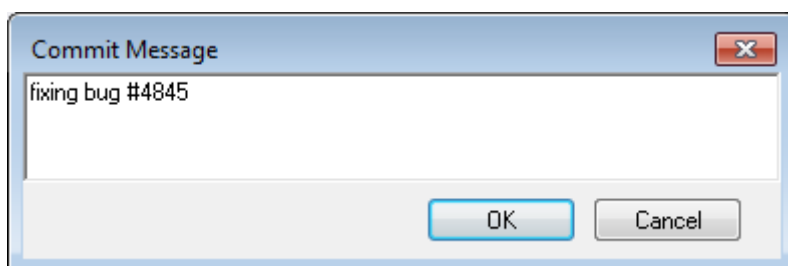


Figure 164: Entering a Commit Comment

2. Optionally enter a comment, and click **OK**.

Performing a Commit Dir action

If you setup WebLOAD to support Git, you perform a Commit dir, that is, save to the Git local repository an entire folder (with all its descendants). This folder must be located under the local repository folder. This functionality is available in both the WebLOAD Recorder and the WebLOAD Console.

To perform a Commit dir:

1. Click **Commit dir** in the **Tools** tab of the ribbon.
2. In the File Explorer window that appears, specify the directory you wish to commit.

A window appears in which you can enter a Commit dir comment.

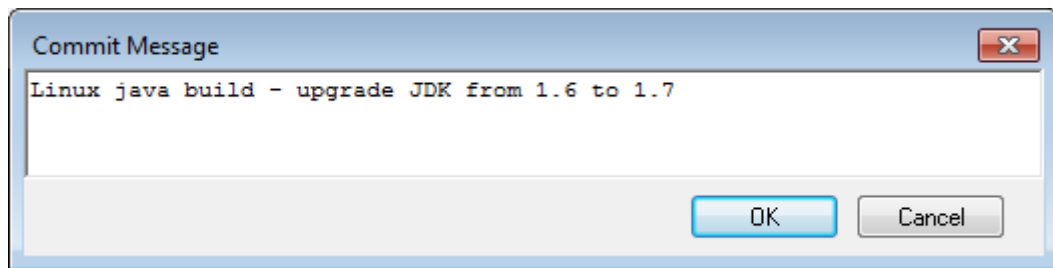


Figure 165: Entering a Commit Dir Comment

3. Optionally enter a comment, and click **OK**.

Performing a Push action

If you setup WebLOAD to support Git, you can perform a Push, which causes all Commits you had made to the Git local repository, to be pushed to the Git remote repository. This functionality is available in both the WebLOAD Recorder and the WebLOAD Console.

To perform a Push:

1. Click **Push** in the **Tools** tab of the ribbon.

A window appears, informing you of the Git Push operation and its success status.

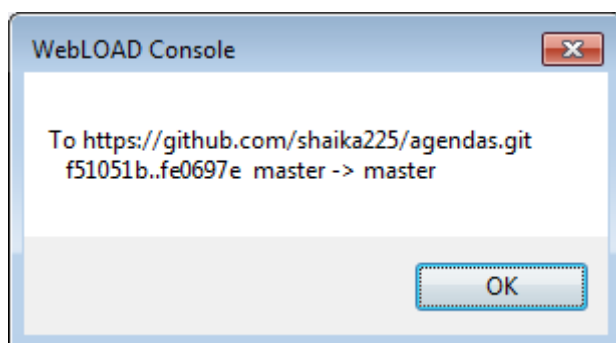


Figure 166: Git Push Operation Message

2. Click **OK**.
3. If the Push operation is rejected, for example because a conflict has arisen due to changes made by two different users, a window appears, displaying the Git rejection message. Follow the instructions in the message, and resolve the rejection using your chosen Git UI.

Performing a Pull action

If you setup WebLOAD to support Git, you can perform a Pull, which pulls data from the remote repository to the local repository. This functionality is available in both the WebLOAD Recorder and the WebLOAD Console.

A Pull operation pulls the entire branch from the remote repository and updates the local repository and directories.

To perform a Pull:

1. Click **Pull** in the **Tools** tab of the ribbon.

A window appears, informing you of the Git Pull operation and its success status.

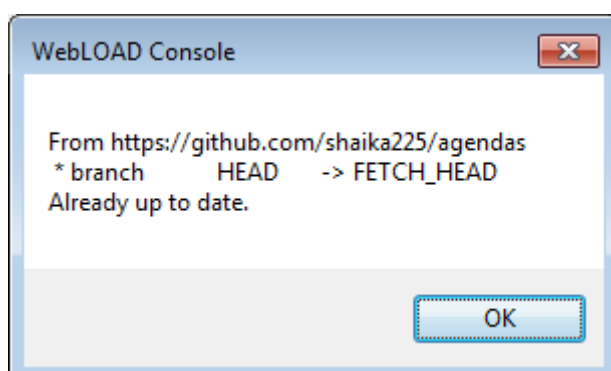


Figure 167: Git Pull Operation Message

2. Click **OK**.

Opening the Git UI

If you setup WebLOAD to support Git, you can launch your Git UI from within WebLOAD. This functionality is available in both the WebLOAD Recorder and the WebLOAD Console.

To launch the Git UI:

1. Click **Open Git Gui** in the **Tools** tab of the ribbon.

The Git UI is launched.

Viewing the Log Window

The Log Window displays log messages detected by WebLOAD Console in run time, in addition to messages generated by the JavaScript compiler and any user messages programmed in the script.

Setting the Log Window Options

Before the Load Session begins, you can set the options that affect the way the data is displayed in the Log Window.

To set the log window options:

1. Click **Global Options** in the **Tools** tab of the ribbon,
-Or-
Select **Global Options** from the Console System button.
2. Select the **General** tab.
3. Click the following radio buttons:
 - **Stop Session after n log message**
 - Stop sending log messages after n log messages
4. Click **OK**.

The following table describes the log window options:

Table 40: Log Window Options

Field Name	Description
Stop Session after n log messages	<p>Stop the Load Session after a defined number of log messages have been received.</p> <p>Once the maximum defined number of log messages is received, a message box displays stating that the Load Session is stopped.</p>

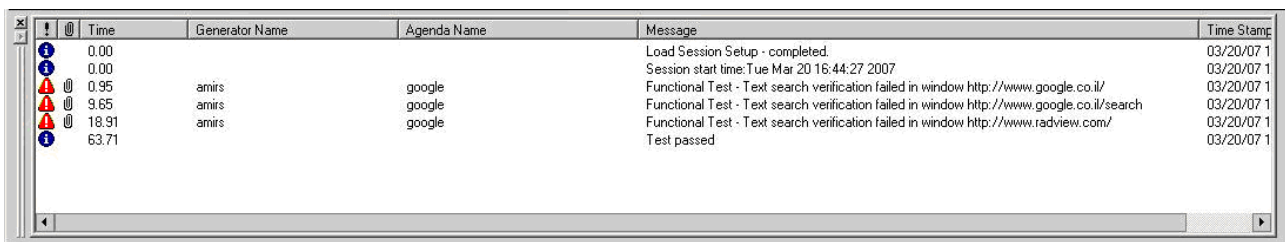
Field Name	Description
Stop sending log messages after n log messages	<p>Stop displaying log messages in the log window after a defined number of log messages have been received.</p> <p>Once the maximum defined number of log messages is received, a message box displays stating that no more log messages are displayed.</p>

Opening the Log Window

To open the log window:

- Check the **Log Window** checkbox in the **Session** tab of the ribbon.

The Log Window appears. The Log window is both floating and dockable, and can be placed in any part of the screen.



Time	Generator Name	Agenda Name	Message	Time Stamp
0.00			Load Session Setup - completed.	03/20/07 1
0.00			Session start time: Tue Mar 20 16:44:27 2007	03/20/07 1
0.95	amirs	google	Functional Test - Text search verification failed in window http://www.google.co.il/	03/20/07 1
9.65	amirs	google	Functional Test - Text search verification failed in window http://www.google.co.il/search	03/20/07 1
18.91	amirs	google	Functional Test - Text search verification failed in window http://www.radview.com/	03/20/07 1
63.71			Test passed	03/20/07 1

Figure 168: Log Window

The Log Window displays:

- A list of the log messages received during the test session.
- The status of each log message.
- A paper-clip symbol which when you double-click it opens the Event Viewer.
- The time each log message was generated (in seconds) from the start of the current Load Session.
- The Load Generator and script for which each log message is generated.

Viewing the Entire Log Message

To view the entire log message:

- Double-click any of the messages displayed in the Log Window.

A Log Message message box appears displaying the complete message. The text displayed in the Log Message message box can be selected and copied to a text editor.

Sorting Error Messages

To sort error messages:

- Click the column heading.
The first time you click a column heading, the error messages list in ascending order. A second click the column heading reverses the order. For example, to sort the error messages by Script Name, click the Script Name column heading.

Opening the Log Window Menu

To open the log window menu:

- Right-click inside the Log Window.
The Log Window pop-up menu appears.

The following table describes the Log Window menu commands:

Table 41: Log Window Menu Commands

Command	Description
Allow Docking	Enables/disables moving the Log Window.
Hide	Does not display the Log Window.
Clear	Erases data in the Log Window.
Show last message	Displays the final messages in the logged in the Log Window.
Print	Prints the data in the Log Window.
Save / Save As	Saves the data in the Log Window.
Report Defect	Opens the Submit Defect dialog box with the selected log message displayed in the Description field.
Float in Main Window	Floats the Log Window in the main WebLOAD Console window.

Printing Error Messages Recorded in the Log Window

To print error messages recorded in the log window:

- Right-click inside the Log Window.
The Log Window pop-up menu appears.
- Click **Print**.

Saving Error Messages Recorded in the Log Window

To save error messages recorded in the log window:

1. Right-click inside the Log Window.

The Log Window pop-up menu appears.

2. Select **Save** or **Save As**.

The Windows Save As dialog box appears.

3. Type the name of the file in the File Name field.

4. Click **Save**.

The file is saved as a * .mdb file. The file can be opened using Microsoft Access or Excel.

Viewing the Message Status

Messages fall into four categories:

- Success / Info Message
- Minor Error
- Error
- Severe Error

The log messages include messages generated by the Console, the JavaScript compiler and any human errors or user messages detected by WebLOAD Console at run-time. You can also add log messages to the script.

Success / Info Message

Success / Info Messages are added to the script by the user. Success / Info Messages can be used in order to facilitate debugging.

Success / Info Messages do not stop an active Load Session.

Minor Error

A Minor Error indicates a suspicious condition. The source of the problem may be generated by the system under test, the connection to the server, networking, etc.

Minor Errors do not stop an active Load Session.

Error

When an Error is detected WebLOAD Console will stop the round and restart the script for the Virtual Client that encountered the error.

Errors do not stop an active Load Session.

Severe Error

Severe Errors cause the Load Generator, on which the error was generated, to stop immediately.

An error message indicating that the process was stopped is displayed.

Clearing the Log Window

You can clear the log window by right-clicking in the Log Window, and selecting **Clear** from the right-click menu.

WebLOAD Console Performance Reports

The WebLOAD Dashboard and WebLOAD Transactions Dashboard display a summary of performance statistics generated during the test session. Integrated Reports enable you to configure a report that combines the following:

- Performance data from the Load Machines and Probing Clients
- Performance data on the SUT, gathered from Performance Monitors
- Existing performance data from a previously saved Load Session with data from the currently running Load Session

Combining several different measurements into a single report gives you a more complete picture of your Web application's performance.

Viewing Test Results on the WebLOAD Dashboard

The WebLOAD Dashboard displays real-time statistical information about the test session for the following categories:

Table 42: Test Session Information Categories

Category	Description
Time	The time during the test that the values displayed on the Dashboard were sampled.
Running Virtual Clients	The total number of Virtual Clients currently running.
Rounds Executed	The total number of times the script has been executed by all Virtual Clients.
Failed Rounds	The total number of rounds with errors.
Failed Hits	The total number of hits with errors.

Category	Description
Hits Per Second	The number of HTTP requests for information (pages, individual images, objects, or frames) made by Virtual Clients to the SUT per second.
Calculated Hits Per Day	The number of HTTP request for information made by Virtual Clients to the SUT per day.
Pages Per Second	The number of upper level request for information made by a Virtual Client to the system under test (SUT) per second.
Calculated Pages Per Day	The number of upper level request for information made by a Virtual Client to the system under test (SUT) per second.
Throughput Per Second	The average number of bytes per second transmitted from the SUT to the Virtual Clients running this script.
Calculated Throughput Per Day	The average number of bytes per day transmitted from the SUT to the Virtual Clients running this script.

To open the Dashboard:

- Click **Dashboard Window** in the **Session** tab of the ribbon.

Viewing Test Results on the WebLOAD Transactions Dashboard

The WebLOAD Transactions Dashboard displays real-time statistical information about the transactions in a test session, in a line graph. Separate lines track each transaction. For additional information about graphs, see *Viewing Integrated Reports* (on page 310).

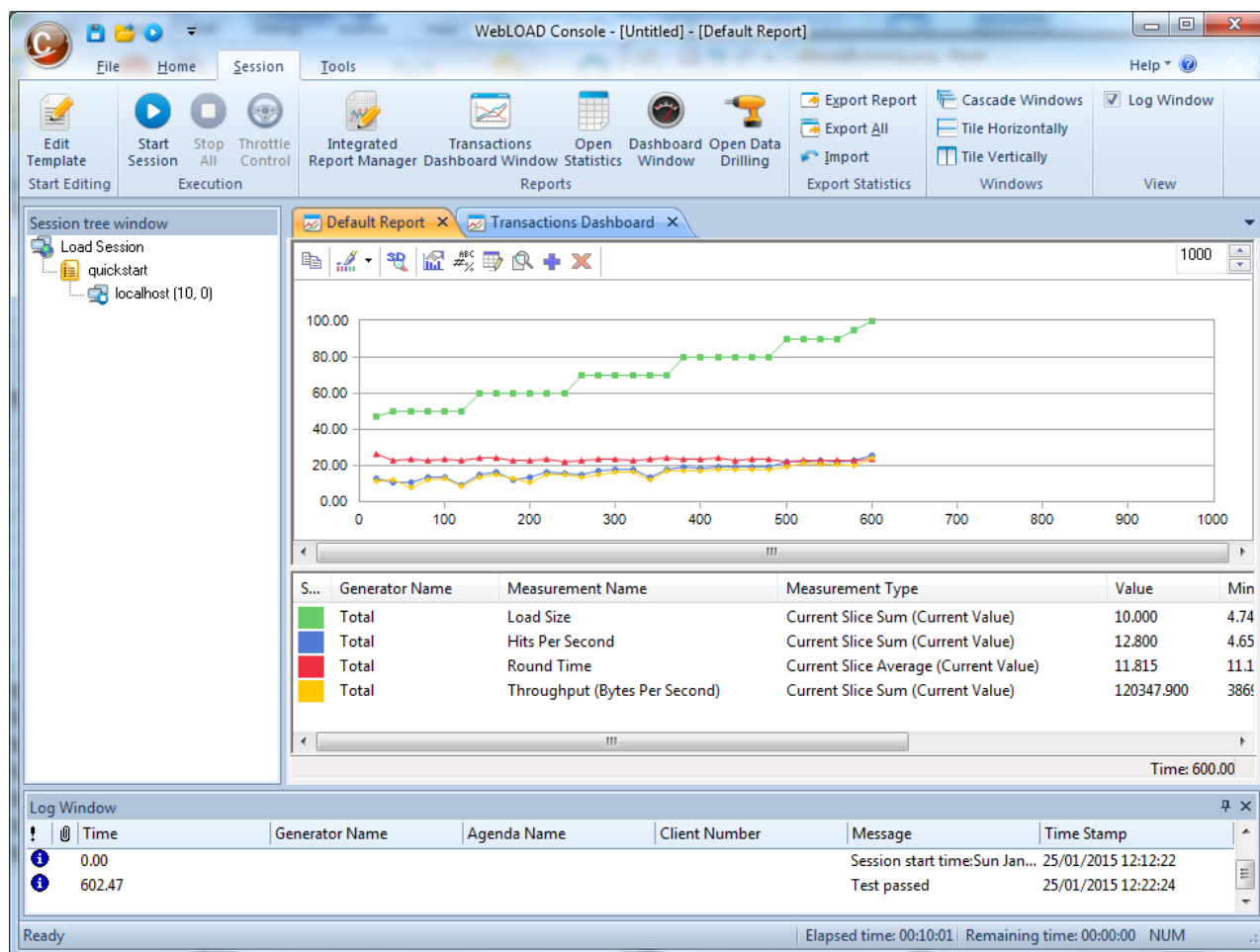


Figure 169: WebLOAD Transactions Dashboard

Table 43: Test Session Information Categories

Category	Description
Transaction Name	The name of the transaction
Value	The last (current) value of the transaction response time.
Total Min	The lowest value reported for this transaction's response time since the beginning of the test.

Category	Description
Total Max	The highest value reported for this transaction's response time since the beginning of the test.
Total Average	The average value of the transaction response time.
Total Successful Count	The number of successful transactions.
Successful TPS	The number of successful transactions per second.
Stdev	The average amount this statistic varies from the average number in this script.
Scale	The scale of the graph's Y-Axis. For information about changing the Scale value, see <i>Changing the Scale of a Report for a Selected Measurement</i> (on page 312).

To open the Transactions Dashboard:

- Click **Transactions Dashboard Window** in the **Session** tab of the ribbon.

Opening Reports

To open reports:

1. Click **Integrated Report Manager** in the **Session** tab of the ribbon.
The Integrated Report dialog box appears.

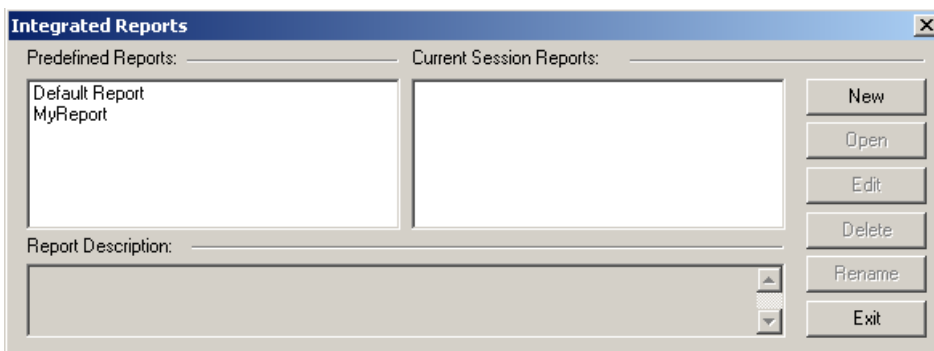


Figure 170: Integrated Reports Dialog Box

2. Click **New**,
-Or-
Select a report and click **Edit**.
The Report dialog box appears.

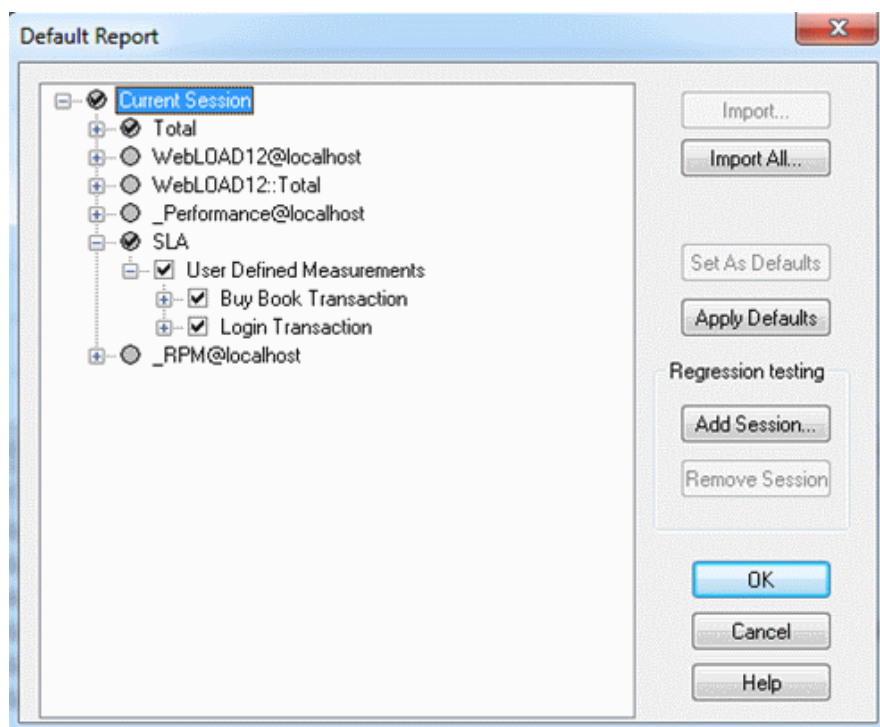


Figure 171: Report Dialog Box

Report data is organized in a tree structure. Report fields are organized into different “report tree branches”, illustrated in the Report Dialog Box figure.

Report tree branches include:

- **Total** – Data collected for the session as a whole, including total load size, total number of rounds, and total number of pages.
- **ScriptName@LoadGenerator** – Data collected for all instances of the named script, running under the named Load Generator. Multiple branches may appear for a single script name, each one linked to a different Load Generator.
- **ScriptName: Total** – Data collected for all instances of the named script, running under all Load Generators participating in this test session.
- **_Performance@LoadGenerator** – Data collected for assessing Load Generator performance during this test session.
- **SLA** – Data collected for all SLA rules defined in the SLA Manager (see *The SLA Manager* on page 204).





Using the Report dialog box you can:

- Configure the items displayed in the test report.
- Rename the report.
- Import an existing report configuration.
- Set a default report configuration.

- Apply a default report configuration.
- Configure a report that compares existing performance data from a previously saved Load Session with data from the currently running Load Session.

The following table describes the buttons in the Report dialog box.

Table 44: Report Dialog Box Fields

Button	Functionality
Import	Imports a report configuration for the selected component in the configuration tree, from a previously saved Load Template. See <i>Importing a Report Configuration for a Selected Component</i> (on page 306).  Note: Select an upper level component in the configuration tree to enable this button.
Import All	Imports a report configuration from a previously saved Load Template, see <i>Importing a Complete Report Configuration</i> (on page 306).
Set as Defaults	Sets the current Integrated Report configuration as the default. This overrides the built-in default configuration.  Note: Select an upper level component in the configuration tree to enable this button.
Apply Defaults	Applies the default configuration. The defaults are set using the Set as Default option. Selecting this option configures the Integrated Reports to the defaults.  Note: Select an upper level component in the configuration tree to enable this button.
<i>Regression Testing</i>	
Add Session	Imports performance data from a previously saved (existing) Load Session, see <i>Importing Report Parameters from a Saved Load Template</i> (on page 306).
Remove Session	Removes the selected Load Session from the performance data available for display in the report.  Note: Select an upper level component in the configuration tree to enable this button.
OK	Confirms the selections made in this dialog box. The selected items are included in the Integrated Report.
Cancel	Closes this dialog box without accepting changes.
Help	Opens the online-help dialog box.

For information on selecting the measurements included in an Integrated Report, see *Selecting Measurements for the Integrated Report* (on page 304).

Opening an Existing Report

To open an existing report:

1. Click **Integrated Report Manager** in the **Session** tab of the ribbon.

The Integrated Reports dialog box appears.

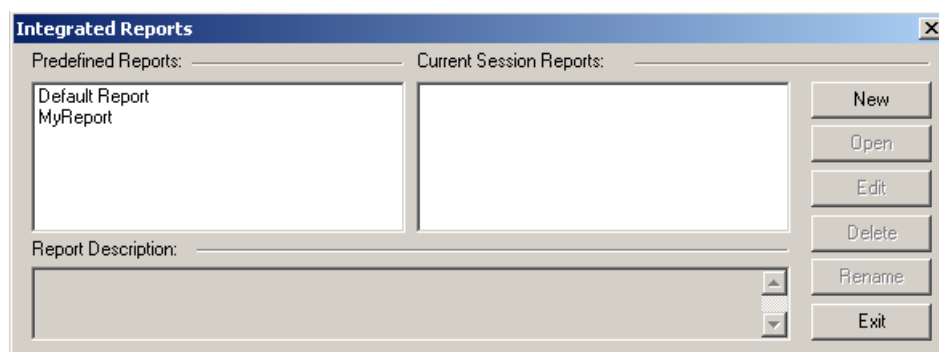



Figure 172: Integrated Reports Dialog Box

2. Select a report.

The Open button is enabled and a description of the measurements included in the report appears at the bottom of the dialog box.

3. Click **Open**.

The Integrated Report displays in the Console Results Window. WebLOAD Console displays the results of your test in a graphical format. Click the **Data Editor**  toolbar button at the top of the Results Window to display your test results in a tabular format.

Editing a Report

To edit a report:

1. Click **Integrated Report Manager** in the **Session** tab of the ribbon.

The Integrated Reports dialog box appears.

2. Select a report and click **Edit**.

The Report dialog box appears.

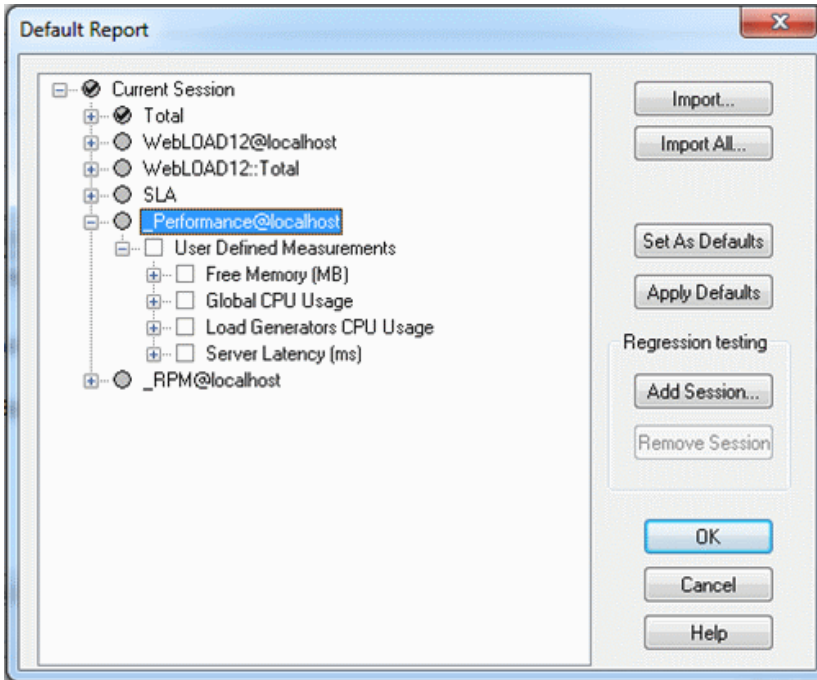


Figure 173: Report Dialog Box

3. Make the necessary changes, and click **OK**.

Renaming a Report

You can assign a meaningful name to a report. The default name for reports is “ReportX” where X is a number that is incremented with each generated report.

To rename a report:

1. Click **Integrated Report Manager** in the **Session** tab of the ribbon.

The Integrated Reports dialog box appears.

2. Select a report to rename.

The Rename button is enabled and a description of the measurements included in the report appears at the bottom of the dialog box.

3. Click **Rename**.

The Rename Report dialog box appears.

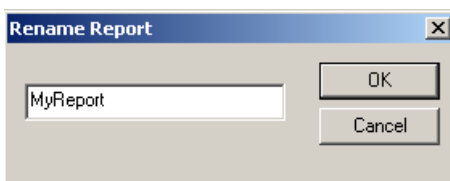


Figure 174: Rename Report Dialog Box

4. Type a new name for the report.
5. Click **OK**.
The selected report is renamed.

Deleting a Report

To delete a report:

1. Click **Integrated Report Manager** in the **Session** tab of the ribbon.
The Integrated Reports dialog box appears.
2. Select a report to delete.
A description of the measurements included in the report appears at the bottom of the dialog box.
3. Click **OK**.
The selected report is deleted.

Configuring an Integrated Report

You can configure an Integrated Report while the test is running. To view data from the WebLOAD Performance Monitor in your Integrated Report, see *Performance Measurements Manager* (on page 363).

To configure an integrated report:

1. Click **Integrated Report Manager** in the **Session** tab of the ribbon.
The Integrated Reports dialog box appears.

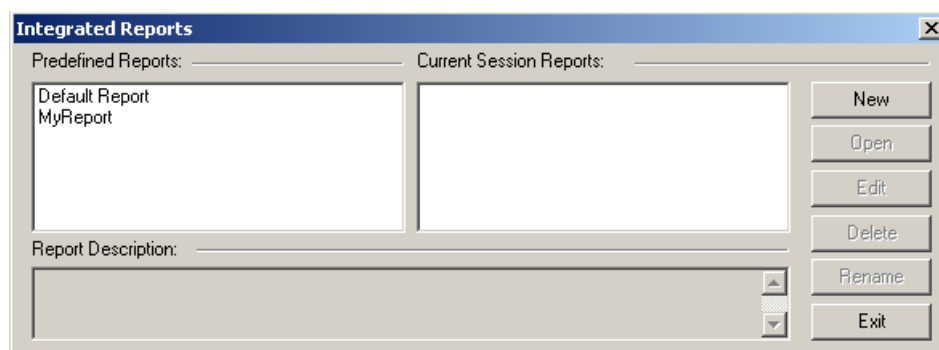


Figure 175: Integrated Reports Dialog Box

2. Click **New**.

The New Report dialog box appears.

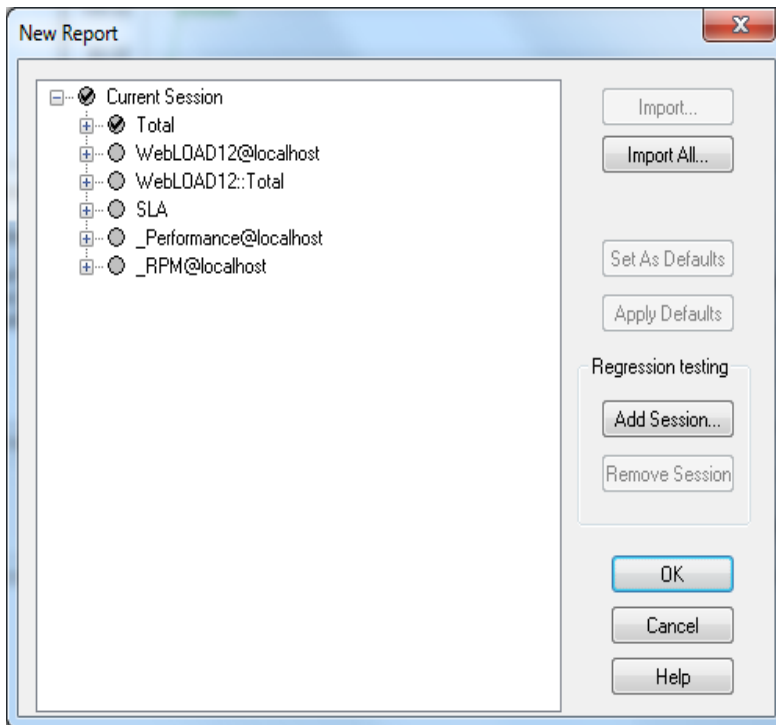


Figure 176: New Report Dialog Box

The full tree of measurements, including all of the items that can be included in the report, are displayed in the Report dialog box. In addition, items configured to be collected in the Performance Measurements Manager are also displayed. Data for all of the items is accumulated during the test run.

Select the parameters to view in the Integrated Report from the tree structure displayed in the Report dialog box. Only the results for the items that are checked are display in the Integrated Report.

Selecting Measurements for the Integrated Report

The first time the Report dialog box appears, none of the items are selected. If you do not select any of the checkboxes, the Integrated Report is blank.

You can configure the Integrated Report in one of several ways:

- Select the items to be included in the report manually.
- Select the checkbox adjacent to the measurement that you want to include in the Integrated Report. The options display in tree format. You can expand or compress the tree using the +/- buttons. If an upper level component is selected, all of the default subcomponents in that tree will be selected.



Note: A gray circle next to a measurement indicates that no default subcomponents are defined (the upper level component cannot be selected). Right-click a measurement to be included in the report to prompt the pop-up menu that enables you to select the item.

- Select a WebLOAD Console default configuration by clicking the **Apply Defaults** button. WebLOAD Console selects the set of default measurements.
- Select **Import All** to apply a previously configured Integrated Reports test set, saved as a Load Template to your current test.



Note: The configuration is imported, even if it was designed for testing a different application.

Changing the Reports Parameters

You can edit the timer options listed in the configuration tree in real time.

While a test is running, you can:

- Add a user-defined timer that was not previously configured.
- Change the scale of the report.

Timer information from Probing Clients and Load Machines are added to the configuration tree automatically in runtime, when the Console receives the timer data.

You can define user-defined timers in the script to measure specific activity. For more information on Timers, see the *WebLOAD Scripting Guide*.

Adding a User-Defined Timer

To add a user-defined timer:

1. Right-click the upper level item in the configuration tree to which you would like to add a timer.
A drop-down menu appears.
2. Click **Add Timer**.
3. Enter the name of the timer in the text box.
4. Select the timer attributes you would like to include in the report.
5. Click **OK**.

The new timer is included in the Integrated Report.

Importing Report Parameters from a Saved Load Template

WebLOAD Console enables you to import a complete report configuration or to import a report configuration for a selected component in the Load Session.

Importing a Complete Report Configuration

To import a complete report configuration:

1. In the **Reports** dialog box, click **Import All**.

The Open dialog box appears.

2. Select the Load Template to import.

3. Click **Open**.

The Import dialog box displays a list of reports contained in the selected Load Template.

4. Select the report that contains the desired configuration.

5. Click **OK**.

The report is configured to display the same measurements.



Note: If the imported configuration contains a measurement not configured in the current report, WebLOAD Console ignores the measurement in the Load Template file and does not add it to the report.

This function imports the Integrated Report test configuration from the selected Load Template file. None of the other information in the Load Template file is imported.

Importing a Report Configuration for a Selected Component

To import a report configuration for a selected component:

1. Select the upper level component in the configuration tree for which you want to import a report configuration.

2. Click **Import**.

The Open dialog box appears.

3. Select the Load Template to import and click **Open**.

The Import dialog box displays a list of reports contained in the selected Load Template.

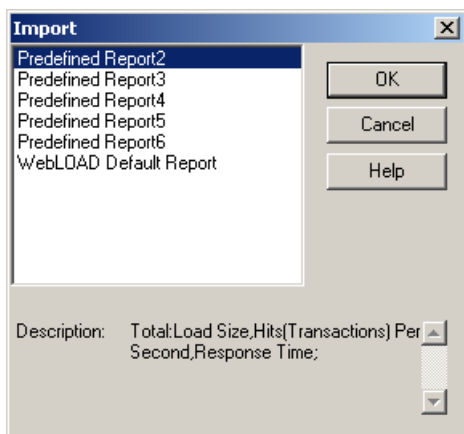


Figure 177: Import Dialog Box

4. Select the report that contains the desired configuration.

WebLOAD Console displays the list of components (Load Machines and Probing Clients) included in that report.

5. Select the component that contains the desired configuration.

The current component is configured to display the same measurements.



Note: If the imported component contains a measurement that is not configured in the current report, WebLOAD Console ignores the measurement in the Load Template file. It does not add it to the report.

This function imports the Integrated Report test configuration from the selected Load Template file. None of the other information in the Load Template file is imported.


Load Generator Performance Statistics

You can configure a report that displays a summary of Load Generator performance statistics generated during the test session. This is useful for analyzing your Load Generator capacity and gauging the load on the Load Generator itself. The Load Generator performance statistics include:

- Free memory – Free memory of the machine hosting the Load Generator
- Global CPU usage – The total CPU usage of the machine hosting the Load Generator
- Load Generator CPU usage - CPU usage of the Load Generator process
- Server Latency

In addition, if the *Show Performance Warnings* option is selected in the **Diagnostics** tab of the **Default/Current Session Options** or **Script Options**, WebLOAD automatically sends warning messages when Load Generators are getting overloaded, so you know how to better utilize the Load Generators in your current and future tests.

To add Load Generator statistics to an integrated report:

1. Open a report, as follows:
 - a. Click **Integrated Report Manager** in the **Session** tab of the ribbon.
The Integrated Reports dialog box appears.
 - b. Click **New** to open a new report, or select an existing report and click **Edit**.
The Report dialog box appears.
2. Alternatively, click Update  in the Integrated Report toolbar, to update the display in the report while a test is in progress.
The Report dialog box appears.

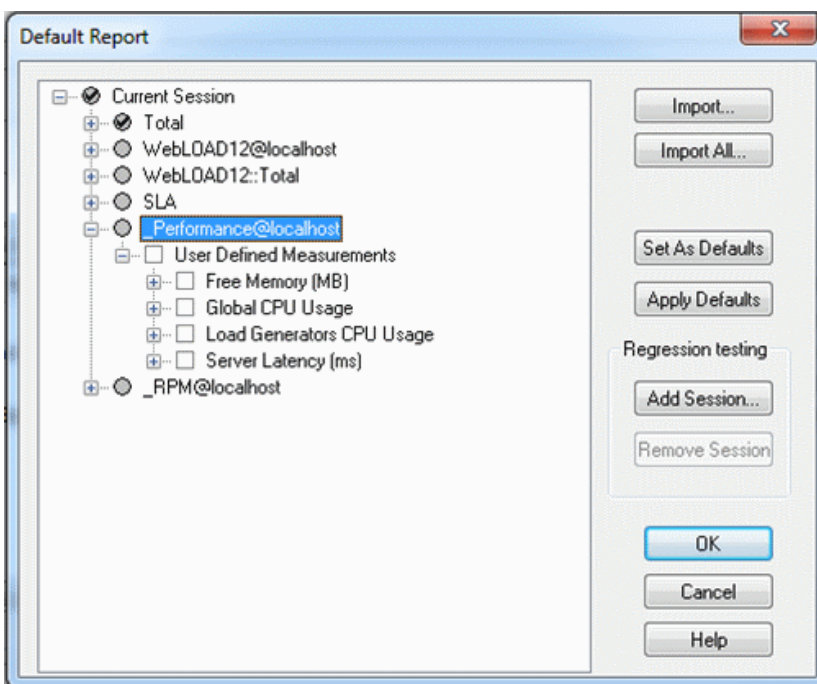


Figure 178: Report Dialog Box

3. Select items under **_Performance@LoadGenerator** to indicate the Load Generator measurements you want displayed in the Integrated Report.

Regression Testing

Regression testing enables you to compare Load Sessions while tests are running, or after the test session is completed. WebLOAD regression testing reports enable you to visualize, verify, and compare results over time to confirm that results only vary in ways that you expect them to.

Regression testing enables you to configure a report that compares:

- Existing performance data from a previously saved or “expected” Load Session with data from the currently running test session.
- Existing performance data from two previously saved Load Sessions.

Combining performance results from several different Load Sessions into a single report enables you to view, verify and compare results over time and ensure that results only vary in ways that you expect them to.

Configuring an Integrated Report to Include Data from Previously Saved Load Sessions

To configure an Integrated Report to include data from previously saved Load Sessions:

1. Click **Integrated Report Manager** in the **Session** tab of the ribbon.
The Integrated Report dialog box appears.
2. Click **New**.
The Reports dialog box appears.
3. In the Reports dialog box, click **Add Session**.
The Open dialog box appears.
4. Select the Load Session (*.l s) file to add and click **Open**.
The Load Session is added to the Report dialog box.
5. Configure the Integrated Report to include the measurements you want to view, see *Configuring an Integrated Report* (on page 303). If a measurement is already selected in the current Load Session it is automatically selected in the Load Session selected to be included in the report.
6. Click **OK**.
The report displays in the Console Results window.

Removing a Previously Saved Load Session from the Integrated Report Configuration

To remove a previously saved Load Session from the Integrated Report configuration:

1. Select the Load Session you want to remove from the Integrated Report configuration in the Reports dialog box.
2. Click **Remove Session**.
The Load Session is removed from the Report dialog box, and any measurements displayed for that Load Session in the Console Results window are removed.

Viewing Integrated Reports

In the Integrated Report, WebLOAD Console displays the results of your test in a line graph. Separate lines track each measurement selected in the report configuration.

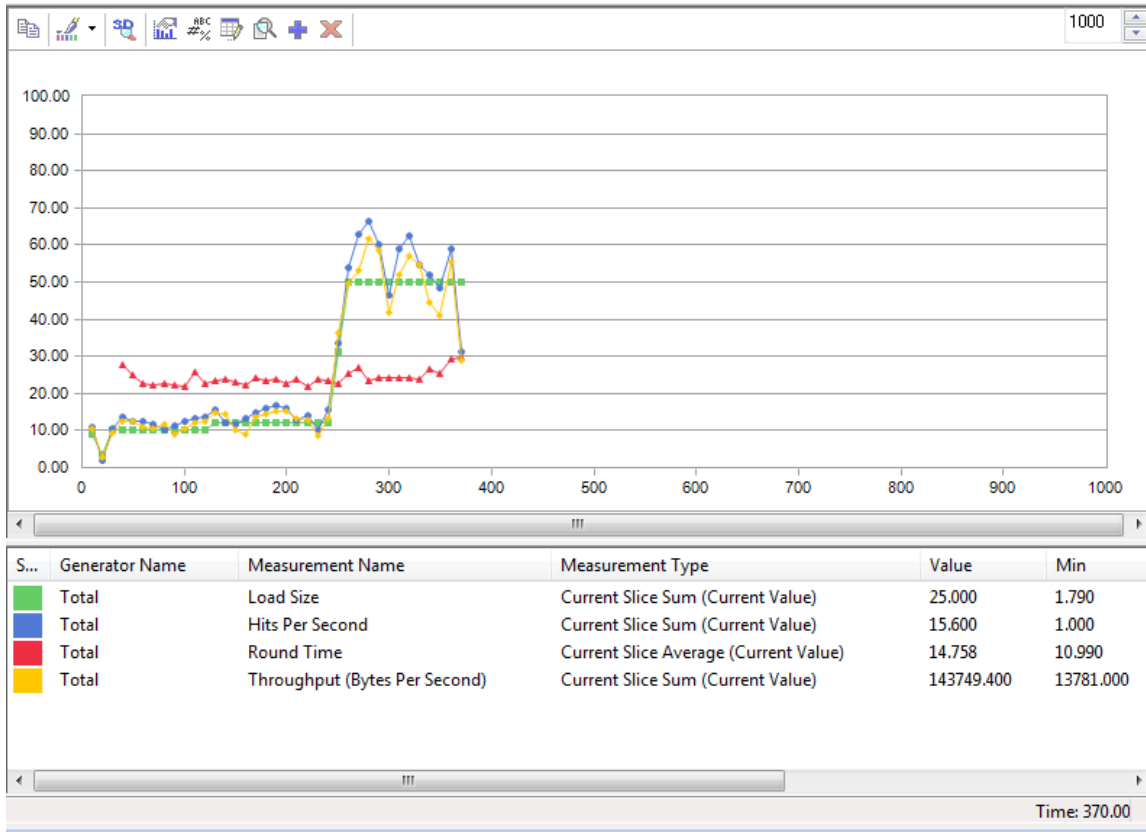


Figure 179: Integrated Report

The report above shows an Integrated Report that includes four measurements. Each is tracked on a different line graph. The line graphs are color-coded to match the measurements on the bottom half of the report.

Features of this graph:

- The upper part of the report displays a line graph for each measurement tracked in the report. The X-Axis represents the elapsed time (in seconds) for this test session. The Y-Axis represents the actual value for each measurement, displayed as a percentage relative to the defined maximum Scale Value.
- Hover over a colored line graph to view the exact value for the selected point.
- The scroll bar at the center of the screen provides you with a quick and easy way to access points in the test session currently not in focus.

- The lower part of the report displays a list of the measurements being tracked for each host and each session, the current value of the measurement and, in the last column on the right, the maximum Scale Value set for each measurement.
- The Time Range field in the top right-hand corner displays the time scale for the X-Axis. In this graph, the X-Axis displays a window of 200 seconds out of the total time span of the Load Session. Using the arrows on the Time Range field, you can change the number of seconds to achieve the report display you desire.
- If you are tracking more than one measurement, the top graph shows multiple graph lines. To bring up a history focusing on a specific measurement, double-click a single measurement line in the Integrated Report.

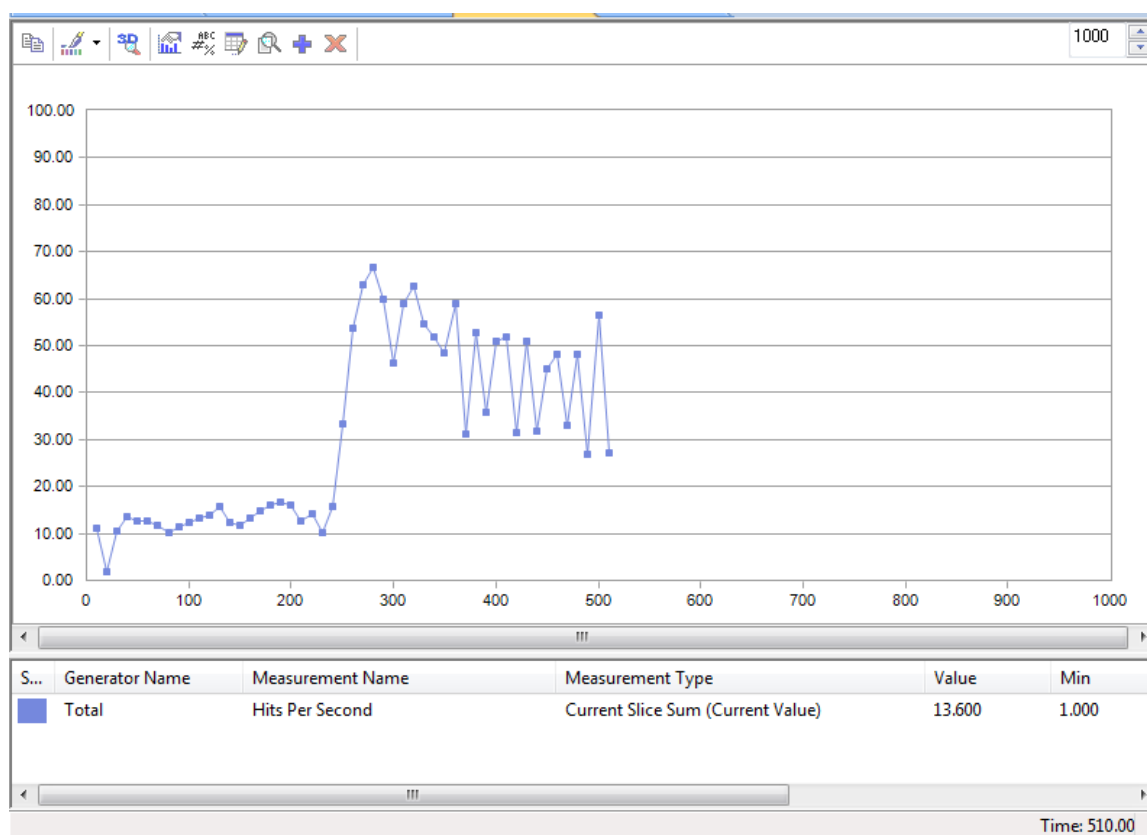


Figure 180: Specific Measurement History

See *Integrated Report Toolbar* (on page 313) for information on the toolbar at the top of the Integrated Report screen in Chart View.

Changing the Scale of a Report for a Selected Measurement

To change the scale of a report for a selected measurement:

1. Right-click a specific measurement from the lower part of the report and select **Change Scale** from the pop-up menu,

-Or-

Double-click a specific measurement from the lower part of the report.

The Change Scale dialog box appears.

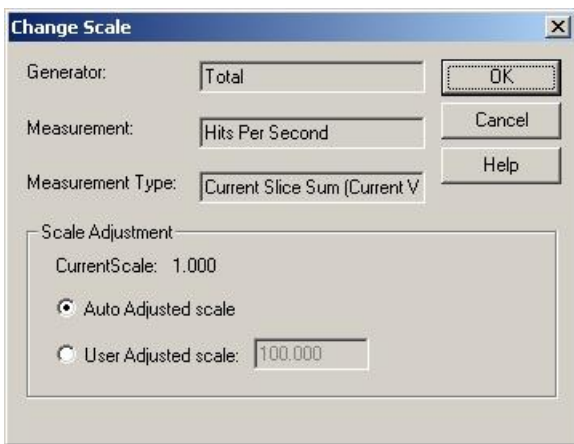


Figure 181: Change Scale Dialog Box

2. Select one of the following:
 - **Auto Adjusted scale** – This uses an automatic scaling algorithm that determines the best way for the measurements to be displayed in the chart view. The scale adjusts itself according to changes in real time and the value adjusts itself according to the scale value. The Y scale of an integrated report for a number of measurements is 0-100. The points of each chart series (specific measurements) are calculated according to the automatic scaling algorithm: $\text{Real value} * 100 / \text{scale value}$.
 - **User Adjusted scale** – This enables you to manually set the scale value.
3. Enter the new scale value.
4. Click **OK**.



The scale of the report for that measurement changes in real time. The scale value for a measurement can also be changed after the test has run.

Viewing the True Value vs. Relative Value of a Measurement

Each measurement displayed in the Integrated Reports has its own expression or value of measurement. For example, Timer results are measured in seconds. CPU is measured in terms of percentage of usage. Throughput is measured in kilobytes per second. All measurements appear as a percentage relative to the defined scale.

To change the scale value of a measurement, see *Changing the Scale of a Report for a Selected Measurement* (on page 312). When you change the maximum value, WebLOAD Console displays the relative value of that measurement rather than the true value of a measurement.

To view a measurement's value:

- Select the Data Editor  toolbar button to display the relative value of the measurements in the chart.
- Select the Point Label  toolbar button to display the real value of the measurements in the chart.



Note: You can display the real value for a single line in the chart by right-clicking the desired line and selecting **Point Label** from the pop-up menu.

Integrated Report Toolbar

At the top of the Integrated Report, find the Integrated Report toolbar. This toolbar offers a variety of options for customizing the Integrated Report display, tailoring the colors, perspective, and dimension as needed.

The following illustration shows the Integrated Report toolbar and the function of each toolbar button.

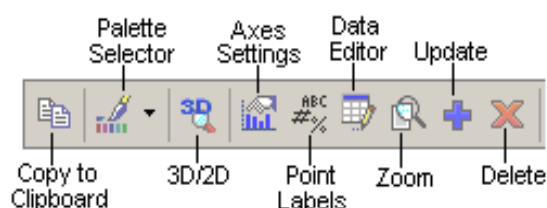
















Figure 182: Integrated Report Toolbar

Table 45: Integrated Report Toolbar Buttons

Icon	Toolbar Button	Function
	Copy to Clipboard	<p>Captures a screen shot of the currently displayed chart to the paste buffer. Select one of the following formats:</p> <ul style="list-style-type: none"> •  As a Bitmap • As a Metafile •  As Text (data only)
	Palette Selector	<p>Changes the color palette of the graph.</p> <ol style="list-style-type: none"> 1. Click the arrow to display a drop-down of palette bars. 2. Select one of the palette bars. The palette colors are applied to the graph.
	3D/2D	Toggles between 2D and 3D view.
	Axes Settings	<p>Modifies the X-Axes and Y-Axes settings of the graph. Perform any of the following:</p> <ul style="list-style-type: none"> • Click  Grid to toggle the Horizontal Grid on and off. • Click Interlaced to display the Horizontal Grid's rows with an alternating background color. • Click  Grid to toggle the Vertical Grid on and off. • Click Vertical Labels to toggle the X-Axes elapsed time labels to be displayed horizontally or vertically. • Click Staggered to toggle the X-Axes elapsed time labels to be displayed in a staggered line or all on the same line. • Click Show Labels to toggle the X-Axes elapsed time labeled on or off. • Click Options... to display the Properties dialog box and configure the graph's advanced settings.
	Point Labels	Toggles whether to display labels for every point in the graph. The Point Labels are the true values of the measurements.

Icon	Toolbar Button	Function
	Data Editor	<p>Displays the relative values of the test results in a tabular format. The Data Editor's features are as follows:</p> <ul style="list-style-type: none"> • The names of all of the measurements and their attributes tracked in the Integrated Report display in the first column at the far left. • The data accumulated during the Load Session, scaled into relative values, displays in tabular format in the table fields. • Hovering over a value in the table highlights the values location in the graph. • Double-clicking a value in the table enables you to modify the test result value. <p> Note: Right-click anywhere in the Data Editor to display a pop-up menu enabling you to select the Data Editor's location on the screen and customize the displayed text.</p>
	Zoom	<p>Toggles between an enlarged section of the graph and the full graph.</p> <ul style="list-style-type: none"> • Click the Zoom button and drag your mouse over an area in the graph to enlarge that area. • Click the Zoom button again to display the full graph.
	Update	<p>Updates items displayed in the Integrated Report while the test is in progress. Click the Update button to open the Report Configuration dialog box. Using this option, you can select the measurement to be displayed in the report.</p>
	Delete	<p>Deletes items from the Integrated Report while the test is in progress. Select the item you want to delete by clicking the Series indicator in the list appearing at the bottom and then click the Delete button. The item is immediately deleted from the report.</p>

WebLOAD Statistics Reports

The Statistics Report is both a real-time monitor that displays the results of your test while you run it and a report that you can save and export.

WebLOAD Console collects approximately 30 different statistics during a test. The Statistics Report displays the values for all of them. You can also create Integrated Reports that display only the statistics that you choose. For information about Integrated Reports, see *WebLOAD Console Performance Reports* (on page 295).

Statistical Measurements and Their Values

The Statistics Report is presented as a table.

Row headings (the labels in the first column on the left) are the names of the statistical measurements.

Column headings categorize the statistics as follows:

- **Total** – Sums the values for each statistic across all the scripts running during the test.
- **ScriptName::Total** – Sums the values for each statistic across all instances of the specified script running during the test.
- **ScriptName@LoadGenerator** – Sums the values for each statistic across all instances of the specified script running under the specified Load Generator during the test.

Measurements (Current Values)	Total	Quickstart:Total	Quickstart@jryakovb	PM@JRYAAKOV
Load Size	1.000	1.000	1.000	-
Round Time	-	-	-	-
Rounds	0.000	0.000	0.000	-
Successful Rounds	0.000	0.000	0.000	-
Rounds Per Second	0.000	0.000	0.000	-
Successful Rounds Per Second	0.000	0.000	0.000	-
Page Time	-	-	-	-
Pages	0.000	0.000	0.000	-
Pages Per Second	0.000	0.000	0.000	-
DNS Lookup Time	-	-	-	-
Hit Time	-	-	-	-
Hits	0.000	0.000	0.000	-
Successful Hits	0.000	0.000	0.000	-
Hits Per Second	0.000	0.000	0.000	-
Successful Hits Per Second	0.000	0.000	0.000	-
Attempted Connections	0.000	0.000	0.000	-
Successful Connections	0.000	0.000	0.000	-
Connect Time	-	-	-	-
Process Time	-	-	-	-
Receive Time	-	-	-	-
Send Time	-	-	-	-
Time To First Byte	-	-	-	-
Throughput (Bytes Per Second)	0.000	0.000	0.000	-
Response Time	-	-	-	-
Response Data Size	0.000	0.000	0.000	-
Responses	0.000	0.000	0.000	-
Content Size	-	-	-	-
GET_http://ad.doubleclick.net	-	-	-	-
GET_http://ad.doubleclick.net	-	-	-	-
GET_http://ad.doubleclick.net	-	-	-	-
GET_http://ad.doubleclick.net	-	-	-	-
GET_http://clients1.google.c	-	-	-	-
GET_http://googleads.g.dou	-	-	-	-
GET_http://googleads.g.dou	-	-	-	-
GET_http://googleads.g.dou	-	-	-	-

Double click on a column or row header or any table cell to view statistical details. Load Size: 1 Time: 1080

Figure 183: Statistics Report Window

The statistics displayed in each row are grouped into the following categories:

- **Per time unit** statistics are ratios that calculate an average value for an action or process. For example: Hits Per Second and Rounds Per Second.
- **Timers** track the amount of time it took the Virtual Clients to complete an action or process and are only measured for successful transactions. For example: Connect Time and Round Time.
- **Counters** track the number of times the Virtual Clients completed an action or process. For example: Hits and Rounds.

By default, WebLOAD Console displays values for its statistics at 20-second intervals. The reported value at each reporting interval is called the *current value*.

Changing the Length of the Reporting Interval

To change the length of the reporting interval:

1. Click **Global Options** in the **Tools** tab of the ribbon,

-Or-

Select **Global Options** from the Console System button.

2. Select the **General** tab.
3. Change the value in the Statistics reporting resolution field to any number of seconds between 1 and 3,600.

Current Value Computation

WebLOAD Console calculates the current value as follows:

- For per time unit and timer statistics, WebLOAD Console averages the values collected during the last reporting interval (current slice). If the reporting interval is 20 seconds (the default), the value for Hits Per Second is the average number of hits per second over the last 20 seconds. The value for Hit Time is the average time it took for a successful hit to be completed during the last 20 seconds.
- For counters, WebLOAD Console reports the number of times the item being counted occurred over the last reporting interval. If the reporting interval is 20 seconds, the value for Hits is the actual number of hits that occurred during the last 20 seconds.

Statistics Definitions

The following list displays the various statistics, by context. Table 46 displays the definitions of the per time unit, timer, and counter statistics, alphabetically.

- *Load Size*
- *Throughput (bytes per second)*
- Page statistics – A page is an upper level request for information made by a Virtual Client to the system under test (SUT). For example, a Get statement for a URL retrieves a page. Page statistics include:
 - *Pages*
 - *Pages Per Second*
 - *Page Time*
- Hit statistics – A hit is a request for information made by a Virtual Client to the system under test (SUT). In other words, an HTTP request. For example, a Get statement for a URL retrieves a page. The page can include any number of graphics and content files. Each request for a gif, jpeg, html file, etc., is a single hit. Hit statistics include:
 - *Hits*
 - *Successful Hits*

- *Failed Hits*
- *Hits Per Second*
- *Successful Hits Per Second*
- *Failed Hits Per Second*
- *Hit Time*
- Rounds statistics – A Round is an entire run of a script. Rounds statistics include:
 - *Rounds*
 - *Successful Rounds*
 - *Failed Rounds*
 - *Aborted Rounds*
 - *Rounds Per Second*
 - *Successful Rounds Per Second*
 - *Failed Rounds Per Second*
 - *Round Time*
- Connection statistics – A Connection is a new TCP/IP connection of a Virtual Client to the SUT. The Connection counter increments every time a TCP/IP connection is opened. This number is always less than or equal to hits/requests, because several hits may use the same HTTP connection if the Persistent Connection option is enabled (**Default Options** ► **Browser Parameters**). Connection statistics include:
 - *Attempted Connections*
 - *Successful Connections*
 - *Failed Connections*
 - *Connect Time*
 - *Connection Speed (Bits Per Second)*
 - *DNS Lookup Time*
- Response statistics – A Response is an HTTP message sent from the Web server as a response to an HTTP Request. Response statistics include:
 - *Responses*
 - *Response Data Size*
 - *Response Time*
 - *Send Time*
 - *Process Time*
 - *Receive Time*
 - *Time to First Byte*

- *HTTP Response Status*
- *Content Size*
- **Report statistics** – The statistics collected by the reportStatistics command in a Selenium script. Refer to the *Navigation Timing Statistics* table in the *Selenium Integration* appendix of the *WebLOAD™ Recorder User's Guide*.
- **<User-Defined> statistics:**
 - *Automatic Data Collection*
 - *Timer*
 - *Counters*
 - *Transaction Timers*
 - *Total <transaction name> TPS*
 - *Transaction Counters*
- *Java and ActiveX Timers*
- *Java and ActiveX Counters*
- *Server Performance Measurements*

Table 46: Statistics

Statistic	Type	Definition
Aborted Rounds	Counter	The number of times the Virtual Clients started to run a script but did not complete the script, during the last reporting interval. This might be due to session being stopped either automatically or manually by the user.
Attempted Connections	Counter	The total number of times the Virtual Clients attempted to connect to the SUT during the last reporting interval.
<User-Defined> Automatic Data Collection	Timer and counter	If you have Automatic Data Collection enabled (Default Options > Functional Testing), WebLOAD Console creates three counters for each GET and POST statement in the script. The counters are the total number of times the Get and Post statements occurred, the number of times the statements succeeded, and the number of times the statements failed during the last reporting interval.

Statistic	Type	Definition
Connect Time	Timer	<p>The time it takes for a Virtual Client to connect to the SUT, in seconds. In other words, the time it takes from the beginning of the HTTP request to the TCP/IP connection.</p> <p>The value posted in the Current Value column is the average time it took a Virtual Client to connect to the SUT during the last reporting interval.</p> <p>If the Persistent Connection option is enabled (Default Options ► Browser Parameters), there may not be a value for Connect Time because the HTTP connection remains open between successive HTTP requests.</p>
Connection Speed (Bits Per Second)	Per-time unit	<p>The number of bits transmitted back and forth between the Virtual Clients and the SUT divided by the time it took to transmit those bits, in seconds.</p> <p>You can set the Virtual Clients to emulate a particular connection speed during the test either by using the Variable Connection Speed settings (Default Options ► Connection) or by coding the connection speed in the script.</p> <p>If a connection speed is specified for the test, WebLOAD Console reports it in the Statistics Report.</p> <p>The value posted in the Current Value column is the number (sum) of bits passed per second during the last reporting interval. It should match, very closely, the connection speed you specified for the test.</p> <p>For example, if:</p> <ul style="list-style-type: none"> • Reporting interval = 20 sec. • Number of bits transmitted during last interval = 96,000 <p>Connection Speed = $96,000 / 20 = 4,800$ bits per second.</p>
Content Size	Counter	The size of the graphics and content files included in a page.
<User-Defined> Counters	Counter	<p>You can add your own counters to scripts with the <code>SendCounter ()</code> and the <code>SendMeasurement ()</code> functions (see the <i>WebLOAD Scripting Guide</i>). If there is a user-defined counter in the script that you are running, WebLOAD Console reports the counter's values in the Statistics Report.</p> <p>The row heading is the name (argument) of the counter, that is, the string in parenthesis in the <code>SendCounter ()</code> or <code>SendMeasurement ()</code> function call.</p> <p>The value reported is the number of times the counter was incremented during the last reporting interval.</p>
DNS Lookup Time	Timer	The time it takes to resolve the host name and convert it to an IP by calling the DNS server.

Statistic	Type	Definition
Failed Connections	Counter	<p>The total number of times the Virtual Clients tried to connect to the SUT but were unsuccessful, during the last reporting interval.</p> <p>This number is always less than or equal to the number of failed hits because hits can fail for reasons other than a failed connection.</p>
Failed Hits	Counter	<p>The total number of times the Virtual Clients made HTTP requests but did not receive the correct HTTP response from the SUT during the last reporting interval. Each request for a gif, jpeg, html file, etc. is a single hit.</p>
Failed Hits Per Second	Per-time unit	<p>The number of times the Virtual Clients did not obtain the correct HTTP response divided by the elapsed time, in seconds.</p> <p>The value posted in the Current Value column is the number (sum) of unsuccessful HTTP requests per second during the last reporting interval.</p> <p>For example, if:</p> <ul style="list-style-type: none"> Elapsed time of test = 10 min. (600 sec.) Number of hits that fail to return a response = 1,200 <p>Successful Hits Per Second = $1,200 / 600 = 2$.</p>
Failed Rounds	Counter	<p>The total number of times the Virtual Clients started but did not complete the script during the last reporting interval.</p>
Failed Rounds Per Second	Per-time unit	<p>The number of times the Virtual Clients started but did not complete an iteration of this script divided by the elapsed time, in seconds.</p> <p>The value posted in the Current Value column is the number (sum) of failed iterations of the script per second during the last reporting interval.</p> <p>For example, if:</p> <ul style="list-style-type: none"> Elapsed time of test = 10 min. (600 sec.) Number of failed rounds = 20 <p>Failed Rounds Per Second = $20 / 600 = 0.033$.</p>
Hit Time	Timer	<p>The time it takes to complete a successful HTTP request, in seconds. The Hit Time is the sum of the Connect Time, Send Time, Response Time, and Process Time for an individual page hit.</p> <p>The value posted in the Current Value column is the average time it took to make an HTTP request and process its response during the last reporting interval.</p>

Statistic	Type	Definition
Hits	Counter	The total number of times the Virtual Clients made HTTP requests to the system under test (SUT) during the last reporting interval. Each request for a gif, jpeg, html file, etc. is a single hit.
Hits Per Second	Per-time unit	<p>Hits Per Second is the number of times the Virtual Clients made an HTTP request divided by the elapsed time, in seconds.</p> <p>The value posted in the Current Value column is the number (sum) of HTTP requests per second during the last reporting interval.</p> <p>For example, if:</p> <ul style="list-style-type: none"> • Elapsed time of test = 10 min. • Number of hits = 12,600 <p>Hits Per Second = 12,600 / 600 = 21.</p>
HTTP Response Status	Counter	<p>WebLOAD Console creates a row in the Statistics Report for each kind of HTTP status code it receives as an HTTP response from the SUT (redirection codes, success codes, server error codes or client error codes).</p> <p>The value posted is the number of times the Virtual Clients received that status code during the last reporting interval.</p> <p>For example:</p> <p>The first digit of the Status-Code defines the class of response. The last two digits do not have any categorization role. There are five categories:</p> <ul style="list-style-type: none"> • 1xx: Informational – Not used, but reserved for future use. • 2xx: Success – The action was successfully received, understood, and accepted. • 3xx: Redirection – Further action must be taken in order to complete the request. • 4xx: Client Error – The request contains incorrect syntax or cannot be fulfilled. • 5xx: Server Error – The server failed to fulfill an apparently valid request.

Statistic	Type	Definition
Java and ActiveX Counters	Counter	<p>You can add function calls to your scripts that enable you to instantiate and call methods and properties in Java and ActiveX components (see the <i>WebLOAD Scripting Guide</i>). If there are ActiveX or Java function calls in the Script that you are running, WebLOAD Console reports three counters for them in the Statistics Report.</p> <p>The row heading is the name of the function call. The counters are the total number of times it occurred, the number of times it succeeded, and the number of times it failed during the last reporting interval.</p>
Java and ActiveX Timers	Timer	<p>You can add function calls to your scripts that enable you to instantiate and call methods and properties in Java and ActiveX components (see the <i>WebLOAD Scripting Guide</i>). If there are ActiveX or Java function calls in the script you are running, WebLOAD Console reports timers for them in the Statistics Report.</p> <p>The row heading is the name of the function call. The value posted is the average amount of time it took to complete the function call, in seconds, during the last reporting interval.</p>
Load Size	Counter	The number of Virtual Clients running during the last reporting interval.
Page Time	Timer	<p>The time it takes to complete a successful upper level request, in seconds. The Page Time is the sum of the Connection Time, Send Time, Response Time, and Process Time for all the hits on a page.</p> <p>The value posted in the Current Value column is the average time it took the Virtual Clients to make an upper level request and process its response during the last reporting interval.</p>
Pages	Counter	The total number of times the Virtual Client made upper level requests (both successful and unsuccessful) during the last reporting interval.
Pages Per Second	Per-time unit	<p>Pages Per Second is the number of times the Virtual Clients made upper level requests divided by the elapsed time, in seconds.</p> <p>The value posted in the Current Value column is the number (sum) of requests per second during the last reporting interval.</p> <p>For example, if:</p> <ul style="list-style-type: none"> Elapsed time of test = 10 min. (600 sec.) Number of pages = 750 <p>Pages Per Second = $750 / 600 = 1.25$.</p>

Statistic	Type	Definition
Process Time	Timer	<p>The time it takes WebLOAD Console to parse an HTTP response from the SUT and then populate the DOM, in seconds.</p> <p>The value posted in the Current Value column is the average time it took WebLOAD Console to parse an HTTP response during the last reporting interval.</p>
Receive Time	Timer	The elapsed time between receiving the first byte and the last byte.
Response Data Size	Counter	<p>The size, in bytes, of all the HTTP responses sent by the SUT during the last reporting interval.</p> <p>WebLOAD Console uses this value to calculate Throughput (bytes per second).</p>
Response Time	Timer	<p>The time it takes the SUT to send the object of an HTTP request back to a Virtual Client, in seconds. In other words, the time from the end of the HTTP request until the Virtual Client has received the complete item it requested (Time to first byte + Receive Time - Process Time).</p> <p>The value posted in the Current Value column is the average time it took the SUT to respond to an HTTP request during the last reporting interval.</p>
Responses	Counter	<p>The number of times the SUT responded to an HTTP request during the last reporting interval.</p> <p>This number should match the number of successful hits.</p>
Round Time	Timer	<p>The time it takes one Virtual Client to complete an entire iteration of a script, in seconds.</p> <p>The value posted in the Current Value column is the average time it took the Virtual Clients to complete an entire iteration of the script during the last reporting interval.</p> <p>For example, if:</p> <ul style="list-style-type: none"> Virtual Clients = 2. First Virtual Client completes a round in 60 sec. Second Virtual Client completes a round in 85 sec. Both Virtual Clients complete the rounds within the current reported interval. <p>Round Time = 72.5 sec.</p>
Rounds	Counter	<p>The total number of times the Virtual Clients attempted to run the script during the last reporting interval.</p> <p>For example:</p> <p>Rounds = Failed Rounds + Successful Rounds + Aborted Rounds.</p>

Statistic	Type	Definition
Rounds Per Second	Per-time unit	<p>The number of times the Virtual Clients attempted to run the script divided by the elapsed time, in seconds.</p> <p>The value posted in the Current Value column is the number (sum) of attempts (both successful and unsuccessful) per second during the last reporting interval.</p> <p>For example, if:</p> <ul style="list-style-type: none"> Elapsed time of test = 10 min. (600 sec.) Number of rounds = 200 <p>Rounds Per Second = $200 / 600 = 0.33$.</p>
Send Time	Timer	<p>The time it takes the Virtual Clients to write an HTTP request to the SUT, in seconds.</p> <p>The value posted in the Current Value column is the average time it took the Virtual Clients to write a request to the SUT during the last reporting interval.</p>
Server Performance Measurements	Timers	<p>If you selected Performance Monitor statistics for the report, WebLOAD Console creates a row for them and reports their values in the Statistics Report.</p> <p>For example:</p> <p>Sessions Errored Out, % Processor Time, Current Disk Queue Length, and so on.</p> <p>For definitions of the statistics, see the Performance Measurements Manager (click Edit Template in the Session tab of the ribbon so that you are in edit mode, then click Performance Measurements Manager).</p> <p>Be selective when selecting server performance measurements or the system resources required to manage the data might affect the Console.</p>
Successful Connections	Counter	<p>The total number of times the Virtual Clients were able to successfully connect to the SUT during the last reporting interval.</p> <p>This number is always less than or equal to the number of successful hits because several hits might use the same HTTP connection if the Persistent Connection option is enabled (Default Options > Browser Parameters).</p>
Successful Hits	Counter	<p>The total number of times the Virtual Clients made HTTP requests and received the correct HTTP response from the SUT during the last reporting interval. Each request for a gif, jpeg, html file, etc. is a single hit.</p>

Statistic	Type	Definition
Successful Hits Per Second	Per-time unit	<p>The number of times the Virtual Clients obtained the correct HTTP response to their HTTP requests divided by the elapsed time, in seconds.</p> <p>The value posted in the Current Value column is the number (sum) of successful HTTP requests per second during the last reporting interval.</p> <p>For example, if:</p> <ul style="list-style-type: none"> Elapsed time of test = 10 min. (600 sec.) Number of hits that return a successful response = 11,400 <p>Successful Hits Per Second = $11,400 / 600 = 19$.</p>
Successful Rounds	Counter	The total number of times the Virtual Clients completed one iteration of the script during the last reporting interval.
Successful Rounds Per Second	Per-time unit	<p>The number of times the Virtual Clients completed an entire iteration of the script divided by the elapsed time, in seconds.</p> <p>The value posted in the Current Value column is the number (sum) of successful iterations of the script per second during the last reporting interval.</p> <p>For example, if:</p> <ul style="list-style-type: none"> Elapsed time of test = 10 min. (600 sec.) Number of successful rounds = 180 <p>Successful Rounds Per Second = $180 / 600 = 0.3$.</p>
Throughput (bytes per second)	Per-time unit	<p>The average number of bytes per second transmitted from the SUT to the Virtual Clients running this script during the last reporting interval.</p> <p>In other words, this is the amount of the Response Data Size (sum) divided by the number of seconds in the reporting interval.</p> <p>For example, if:</p> <ul style="list-style-type: none"> Elapsed time of test = 10 min. Reporting interval = 20 sec. Data size in bytes sent during last interval = 6,000 <p>Throughput = $6,000 / 20 = 300$.</p>
Time to First Byte	Timer	The time it takes from when a request is sent until the Virtual Client receives the first byte of data.

Statistic	Type	Definition
Total <transaction name> TPS	Per-time unit	<p>The number of times the Virtual Client encountered the transaction divided by the elapsed time, in seconds.</p> <p>The value posted in the Current Value column is the number of times the Virtual Client encountered the transaction during the last reporting interval.</p> <p>For example, if:</p> <ul style="list-style-type: none"> Elapsed time of test = 10 min. (600 sec.) Number of times the VC encountered the transaction = 11,400 <p>Total Transactions Per Second = $11,400 / 600 = 19$.</p>
<User-Defined> Timer	Timer	<p>You can add timers to scripts to keep track of the amount of time it takes to complete specific actions (see the <i>WebLOAD Scripting Guide</i>). If there are any timers in the scripts that you are running, WebLOAD Console reports their values in the Statistics Report.</p> <p>The row heading is the name (argument) of the timer, that is, the string in parenthesis in the <code>SetTimer()</code> function call. The timer represents the time it takes to complete all the actions between the <code>SetTimer()</code> call and its corresponding <code>SendTimer()</code> call, in seconds.</p> <p>The value posted is the average time it took a Virtual Client to complete the actions between the pair of timer calls, in seconds, during the last reporting interval.</p>
<User-Defined> Transaction Counters	Counter	<p>You can add user-defined transaction functions to scripts for functional tests (see the <i>WebLOAD Scripting Guide</i>). If there is a user-defined transaction function in the script that you are running, WebLOAD Console reports three counters for it in the Statistics Report.</p> <p>The row heading is the name (argument) of the user-defined transaction, that is, the string in parenthesis in the <code>BeginTransaction()</code> function call.</p> <p>The counters are the total number of times the transaction occurred, the number of times the transaction succeeded, and the number of times the transaction failed during the last reporting interval.</p>

Statistic	Type	Definition
<User-Defined> Transaction Timers	Timer	<p>You can add user-defined transaction functions to scripts for functional tests (see the <i>WebLOAD Scripting Guide</i>). If there is a user-defined transaction function in the script that you are running, WebLOAD Console reports a timer for it in the Statistics Report. Transaction timers are only measured for successful transactions.</p> <p>The row heading is the name (argument) of the user-defined transaction, that is, the string in parentheses in the <code>BeginTransaction()</code> function call.</p> <p>The timer represents the average time it took to complete all the actions between the <code>BeginTransaction()</code> call and its corresponding <code>EndTransaction()</code> call, in seconds, during the last reporting interval.</p>

Statistics for Virtual Clients and Probing Clients

There are differences in the statistics reported for Virtual Clients and Probing Clients:

- The values reported for Virtual Clients are average values based on the data reported by the total number of Virtual Clients running that script.
- The values reported for a Probing Client are the specific values for that single, representative user.

By default, Virtual Clients are single-threaded so the Load Machines can support more Virtual Clients overall. Probing Clients run with four threads in order to emulate a single user as closely as possible.

Therefore, you should use the statistics that you obtain from the different types of emulated users in different ways. For example, if you are interested in the exact experience of a typical user of your site, examine the timers like Round Time, Hit Time, Connect Time, and so on from a Probing Client. For performance or load issues, however, examine statistics (such as Successful and Failed hits, Successful and Failed Rounds, Rounds Per Second, Hits Per Second, etc.) from Virtual Clients.

Summary Reports

Each cell in the Statistics Report is an access point for another report:

- Double-click a row heading to display a Summary by Measurement Report.
- Double-click a column heading to display a Summary by script or Summary by Total Report.
- Double-click a cell to display a summary report for that statistical measurement.

Script Names	Test Summary Min	Test Summary Max	Test Summary Sum	Test Summary Aver...	Current Slice Sum
Total	38.000	40.000	558.000	39.857	40.000
Test::Total	38.000	40.000	558.000	39.857	40.000
Test@10.0.1.186	38.000	40.000	558.000	39.857	40.000
_Performance@10.0.1.186	-	-	-	-	-
_RPM@10.0.1.186	-	-	-	-	-

Double-click on a Script name (left-hand column) to view the History Report of this Measurement, for the selected Load Size: 1 Time: 280

Figure 184: Summary Reports

Summary reports provide the following values:

Table 47: Summary Report Values

Value	Definition
Current Slice/Current Value	WebLOAD Console displays values for its statistics at 20-second intervals (you can modify this in Global Options ► General , by changing the value in the Statistics reporting resolution field. The reported value at each reporting interval is called the current value.
Sum	The aggregate or total value for this statistic (throughout all scripts and Virtual Clients).
Min	The lowest value reported for this statistic since the beginning of the test.
Max	The highest value reported for this statistic since the beginning of the test.

Value	Definition
Average	<p>For per time unit statistics and counters, average is the total of all of the current values divided by the number of reporting intervals.</p> <p>For timers, average is the total amount of time counted by the timer (not the elapsed time) divided by the Count (the total number of readings). For example, the average for Hit Time is the amount of time it took to complete all the successful hits divided by the number of successful hits (the Count).</p>
STDev	The average amount this statistic varies from the average number in this script.

The Statistics Report displays all of the statistics that WebLOAD Console collects unless there were no readings for a statistic during a test. For example, if there were no unsuccessful rounds during a test, the table would not display a row for Failed Rounds.

Additionally, if a measurement has no meaning for a specific statistic, WebLOAD Console displays a dash (-) in the cell. For example, if there were two scripts running but only one of them returned an HTTP Response Status code, the table would include a row for that Response Status and would display a dash in the cell for the script that did not return the code.

Using the Statistics Report

Open specific statistical values by double-clicking a cell. You can click any **Script Name** or **Total** in the column headers, any **Measurement Name** in the left-hand column, or any cell at the intersection of any measurement row and script or Total column.

Table 48: Statistics Report Options

Double click	To
Script Name	Displays a Summary Report table containing all measurement attribute values for the selected script.
Total	Displays a Summary Report table containing the sum of all measurement attribute values for all current scripts.
Measurement Name	Displays a Summary Report table containing the attribute values for this measurement for all current scripts.

Double click	To
Intersection, Measurement & Script	Displays a History Report containing all of the Attribute values recorded from the start of the current test session. These values are specific to the selected measurement and script. The complete History Report can also be accessed via the Summary Reports.
Intersection, Measurement & Total	Displays a History Report containing all of the Attribute values recorded for the selected Measurement totaled for all current scripts. The complete History Report can also be accessed via the Summary Reports.

Statistics Report-Summary by Script

Double-click a specific script name in the Statistics Report main window to call up a summary table. When you select a script, the report contains the values for all the attributes collected for that script.

The following example contains all of the values collected for the script, Quickstart:

Statistical Measurement	Test Summary Min	Test Summary Max	Test Summary Count	Test Summary Sum	Test Summary Average	Test Summary STDev	Current Slice Min	Current Slice Max	Current Slice Count	Current Slice Sum
Load Size	0.000	1.000	-	0.891	-	-	-	-	-	1.000
Round Time	425.894	471.323	4.000	-	450.720	-	-	-	-	-
Rounds	0.000	1.000	-	7.000	0.067	-	-	-	-	0.000
Successful Rounds	0.000	1.000	-	4.000	0.038	-	-	-	-	0.000
Aborted Rounds	0.000	1.000	-	1.000	0.056	-	-	-	-	0.000
Failed Rounds	0.000	1.000	-	2.000	0.022	-	-	-	-	0.000
Rounds Per Second	0.000	0.050	-	-	0.003	-	-	-	-	0.000
Successful Rounds Per Second	0.000	0.050	-	-	0.002	-	-	-	-	0.000
Failed Rounds Per Second	0.000	0.050	-	-	0.001	-	-	-	-	0.000
Page Time	0.547	100.344	244.000	-	2.778	-	-	-	-	-
Pages	0.000	13.000	-	244.000	1.937	-	-	-	-	0.000
Pages Per Second	0.000	0.650	-	-	0.097	-	-	-	-	0.000
DNS Lookup Time	0.000	0.054	18.000	-	0.028	-	-	-	-	-
Hit Time	0.365	100.344	600.000	-	1.130	-	-	-	-	-
Hits	0.000	19.000	-	600.000	4.762	-	-	-	-	0.000
Successful Hits	0.000	19.000	-	599.000	4.754	-	-	-	-	0.000
Failed Hits	0.000	1.000	-	1.000	0.011	-	-	-	-	0.000
Hits Per Second	0.000	0.950	-	-	0.238	-	-	-	-	0.000
Successful Hits Per Second	0.000	0.950	-	-	0.238	-	-	-	-	0.000
Failed Hits Per Second	0.000	0.050	-	-	0.001	-	-	-	-	0.000
Attempted Connections	0.000	19.000	-	614.000	4.873	-	-	-	-	0.000
Successful Connections	0.000	19.000	-	599.000	4.754	-	-	-	-	0.000
Failed Connections	0.000	2.000	-	15.000	0.155	-	-	-	-	0.000
Connect Time	0.088	1.703	599.000	-	0.311	-	-	-	-	-
Process Time	0.000	0.005	599.000	-	0.000	-	-	-	-	-
Receive Time	0.000	3.351	599.000	-	0.144	-	-	-	-	-
Send Time	0.000	0.007	599.000	-	0.001	-	-	-	-	-
Time To First Byte	0.110	0.778	599.000	-	0.422	-	-	-	-	-
Throughput (Bytes Per Second)	0.000	20629.600	-	-	1628.541	-	-	-	-	0.000
Response Time	0.255	3.757	599.000	-	0.567	-	-	-	-	-
Response Data Size	0.000	412592.000	-	4103923.000	32570.817	-	-	-	-	0.000
Responses	0.000	19.000	-	599.000	4.754	-	-	-	-	0.000
Content Size	449.000	75129.250	-	-	6851.290	-	-	-	-	-
Failed	0.000	0.050	-	-	0.001	-	-	-	-	0.000
Failed	0.000	1.000	-	1.000	0.011	-	-	-	-	0.000

Double-click on a Measurement (left-hand column) to view the History Report for the selected Measurement for this agenda. Load Size: 1 Time: 2520

Figure 185: Summary Table

The names of all the measurements tracked during the current Load Session are listed in the column at the far left. The names of all the Attributes totaled for the relevant measurements are displayed in the header at the top of the table.

Example:

Figure 185 displays the Response Time measurement values for the script, Quickstart.



Note: The minimum time needed for the SUT to respond to requests from the script during the current test session is 1.246 seconds. The maximum time needed for the SUT to respond to the script run by Generator1 during the current slice is 0.461 seconds. The average Response Time to the script in the current slice is 1.398 seconds.

Getting More Detailed Information

Double-click a measurement name in the left-hand column to call up a History Report for this measurement in the script. This History Report can also be accessed from the Statistics Report main window by double-clicking the button located at the intersection of the selected measurement and script. This History Report is described in *Statistics Report-History by Measurement and* (on page 339).

Example:

Double-click Response Time in the left-hand column to call up a History Report. This History Report lists all of the Response Time Attribute values recorded for the script run by Load Generator1 from the start of the current test session. This History Report can also be accessed from the Statistics Report main window by double-clicking the button located at the intersection of the Response Time row and mix1 column.

Time	Total Load Size	Current Value (Current Slice Average)	Test Summary Min	Test Summary Max	Test Summary Count	Test Summary Average	Current Slice Average (Current Value)
1900.00	1.000	-	0.255	3.757	480.000	0.591	-
1920.00	1.000	0.685	0.255	3.757	487.000	0.592	0.685
1940.00	1.000	0.335	0.255	3.757	494.000	0.589	0.335
1960.00	1.000	0.458	0.255	3.757	507.000	0.585	0.458
1980.00	1.000	0.312	0.255	3.757	514.000	0.582	0.312
2000.00	1.000	0.458	0.255	3.757	524.000	0.579	0.458
2020.00	1.000	0.562	0.255	3.757	537.000	0.579	0.562
2040.00	1.000	0.471	0.255	3.757	544.000	0.578	0.471
2060.00	1.000	0.361	0.255	3.757	563.000	0.570	0.361
2080.00	1.000	0.637	0.255	3.757	578.000	0.572	0.637
2100.00	1.000	0.446	0.255	3.757	596.000	0.568	0.446
2120.00	1.000	-	0.255	3.757	596.000	0.568	-
2140.00	1.000	-	0.255	3.757	596.000	0.568	-
2160.00	1.000	-	0.255	3.757	596.000	0.568	-
2180.00	0.552	-	0.255	3.757	596.000	0.568	-
2200.00	0.000	-	0.255	3.757	596.000	0.568	-
2220.00	0.000	-	0.255	3.757	596.000	0.568	-
2240.00	0.000	-	0.255	3.757	596.000	0.568	-
2260.00	0.000	-	0.255	3.757	596.000	0.568	-
2280.00	0.000	-	0.255	3.757	596.000	0.568	-
2300.00	0.000	-	0.255	3.757	596.000	0.568	-
2320.00	0.000	-	0.255	3.757	596.000	0.568	-
2340.00	0.000	-	0.255	3.757	596.000	0.568	-
2360.00	0.000	-	0.255	3.757	596.000	0.568	-
2380.00	0.000	-	0.255	3.757	596.000	0.568	-
2400.00	0.000	-	0.255	3.757	596.000	0.568	-
2420.00	0.000	-	0.255	3.757	596.000	0.568	-
2440.00	0.000	-	0.255	3.757	596.000	0.568	-
2460.00	0.753	0.277	0.255	3.757	599.000	0.567	0.277
2480.00	1.000	-	0.255	3.757	599.000	0.567	-
2500.00	1.000	-	0.255	3.757	599.000	0.567	-
2520.00	1.000	-	0.255	3.757	599.000	0.567	-

Figure 186: History Report

Statistics Report-Summary by Total

Double-click the heading **Total** at the top of the Total column in the Statistics Report main window to call up the Summary by Total Report.

This table displays the sum of all measurement values for all current scripts.

Statistical Measurement	Test Summary Max	Test Summary Count	Test Summary Sum	Test Summary Average	Test Summary STDev	Current Slice Min	Current Slice Max	Current Slice Count	Current Slice Sum	Current Slice Average	Current Slice STDev
Page Time	92.416	37.000	-	28.253	-	-	-	-	-	16.266	-
Pages	5.000	-	37.000	1.423	-	-	-	-	1.000	-	-
Pages Per Second	0.250	-	-	0.071	-	-	-	-	0.050	-	-
Hit Time	2.288	822.000	-	1.272	-	-	-	-	-	0.414	-
Hits	96.000	-	822.000	27.400	-	-	-	-	2.000	-	-
Successful Hits	96.000	-	822.000	27.400	-	-	-	-	2.000	-	-
Hits Per Second	4.800	-	-	1.370	-	-	-	-	0.100	-	-
Successful Hits Per Second	4.800	-	-	1.370	-	-	-	-	0.100	-	-
Attempted Connections	56.000	-	524.000	17.467	-	-	-	-	0.000	-	-
Successful Connections	56.000	-	524.000	17.467	-	-	-	-	0.000	-	-
Connect Time	0.627	822.000	-	0.289	-	-	-	-	-	0.000	-
Process Time	0.003	822.000	-	0.001	-	-	-	-	-	0.000	-
Receive Time	0.969	822.000	-	0.498	-	-	-	-	-	0.203	-
Send Time	0.001	822.000	-	0.000	-	-	-	-	-	0.000	-
Time To First Byte	0.646	822.000	-	0.448	-	-	-	-	-	0.211	-
Throughput (Bytes Per Second)	60190.700	-	-	12557.500	-	-	-	-	689.500	-	-
Response Time	1.560	822.000	-	0.945	-	-	-	-	-	0.414	-
Response Data Size	1203814.000	-	7534500.000	251150.000	-	-	-	-	13790.000	-	-

Double-click on a Measurement (left-hand column) to view the History Report for the selected Measurement for this agenda. Load Size: 10 Time: 600

Figure 187: Summary by Total Report

The names of all the measurements tracked during the current Load Session are listed in the column at the far left. The names of all the attributes totaled for the relevant measurements for all current scripts are displayed in the header at the top of the table.

Getting More Detailed Information

Double-click a measurement name in the left-hand column to call up a History Report. This History Report can also be accessed from the Statistics Report main window by double-clicking the button located at the Intersection of the selected Measurement row and Total column. This History Report is described in *Statistics Report-History by Measurement and Total* (on page 340).

Example:

Double-click **Response Time** in the left-hand column to call up a History Report. This History Report lists all of the Response Time Attribute values recorded for all scripts from the start of the current test session. This is the same Summary Report accessed from the Statistics Report main window by double-clicking the button located at the intersection of the Response Time row and Total column.

Time	Total Load Size	Current Value (Current Slice Average)	Test Summary Min	Test Summary Max	Test Summary Count	Test Summary Average	Current Slice Average (Current Value)
300.00	7.029	0.144	0.144	1.560	542.000	1.190	0.144
320.00	8.000	0.474	0.144	1.560	604.000	1.116	0.474
340.00	8.000	0.480	0.144	1.560	622.000	1.098	0.480
360.00	8.000	0.188	0.144	1.560	624.000	1.095	0.188
380.00	8.000	0.440	0.144	1.560	651.000	1.068	0.440
400.00	8.028	0.446	0.144	1.560	658.000	1.061	0.446
420.00	9.000	0.586	0.144	1.560	683.000	1.044	0.586
440.00	9.000	0.464	0.144	1.560	706.000	1.025	0.464
460.00	9.000	-	0.144	1.560	706.000	1.025	-
480.00	9.000	0.464	0.144	1.560	727.000	1.009	0.464
500.00	9.027	0.430	0.144	1.560	742.000	0.997	0.430
520.00	10.000	0.470	0.144	1.560	772.000	0.976	0.470
540.00	10.000	0.468	0.144	1.560	787.000	0.967	0.468
560.00	10.000	-	0.144	1.560	787.000	0.967	-
580.00	10.000	0.456	0.144	1.560	820.000	0.946	0.456
600.00	9.881	0.414	0.144	1.560	822.000	0.945	0.414

Load Size: 10 Time: 600

Figure 188: History Report

Statistics Report-Summary by Measurement

Double-click a specific measurement name in the Statistics Report main window to call up the following summary table.

This table displays all attribute values for the selected measurement for the current scripts.

Script Names	Test Summary Min	Test Summary Max	Test Summary Sum	Test Summary Aver...	Current Slice Sum
Total	38.000	40.000	558.000	39.857	40.000
Test::Total	38.000	40.000	558.000	39.857	40.000
Test@10.0.1.186	38.000	40.000	558.000	39.857	40.000
_Performance@10.0.1.186	-	-	-	-	-
RPM@10.0.1.186	-	-	-	-	-

Double-click on a Script name (left-hand column) to view the History Report of this Measurement, for the selec Load Size: 1 Time: 280

Figure 189: Summary Table

Figure 189 shows the Response Time values for script, Test. The attributes displayed are for Min, Max, Count, Average and Current Slice. The third line displays the total values for these attributes for all running scripts.

Getting More Detailed Information

Double-click a script name in the left-hand column to call up a History Report. This History Report can also be accessed from the Statistics Report main window by double-clicking the button located at the Intersection of the selected measurement row and script column. This History Report is described in *Statistics Report-History by Measurement and* (on page 339).

Example:

Double-click **Test** in the left-hand column to call up a History Report. This History Report lists all of the Response Time Attribute values recorded for all scripts from the start of the current test session. This is the same History Report accessed from the Statistics Report main window by double-clicking the button located at the intersection of the Response Time row and script column.

Time	Total Load Size	Current Value (Current Slice Average)	Test Summary Min	Test Summary Max	Test Summary Count	Test Summary Average	Current Slice Average (Current Value)
300.00	7.029	0.144	0.144	1.560	542.000	1.190	0.144
320.00	8.000	0.474	0.144	1.560	604.000	1.116	0.474
340.00	8.000	0.480	0.144	1.560	622.000	1.098	0.480
360.00	8.000	0.188	0.144	1.560	624.000	1.095	0.188
380.00	8.000	0.440	0.144	1.560	651.000	1.068	0.440
400.00	8.028	0.446	0.144	1.560	658.000	1.061	0.446
420.00	9.000	0.586	0.144	1.560	683.000	1.044	0.586
440.00	9.000	0.464	0.144	1.560	706.000	1.025	0.464
460.00	9.000	-	0.144	1.560	706.000	1.025	-
480.00	9.000	0.464	0.144	1.560	727.000	1.009	0.464
500.00	9.027	0.430	0.144	1.560	742.000	0.997	0.430
520.00	10.000	0.470	0.144	1.560	772.000	0.976	0.470
540.00	10.000	0.468	0.144	1.560	787.000	0.967	0.468
560.00	10.000	-	0.144	1.560	787.000	0.967	-
580.00	10.000	0.456	0.144	1.560	820.000	0.946	0.456
600.00	9.881	0.414	0.144	1.560	822.000	0.945	0.414

Load Size: 10 Time: 600

Figure 190: History Report

Statistics Report-History by Measurement and Script

To open the History by Measurement and Script Report:

- Double-click a button located at the intersection of a specific measurement row and script column in the Statistics Report main window.
- Double-click a selected script name in the left hand column of a Summary by Measurement Report.
- Double-click a selected measurement name in the far left column of a Summary by Script Report.

The History by Measurement and Script Report displays all the Attribute values recorded from the start of the current Load Session for the selected measurement and script.

The following table displays all of the Response Time Attribute values for the script, Test.

Time	Total Load Size	Current Value (Current Slice Average)	Test Summary Min	Test Summary Max	Test Summary Count	Test Summary Average	Current Slice Average (Current Value)
300.00	7.029	0.144	0.144	1.560	542.000	1.190	0.144
320.00	8.000	0.474	0.144	1.560	604.000	1.116	0.474
340.00	8.000	0.480	0.144	1.560	622.000	1.098	0.480
360.00	8.000	0.188	0.144	1.560	624.000	1.095	0.188
380.00	8.000	0.440	0.144	1.560	651.000	1.068	0.440
400.00	8.028	0.446	0.144	1.560	658.000	1.061	0.446
420.00	9.000	0.586	0.144	1.560	683.000	1.044	0.586
440.00	9.000	0.464	0.144	1.560	706.000	1.025	0.464
460.00	9.000	-	0.144	1.560	706.000	1.025	-
480.00	9.000	0.464	0.144	1.560	727.000	1.009	0.464
500.00	9.027	0.430	0.144	1.560	742.000	0.997	0.430
520.00	10.000	0.470	0.144	1.560	772.000	0.976	0.470
540.00	10.000	0.468	0.144	1.560	787.000	0.967	0.468
560.00	10.000	-	0.144	1.560	787.000	0.967	-
580.00	10.000	0.456	0.144	1.560	820.000	0.946	0.456
600.00	9.881	0.414	0.144	1.560	822.000	0.945	0.414

Figure 191: History by Measurement and Script Report

The times that the attribute values were recorded since the start of the current Load Session are listed in the column at the far left. The names of all the attributes totaled for the selected measurement and script are displayed in the header at the top of the table.

Example:

This table displays all Total Load Size, Current Value, Min, Max, Count, and Average values recorded for Response Time for Test. Values were recorded starting at 20 seconds after the Load Session began.

Statistics Report-History by Measurement and Total

To open the History by Measurement and Total Report:

- Double-click a button located at the intersection of a specific measurement row and the Total column in the Statistics Report main window,
- Or-
- Double-click the specific measurement name in the far left column of a Summary by Total Report.

The History by Measurement and Total Report displays all the Attribute values recorded from the start of the current Load Session for the selected measurement. These values are summed for all scripts:

Time	Total Load Size	Current Value (Current Slice Average)	Test Summary Min	Test Summary Max	Test Summary Count	Test Summary Average	Current Slice Average (Current Value)
300.00	7.029	0.144	0.144	1.560	542.000	1.190	0.144
320.00	8.000	0.474	0.144	1.560	604.000	1.116	0.474
340.00	8.000	0.480	0.144	1.560	622.000	1.098	0.480
360.00	8.000	0.188	0.144	1.560	624.000	1.095	0.188
380.00	8.000	0.440	0.144	1.560	651.000	1.068	0.440
400.00	8.028	0.446	0.144	1.560	658.000	1.061	0.446
420.00	9.000	0.586	0.144	1.560	683.000	1.044	0.586
440.00	9.000	0.464	0.144	1.560	706.000	1.025	0.464
460.00	9.000	-	0.144	1.560	706.000	1.025	-
480.00	9.000	0.464	0.144	1.560	727.000	1.009	0.464
500.00	9.027	0.430	0.144	1.560	742.000	0.997	0.430
520.00	10.000	0.470	0.144	1.560	772.000	0.976	0.470
540.00	10.000	0.468	0.144	1.560	787.000	0.967	0.468
560.00	10.000	-	0.144	1.560	787.000	0.967	-
580.00	10.000	0.456	0.144	1.560	820.000	0.946	0.456
600.00	9.881	0.414	0.144	1.560	822.000	0.945	0.414

Load Size: 10 Time: 600

Figure 192: History by Measurement and Total Report

Listed in the far left-hand column are the times that attribute values were recorded since the beginning of the current Load Session. The names of all the attributes totaled for the selected measurement are displayed in the header at the top of the table.

Data Drilling Reports

The WebLOAD Data Drilling Reports provide a graphic illustration of the problematic tables, making it simple to locate and correct the source of the error.

Data Drilling reports are available after the Load Session has completed.

Data Drilling Reports

Data Drilling provides both a global and detailed account of hit successes and failures allowing you to verify the functional integrity of your Web application at the per-client, per-transactions and per-instance level. The Data Drilling reports provide an extremely detailed yet easily accessible summary of all the statistical, timing and performance information collected over the course of the test session, which includes:

- A named entry for each HTTP action in the script. Information is automatically saved for all actions tested using the Functional Verification Wizard. Information is also saved for all user-defined or manually named transactions.

Click a specific entry to zoom in on the detailed information available for that action. For each HTTP action tested and verified by your test script, WebLOAD Console provides a complete breakdown of all relevant information, including:

- A hierarchical tree of all the 'sub-steps' needed to complete the specified action. For example, a single Get of a simple Web page may include multiple 'sub-Gets' or hits of the various frames found on that page, each with its own pictures and tables, each of which is also listed with its own 'sub-sub-Get'.



Note: WebLOAD Console works with the 'visual set' of Web page elements associated with each action. Only the targeted frames that were actually accessed during a specific action will be saved as part of the DOM for that action, and not necessarily every single peripheral frame associated with the parent Web page.

- Detailed information about the time required for each action and sub-transaction, including low-performance flags for actions that complete correctly, but take longer than a user-set time threshold to complete.

- A hierarchical tree of all the objects accessed or otherwise associated with the specified action, together with detailed information about the time required to access each object. The timing information is broken down into the time required for each HTTP activity associated with that object, including connect time, send time, wait time (time to first byte), and final receive time.
- Entries for every timer triggered during the test session, including user-defined timers, WebLOAD-defined built-in timers, and automatic timers triggered each time ActiveX or Java object methods are activated. Click a timer entry to bring up detailed information about that timer.
- Entries for every error event triggered during the test session by both WebLOAD Console and user-defined verification functions. Click an event entry to bring up detailed information about that event.



Note: Data Drilling is only available to you if:

- You initiated user-defined transactions in the scripting for functional testing.
- You configured Automatic data collection in the **Functional Testing** tab of the Script Options dialog box or Default/Current Session Options dialog boxes.

Opening a Data Drilling Report

To open a Data Drilling report:

- Select **Open Data Drilling** in the **Session** tab of the ribbon.

Viewing the Transaction Grid

The Transaction Grid is the entry point for accessing the Data Drilling reports provided by WebLOAD Console.

Transaction Name	Total Count	Successful Count	Failed Count	Marked Count
SourceAccount	6	6	-	-
UpdateAccount	5	5	-	-

Figure 193: Transaction Grid

The Transaction Grid displays summary information for all user-defined and named transactions incorporated in the Script including:

Table 49: Transactions Grid Display

Attribute	Explanation
Transaction Name	The name of the transaction.
Total Count	The total number of times the transaction was executed.

Attribute	Explanation
Successful Count	The number of successful executions of the transaction.
Failed Count	The number of failed executions of the transaction.
Marked Count	The number of times a lower level transaction, meaning a transaction nested within a higher level transaction, failed within the current transaction.



Note: Some of the transactions include both successes and failures.

Getting More Detailed Information

To see why a particular transaction failed, select the Transaction Name on the Transaction Grid to display a Transaction Reason Failure Grid.

Viewing the Transaction Reason Failure Grid

The Transaction Reason Failure grid lists the reasons for the transaction failure, together with the total failure count for that transaction.

Transaction Name	Total Count	Successful Count	Failed Count	Marked Count
POST_http://www.imeem.com/api/GetRenderedRelat	9	9	-	-
GET_https://www.webloadmpstore.com/checkout.php	4	3	1	-
GET_http://googleads.g.doubleclick.net/pagead/goog	12	12	-	-
GET_http://www.youtube.com/v/GvhJh7vZ7U Page	4	4	-	-
GET_http://www.webloadmpstore.com/login.php Page	4	4	-	-
GET_https://www.webloadmpstore.com/checkout.php	Count			
Transaction timeout	1			

Figure 194: Transaction Reason Failure Grid

In Figure 194 the UpdateAccount transaction is highlighted in the Transaction Grid. The Reason for the UpdateAccount transaction failure, Transfer incomplete displays immediately below the Transaction Grid. The Transaction Failure Reason Grid displays why two of the UpdateAccount transactions failed.

Getting More Detailed Information

To track down and correct the underlying cause of the failure, you want to discover which transactions failed. Double-click the reason failure row to open the Instance Grid.

Viewing the Instance Grid

Severity	Reason	Time	Duration	Total Load Size	Thread Number	Round Number	Load Generator
		130.014	0.328	1	0	1	Quickstart@jryakovb
	Transaction timeout	574.576	100.344	1	0	2	Quickstart@jryakovb
		867.019	0.797	1	0	4	Quickstart@jryakovb
		1329.334	0.625	1	0	5	Quickstart@jryakovb
		1800.078	0.438	1	0	6	Quickstart@jryakovb

Figure 195: Instance Grid

The Instance Grid lists the individual instances of transaction failure, together with specific details about each instance.

The following information is included for each transaction instance displayed in the grid:

Table 50: Transaction Information

Attribute	Explanation
Severity	The severity level of this Transaction Instance failure, based on the return code set by the script in the <code>Verification</code> function.
Reason	The reason that the Transaction Instance failed based on information supplied by the script in the <code>SetFailureReason</code> function.
Time	The time that this Transaction Instance began execution, relative to the beginning of the current test session.
Duration	The time it took to complete the Transaction Instance. If the Transaction Instance failed, the duration is the time until the transaction failed.
Total Load Size	The total number of Virtual Clients running on the Load Generator during this Transaction Instance.
Thread Number	The number of the current thread running during this Transaction Instance.
Round Number	The number of the current round running during this Transaction Instance.
Load Generator	The name of the Load Generator that executed the Transaction Instance.

The Instance Grid incorporates functional testing with performance testing and provides you with a detailed description of exactly which Transaction Instance failed, why it failed, and what else occurred at that point in the script execution for that Load Generator.

Getting More Detailed Information

To track down and correct the underlying cause of the failure you might still need help understanding how all of this information fits together. Double-click one of the Transaction Instances to display a complete Parent Transaction Instance Tree.

Viewing the Parent Transaction Instance Tree

The Parent Transaction Instance Tree:

- Provides a graphic presentation of the relationships between all of the parent and children transactions, providing a detailed breakdown of all sub-events that occurred as part of the selected Transaction. For example, a transaction may involve a series of steps, including opening a connection, sending a signal, waiting, receiving a response, etc.
- Lists all of the details about the selected Instance provided in the Instance Grid.

A transaction is selected in the tree at the left. The fact that this transaction triggered an error is indicated by the blue flag icon next to the transaction and the blue color of the transaction itself.

The properties for the selected Transaction Instance are displayed in a table on the right side of the window. The Transaction Instance Tree provides a broad understanding of how the failed Transaction Instance fits into the general script execution.

Property	Value
Load Generator Name	TransactionTime_TimeoutSeverity.Generator1@natally2000
Reason	Transaction timeout
Time	3.205
Duration	0.030
Total Load Size	1
Thread Number	0
Round Number	12
Severity	WLError

Figure 196: Parent Transaction Instance Tree

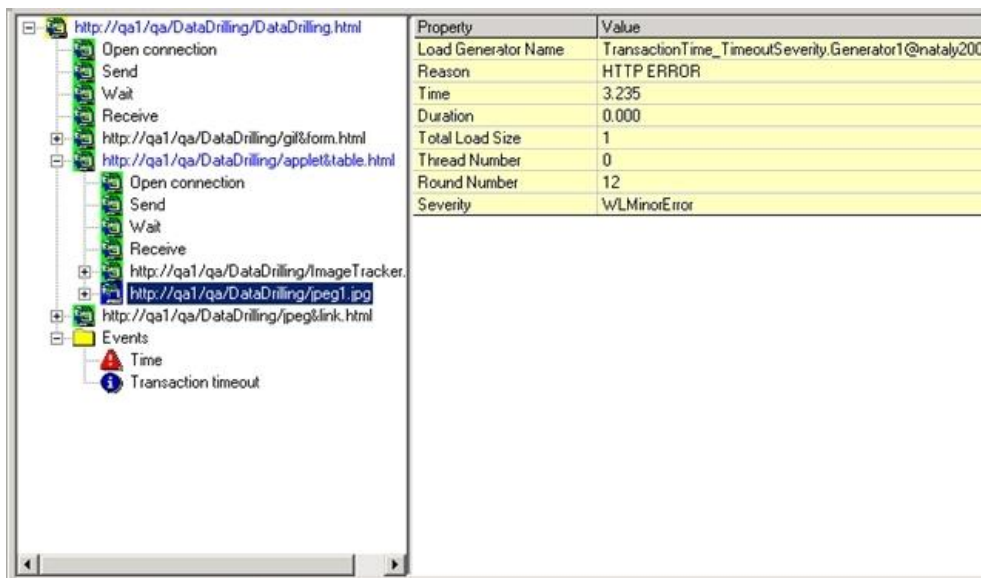
Getting More Detailed Information

For even more detailed information you may wish to see a breakdown of the transaction steps that were part of the parent transaction. For example, one of the child branches in the preceding Transaction Instance Tree is also marked in blue, indicating that an error occurred at that point. Click the error-flagged child branch in the Transaction Instance Tree to see a detailed breakdown of that specific event.

Viewing the Child Transaction Instance Tree

Child transactions are selected on the tree at the left. The fact that the transaction triggered an error is indicated by the blue flag on the icon next to the transaction and the blue color of the transaction itself in the tree. Details of the error are provided in the table at the right.

In the Figure 197 below, the selected child transaction triggered an HTTP ERROR of severity Minor Error. The corresponding error entry in the Log Window is highlighted.



Property	Value
Load Generator Name	TransactionTime_TimeoutSeverity.Generator1@nataly2000
Reason	HTTP ERROR
Time	3.235
Duration	0.000
Total Load Size	1
Thread Number	0
Round Number	12
Severity	WLMInorError

Figure 197: Child Transaction Instance Tree

To see more information about the actual event that triggered this error, double-click the child transaction branch in the tree to expand its own sub tree, including its own Events sub-branch. Double-click more entries under the Events sub-branch to display more information about each selected item on the table at the right.

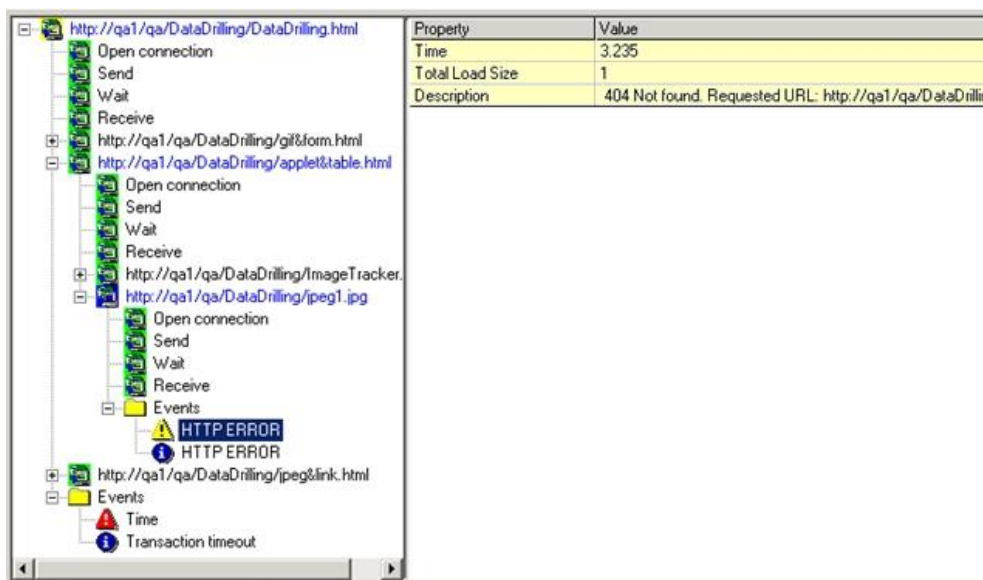


Figure 198: Child Transaction Instance Sub-Tree

By working your way through the multiple levels of the Data Drilling reports you can achieve a complete picture of the exact point at which your script failed, along with a complete history of the general system and events that led up to the failure. This level of detailed information simplifies your work when testing and debugging your website.

Exporting Performance Reports

Performance Reports are the product of your test session. Performance Reports enable you to combine several different measurements into a single report. You are able to view the data accumulated in the NT and Unix Performance Monitors together with the performance data from the Load Generators and Probing Clients. All of the data displayed in the Integrated Reports and Statistics can be exported to HTML, Microsoft Excel and to Tab files. Statistics from external files can also be imported to your reports.

Report Export / Import Options

WebLOAD Console enables you to export the following types of reports, and import external statistics:

- Reports containing only the data specified.
- General reports containing all the data from your test session.

Access the export / import options through the Reports menu in the menu bar.

The following table describes each export / import option.

Table 51: Export / Import Options

Category	Description
Export	Exports selective WebLOAD Console report data.
Export All	Exports all of the statistical data gathered by WebLOAD Console throughout the test session.
Import External Statistics	Imports statistical data from an external file into your report.

For information on importing external statistics, see *Importing External Statistics* on page 407.

Exporting Selective WebLOAD Report Data

The following is a graphic presentation of an Integrated Report. This report displays the data for Load Size, Hits Per Second, Response Time, and Process Time.

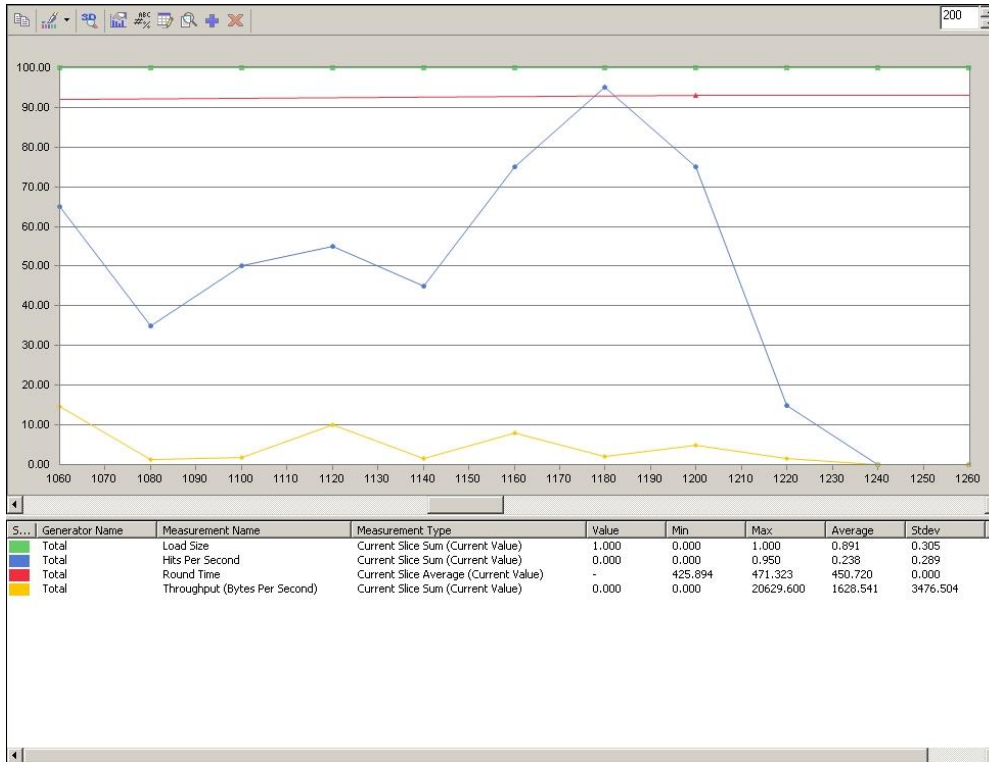


Figure 199: Integrated Report Graphic Representation

The screen below displays the data for the Response Time in tabular format.

Time	Total Load Size	Current Value (Current Slice Average)	Test Summary Min	Test Summary Max	Test Summary Count	Test Summary Average	Current Slice Average (Current Value)
40.00	5.000	0.806	0.806	0.827	152.000	0.816	0.806
60.00	5.000	0.836	0.806	0.836	230.000	0.822	0.836
80.00	5.000	0.942	0.806	0.942	245.000	0.830	0.942
100.00	5.000	0.959	0.806	0.959	250.000	0.832	0.959
120.00	5.998	0.483	0.483	0.959	405.000	0.699	0.483
140.00	6.000	0.502	0.483	0.959	450.000	0.679	0.502
160.00	6.000	0.789	0.483	0.959	458.000	0.681	0.789
180.00	6.000	-	0.483	0.959	458.000	0.681	-
200.00	6.000	0.459	0.459	0.959	491.000	0.666	0.459
220.00	6.994	0.659	0.459	0.959	569.000	0.665	0.659
240.00	7.000	0.570	0.459	0.959	685.000	0.649	0.570
260.00	7.000	0.686	0.459	0.959	735.000	0.652	0.686
280.00	7.000	0.601	0.459	0.959	757.000	0.650	0.601
300.00	7.000	0.488	0.459	0.959	933.000	0.619	0.488
320.00	7.999	0.849	0.459	0.959	941.000	0.621	0.849
340.00	8.000	0.657	0.459	0.959	968.000	0.622	0.657
360.00	8.000	0.736	0.459	0.959	979.000	0.624	0.736
380.00	8.000	1.899	0.459	1.899	981.000	0.626	1.899
400.00	8.000	0.796	0.459	1.899	993.000	0.628	0.796
420.00	8.992	0.855	0.459	1.899	1038.000	0.638	0.855
440.00	9.000	0.690	0.459	1.899	1151.000	0.643	0.690

Figure 200: Response Time Data Table

All examples in this section refer to these reports.

Exporting Performance Reports to Excel

You can view all of the data available in the Performance Reports from a single Excel spreadsheet. After you have exported a Performance Report to Excel, you work with the exported spreadsheet as you would with any other Excel spreadsheet.

To export Performance Reports to Excel:

1. Display the Integrated Report you want to export. If there is more than one report open, focus on the report you want to export.
2. Click **Export Report** ► **Export to Excel** in the **Session** tab of the ribbon.

The report is sent to Excel, which opens automatically and displays the Integrated Report.



Notes:

If Excel is not installed, you receive the following error: Cannot Open Excel.

Exporting a large amount of data to Excel may take some time. Wait until the export to Excel has completed before continuing work with the Console.

3. Save the Excel file.

The following screen shows an Integrated Report after it is exported to an Excel spreadsheet. This screen shows a tabular presentation for a graphic screen displayed in WebLOAD Console.

	A	B	C	D	E	F
1	Time	Total/ Load Size/ Current Value	Total/ Hits Per Second/ Current Value	Total/ Response Time/ Current Value	Total/ Process Time/ Current Value	
2	20	3.855	0.55	2.746	0.021	
3	40	4	1.6	2.667	0.011	
4	60	4	3.25	1.246	0.001	
5	80	7.955	1.7	1.587	0.001	
6	100	8	1.1	8.101	0.011	
7	120	8	2	2.638	0.001	
8	140	11.92	1.6	3.345	0	
9	160	12	1.55	4.773	0.002	
10	180	12	2.65	6.977	0.014	
11	200	15.96	3.1	3.867	0.003	
12	220	16	3.35	2.746	0.001	
13	240	16	2.75	5.613	0.005	
14	260	16	0.15	2.393	0	
15						

Figure 201: Integrated Report in Excel Spreadsheet

Features of this report:

- The first column contains the times (in seconds) that the measurement values were recorded.
- The top row contains the name of each measurement in the following format: Script / Measurement / Attribute. The names of the measurements are color-coded to match the measurements in the Chart View of the report.
- Integrated Reports displayed in Chart View are exported to Excel in tabular format.

See *Customizing the Export File* (on page 360), for information on configuring the sampling frequency and the delimiters of your exported report.

Exporting Performance Reports to a Tab File

To export Performance Reports to a Tab file:

1. Display the report you want to export. If there is more than one report open, focus on the report you want to export.
2. Click **Export Report** ► **Export to Tab File** in the **Session** tab of the ribbon.
The Windows Save As dialog box appears.
3. Enter the name of your new Tab file. The file has a *.tab extension.
4. Click **Save**.

Once the file is saved, it is displayed. Configure the viewer and the view options using the Export options dialog box, see *Customizing the Export File* (on page 360).

Figure 202 shows an Integrated Report after it is exported to a Tab file and opened in Microsoft Word (using tab as a delimiter). You can change the presentation of this file using the Export options dialog box.

Time	Total/LoadSize	Current value	Min	Max	Count	Average
20	3.855	2.746	2.746	2.746	11.000	2.746
40	4.000	2.667	2.667	2.746	43.000	2.687
60	4.000	1.246	1.246	2.746	108.000	1.820
80	7.955	1.587	1.246	2.746	142.000	1.764
100	8.000	8.101	1.246	8.101	164.000	2.614
120	8.000	2.638	1.246	8.101	204.000	2.619
140	11.920	3.345	1.246	8.101	236.000	2.717
160	12.000	4.773	1.246	8.101	267.000	2.956
180	12.000	6.977	1.246	8.101	320.000	3.622
200	15.960	3.867	1.246	8.101	382.000	3.662
220	16.000	2.746	1.246	8.101	449.000	3.525
240	16.000	5.613	1.246	8.101	504.000	3.753
260	16.000	2.393	1.246	8.101	507.000	3.745

Figure 202: Integrated Report in Tab File

Features of this Report:

- The first column contains the times (in seconds) that the measurement values were recorded.
- The top row contains of the name of each measurement in the following format: Script / Measurement / Attribute.
- Integrated Reports displayed in Chart View is exported to Tab files in tabular format.

See *Customizing the Export File* (on page 360), for information on configuring the sampling frequency, the delimiters of your exported table, the viewer, and the view options.

Exporting Performance Reports to HTML

To export Performance Reports to HTML:

1. Display the report you want to export. If there is more than one report open, focus on the report you want to export.
2. Click **Export Report** ► **Export to HTML File** in the **Session** tab of the ribbon.

The report is displayed in HTML.

Figure 203 shows an Integrated Report after it is exported to HTML.

Time	Total/LoadSize	Current value	Min	Max	Count	Average
20	3.855	2.746	2.746	2.746	11.000	2.746
40	4.000	2.667	2.667	2.746	43.000	2.687
60	4.000	1.246	1.246	2.746	108.000	1.820
80	7.955	1.587	1.246	2.746	142.000	1.764
100	8.000	8.101	1.246	8.101	164.000	2.614
120	8.000	2.638	1.246	8.101	204.000	2.619
140	11.920	3.345	1.246	8.101	236.000	2.717
160	12.000	4.773	1.246	8.101	267.000	2.956
180	12.000	6.977	1.246	8.101	320.000	3.622
200	15.960	3.867	1.246	8.101	382.000	3.662
220	16.000	2.746	1.246	8.101	449.000	3.525
240	16.000	5.613	1.246	8.101	504.000	3.753
260	16.000	2.393	1.246	8.101	507.000	3.745

Figure 203: Integrated Report in HTML

Features of this Report:

- The first column contains the times (in seconds) that the measurement values were recorded.
- The top row contains of the name of each measurement in the following format: Script / Measurement / Attribute.
- Integrated Reports displayed in Chart View is exported to HTML in tabular format.

See *Customizing the Export File* (on page 360), for information on configuring the sampling frequency, the delimiters of your exported table, the viewer, and the view options.

Exporting All Statistical Data

WebLOAD Console provides a wealth of statistical information about the performance of the Web system under test. All of the statistical data gathered by WebLOAD Console can be exported to Microsoft Excel, Tab files and HTML. Exporting the WebLOAD Console statistical data allows you to view and manipulate your data and perform cross report comparisons.

Exporting All Statistics to Excel

To export all Statistics to Excel:

- Click **Export All** ► **Export All to Excel** in the **Session** tab of the ribbon.

Excel opens automatically and displays all of the Statistics gathered by WebLOAD Console.

The following screen shows the Statistics as they appear when exported to Excel.

Time	Agenda1.Generator1@localhost / Transactions Per Second/Min	Agenda1.Generator1@localhost / Transactions Per Second/Max	Agenda1.Generator1@localhost / Transactions Per Second/Average	Agenda1.Generator1@localhost / Transactions Per Second/Current Value
20	0.55	0.55	0.55	0.55
40	0.55	1.6	1.075	1.6
60	0.55	3.25	1.8	3.25
80	0.55	3.25	1.775	1.7
100	0.55	3.25	1.64	1.1
120	0.55	3.25	1.7	2
140	0.55	3.25	1.685	1.6
160	0.55	3.25	1.668	1.55
180	0.55	3.25	1.777	2.65
200	0.55	3.25	1.91	3.1
220	0.55	3.35	2.04	3.35
240	0.55	3.35	2.1	2.75
260	0.15	3.35	1.95	0.15

Figure 204: Statistics in Excel

Features of this report:

- The first column contains the times (in seconds) that the measurement values were recorded.
- The top row contains the name of each measurement in the following format: Script / Measurement / Attribute.
- Data is exported to Excel by columns. When the end of a sheet is reached, a new Excel sheet is opened.

See *Customizing the Export File* (on page 360), for information on configuring the sampling frequency and the delimiters of your exported table.

Exporting All Statistics to a Tab File

To export all statistics to a Tab file:

1. Click **Export All** ► **Export All to Tab File** in the **Session** tab of the ribbon.
The Windows Save As dialog box appears.
2. Enter the name of your new Tab file. The file has a *.tab extension.
3. Click **Save**.

Once the file is saved, the data is displayed.

Configure the viewer and the view options in the Export options dialog box, see *Customizing the Export File* (on page 360).

Change the presentation of this file using the Export options dialog box.

- The left column contains the times (in seconds) that the measurement values were recorded.
- The top row contains the name of each measurement in the following format: Script / Measurement / Attribute.

See *Customizing the Export File* (on page 360), for information on configuring the sampling frequency, the delimiters of your exported table, the viewer, and the view options.

Exporting All Statistics to HTML

To export all statistics to HTML:

- Click **Export All** ► **Export All to HTML File** in the **Session** tab of the ribbon.

Change the presentation of this file using the Export options dialog box.

- The left column contains the times (in seconds) that the measurement values were recorded.
- The top row contains the name of each measurement in the following format: Script / Measurement / Attribute.

See *Customizing the Export File* (on page 360), for information on configuring the sampling frequency, the delimiters of your exported table, the viewer, and the view options.

Customizing the Export File

Before data is exported you can define the way the data is displayed in the exported file.

To customize the export file:

1. Click **Global Options** in the **Tools** tab of the ribbon,
-Or-
Select **Global Options** from the Console System button.
The Global Options dialog box opens.
2. Select the **Export** tab.
The Export tab is displayed.

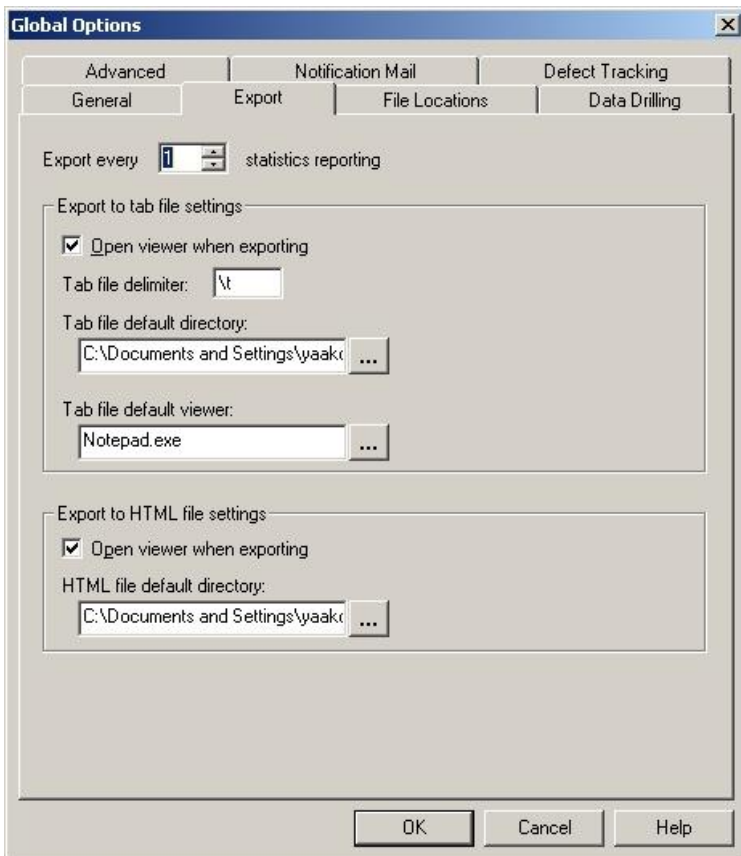


Figure 205: Global Options Export Tab

The Export tab contains the following fields:

Table 52: Export Dialog Box Fields

Field Name	Description
Export every __ statistics reporting	The frequency of the number of samples exported. For example, if 1 is entered for this value, WebLOAD Console exports every sample taken. If 2 is entered for this value, WebLOAD Console exports every second sample. When configuring this option, you must consider the setting selected for sampling. If for example, the Statistics Reporting Resolution is set to 20 seconds and Export Every Statistics Reporting is set to 2, the measurement exported reflects a sampling every 40 seconds. The measurements are the measurements taken at the intervals: 20, 60, 100, 140... This option is useful in cases where there is a large amount of data due to a long test session.
Open viewer when exporting	Select this checkbox to launch the Tab File and HTML Default Viewer after the Tab or HTML file is exported. This is selected by default
Tab File delimiter	Definition of the delimiter used when exporting the statistical data to a tab file. All strings can be used as delimiters by entering the string in the Tab File Delimiter field. <ul style="list-style-type: none"> • To use tab characters as delimiters enter \t. • To use a new line as a delimiter, enter \r or \n. • To use a space as a delimiter, use \s.
Tab\HTML File default directory	The default directory for saving Tab and HTML files.
Tab File default viewer	The default viewer to open after the Tab file is exported.

3. Click **OK**.

Performance Measurements Manager

WebLOAD Console provides the Performance Measurements Manager (PMM) for monitoring the performance of various server-side applications, databases, stream technology, system, and Web server measurements in real-time while your test is running.

Using the Performance Measurements Manager

Using the PMM, you configure WebLOAD Console to collect data for the hosts and services available from the server, to provide a complete picture of application performance while you are testing your Web application. Thus, in addition to the data accumulated from load testing, you receive data on the performance of your server.

Using the PMM, you can monitor:

- Application Server Resources
- Database Resources
- Stream Technology Resources
- System Resources
- Web Server Resources

About the Performance Measurements Manager

Using the PMM you can monitor a wide range of performance objects such as application data, database, Web server, stream technology, and system performance statistics in real-time during your test sessions. Each performance object, or data source, provides measurements that represent detailed data on specific aspects of the system or service.

Use the performance data collected to:

- Understand the effect of the load on your system's resources.
- Observe changes in resource usage so you can plan for future upgrades.

- Test application changes by monitoring the results.
- Diagnose problems and target components or processes for optimization.

You set up the data sources to monitor in the PMM, including the hosts and measurements. You can maintain multiple configurations of servers, hosts and measurements to monitor.

Each configuration includes:

- Data sources
- Hosts
- Selected measurements from each data source on each host

To activate the monitors, you must configure the list of data sources, hosts and measurements to monitor before running your load test. While running your test, you can create Integrated Reports to display performance statistics for any monitors configured in the PMM. Use the PMM Wizard to add data sources to the PMM main window and create configuration files.

The PMM can be run from within WebLOAD Console or as a standalone tool. Run the PMM from within WebLOAD Console to create and define the performance objects to monitor and assign the configuration to a WebLOAD template, or run the standalone tool to create a library of configuration files for use in future WebLOAD tests.

Opening the Performance Measurements Manager

The PMM can be opened as a standalone tool for configuring the data to monitor before you even define your test or from within WebLOAD Console to assign PMM configurations to your test template.

- To open the PMM from within WebLOAD Console, click **Performance Measurements Manager** in the **Session** tab of the ribbon.
- To open the PMM as a standalone tool select **Start > Programs > RadView > WebLOAD > Utilities > Performance Measurements Manager**.

The PMM opens with no configured data sources listed.

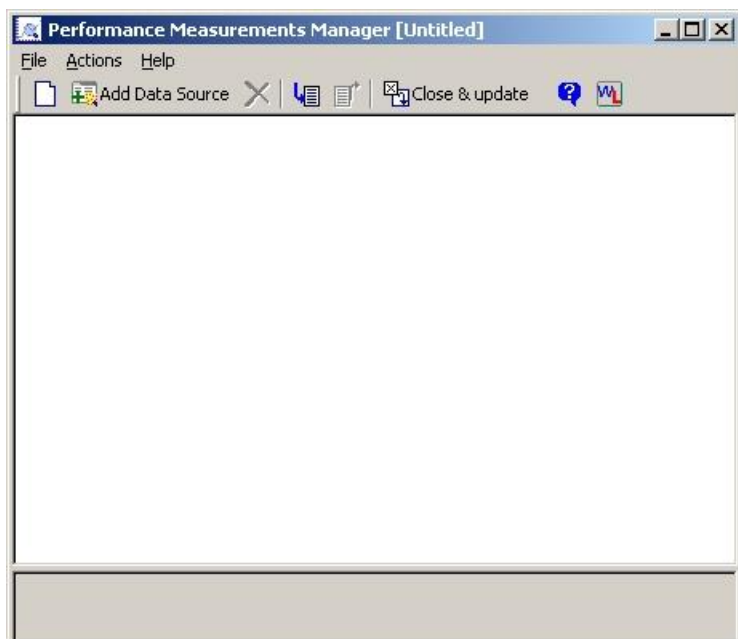


Figure 206: Performance Measurements Manager

The Performance Measurements Manager Main Window

The PMM main window is where you set up the configuration file of data sources, hosts, and performance measurements to monitor. The main window displays the data sources configured to be monitored together with the hosts and the performance measurements being monitored on each host.

The main window consists of the following:

- Menu bar
- Toolbar
- Configuration window
- Description pane

Menu Bar

The PMM menu bar is located at the top of the window.



Figure 207: PMM Menu Bar

This section discusses each of the PMM menus, their options, and their related toolbar buttons.

File Menu



The File menu includes the following functions for PMM configuration file management.



Note: Menu entries vary depending if PMM is run from within WebLOAD Console or as a standalone tool.

Table 53: File Menu






Menu Entry	Icon	Description
New Configuration / New		Clears the configuration in the PMM main window.
Open (available only in the PMM standalone tool)		Opens the Open dialog box, enabling you to select the PMM file to open.
Import PMM		Opens an existing PMM configuration. If a configuration is already defined in the Configuration pane, PMM prompts you to Append the selected file to the existing configuration, or replace the existing configuration.
Export PMM		Saves the current configuration to a *.pmm file. The following information is saved: <ul style="list-style-type: none"> • Data sources • Hosts • Selected measurements from each data source on each host <p> Note: Exporting a configuration from the PMM within WebLOAD Console does not assign the configuration to the current template. To update the template with the configuration displayed, click Close and update Template.</p>
Save / Save As (available only in the PMM standalone tool)		Saves the current configuration to a *.pmm file.

Menu Entry	Icon	Description
Close and Update Template (available only in the PMM initiated through the WebLOAD Console)		Closes PMM and updates the current WebLOAD template with the current PMM configuration.  Note: The PMM configuration is not saved outside of the template.
Exit	none	Closes PMM.

Actions Menu

The Actions menu includes the following functions for creating a PMM configuration file and controlling the Configuration pane.

Table 54: Actions Menu

Menu Entry	Icon	Description
Add Data Source		Opens the PMM Wizard for configuring new data sources, hosts and measurements to monitor.
Delete		Deletes the currently selected item and all sub-items from the configuration.
Clear All (available only in the PMM standalone tool)		Removes all elements from the current configuration.
Expand Tree		Expands the configuration tree.
Collapse Tree		Collapses the configuration tree.

Help Menu

Use the PMM Help menu to access online help, the WebLOAD Community, as well as copyright and version information about PMM.

Configuration Pane

The Configuration pane displays the performance objects that are configured for monitoring, including hosts, data sources and measurements.

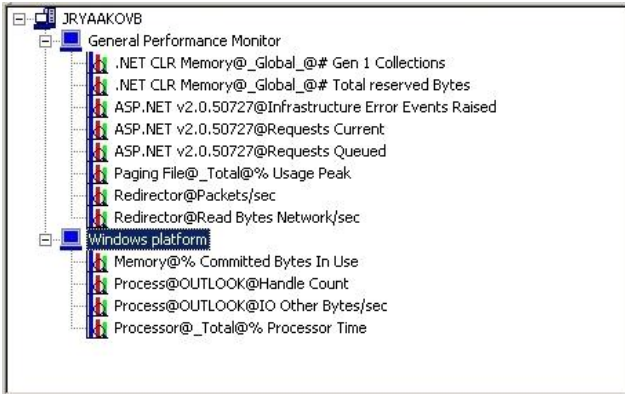


Figure 208: Configuration Pane

Description Pane

The Description pane displays information about the item selected in the Configuration window. As you click different items in the Configuration window, the information displayed in the description pane changes appropriately.

The PMM Configuration Tree Structure

The PMM Configuration tree gives you a complete graphical overview of the performance objects configured for monitoring, including hosts, data sources and measurements. The icons adjacent to the tree items enable you to view your current configuration at a glance.

The following figure shows the PMM configuration tree, configured to collect data from two hosts, each with one data source and multiple measurements.

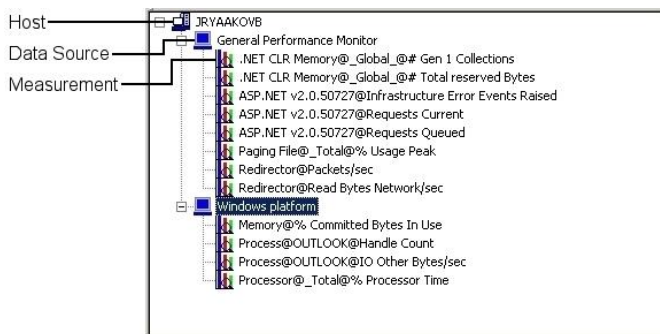


Figure 209: PMM Configuration Tree

Configuration Tree Structure




The first level branches display the hosts for which the data sources and measurements are defined. The second level branches display the data sources configured on each host and the third level branches display the measurements configured for each data source on each host.

Configuration Tree Icons

PMM displays icons on each line of the Configuration Tree, making it easy to view the performance objects configured for collection during your test session.

The Configuration Tree icons are described in the following table:

Table 55: Configuration Tree Icons

Icon	Performance Object
	Host
	Data Source
	Measurement

The following options can be accessed through the right mouse button:

Table 56: Configuration Tree Right-Mouse Options

Tree Item	Right-Mouse Option	Description
Host	Add data source	Opens the PMM Wizard, enabling you to select additional data sources to monitor on the selected host.
	Delete	Removes the selected host from the configuration.
Data Source	Add measurements	Opens the PMM Wizard to enabling you to select additional measurement to monitor on the selected data source.
	Delete	Removes the selected data source from the configuration.
Measurement	Delete	Removes the selected measurement from the configuration.

Navigating the PMM Configuration Tree

You can move through a PMM configuration tree by expanding and collapsing nodes as needed with the plus \oplus and minus \ominus keys, or by clicking and double-clicking with the mouse.

The following table lists the mouse actions you can use to move through a PMM configuration tree.

Table 57: PMM Configuration Tree Actions

Action	Result
Click	Selects an item and displays a description for the selected item.
Double-click	Displays or hides items contained by the selected item.
Right-click	Displays the Actions shortcut menu for the selected item.

The following table lists the keystrokes you can use to move in the PMM main window.

Table 58: PMM Configuration Tree Keystrokes

Key	Result
UP ARROW	Moves the selection up one item in the configuration tree.
DOWN ARROW	Moves the selection down one item in the configuration tree.
RIGHT ARROW	Expands the selected item. If the selected item does not contain hidden items, behaves like down arrow.
LEFT ARROW	Collapses the selected item. If the selected item does not contain exposed items, behaves like up arrow.
PAGE UP / HOME	Moves the selection to the first item in the configuration tree.
PAGE DOWN	Moves the selection to the bottom item visible in the window.
END	Moves the selection to the last item in the configuration tree.
Plus sign "+" on the numeric keypad	Expands the selected node.
Minus sign "-" on the numeric keypad	Collapses the selected node.
Asterisk "*" on the numeric keypad	Expands the entire tree below the selected node.

Using PMM Configuration Files


PMM configuration files can be created and attached to WebLOAD Console templates for monitoring performance in a test session, or configured and saved for use in future WebLOAD Console sessions.

Creating a New Configuration File

When PMM is first opened the main configuration window is empty, enabling you to create a new configuration file. For directions on how to define the components to monitor, see *Specifying the Data Sources, Hosts, and Measurements to Monitor* (on page 375).

If the PMM currently displays a configuration file, creating a new configuration file requires clearing the current configuration information from the display window.


To clear an existing configuration from the configuration window in order to define a new configuration file:

1. Click the **New**  toolbar button.
A dialog box appears asking for confirmation.
2. Click **Yes**.
The PMM configuration window is cleared.

Opening / Importing Saved PMM Configuration Files

PMM configurations can either be opened alone, or they can also be added to any existing configuration in the PMM configuration window, letting you reuse existing configuration files within a new test configuration.

To open an existing configuration file:

1. **File** ► **Import PMM**,
-Or-
Click the **Import**  toolbar button.
The Open dialog box opens for selecting an existing PMM configuration file.
2. Browse to the location of the PMM configuration files, select a file, and click **Open**.
If a configuration is already defined and displayed within the PMM configuration window, a message box displays:

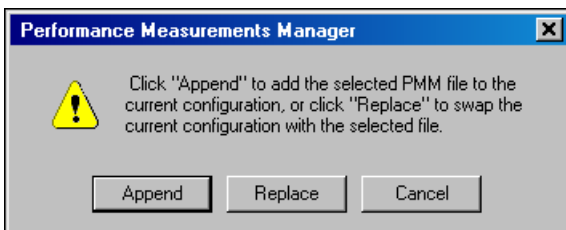


Figure 210: PMM Append/Replace Message Box

The message box has the following options:

- **Append** – Add the selected PMM configuration to the current configuration in the PMM configuration window.
- **Replace** – Remove the current configuration in the PMM configuration window, and replace it with the selected configuration.
- **Cancel** – Aborts the PMM configuration file opening/importing process.

The selected configuration is added to the PMM configuration window.


Assigning a PMM Configuration to a WebLOAD Console Template

PMM configurations can only be attached to WebLOAD Console templates when working with the PMM from within WebLOAD Console.

To assign a PMM configuration to a template:

1. Ensure that the template to which you want to assign the PMM configuration is open in WebLOAD Console.
2. Open / define the PMM configuration you want to attach to the template. For information on defining PMM configurations, see *Specifying the Data Sources, Hosts, and Measurements to Monitor* (on page 375).
3. Select **File > Close and update Template**,

-Or-

Click the  **Close & update** toolbar button.

PMM closes and the configuration file is assigned to the current template.

Saving and Exporting PMM Configurations

After configuring the data sources, hosts and measurements to monitor you can save the configuration for use in WebLOAD Console tests. PMM lets you save multiple configurations to save you the time of reconfiguring the measurements to monitor each test. PMM configuration files are saved with the extension * .pmm.

PMM configuration files include the following information to monitor:

- Data sources
- Hosts
- Measurements

Exporting PMM Configuration Files

When working with PMM from within WebLOAD Console you create configurations for use with the current WebLOAD Console template. To save the configuration, you must export it to a PMM file.

To export PMM files:

1. From the PMM activated in WebLOAD Console, select **File ► Export PMM**,

-Or-

Click the **Export**  toolbar button.

The Save As dialog box opens.

2. Browse to the location you want to save your PMM file, enter a name for the file in the File Name field, and click **Save**.

Saving PMM Configuration Files

When working with PMM as a standalone tool, you can save configurations for use in WebLOAD Console tests. The configuration is not attached to any WebLOAD Console template.

To save PMM files:

1. From the PMM standalone tool, select **File ► Save**,

-Or-

Click the **Save**  toolbar button.

The Save As dialog box opens.

2. Browse to the location you want to save your PMM file, enter a name for the file in the File Name field, and click **Save**.

Specifying the Data Sources, Hosts, and Measurements to Monitor

PMM can be used for adding new data sources to the PMM configuration or updating an existing configuration. You select the type of data source to monitor and the measurements to monitor on each host. When you connect to a host server using Perfmon (Windows), you must have administrator privileges on that server in order to successfully monitor it. For more information about configuring administrator privileges on a host server using Perfmon (Windows), see *Configuring Administrator Privileges for Perfmon (Windows)* (on page 385).

Some data sources require initial configuration to enable monitoring by the PMM. For more information, refer to *Enabling Data Sources Monitoring* on page 387.

Adding Data Sources

Data sources are the performance objects you want to monitor.

Starting the Performance Measurements Manager Wizard

To add new data sources:

1. Click the  **Add data source** toolbar button.

The Performance Measurements Manager Wizard opens.



Figure 211: Performance Measurements Manager Wizard

2. Check or clear **Do not display this page again.**
3. Click **Next.**

The Data Source screen appears.

Selecting the Data Source

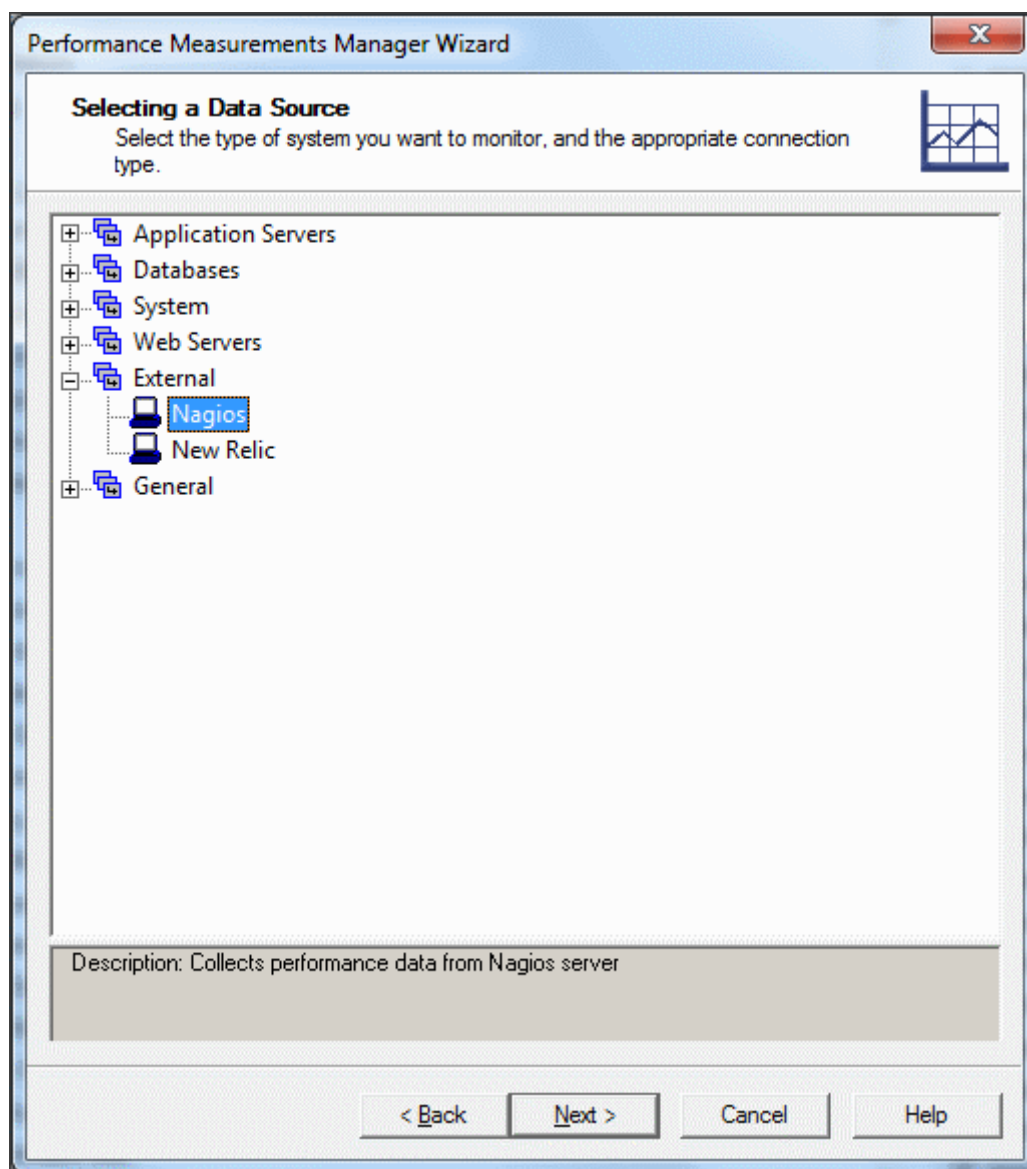


Figure 212: Data Source Screen

To select a data source:

1. Expand the tree to view all available data sources. PMM supports the following types of data sources:

Application Servers:

- Adobe LiveCycle ES using JMX
- GlassFish AS using JMX
- IBM WebSphere 5 using WAS
- JBoss 4.x-5.x GA using JMX

- JOnAS using JMX
- Microsoft Active Server Pages
- Microsoft ASP .Net
- Oracle WebLogic 10.0 using JMX
- Oracle WebLogic using SNMP
- SAP NetWeaver using JMX
- Tomcat 6.x using JMX
- WebSphere 6.1 using JMX
- WebSphere 7.0 using JMX

Databases:

- Microsoft SQL Server
- Microsoft SQL Server using SNMP (does not support SQL 2005 or 2008)
- MySQL database using SNMP
- Oracle

System:

- UNIX/Linux platforms Network data using SNMP
- UNIX/Linux platforms using RSTATD
- UNIX/Linux UC-Davis
- Windows .Net Framework
- Windows platform

Web Servers:

- Apache 2 web server using SNMP
- Apache web server using SNMP
- Microsoft Internet Information Server (IIS)
- Sun One (former iPlanet 6)

General:

- General JMX – General JMX can be used to connect to any server that supports both JMX (JSR-3) and JMX Remoting (JSR-160).
- General Performance Monitor – General Performance Monitor can be used just like the System > Windows Platform. The difference is that it gives access to all Performance Monitor counters.
- General SNMP – General SNMP can be used similarly to other SNMP-based PMs. The difference is that it gives access to all MIBs served by the SNMP agent on that machine.

- General UNIX – General UNIX can be used similarly to the System > UNIX platform using RSTATD. The difference is that it gives access to all RSTATD counters.

External

- Nagios – Collects performance data from an installed Nagios server
 - New Relic – Collects performance data from an installed New Relic server
2. Select a server or system as a data source, and click **Next**.

The Host Selection screen appears.

Selecting a Host

Performance Measurements Manager Wizard

Selecting a Host
Select the host to monitor and click Next to configure the specific performance measurement to report.

Enter the name of the host to monitor, or click the Browse button to select from the available hosts.

Note: Identification of hosts running in a Windows environment is limited to 15 characters.

Name	Value
Server type	LiveCycle
Port	

< Back Next > Cancel Help

Figure 213: Host Selection Screen

To select a host:

1. Click **Browse** to select a host to monitor.

The Host Selection dialog box opens.

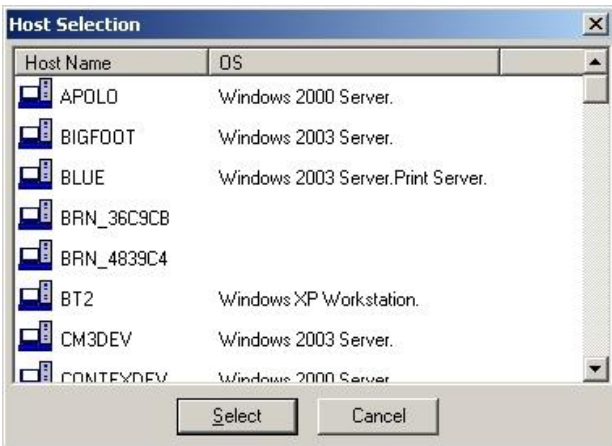


Figure 214: Host Selection Dialog Box

2. Select the host to monitor from the list and click **Select**.

The host is added to the Selecting a Host screen.

3. Some application server data sources require that you enter the application server’s port value. Contact your system administrator for the information. Following are port numbers commonly used by some application server data sources:

- Tomcat – 8080.
- JBoss – 1099.
- WebSphere – 9080.
- WebLogic – 7001.

4. Click **Next**.

PMM attempts to connect to the host. If you are accessing this host for the first time within this Load Template, PMM prompts you for your username and password on the host.

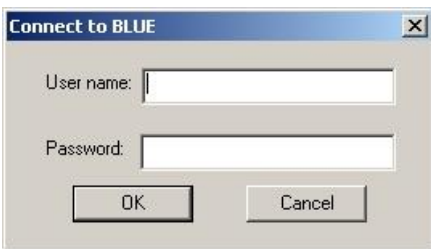


Figure 215: Prompt for Accessing the Host Computer

Enter your username and password.

If a connection is successfully made, the Wizard progresses to the Measurements to Monitor screen.

If a connection cannot be established, the Wizard returns to the Host Selection screen, enabling you to select an alternate host.

Selecting the Measurements to Monitor

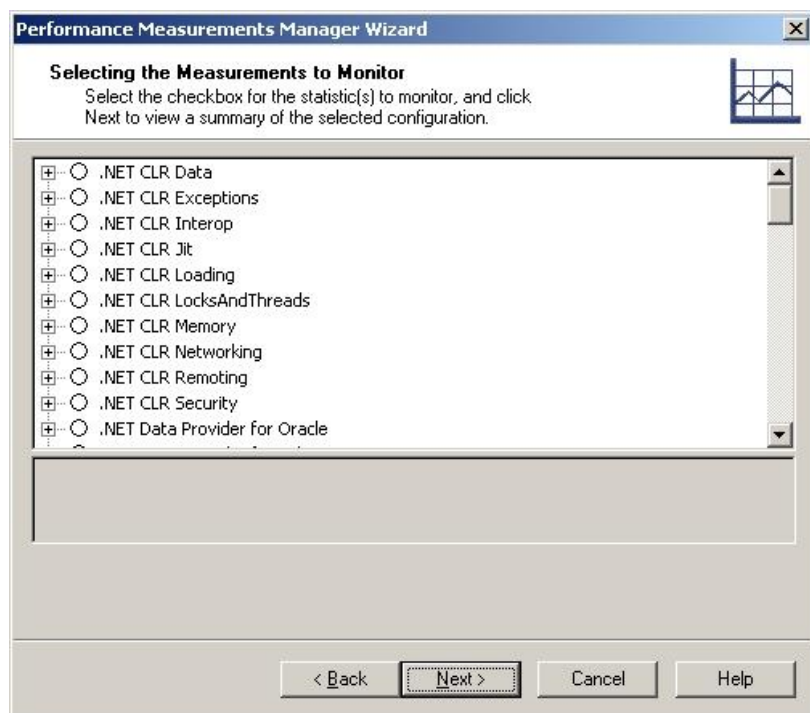


Figure 216: Selecting the Measurements to Monitor Screen

To select the measurements to monitor:

1. Select one or more measurements to monitor by clicking the checkbox adjacent to the item. PMM only collects the data for measurements that are selected. All other data is not collected.

Explanations of each measurement are available by clicking the measurement name. The explanation appears at the bottom of the window.

The list of available measurements varies depending upon the type of data source. Each data source is supplied with a default set of measurements to monitor.

**Notes:**

An empty circle ○ next to a measurement indicates that the measurement contains sub-components and that no default sub-components are defined (the upper level component cannot be selected). To select sub-components, click the + to expand the tree.

A circle with a checkmark ☑ next to a measurement indicates that the measurement contains sub-components and that default sub-components are defined. To view and configure the configured subcomponents, click the + to expand the tree.

2. Click **Next**.

The Wizard displays a summary of the host, data source, and measurements configured for monitoring.

Finishing the Configurations



Figure 217: PMM Wizard Configuration Summary

To finish the configurations:

1. To accept the configuration, click **Finish**.

The PMM Wizard closes and the selected configuration is added to the PMM main window.

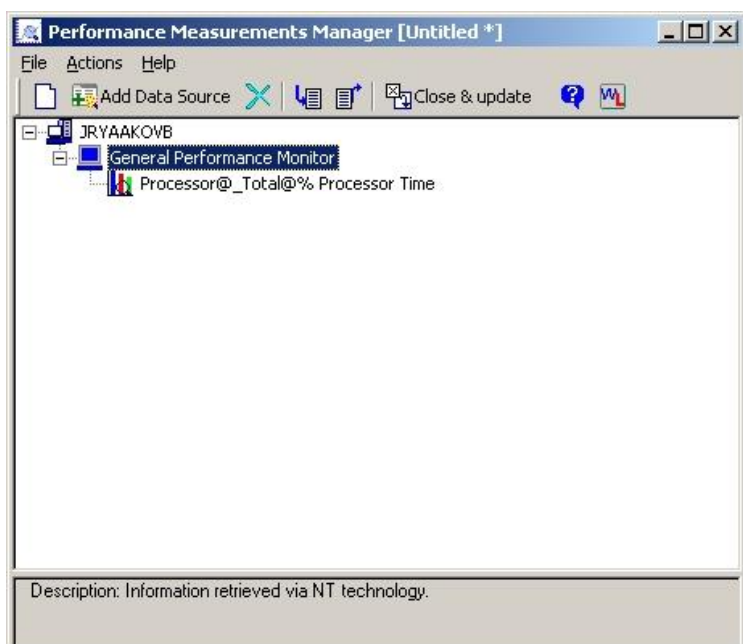



Figure 218: PMM Main Window

- To add additional data sources, hosts, and measurements to monitor, click the  **Add data source** toolbar button and repeat the process.

Adding Measurements to a Configured Data Source

Measurements can be added quickly and easily to a data source configured in the PMM window.

To add measurements to a data source already configured in the PMM:

- Right-click the data source to which you want to add measurements and click **Add measurements** from the pop-up menu.

The Measurements to Monitor screen for the selected data source appears.

- Select all of the measurements you want to monitor for the data source by selecting the checkbox adjacent to the measurement names.

Explanations for each measurement are available by clicking the measurement name. The explanation appears at the bottom of the window.

- Click **Next**.

A summary of the current configuration appears.

- To accept the configuration, click **Finish**.

The PMM Wizard closes and the configuration is updated in the PMM main window.

Adding Data Sources to a Configured Host

Data sources can be added quickly and easily to a host configured in the PMM window.

To add data sources to a host already configured in the PMM:

1. Right-click the host in the PMM main window to which you want to add a data source, and click **Add data source** from the pop-up menu.

The Select a Data Source Wizard screen appears.

2. Select the data-source to add to the configuration and click **Next**.

The PMM Wizard progresses to the Measurements to Monitor screen.

3. Select all of the measurements you want to monitor for the data source.

Explanations of each measurement are available by clicking the measurement name. The explanation appears at the bottom of the window.

4. Click **Next**.

A summary of the current configuration appears.

5. To accept the configuration, click **Finish**.

The PMM Wizard closes and the configuration is updated in the PMM main window.

Deleting a Data Source

To delete a data source from PMM configuration:

1. Right-click the data source you wish to delete in the PMM main window.
2. Select **Delete** from the pop-up menu.

A message box appears asking for confirmation.

3. Click **Yes**.

The data source is removed from the configuration and is no longer monitored.

Deleting a Monitored Host

To delete a host from the PMM configuration:

1. Right-click the host you wish to delete in the PMM main window.
2. Select **Delete** from the pop-up menu.

A message box appears asking for confirmation.

3. Click **Yes**.

The host, and all data sources and measurements configured for the host are removed from the configuration and are no longer monitored.

Deleting a Measurement

To delete a measurement from the PMM configuration:

1. Right-click the measurement you wish to delete in the PMM main window.
2. Select **Delete** from the pop-up menu.

A message box appears asking for confirmation.

3. Click **Yes**.

The measurement is removed from the configuration and no longer monitored.

Configuring Administrator Privileges for Perfmon (Windows)

In order to successfully monitor a host server using Perfmon (Windows), you must have administrator privileges on that server. There are three methods for configuring Administrator privileges:

- *Configuring Administrator Privileges on a Server* (on page 385)
- *Configuring Administrator Privileges on all Servers Simultaneously* (on page 386)
- *Configuring Administrator Privileges on non-Server PCs* (on page 386)

Configuring Administrator Privileges on a Server

To configure Administrator privileges on a server:

1. Log into the server to be tested.
2. Right-click **My Computer** and click **Manage**. The Computer Management screen appears.
3. In the Local Users and Groups area, add your WebLOAD Console username to the following groups:
 - Performance Log Users
 - Performance Monitor Users

Configuring Administrator Privileges on all Servers Simultaneously

To configure Administrator privileges on all servers simultaneously:

1. Log into the Active Directory computer in the domain to be tested.
2. Navigate to **Administrative Tools > Active Directory Users and Computers**.
3. Select your WebLOAD Console username and click **Properties**.
4. Click the Member of tab and add the user to the following groups:
 - Performance Log Users
 - Performance Monitor Users

Configuring Administrator Privileges on non-Server PCs

The following method must be used for non-server PCs, but can also be used for server PCs.

To configure Administrator privileges on a non-server PC:

1. Grant at least Read permission to the following files:
 - <Windows directory>\system32\perfcxxx.dat
 - <Windows directory>\system32\perfhxxx.dat
 where xxx represents the system language (e.g., 009 for English).
2. Open regedit32 and find the following keys:
 - HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurePipeServers\Winreg
 - HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\NT\CurrentVersion\Perflib
3. Grant your user at least Read permission to these keys and their sub-keys and close regedit32.

Your user has access to monitor the performance of the host server using Perfmon (Windows).

For more information about configuring administrator privileges on a host server using Perfmon (Windows), see *Controlling remote Performance Monitor access to Windows NT servers*, located at

<http://support.microsoft.com/default.aspx?scid=kb;en-us;164018>.

Enabling Data Sources Monitoring

PMM enables you to monitor various data sources during runtime to view performance data in real time. However, some data sources require preliminary configuration of the data source and/or of WebLOAD. The following sections describe the various data sources and, where applicable, the necessary preliminary configuration.

Enabling Application Server Monitoring

PMM enables you to monitor application servers during runtime to view performance data from the application server in real-time. Before running your test session, you may need to perform some configuration tasks, as described below.



Note: No configuration is necessary if you are using the following servers: GlassFish AS using JMX, JOnAS using JMX, Microsoft Active Server Pages, Microsoft ASP.NET.

Adobe Life Cycle Enterprise Server using JMX – Configuration Tasks

Adobe Life Cycle Data Services Enterprise Server JMX configuration depends on the type of server being used: JRun or Tomcat.

Configuring a JRun Server

The JRun configuration file is `\lcds\jrun4\bin\jvm.config`.

In order to enable JMX, set the following in the configuration file:

```
java.args=-Xms32m -Xmx384m
-Dsun.io.useCanonCaches=false
-Dcom.sun.management.jmxremote
-Dcom.sun.management.jmxremote.port=<port>
-Dcom.sun.management.jmxremote.ssl=false
-Dcom.sun.management.jmxremote.authenticate=false
```

Configuring a Tomcat Server

The Tomcat configuration file is `\lcds\tomcat\bin\catalina.bat`.

In order to enable JMX, set the following in the configuration file:

```
set CATALINA_OPTS=%CATALINA_OPTS% -Dcom.sun.management.jmxremote
-Dcom.sun.management.jmxremote.port=<port>
-Dcom.sun.management.jmxremote.ssl=false
-Dcom.sun.management.jmxremote.authenticate=false
-Djava.rmi.server.hostname=<hostname>
```

IBM WebSphere 5 Using WAS – Configuration Tasks

To monitor the IBM WebSphere 5 application server in the Performance Measurements Manager Wizard, you need to install the supported IBM WebSphere 5 software and configure the IBM WebSphere 5 server.

WebLOAD Configuration

To enable the PMM to monitor the IBM WebSphere 5 application server, you need to install the IBM WebSphere Application Server WAS, Version 5 Application Client software.

For information about obtaining and installing the WebSphere 5 Application Client software, refer to the WebSphere info center at <http://www.ibm.com/software/webservers/appserv/infocenter.html> or contact IBM.

When this software package is not installed or has not been installed properly, the ability to monitor WebSphere 5 using WAS is not available and WebLOAD might show one of the following error messages:

- The IBM J2EE Client version x (required for probing WebSphere) is not installed. Please install the IBM J2EE Client.
- The IBM J2EE Client version x (required for probing WebSphere) is not installed properly. Please reinstall the IBM J2EE Client.
- The Java program associated with the IBM J2EE Client (required for probing WebSphere) is not found. Please reinstall the IBM J2EE Client.
- The IBM J2EE Client version x (required for probing WebSphere) is not installed. Before continuing, please install the IBM J2EE Client.
- The file [name], essential for probing WebSphere, could not be found in the J2EE directory. Before continuing, please install the IBM J2EE Client version x.

Server Side Configuration

- Enable data collection as described in:
<http://www.ibm.com/software/webservers/appserv/doc/v35/ae/infocenter/was/atsepmcf.html>.
- Turn off the security settings on the application server, as follows:
 - a. Log in to the server by typing the following command line directly on the server:


```
wsadmin.bat -user <user name> -password <user password>
```
 - b. Type the following command to disable the security settings:


```
$AdminTask setGlobalSecurity {-interactive}"
```
 - c. Type `$AdminConfig save` to save the new configuration.
 - d. Restart the application server.

JBoss 4.x – 5.x using JMX – Configuration Tasks

To enable the PMM to monitor the JBoss 4.x – 5.x using JMX, you need to configure the JBoss 4.x – 5.x application server to use JMX, and configure WebLOAD to use JBoss 4.x and 5.x.

Server Side Configuration:

To configure JBoss to use JMX:

1. Edit the server execution file (for example, `/usr/local/jboss/bin/run.sh`) by adding the following flags:

```
JAVA_OPTS=$JAVA_OPTS -Dcom.sun.management.jmxremote=true
JAVA_OPTS=$JAVA_OPTS -
Dcom.sun.management.jmxremote.port=6789
JAVA_OPTS=$JAVA_OPTS -
Dcom.sun.management.jmxremote.authenticate=false
JAVA_OPTS=$JAVA_OPTS -
Dcom.sun.management.jmxremote.ssl=false
```

where `jmxremote.port` is some arbitrary free TCP port.

2. Run JBoss with `-b0.0.0.0`. This instructs the server not to bind exclusively to localhost.



Note: By default, the JBoss Naming Service (JNDI) is listening on TCP port 1099. This port is the default port, retrieved from the Service Binding Manager. You can change the default port by changing the following configuration file:

```
/usr/local/jboss/server/all/conf/jboss-service.xml.
```

WebLOAD Configuration

To configure WebLOAD to use JBoss 4.x.x:

1. Put the jars `jboss-management.jar` and `jbossall-client.jar` into `<WebLoad-ProgramData-Dir>\extensions\Java`.
2. Try to connect to port 1099 (JNDI) with the default installation.

To configure WebLOAD to use JBoss 5.x.x:

1. Put the following jars into `<WebLoad-ProgramData-Dir>\extensions\Java`:
 - `lib/jboss-common-core.jar`
 - `client/jmx-invoker-adaptor-client.jar`
 - `client/jbossjmx-ant.jar`
 - `client/jboss-logging-spi.jar`
 - `client/jboss-client.jar`
 - `common/lib/jboss-serialization.jar`
 - `common/lib/jboss-remoting.jar`
 - `common/lib/jboss-security-spi.jar`
 - `common/lib/jboss-integration.jar`
 - `common/lib/jboss-javaee.jar`
 - `common/lib/jboss-management.jar`
2. Try to connect to port 1099 (JNDI) with the default installation.

Oracle WebLogic 10.0 Using JMX – Configuration Tasks

To enable the PMM to monitor Oracle WebLogic 10.0 Using JMX, you need to configure the Oracle WebLogic server.

Server Side Configuration:

To configure the Oracle WebLogic server:

In the server's start-up script (for example, `startWebLogic.cmd`), add the following rows:

```
set JAVA_OPTIONS=%JAVA_OPTIONS% -
Dcom.sun.management.jmxremote=true

set JAVA_OPTIONS=%JAVA_OPTIONS% -
Dcom.sun.management.jmxremote.port=6790

set JAVA_OPTIONS=%JAVA_OPTIONS% -
Dcom.sun.management.jmxremote.authenticate=false

set JAVA_OPTIONS=%JAVA_OPTIONS% -
Dcom.sun.management.jmxremote.ssl=false
```

Oracle WebLogic Using SNMP – Configuration Tasks

To enable the PMM to monitor Oracle WebLogic using SNMP, you need to configure the Oracle WebLogic to set up a server side SNMP agent and configure the PMM to use Oracle WebLogic.

Server-side Configuration:

To set up a server-side SNMP agent:

1. Log into the WebLogic Administration Console.
2. In the Domain Structure tree, expand the **Diagnostics** node and click the **SNMP** node.
3. Perform the following in the **Server SNMP Agents** table:
 - a. Click **New** to add and configure a new SNMP Server agent.
 - b. Specify a name.
 - c. Click **OK** to continue.
The new SNMP Server Agent is added with the following settings:
Enabled = false, UDP Port = 161.
 - d. Click the name of the new SNMP Server Agent to configure its settings.

- e. Check the **Enabled** option.
 - f. Change the SNMP UDP port number. You can set it to be any free UDP port, such as 3161.
 - g. Change the Trap Version to V2.
 - h. Click **Save** to continue.
4. Set the target server for monitoring, as follows:
 - a. Click the **Targets** tab.
 - b. Select the relevant servers from the list. Select only one server per agent.
 - c. Click **Save** to continue.

The basic configuration of an SNMP Server agent is complete. No restart is needed.

WebLOAD Configuration:

In the PMM wizard, enter the following parameters:

1. In the Selecting a Host window, it is preferred to specify the IP address of the active network interface.
2. System Entry Point – for Oracle WebLogic the default system entry point is **.1.3.6.1.4 (iso.org.dod.internet.private.enterprises)** or **.1.3.6.1.4. 140.625.105.1.1.16**.

SAP NetWeaver Using JMX – Configuration Tasks

To enable the PMM to monitor SAP NetWeaver Using JMX, you need to configure the PMM to use SAP NetWeaver.

WebLOAD Configuration:

1. Copy the following jars for SAP NetWeaver installation to
C:\Program Files\RadView\WebLOAD\extensions\java:
 - `com_sap_pj_jmx.jar` – the SAP-JMX library.
 - `client.jar` – the SAP AS Java client API (includes the JMX Adapter).
 - `exception.jar` – the SAP exception framework.
 - `logging.jar` – the SAP Logging API.
2. In the PMM wizard, perform the following:
 - a. In the Selecting a Host window, specify the RMI-P4 port.
 - b. When you are prompted for your username and password on the host, enter the credentials of a user with a default administrator role.

Tomcat 6.0 Using JMX – Configuration Tasks

To enable the PMM to monitor Tomcat 6.0 Using JMX, you need to configure the Tomcat 6.0 application server.

Server-side configuration

In the server's execution script (for example, `\bin\catalina.bat`), add the following rows:

```
set CATALINA_OPTS=%CATALINA_OPTS% -Dcom.sun.management.jmxremote
-Dcom.sun.management.jmxremote.port=<port> -
Dcom.sun.management.jmxremote.ssl=false
-Dcom.sun.management.jmxremote.authenticate=false -
Djava.rmi.server.hostname=<hostname>
```

WebSphere 6.1 Using JMX – Configuration Tasks

To enable the PMM to monitor WebSphere 6.1 using JMX, you need to configure the WebSphere 6.1 application server and configure WebLOAD to support WebSphere 6.1.

Server-side configuration

1. Add `-Djavax.management.builder.initial= -Dcom.sun.management.jmxremote` to the Generic JVM Argument, as follows:
 - a. Launch the WebSphere Application Server.
 - b. Connect to the Administration Console.
 - c. Select **Servers > Server Types > WebSphere application servers**, and specify which server to manage (default: server1).
 - d. In the Configuration tab, select **Server Infrastructure > Java and Process Management > Process Definition**.
 - e. Select **Additional Properties > Java Virtual Machine**.
 - f. Set the Generic JVM arguments to `-Djavax.management.builder.initial=-Dcom.sun.management.jmxremote`.
2. Add the following lines to `/opt/IBM/WebSphere/AppServer/java/jre/lib/management/management.properties`:


```
com.sun.management.jmxremote.port=9999
com.sun.management.jmxremote.authenticate=false
com.sun.management.jmxremote.ssl=false
```

WebLOAD Configuration:

Copy the following libraries from the application server's installation directory to the <WebLOAD_DIR>\ extensions\java\ directory:

- com.ibm.ws.admin.client_6.1.0.jar
- com.ibm.ws.runtime_6.1.0.jar
- ibmorb.jar

WebSphere 7.0 Using JMX – Configuration Tasks

To enable the PMM to monitor WebSphere 6.1 Using JMX, you need to configure the WebSphere 6.1 application server and configure WebLOAD to support WebSphere 6.1.

Server-side configuration

1. Add `-Djavax.management.builder.initial= -Dcom.sun.management.jmxremote` to the Generic JVM Argument, as follows:
 - a. Launch the WebSphere Application Server.
 - b. Connect to the Administration Console.
 - c. Select **Servers > Server Types > WebSphere application servers**, and specify which server to manage (default: server1).
 - d. In the Configuration tab, select **Server Infrastructure > Java and Process Management > Process Definition**.
 - e. Select **Additional Properties > Java Virtual Machine**.
 - f. Set the Generic JVM arguments to `-Djavax.management.builder.initial= -Dcom.sun.management.jmxremote`.
2. Add the following lines to `/opt/IBM/WebSphere/AppServer/java/jre/lib/management/management.properties`:


```
com.sun.management.jmxremote.port=9999
com.sun.management.jmxremote.authenticate=false
com.sun.management.jmxremote.ssl=false
```

WebLOAD Configuration:

Copy the following libraries from the application server's installation directory to the <WebLOAD_DIR>\ extensions\java\ directory:

- [WebSphere]/AppServer/runtimes/com.ibm.ws.admin.client_7.0.0.jar
- [WebSphere]/AppServer/plugins/com.ibm.ws.runtime.jar

- [WebSphere]/AppServer/runtimes/com.ibm.ws.webservices.thincli
ent_7.0.0.jar
- [WebSphere]/AppServer/java/jre/lib/ibmorb.jar
- [WebSphere]/AppServer/java/jre/lib/ibmjgssprovider.jar
- [WebSphere]/AppServer/deploytool/itp/plugins/com.ibm.webspher
e.v7_7.0.0.v20080817/wasJars/ibmjsseprovider2.jar

Where [WebSphere] is the application server's installation directory.

Enabling Database Monitoring

PMM enables you to monitor database servers during runtime. This allows you to view performance data from the database in real-time. Before running your test session, you may need to perform some configuration tasks, as described below.



Note: No configuration is necessary if you are using the following database: Microsoft SQL server.

Oracle Monitor – Configuration Tasks on Windows

The PMM Oracle database monitor uses SNMP to retrieve statistics.

To use this monitor on Windows you must:

1. Install SNMP support for Windows.
2. Install SNMP support for Oracle using the Oracle Database installation.
3. Specify the port where the Oracle master agent is listening.

The required port is specified in the `TRANSPORT` section of the `MASTER.CFG` file located in the `\ADMIN` directory.

For example, add the following section to the `MASTER.CFG` file:

```
TRANSPORT ordinary SNMP
OVER UDP SOCKET
AT PORT 161
```

Use port 161 for the Oracle Peer SNMP master agent.

4. Specify the authentication in the `COMMUNITY` section of the `MASTER.CFG` file.

For example:

```
COMMUNITY public
ALLOW ALL OPERATIONS
USE NO ENCRYPTION
```

- Specify an unused port where the encapsulated agent, such as Microsoft SNMP service, should listen. Microsoft SNMP service typically uses port 1161. The port is specified in the `SERVICES` file located in the `NT_HOME|\SYSTEM32\DRIVERS\ETC` directory.

For example, modify the `nt_home\system32\drivers\etc\services` file from:

```
...
snmp 161/udp
```

```
...
```

To:

```
snmp 1161/udp
```

- Specify which non-PEER master agents are to be encapsulated by editing the encapsulator configuration file, `ENCAPS.CFG`, located in the `\ADMIN` directory. You must add the `AGENT` entry, including MIB-subtrees manageable by NMS, for the encapsulated master agent.

For example, modify the `encaps.cfg` file as follows:

```
AGENT AT PORT 1161 WITH COMMUNITY public
SUBTREES 1.3.6.1.2.1.1,
1.3.6.1.2.1.2,
1.3.6.1.2.1.3,
1.3.6.1.2.1.4,
1.3.6.1.2.1.5,
1.3.6.1.2.1.6,
1.3.6.1.2.1.7,
1.3.6.1.2.1.8,
1.3.6.1.4.1.77
FORWARD ALL TRAPS;
```



Note: The port (1161) must match the unused port specified in step 5.

- Make the “OracleSNMPPeerEncapsulator” service automatic and start it.
- Reboot your computer.

Oracle Monitor – Configuration Tasks on UNIX

The PMM Oracle database monitor uses an SNMP Master Agent to communicate with the Oracle Intelligent Agent. The SNMP Master Agent and the Oracle Intelligent Agent must be configured correctly before the Oracle Intelligent Agent can communicate over SNMP to the Master Agent on a UNIX platform. For detailed information on configuring the Intelligent Agent, see the *Oracle Enterprise Manager Configuration Guide*.

To use this monitor on UNIX you must:

1. Configure the SNMP Master Agent.
2. Start the SNMP Agents.

Configuring the SNMP Master Agent

The necessary SNMP files are installed automatically when you install the Oracle Intelligent Agent. After installing the Oracle Intelligent Agent, you will have to edit the following files as described below:

- `$(ORACLE_HOME)/network/snmp/peer/CONFIG.master`
- `$(ORACLE_HOME)/network/snmp/peer/CONFIG.encap`
- `$(ORACLE_HOME)/network/snmp/peer/start_peer`
- `/etc/snmpd.config`

CONFIG.Master (Peer Master Agent file)

To edit the CONFIG.master file:

- Find the line beginning with `MANAGER` and change the IP address coded in this line to match the IP address of the machine where the SNMP traps will be sent.

CONFIG.encap (Peer Encapsulator file)

To edit the CONFIG.encap file:

- Find the line `AGENT AT PORT`. It normally reads `AGENT AT PORT 1161 WITH COMMUNITY public`.

If you modify the port number from 1161, you must also modify the `start_peer` script.

*start_peer Script***To edit the start_peer script:**

1. Find the line `NEW_SNMPD_PORT=` and verify that it is using the same port number listed above in the `CONFIG.encap` file.
2. Find the line `NEW_TRAPD_PORT=` and verify the port number is different from `NEW_SNMPD_PORT=`.

*snmpd.config File***To edit the snmpd.config file:**

1. Add the following line to the file:
`trap <hostname or ipaddress>`
2. Replace the information in angle brackets with the actual hostname or IP address of the local host where the file is located.

Starting the SNMP Agents**To start the SNMP agents:**

1. Make sure no SNMP components are running. The three main components are the `master_peer`, `encap_peer`, and `snmpd`. If any of these processes are running, use the `ps` command to find them, and the `kill` command to terminate the processes.

For example:

```
ps -ef | grep snmp
```

This command checks to see if the SNMP Master Agent is running.

2. Start the PEER Master Agent, PEER encapsulator, and native Digital UNIX SNMP Agent:

```
cd $ORACLE_HOME/network/snmp/peer
```

```
su root
```

```
./start_peer -a
```

This command starts all three processes.

3. Now use the `ps` command to verify that all three processes were started:

```
ps -aux |grep peer
```

```
ps -aux |grep snmpd
```

```
ps -ef | grep snmp
```


MySQL database using SNMP – Configuration Tasks

The PMM for MySQL enable monitoring various components of MySQL using the SNMP Micro Agent for MySQL provided by AdventNet (commercial product).

For more information on the AdventNet SNMP Micro Agent for MySQL, see http://www.adventnet.com/products/mysql_agent/index.html.

Microsoft SQL Server Using SNMP – Configuration Tasks

For the Microsoft **SQL Server 2000**, enable SNMP monitoring by selecting the **Enable SNMP** check box in the SQL Server Network Utility dialog box.

Microsoft **SQL Server 2005** and **SQL Server 2008** do not support SNMP. Use the Windows Performance Monitor instead.

Enabling Systems Monitoring

PMM enables you to monitor operating systems during runtime. This allows you to view performance data from the systems in real-time. Before running your test session, you may need to perform some configuration tasks, as described below.



Note: No configuration is necessary if you are using the following systems: Windows .Net Framework, Windows platform.

Enabling Web Servers Monitoring

PMM enables you to monitor Web servers during runtime to view performance data from the server in real-time. Before running your test session, you may need to perform some configuration tasks, as described below.



Note: No configuration is necessary if you are using the following Web server: Microsoft Internet Information Server (IIS).

Apache Web Server Using SNMP – Configuration Tasks

The PMM uses SNMP to monitor different status values of the Apache Web server.

The Apache SNMP monitoring is based on Mod-Apache-Snmp. For download and installation instructions, refer to <http://mod-apache-snmp.sourceforge.net/english/index.htm>.

Apache 2 Web Server Using SNMP – Configuration Tasks

The PMM uses SNMP to monitor different status values of the Apache 2 Web server.

The Apache 2 SNMP monitoring is based on **Mod-Apache-Snmp**. For download and installation instructions, refer to <http://mod-apache-snmp.sourceforge.net/english/index.htm>.

Sun One (iPlanet) – Configuration Tasks

The PMM Sun One monitor uses SNMP to retrieve server statistics.

To use this monitor you must:

1. Install SNMP support for Windows.
2. Ensure that the iPlanet SNMP agent is installed and activated on the iPlanet server host.
Configure the Sun One server to be a proxy for the SNMP agent.
3. Configure the Sun One server host to use port 1161 as the SNMP message port.
4. Enable **SNMP Statistics Collection** on the SNMP Subagent Configuration page of the Server Manager.

Setting up the SNMP Agent and Configuring the Sun One Server as a Proxy for the SNMP Agent

To enable PMM to monitor performance data from the Sun One application server you must first set up the Sun One SNMP agent. The Administration Interface will not start the OS master agent on ports other than 161. The workaround is to use the Sun One SNMP master agent, modify the native master agent to use a different port, and enable the Sun One SNMP subagent to gather information about the server.

To set up the Sun One SNMP agent on Solaris:

1. Check whether an SNMP daemon (snmpd) is running on port 161.
If an SNMP daemon is running:
 - a. Make sure you know how to restart it and which MIB it supports.
 - b. Kill its process.
2. Edit the CONFIG file located in the `plugins/snmp/magt` server root directory from:


```
...
snmp 161/udp
...
To:
...
```

```
snmp 1161/udp
```

```
...
```

3. Manually activate the Sun One master agent and subagent.

- a. At the command prompt, enter the commands:

```
cd <iws_install>/Servers/plugins/snmp
```

```
cd maget
```

```
./magt CONFIG INIT&
```

```
cd ../sagt
```

```
./sagt -c CONFIG &
```

- b. Restart the Sun One Server and the subagent from the Web server Administration.

Enabling Monitoring of Application Servers that use Generic Monitoring Protocols

PMM enables you to monitor application servers that use generic monitoring protocols. You can monitor the application servers during runtime to view performance data in real-time. Before running your test session, you may need to perform some configuration tasks, as described below.

General JMX – Configuration Tasks

The Java Management Extensions (JMX) technology provides the tools for building distributed, Web-based, modular and dynamic solutions for managing and monitoring devices, applications, and service-driven networks.

General JMX can connect to any device that supports JMX (JSR-003) and JMX Remote API (JSR-160). Refer to your application server's documentation to see how to configure JMX for your server.

General Performance Monitor – Configuration Tasks

General Performance Monitor can be used to connect and collect statistics using Windows Performance Monitor protocol (Perfmon).

When you connect to a host server using Perfmon (Windows), you must have administrator privileges on that server in order to successfully monitor it. For more information about configuring administrator privileges on a host server using Perfmon (Windows), see *Configuring Administrator Privileges for Perfmon (Windows)* (on page 385).

General SNMP – Configuration Tasks

General SNMP can be used to connect to all servers supporting SNMP monitoring.

Refer to your server documentation for instructions on how to enable SNMP on your server.

General UNIX – Configuration Tasks

RSTAT monitors enable you to monitor the **rpc.rstatd** daemon counters. This daemon is already installed and running on most Solaris and Linux machines. If not, you can obtain it at <http://rstatd.sourceforge.net/>.

This protocol uses RPC calls on port 111. You may need to open this port in the firewall to allow PMM to access the rstatd daemon.

Monitoring Generic SNMP Data Sources

WebLOAD Console currently supports a variety of SNMP monitors by default, including basic SNMP for Windows, Solaris, and UNIX/Linux, along with WebLogic, and Oracle servers. WebLOAD Console's SNMP monitoring support can be easily extended by updating WebLOAD Console's `UserMibFilter.ini` file. For more information about extending SNMP monitoring support, see *PMM Extension* in the *WebLOAD Extensibility SDK*.

Remote Performance Measurement Manager

This feature enables you to collect statistics on remote machines using TestTalk. With other methods of testing a remote machine, security and firewall issues may block communication between the WebLOAD Console and the remote machine. Using TestTalk to measure performance on remote machines avoids this problem.

To enable the Remote Performance Measurement Manager, you must add a special protocol extension to the Load Generator, called the Remote Machine Load Generator. This extension monitors the Windows Perfmon and sends the information back as Load Generator statistics.

Once the Remote Machine Load Generator is installed, you can add a Remote Performance Measurement Manager script to your template. When you run the template, statistics from the remote machine are gathered through the Remote Machine Load Generator and the script you configure.

Setting the Remote Machine Load Generator

In order to collect statistics from a remote machine using the Remote Performance Measurement Manager, you must first add the Remote Machine Load Generator extension to your Load Generator. The following sections explain how to install the Remote Machine Load Generator extension.

Installing the Remote Machine Load Generator Extension

Install WebLOAD on any Windows machine that can access the server you want to probe. For more information, see the *WebLOAD Installation Guide*.

Setting INI Properties

Set the following parameters in `weblload.ini` of the Load Generator installation (<Install Dir>\bin):

```
PROTOCOLS="pmprobeprotocol.dll;JSCOMObject.dll"  
PROTOCOLS_NAMES="Performance Monitor Probe;COM Object"
```

Setting the Probed Measurements

You can run the Remote Performance Measurement Manager utility to select the servers you want to probe and the statistics you want to collect. For instructions on how to use the Performance Measurements Manager utility to select a server and statistics, see *Specifying the Data Sources, Hosts, and Measurements to Monitor* (on page 375).



Note: To use the Remote PMM, you must obtain a valid license file for the remote machine. To do so, open the Update License utility after performing a full WebLOAD installation, and send the HostID to license@radview.com with a request for a license for setting up a Remote PMM only.

To select a server and statistics:

4. From the Start menu, select **Start > Programs > <WebLOAD Installation> > Utilities > Performance Measurements Manager**, to select the servers you want to probe and the measurements that should be received.
5. Save the selected definitions in any file you choose, for example: `myload.pmm`.
6. Modify the `C:\Program Files\RadView\WebLOAD\bin\webload.ini` file, as follows:
 - a. Locate the `LOAD_PMM_FILE_PATH=""` parameter.
 - b. Insert the path and name of your selected definitions file. For example:
`LOAD_PMM_FILE_PATH="C:\\temp\\myload.pmm"`



Note: Currently, only the "Perfmon" measurements for probing Windows are supported when using the Remote PMM feature.

Firewall Settings

To configure the firewall settings:

- Open port 9000 in the firewall.

Note: Port 9000 is the default port for TestTalk, WebLOAD Console's network agent. You can change the port number through TestTalk's configuration file.

To change the default port:

1. Open the `C:\Program Files\RadView\WebLOAD\bin\webload.ini` file.
2. Modify the following parameter value from "9000" to a different port number, for example: "9011":
`TESTTALK_NETWORK_PORT="9011"`.
3. Save and close the file.
4. Exit TestTalk.
5. Restart WebLOAD Console.

Setting the Console

Once you have installed and configured the Remote Machine Load Generator, you can add a Remote Performance Measurement Manager script to your template. This script is run by the Remote Machine Load Generator when you run your template.

To add a Remote PMM script to the template:

1. Click **Add Remote PMM** in the **Home** tab of the Console ribbon. The Add PMM Probing Client dialog box is displayed.

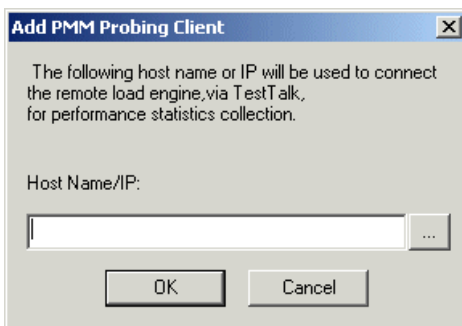



Figure 219: Add PMM Probing Client Dialog Box

2. Enter the remote host's name or IP address,

-Or-

Select a host from a list of available hosts by clicking . The Host Selection dialog box is displayed.

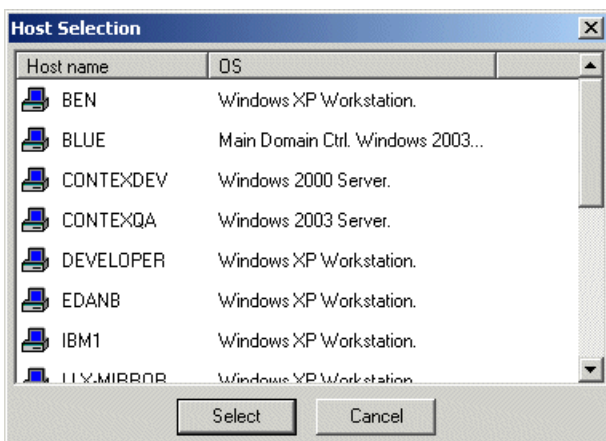


Figure 220: Host Selection Dialog Box

- a. Select a host.
- b. Click **Select**. The selected host name is displayed in the Host Name/IP field of the Add PMM Probing Client dialog box.

- Click **OK**. A new script called RemotePMMScript is automatically added to the template.

Note: You can rename the new script by right-clicking on the new script and typing in a new name.

Importing External Statistics

WebLOAD Console enables you to import statistical data from an external comma-delimited values file (.csv) into your report, while running a Load Session or upon test completion. Imported counters are added to the integrated reports tree, under `Import::<File name>`, where <File name> is the name of the file containing the external statistics. For more information about integrated reports, see *WebLOAD Console Performance Reports* (on page 295). All imported data is integrated with the Load Session's database and the WebLOAD Console statistics are also calculated for this information. For more information about WebLOAD Console statistics, see *WebLOAD Statistics Reports* (on page 317).



Imported data is also available for use with WebLOAD Analytics.

To import external statistics:

- Select **Import External Statistics** from the **Session** tab of the ribbon.

The Import External Statistics dialog box appears.

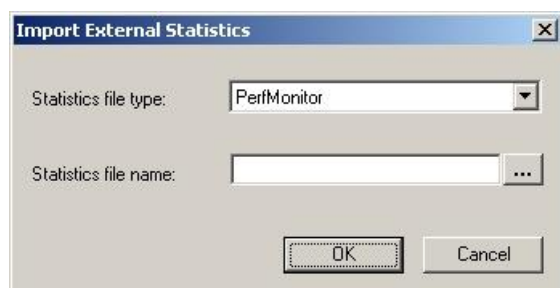


Figure 221: Import External Statistics Dialog Box

- In the Statistics File Type drop-down list, select one of the following types of statistics files:
 - Anue System**
 - PerfMonitor**
- In the Statistics File Name field, browse for or enter the name of the external statistics file. The file has a *.csv extension.
- Click **OK**.

The external statistics are imported into the report.

Integrating with AppDynamics

This section describes how to integrate WebLOAD with AppDynamics.

AppDynamics is an Application Performance Management (APM) solution that monitors, troubleshoots, and diagnoses problems in mission-critical applications.

By integrating WebLOAD scripts with AppDynamics you will be able to identify slow or problematic transactions of scripts run by WebLOAD test sessions. The integration consists of two tasks:

1. In WebLOAD – Specifying that a WebLOAD script should add a line in all transaction headers which will identify them as WebLOAD transactions. Refer to *Configuring WebLOAD Scripts for AppDynamics* (on page 409).
2. In AppDynamics – Configuring AppDynamics to enable it to read the headers. The additional line enables AppDynamics to identify the corresponding transactions as WebLOAD transactions. Refer to *Configuring AppDynamics to Recognize WebLOAD Transactions* (on page 410).

Configuring WebLOAD Scripts for AppDynamics

To enable WebLOAD for AppDynamics:

1. Add the line `wlGlobals.AddWebLoadHeader=true` in the JavaScript code of the script.

This causes WebLOAD to send extra data upon every request of the script. The extra data identifies the request as a WebLOAD request, lists the name of the currently running script, and lists the name of the corresponding transaction.

Configuring AppDynamics to Recognize WebLOAD Transactions

This section describes how to configure AppDynamics Lite and AppDynamics Pro to read WebLOAD transaction headers.

Configuring AppDynamics Lite to Recognize WebLOAD Transactions

To configure AppDynamics Lite to recognize WebLOAD transactions:

1. In AppDynamics Lite, select **Configure > Transaction Detection**.

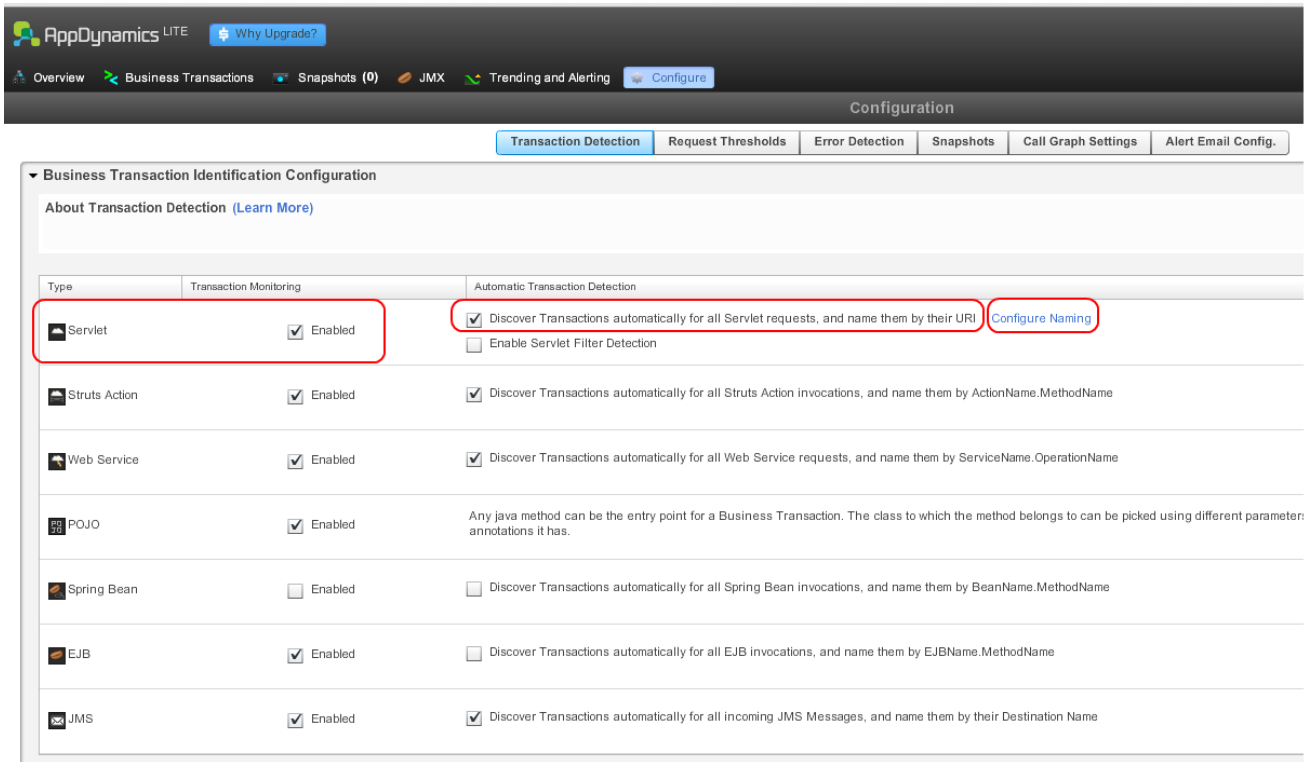


Figure 222: Transaction Detection Tab

2. In the **Servlet** transaction type:
 - a. Check the option **Enabled**.
 - b. Check the option **Discover Transactions automatically for all Servlet requests, and name them by their URI**.
3. Click **Configure Naming** (adjacent to **Discover Transactions automatically for all Servlet requests, and name them by their URI**).

The Servlet Transaction Naming Configuration window appears.

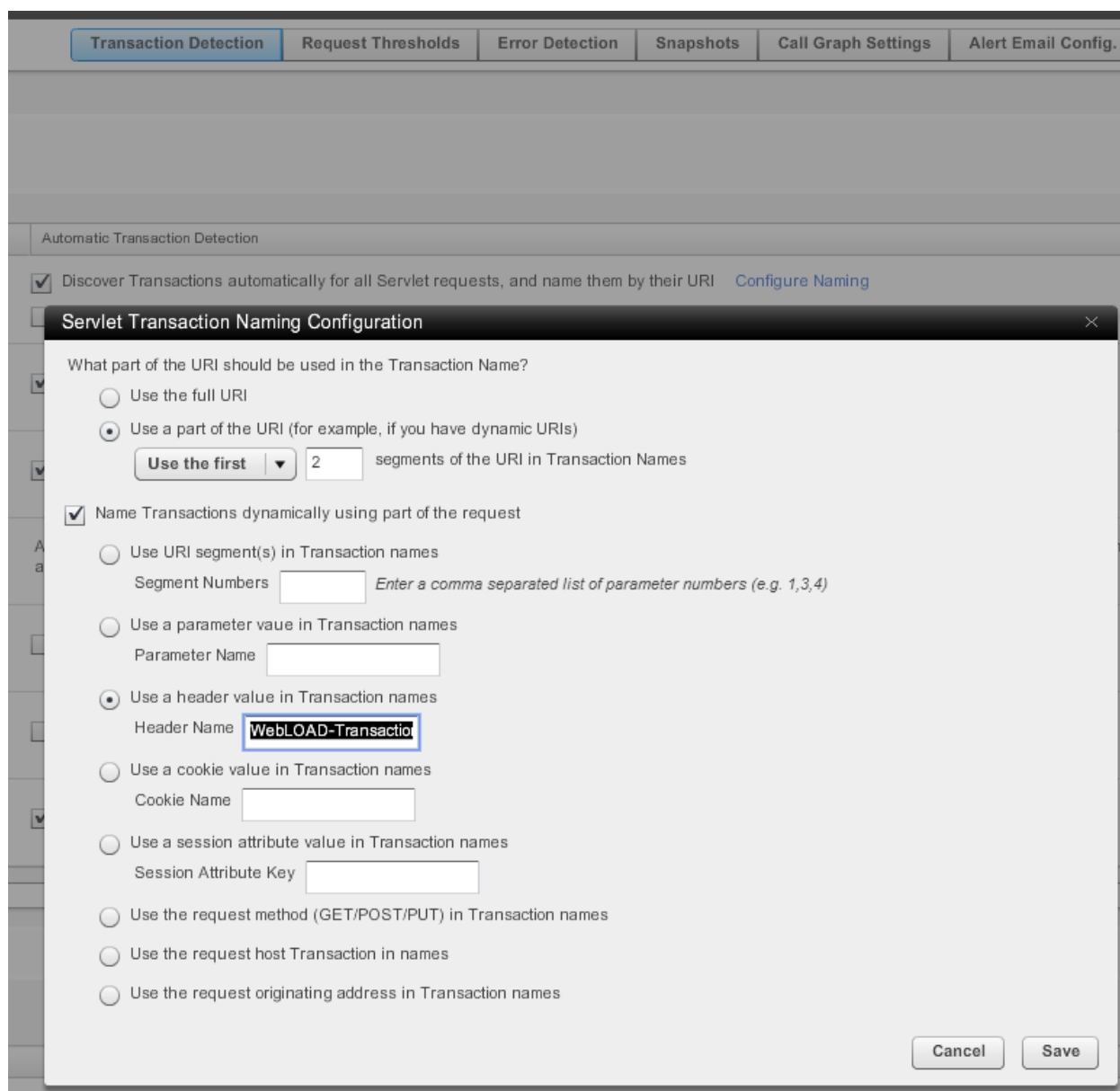


Figure 223: Servlet Transaction Naming Configuration Window

4. In the Servlet Transaction Naming Configuration window, specify the WebLOAD naming convention by performing the following:
 - a. Select **Use the full URI** or **Use a part of the URI**.
 - b. Select **Name Transactions dynamically using part of the request**.
 - c. Select **Use a header value in Transaction names**.
 - d. In the **Header Name** field, enter `WebLOAD-TransactionName`.
5. Click **Save**.

When you run a WebLOAD test, AppDynamics displays the WebLOAD transactions in its Business Transactions list, as described in *Viewing WebLOAD Performance in AppDynamics* (on page 417).

Configuring AppDynamics Pro to Recognize WebLOAD Transactions

In AppDynamics Pro, in addition to configuring AppDynamics to recognize WebLOAD transactions, you can also configure AppDynamics to add a "WebLOAD-TransactionName" prefix to each transaction name.

To configure AppDynamics Pro to recognize WebLOAD transactions:



Note: As a prerequisite, you must first create an account in AppDynamics. In the example shown below, the account name is called **WebLOADTest**.

1. In AppDynamics Pro, access the **Applications** menu and select the WebLOAD application.

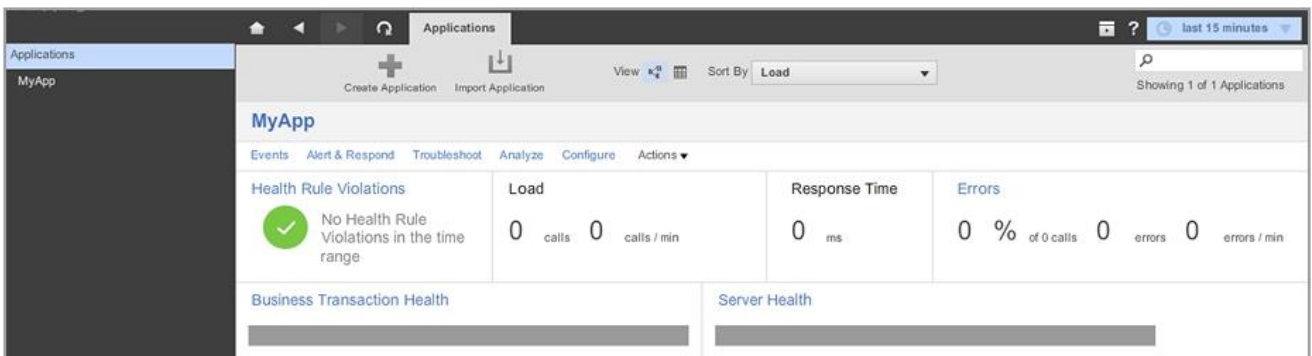


Figure 224: Applications Menu

2. Select **Configure > Instrumentation > Transaction Detection**.

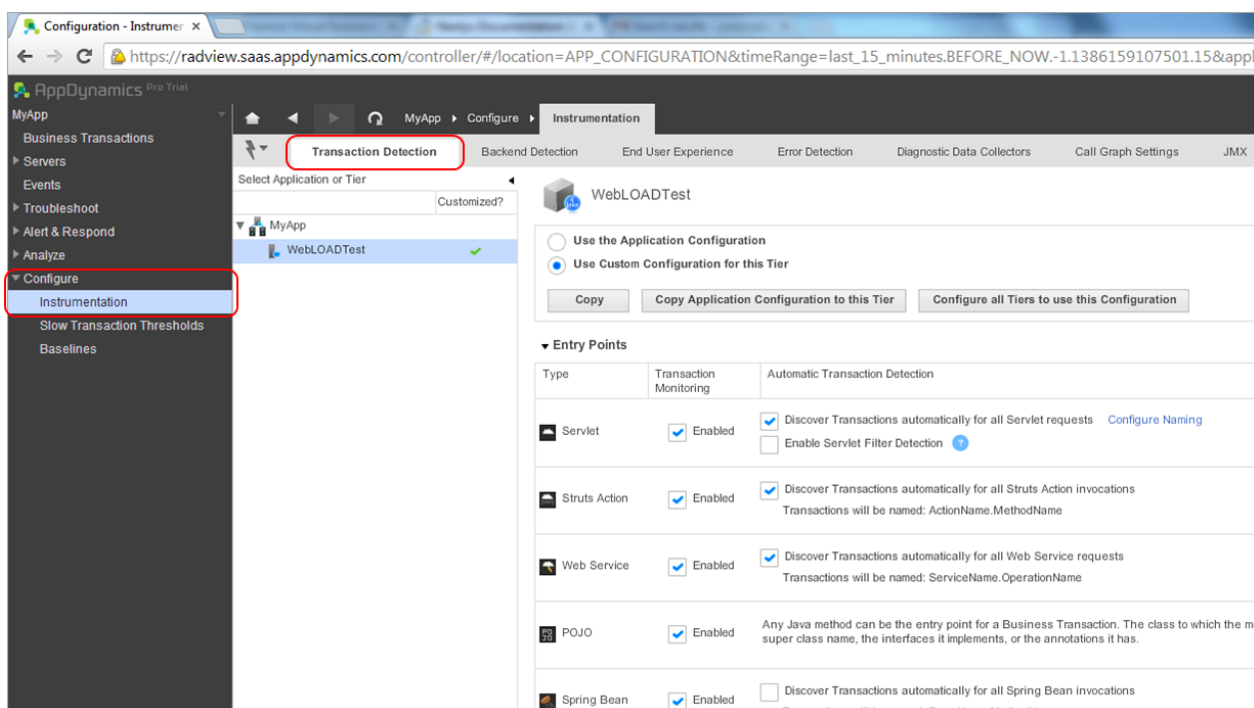


Figure 225: Transaction Detection Page

3. In the **Transaction Detection** page, select the WebLOAD node (called **WebLOADTest** in our example) and select **Use Custom Configuration for this Tier**.

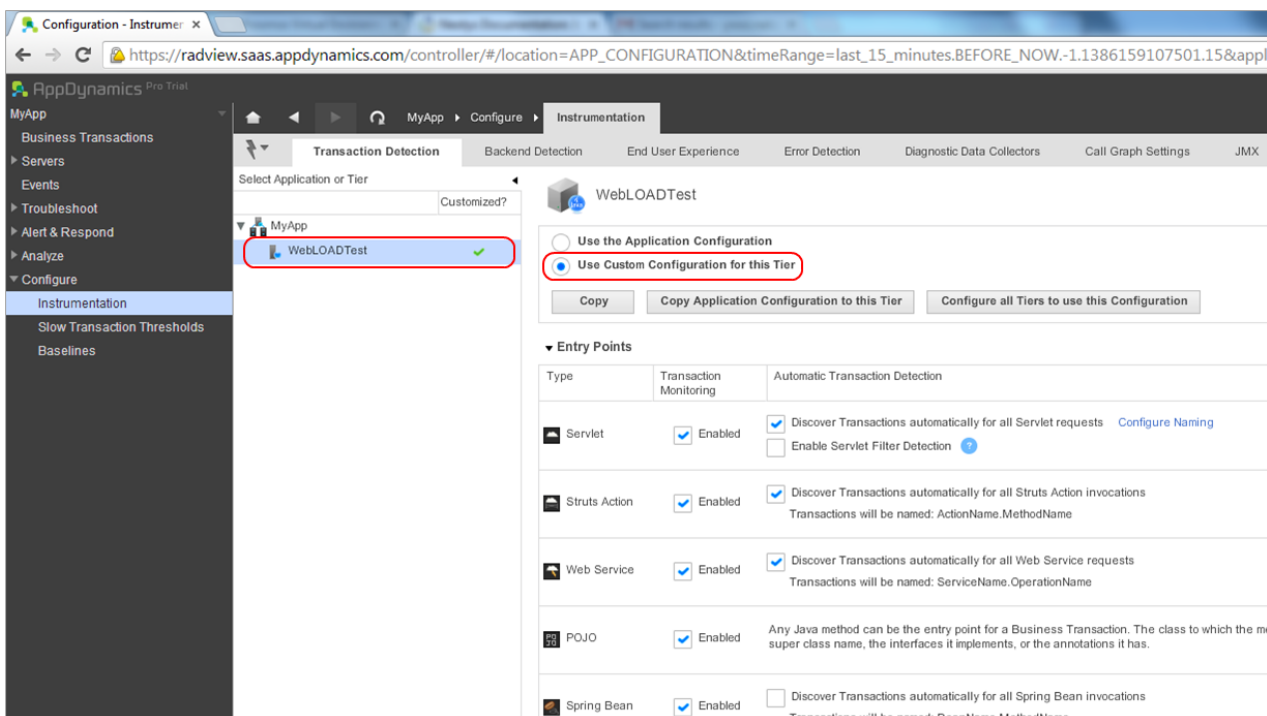


Figure 226: Transaction Detection – Use Custom Configuration

- In **Custom Rules**, click the green + (plus) icon.

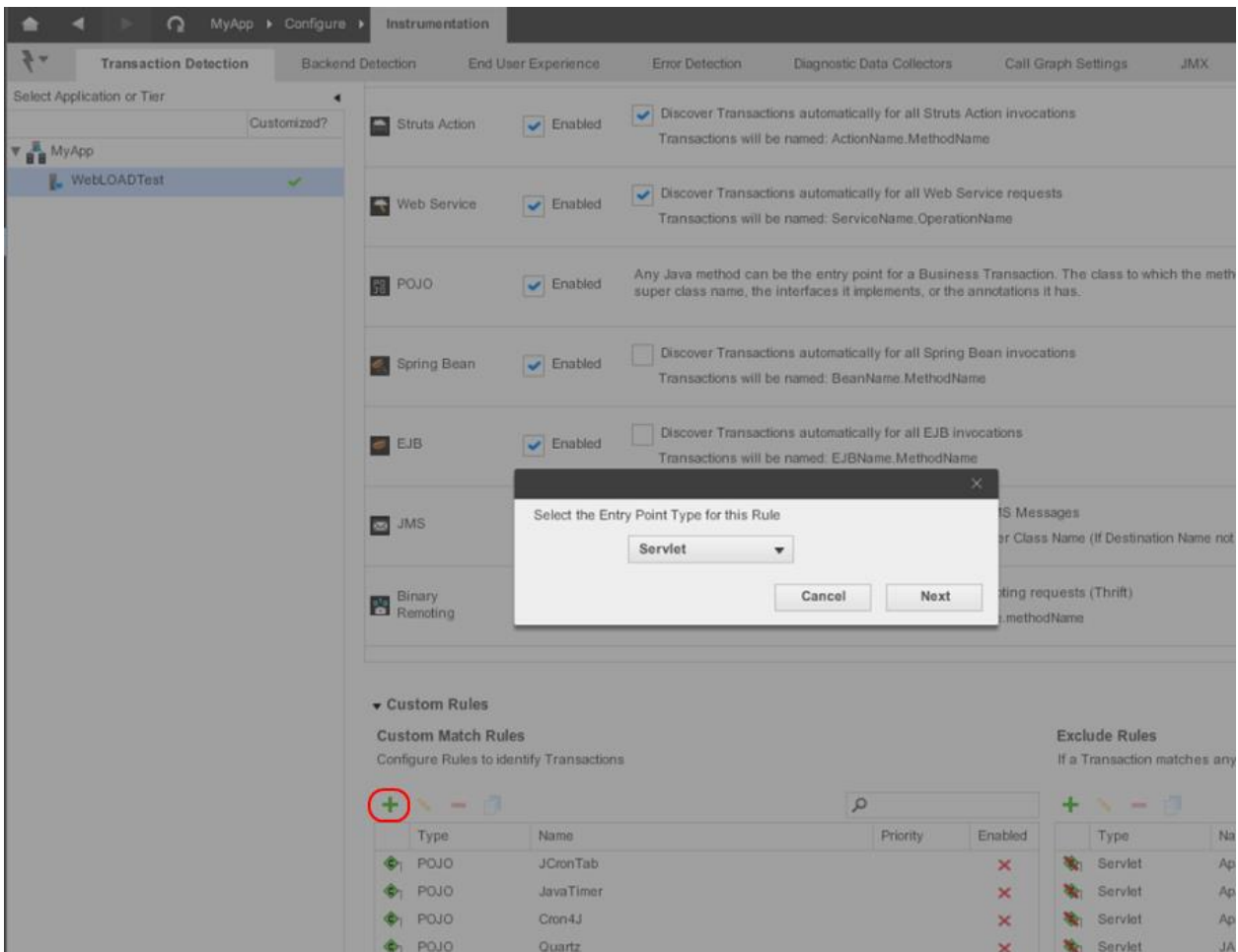


Figure 227: Selecting an Entry Point Type for a Rule

- In the Select the Entry Point Type for this Rule window that appears, select **Servlet** and click **Next**.

A New Business Transaction Match Rule – Servlet window appears, for defining a new rule.

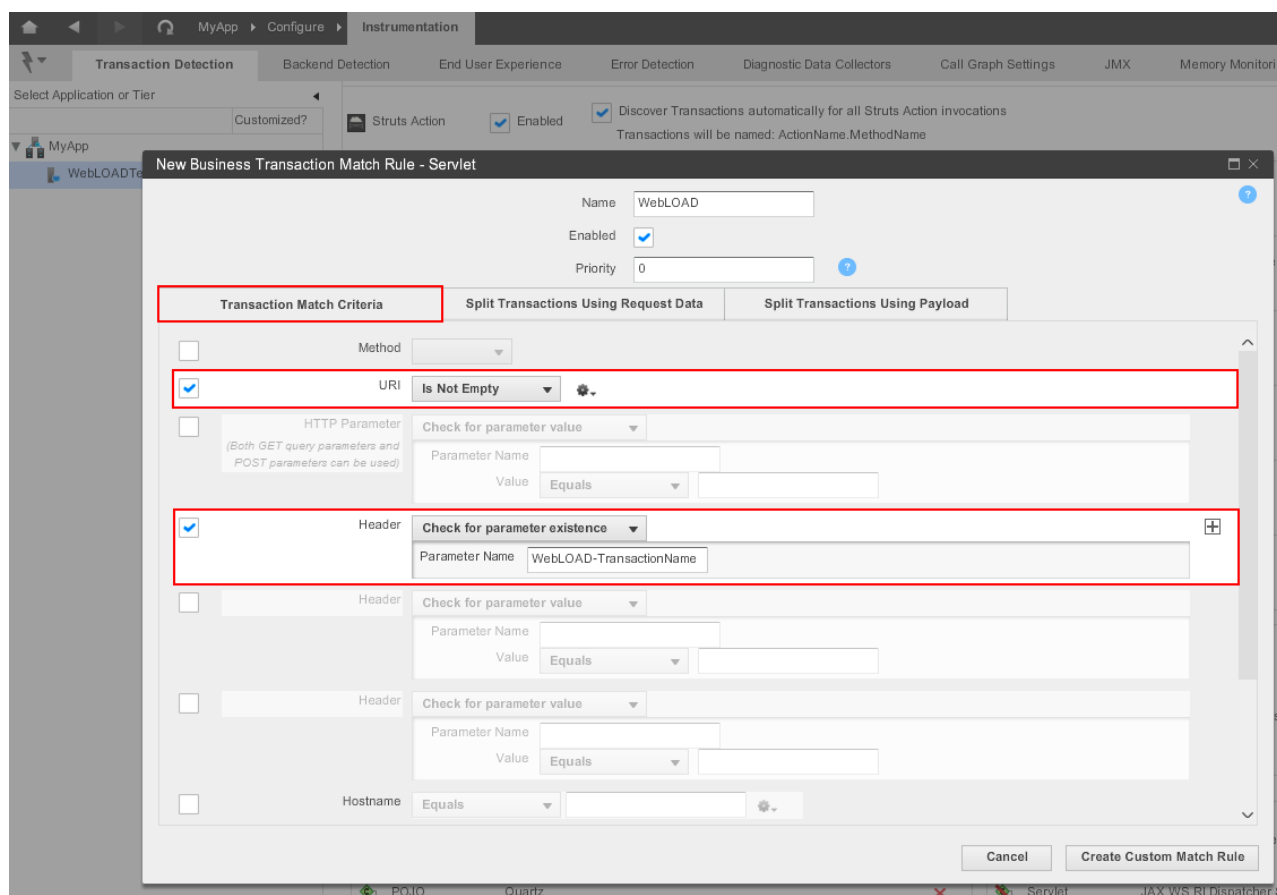


Figure 228: Defining a New Business Transaction Match Rule for a Servlet

6. Select the **Transaction Match Criteria** tab.
7. In the **Transaction Match Criteria** tab:
 - a. In **URI**, select **Is Not Empty**.
 - b. In **Header**:
 - Select **Check for parameter existence**.
 - In **Parameter Name**, enter **WebLOAD-TransactionName**.
 - c. Click **Create Custom Match Rule**.
8. Select the new rule in the **Custom Rules** list.

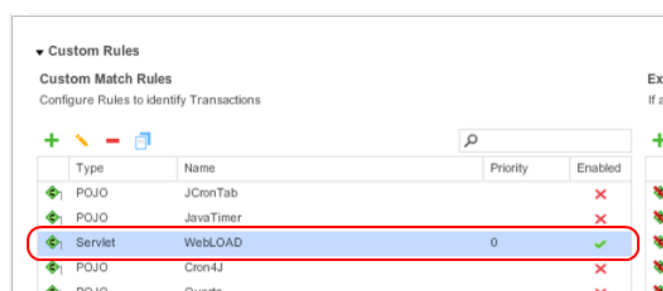


Figure 229: Selecting the WebLOAD Rule

9. In the Business Transaction Match Rule window that appears, select the **Split Transactions Using Request Data** tab.

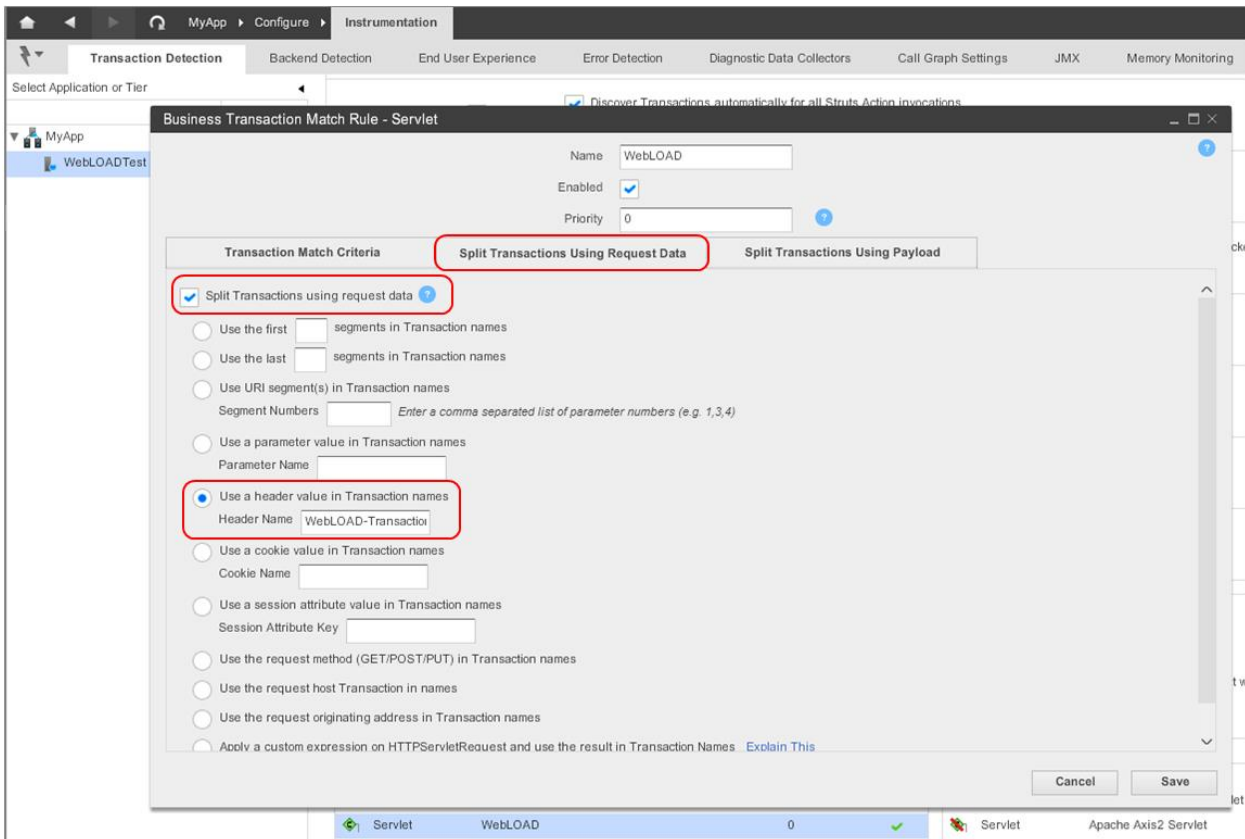


Figure 230: Specifying a WebLOAD Prefix for WebLOAD Transaction Names

10. In the **Split Transactions Using Request Data** tab:
 - a. Check the option **Split Transactions using request data**.
 - b. Select the option **Use a header value in Transaction names** and enter WebLOAD-TransactionName as the **Header Name**.

When you run a WebLOAD test, AppDynamics displays the WebLOAD transactions in its Business Transactions list, as described in *Viewing WebLOAD Performance in AppDynamics* (on page 417).

Integrating with Dynatrace

This section describes how to integrate WebLOAD with Dynatrace.

Dynatrace is a suite of Application Performance Monitoring (APM) tools that enables in-depth real-time application monitoring.

By integrating WebLOAD scripts with Dynatrace you can rapidly resolve and proactively prevent application performance problems. The integration consists of two tasks:

- Installing Dynatrace on your local computer.
- In WebLOAD – Enabling Dynatrace in the **Dynatrace** tab of the **Global Options** window (refer to *The Dynatrace Tab* on page 197). When Dynatrace is enabled, WebLOAD adds an identifier in all request headers which identifies them as WebLOAD requests.

After integrating WebLOAD with Dynatrace, you can perform the following:

- *Monitoring a WebLOAD Load Session in Dynatrace*
- *Viewing in Dynatrace a WebLOAD Point of Interest*
- *Viewing in Dynatrace the Transactions Related to WebLOAD Errors*

Monitoring a WebLOAD Load Session in Dynatrace

To monitor a WebLOAD Load Session in Dynatrace:

1. Enable Dynatrace in *The Dynatrace Tab* (on page 197).
2. In the Dynatrace **Monitoring** page, click the **Synthetic Request** node.

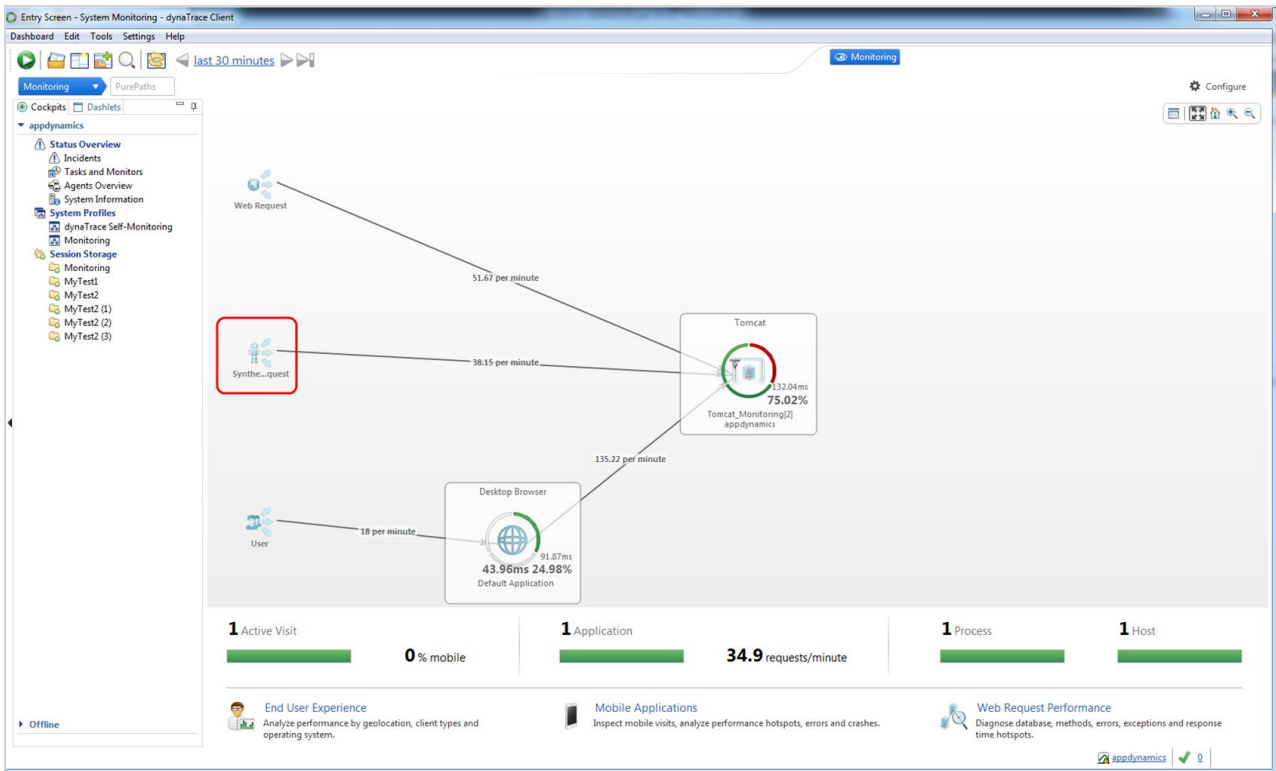


Figure 232: Selecting the Synthetic Request Node in Dynatrace

A dropdown list of options appears, enabling you to access the various monitoring and drill-down options offered by Dynatrace.

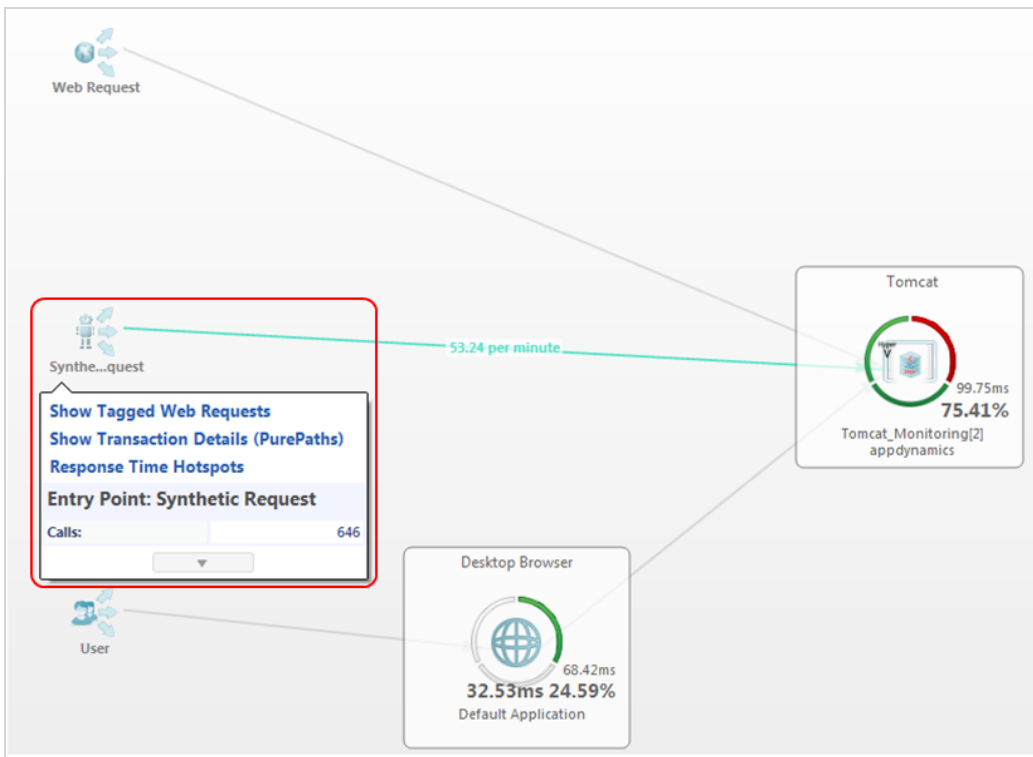


Figure 233: Monitoring Options in Dynatrace

Viewing in Dynatrace a WebLOAD Point of Interest

You can access Dynatrace directly from a WebLOAD's report window to view in Dynatrace the monitoring data of any point in a WebLOAD graph.

To view Dynatrace data for a WebLOAD point of interest:

1. Enable Dynatrace in *The Dynatrace Tab* (on page 197).
2. In a WebLOAD report, right-click a point in a graph and select **DynaTrace**.

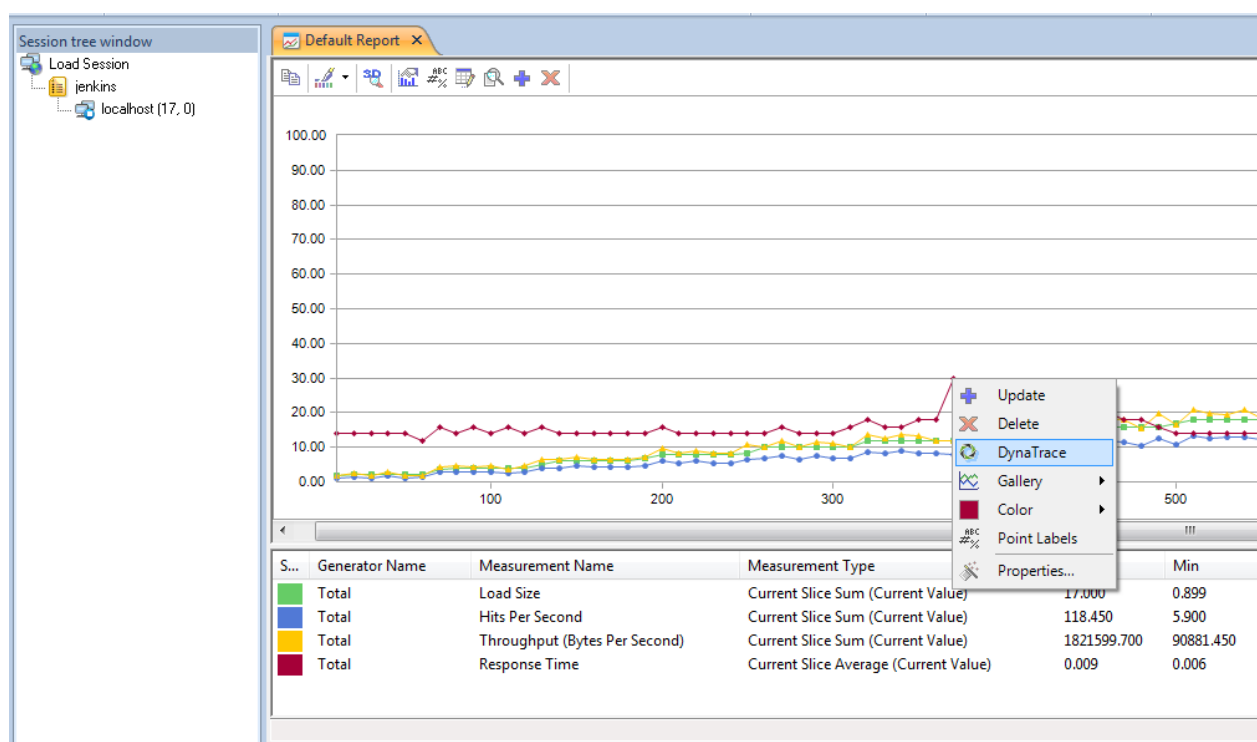


Figure 234: Selecting DynaTrace for a Point in a Graph

3. A Dynatrace window appears, showing data for the time period starting from 10 minutes prior to the point of interest, and ending 10 minutes after the point of interest.

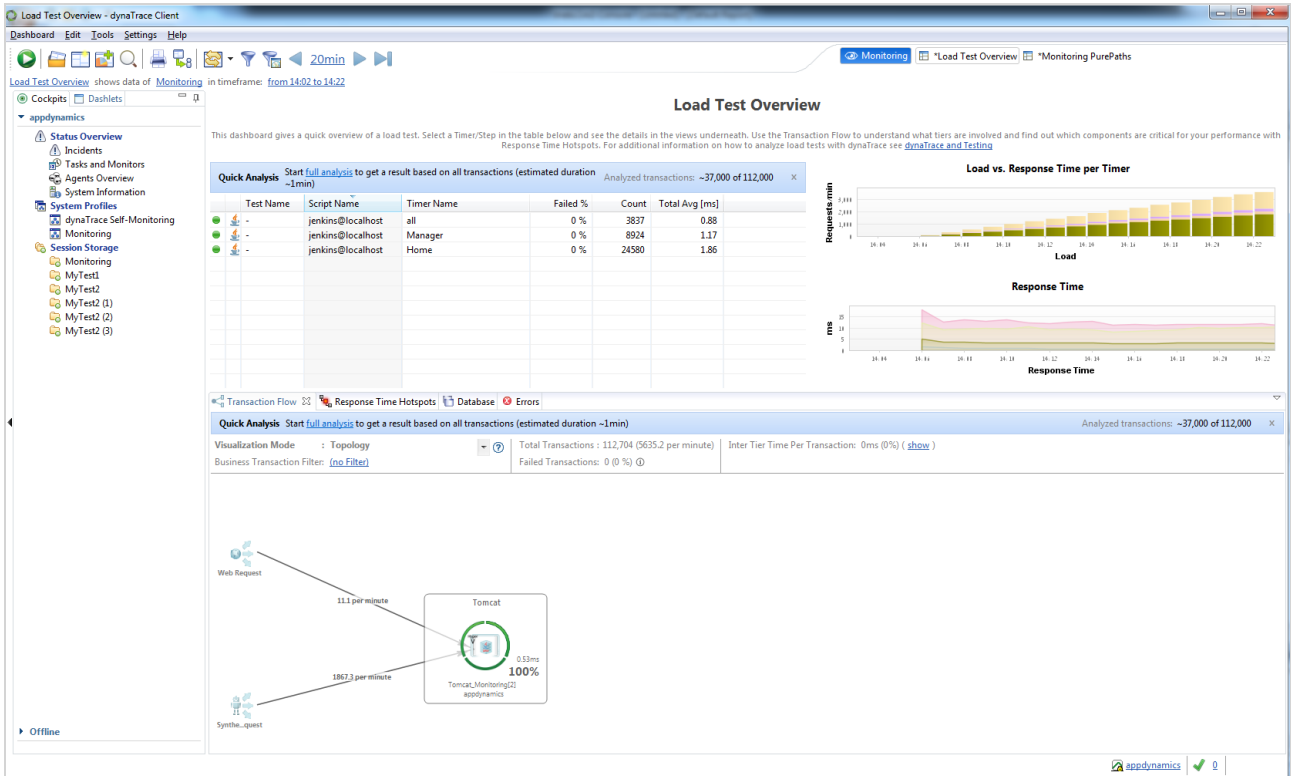


Figure 235: Dynatrace Data for the Interval Around a Selected Point

Specifying the Dynatrace System Profile

When you enable Dynatrace in WebLOAD (refer to *The Dynatrace Tab* on page 197), by default WebLOAD specifies that points of interest viewed in Dynatrace will be viewed in the **Monitoring** system profile. If you wish to change this default setting, do the following:

1. Click **Global Options** in the **Tools** tab of the ribbon,

-Or-

Select **Global Options** from the Console System button.

2. Select the **Dynatrace** tab.

3. In the **Timestamp** field, change:

`rest/integration/opendashboard?source=live:Monitoring&filter=tf:CustomTimeframe?`

To:

`rest/integration/opendashboard?source=live:<Your-desired-system-profile>&filter=tf:CustomTimeframe?`

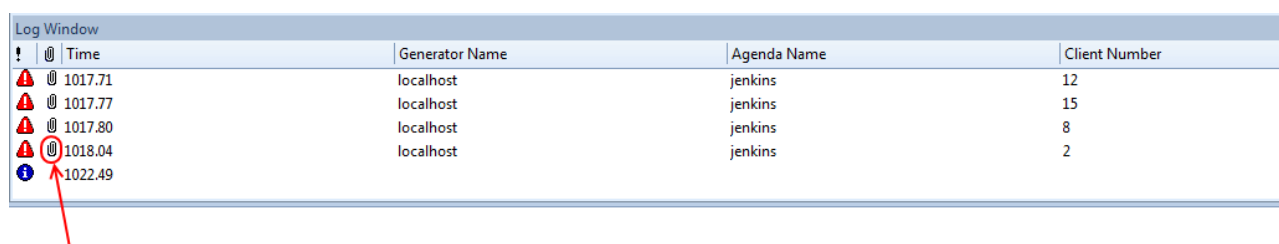
Whenever you click a point of interest in WebLOAD and select **DynaTrace**, the point of interest will be displayed in the Dynatrace system profile you specified in the **Timestamp** field.

Viewing in Dynatrace the Transactions Related to WebLOAD Errors

If an error occurs at any time during a test session, an error message displays in the Log Window. If you installed Dynatrace, you can access Dynatrace directly from WebLOAD's Log Window to view in Dynatrace the transaction associated with the error.

To view a transaction in Dynatrace that is associated with a WebLOAD error:

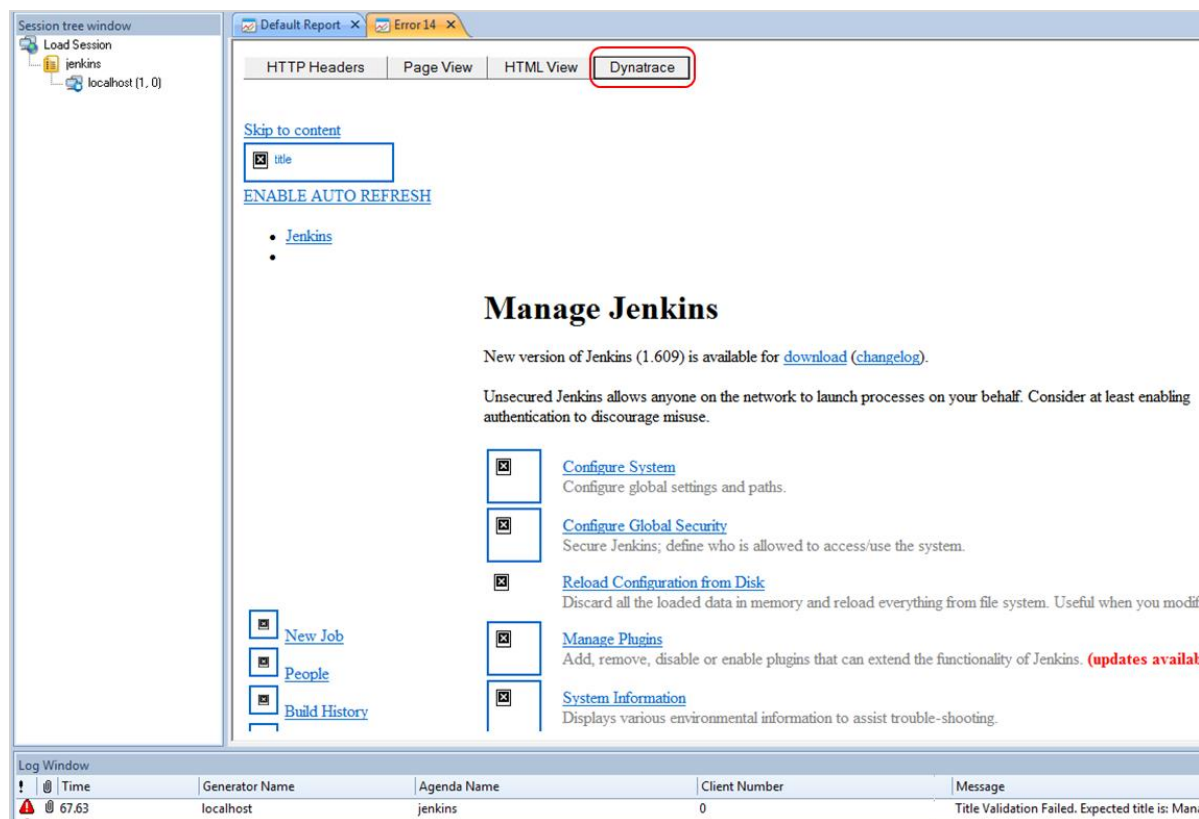
1. Enable Dynatrace in *The Dynatrace Tab* (on page 197).
2. In the Log Window, double-click the paperclip icon adjacent to the error.



!	@	Time	Generator Name	Agenda Name	Client Number
!	@	1017.71	localhost	jenkins	12
!	@	1017.77	localhost	jenkins	15
!	@	1017.80	localhost	jenkins	8
!	@	1018.04	localhost	jenkins	2
!	@	1022.49			

Figure 236: Selecting an Attachment of an Error in the Log Window

3. In the Event Viewer that appears, click the **Dynatrace** button



Session tree window

- Load Session
 - jenkins
 - localhost (1, 0)

Default Report - x Error 14 - x

HTTP Headers Page View HTML View **Dynatrace**

[Skip to content](#)

[ENABLE AUTO REFRESH](#)

- [Jenkins](#)
-

Manage Jenkins

New version of Jenkins (1.609) is available for [download](#) ([changelog](#)).

Unsecured Jenkins allows anyone on the network to launch processes on your behalf. Consider at least enabling authentication to discourage misuse.

- [Configure System](#)
Configure global settings and paths.
- [Configure Global Security](#)
Secure Jenkins; define who is allowed to access/use the system.
- [Reload Configuration from Disk](#)
Discard all the loaded data in memory and reload everything from file system. Useful when you modify
- [Manage Plugins](#)
Add, remove, disable or enable plugins that can extend the functionality of Jenkins. **(updates available)**
- [System Information](#)
Displays various environmental information to assist trouble-shooting.

[New Job](#)

[People](#)

[Build History](#)

Log Window

!	@	Time	Generator Name	Agenda Name	Client Number	Message
!	@	67.63	localhost	jenkins	0	Title Validation Failed. Expected title is: Man

Figure 237: Selecting Dynatrace in the Event Viewer

4. A Dynatrace window appears, displaying the details of the relevant transaction.

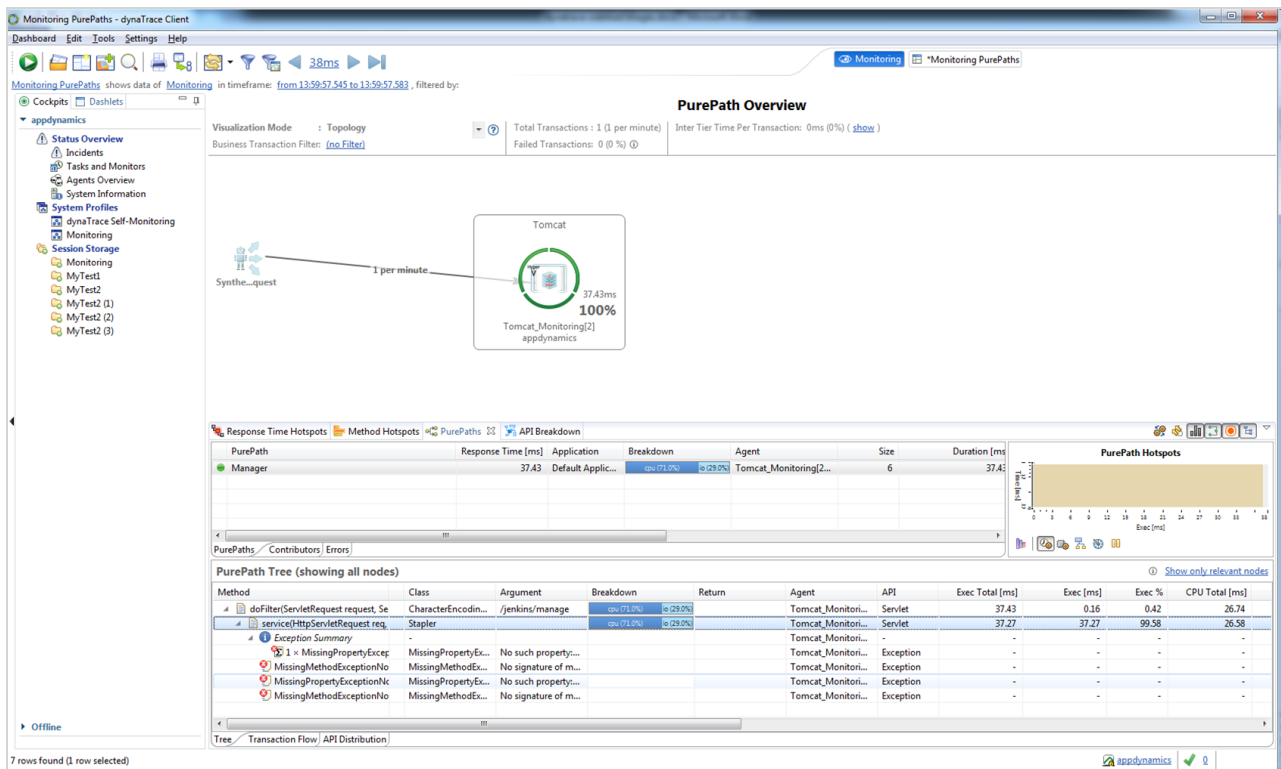


Figure 238: Viewing in Dynatrace the Transaction Related to a WebLOAD Error

Integrating with Nagios

This section describes how to integrate WebLOAD with Nagios. After integrating WebLOAD with Nagios, you can utilize data collected by Nagios to be used for WebLOAD monitoring purposes

Nagios is a free and open source computer-software application that monitors systems, networks and infrastructure. Nagios offers monitoring and alerting services for servers, switches, applications and services. It alerts users when things go wrong and alerts them a second time when the problem has been resolved.

The integration assumes you have installed and set up Nagios monitoring in your network.

Instructing PMM to Collect Data from Nagios

Nagios integration is carried out by adding the Nagios data source in the Performance Measurements Manager (PMM), and instructing the PMM which data, collected by Nagios, to retrieve.

To instruct PMM to collect data from Nagios:

1. Run the Performance Measurements Manager wizard, as described in [Opening the Performance Measurements Manager](#) (on page 364).
2. In the [Selecting a Data Source](#) screen, select **External > Nagios**.

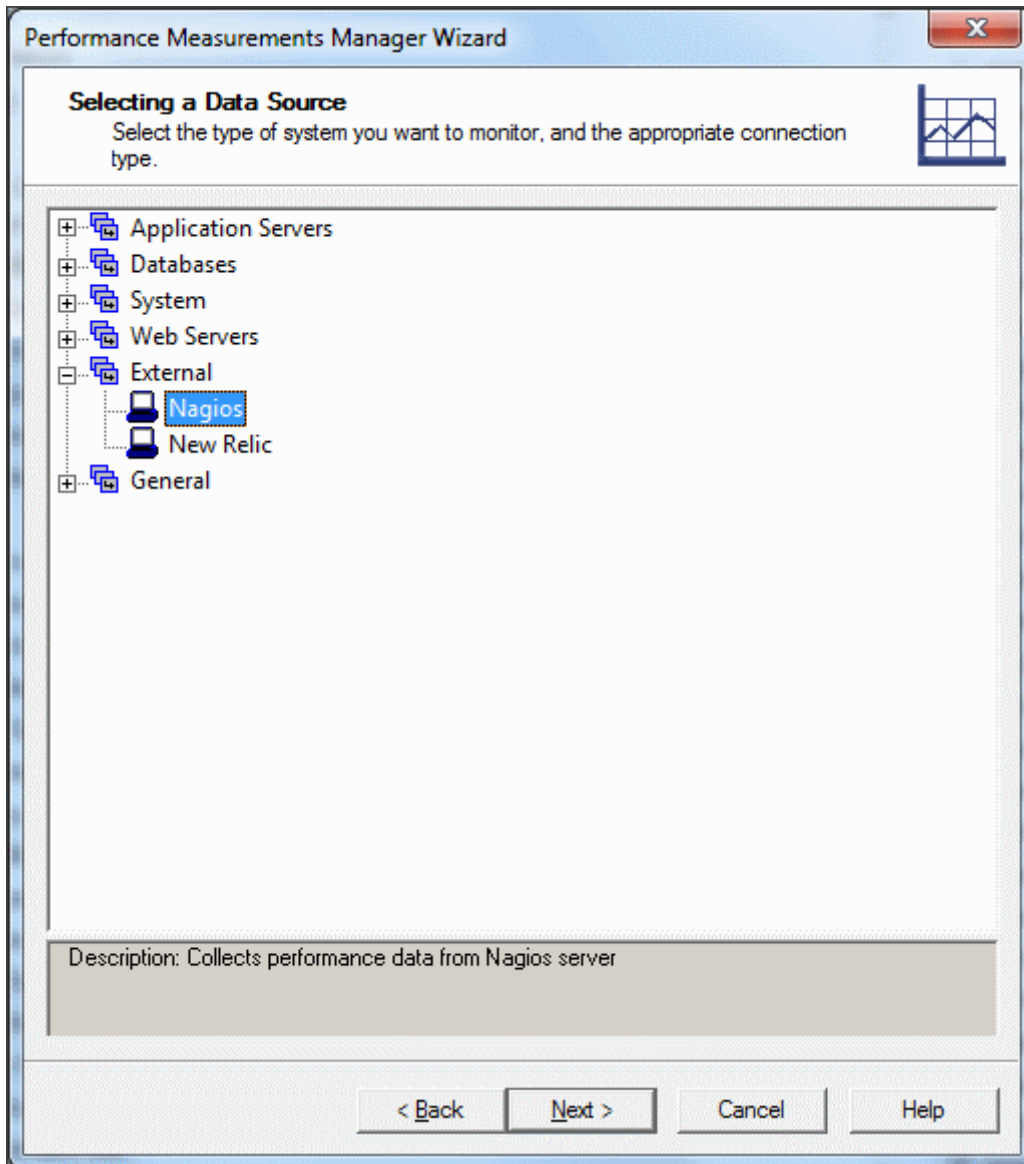


Figure 239: Selecting Nagios in Data Source Screen

3. In the Selecting a Host screen, enter the Nagios host name, username and password.

Performance Measurements Manager Wizard

Selecting a Host
Select the host to monitor and click Next to configure the specific performance measurement to report.

Enter the name of the host to monitor, or click the Browse button to select from the available hosts.

Note: Identification of hosts running in a Windows environment is limited to 15 characters.

Connection settings

PMM Provider	Nagios
URL	http://{host}
User name	<input type="button" value="Show username"/>
Password	*****

Password
Enter the password.

Figure 240: Specifying Nagios Server in Host Selection Screen

4. In the Selecting the Measurements to Monitor screen, do the following for each site/OS/host/device you wish to monitor
 - a. Expand a site/OS/host/device by clicking the + adjacent to the item.

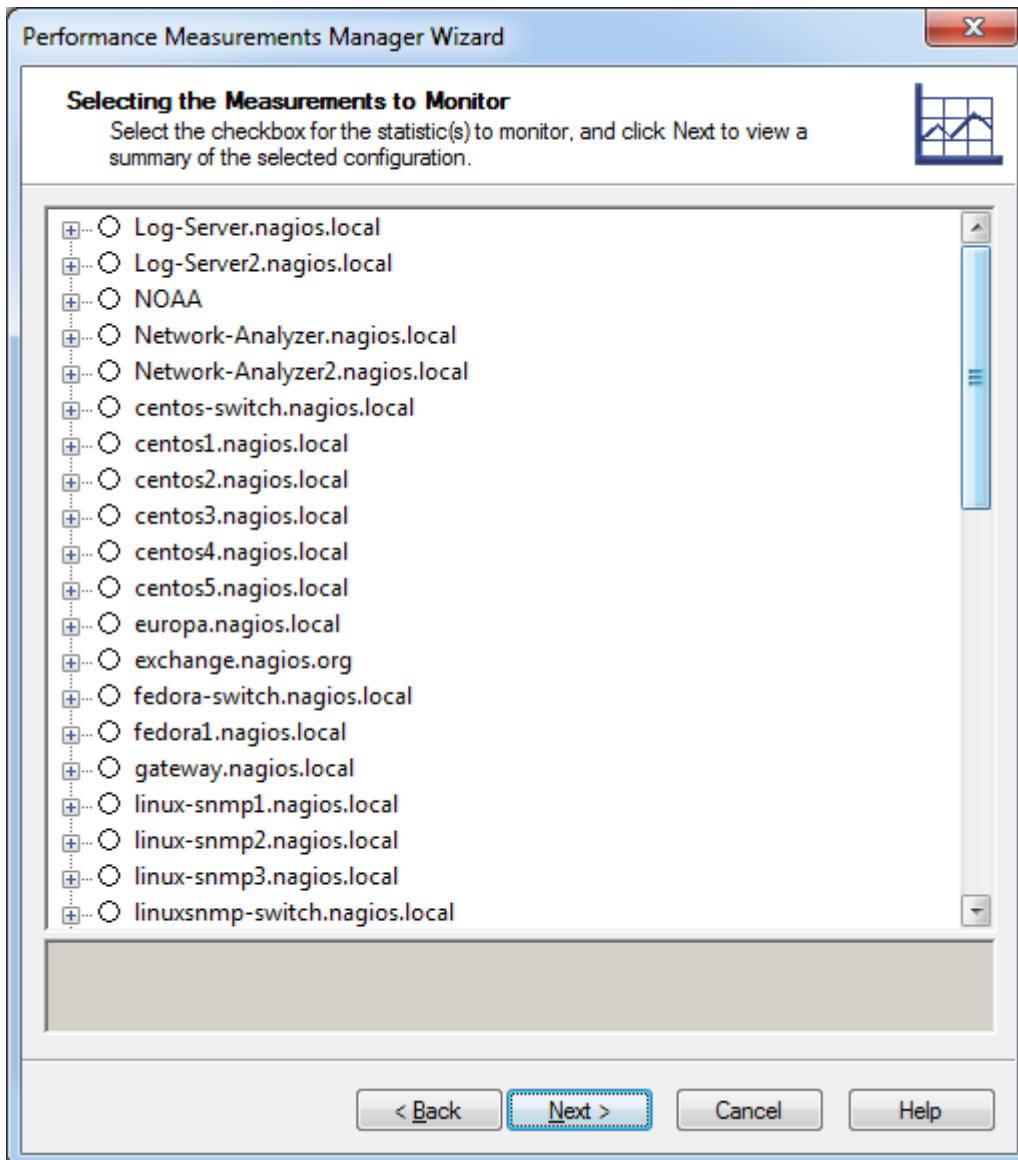


Figure 241: Selecting the Item to Monitor

- b. The sub components list all the statistics collected by Nagios for this item. Select the desired statistics.

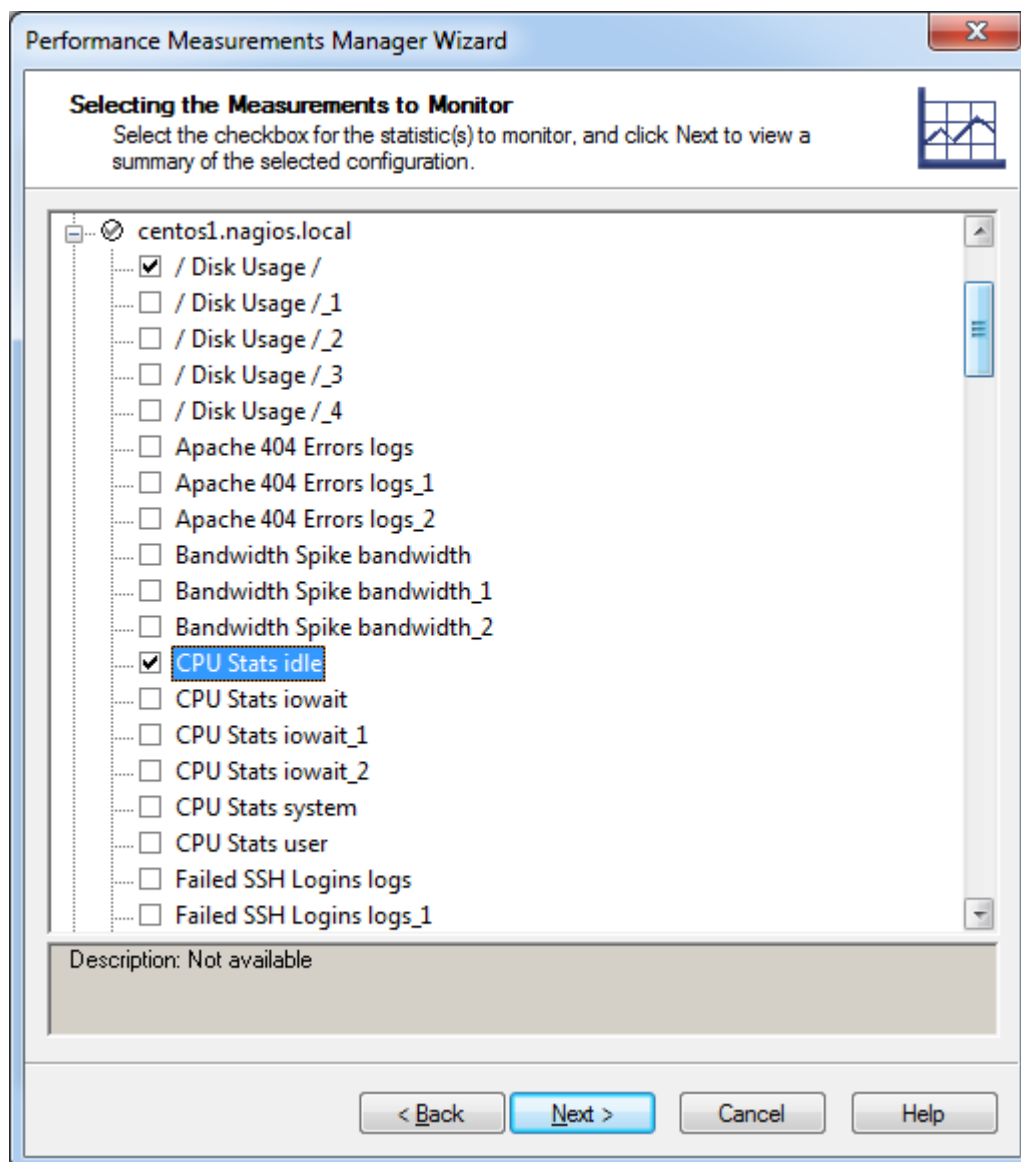


Figure 242: Selecting the Measurements to Monitor Screen

5. After all the desired items and statistics are selected, click **Next**. The Wizard displays a summary of the host, data source, and measurements configured for monitoring.
6. To accept the PMM configuration, click **Finish**.

The PMM Wizard closes and the selected configuration is added to the PMM main window.

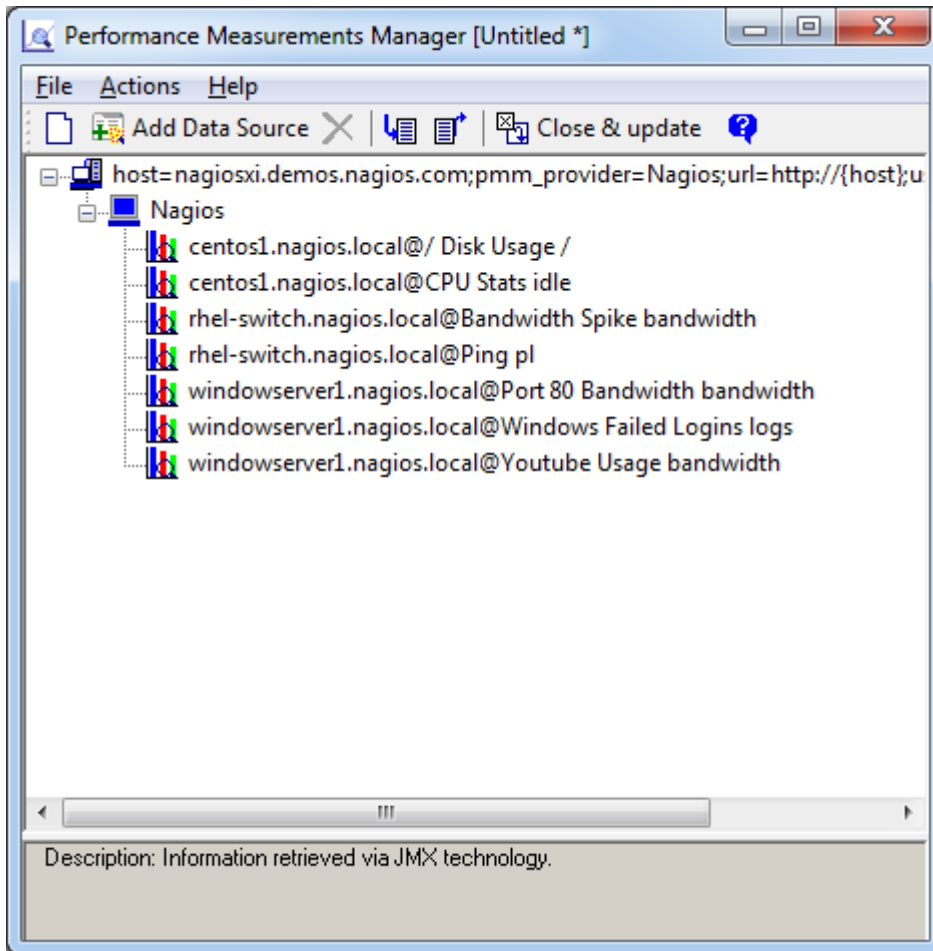


Figure 243: PMM Wizard Configuration Summary

Selecting the Nagios Statistics to Display in Reports

After integrating WebLOAD with Nagios, you can specify, while running a session, which Nagios statistics to view in the report view.

1. Open a report as described in *Opening Reports* (on page 298).
2. In the **PM@<Nagios-host>** node, select the statistics you wish to display in the report.

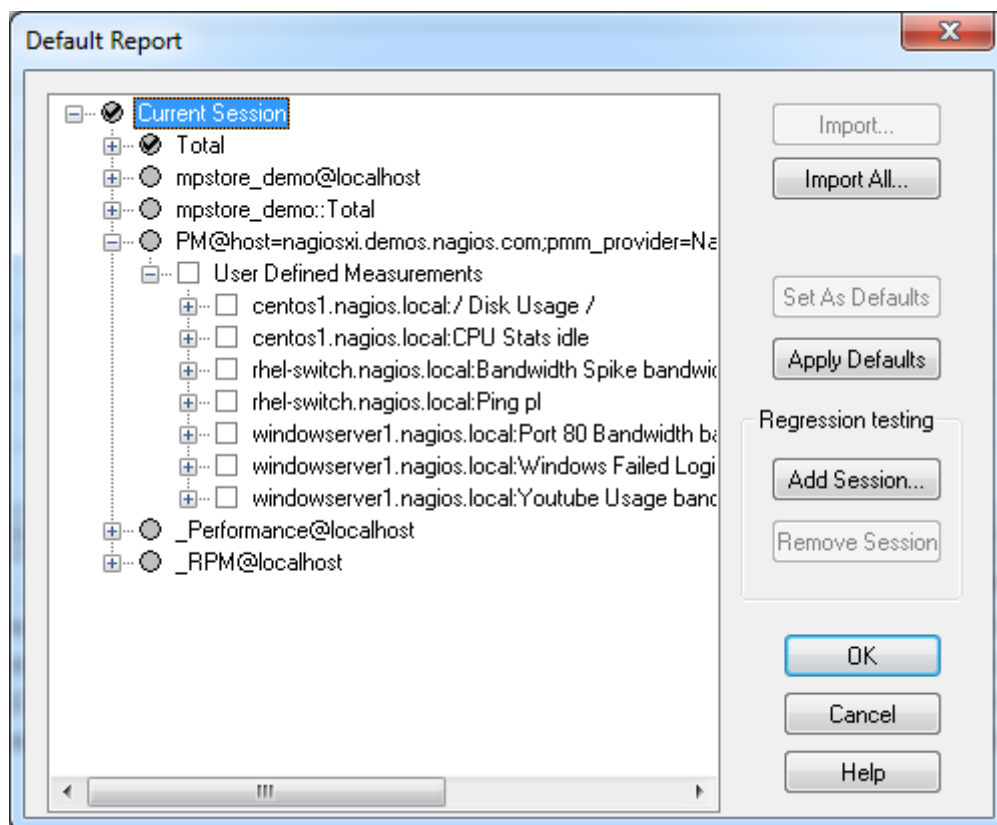


Figure 244: Selecting the Statistics to Display in a Report

The following figure shows a report in Report view, displaying the statistics collected by Nagios that were selected to appear in the report.

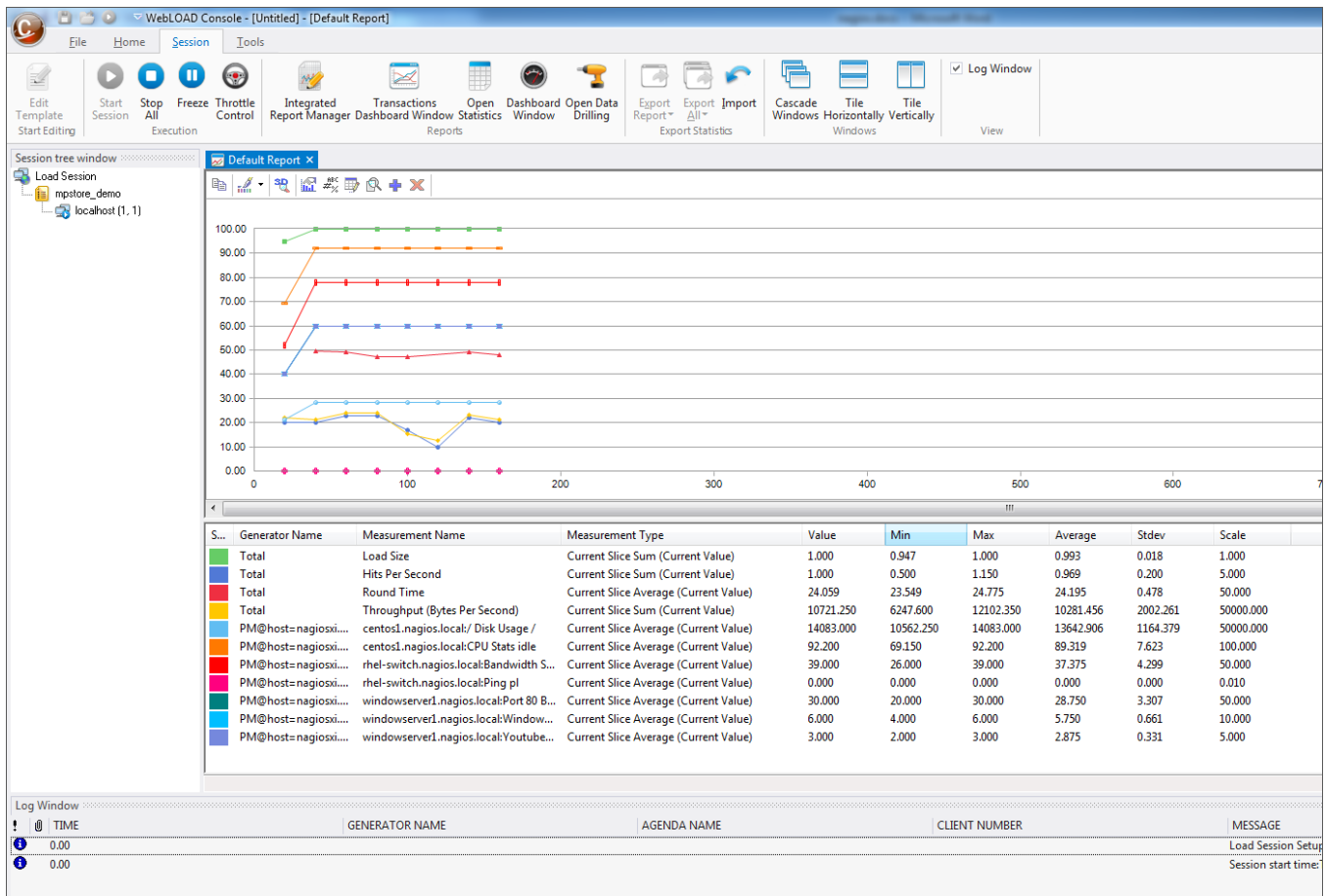


Figure 245: Viewing Nagios Statistics in Report View

Integrating with New Relic

This section describes how to integrate WebLOAD with New Relic. After integrating WebLOAD with New Relic, you can utilize data collected by New Relic to be used for WebLOAD monitoring purposes.

New Relic is a cloud-based performance management solution for measuring and monitoring the performance of applications and infrastructure.

Prerequisites

Prior to integrating New Relic in WebLOAD, make sure you fulfill the following prerequisites:

- You have installed and set up New Relic monitoring in your network.
- You have at least a New Relic PRO license.
- You have a New Relic API key, as described in *Get or Generate a New Relic API Key*.

Get or Generate a New Relic API Key

To get or generate a New Relic API key:

1. Access New Relic as a Pro user.
2. Navigate to **User > Account Settings**.

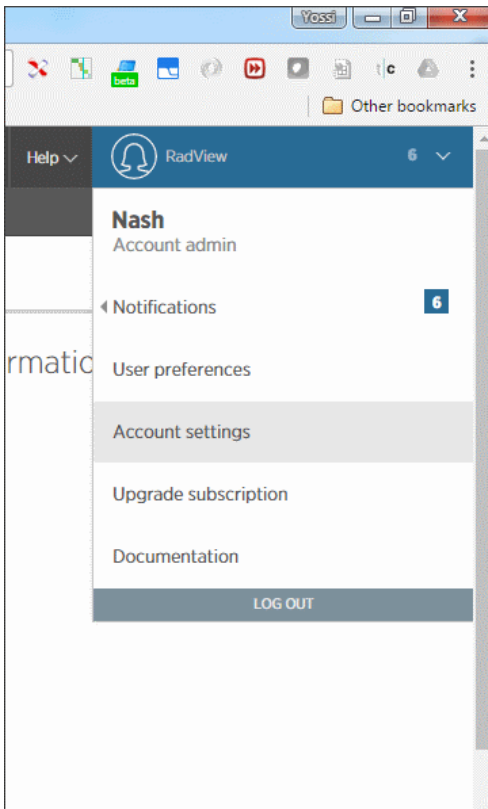


Figure 246: New Relic – selecting Account Settings

3. Select **Integrations** > **API Keys**.

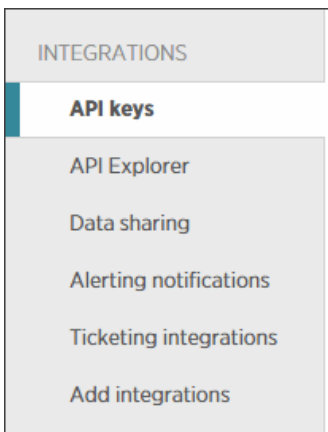


Figure 247: New Relic – selecting the API Keys option

4. In the **API Keys** page, in the **Admin's API Key** column, click **(Show key)** in the appropriate line.

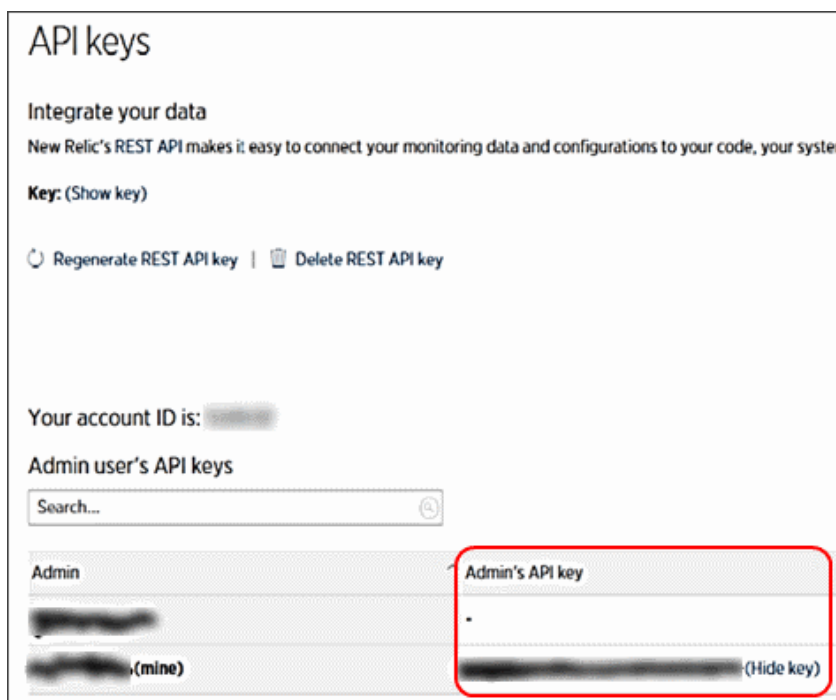


Figure 248: New Relic – Displaying the API Key value

5. Record the value of the API key. You will need to enter it in the PMM later on.

Instructing PMM to Collect Data from New Relic

New Relic integration is carried out by adding the New Relic data source in the Performance Measurements Manager (PMM), and instructing the PMM which data, collected by New Relic, to retrieve.

To instruct PMM to collect data from New Relic:

1. Run the Performance Measurements Manager wizard, as described in [Opening the Performance Measurements Manager](#) (on page 364).
2. In the [Selecting a Data Source](#) screen, select **External > New Relic**.

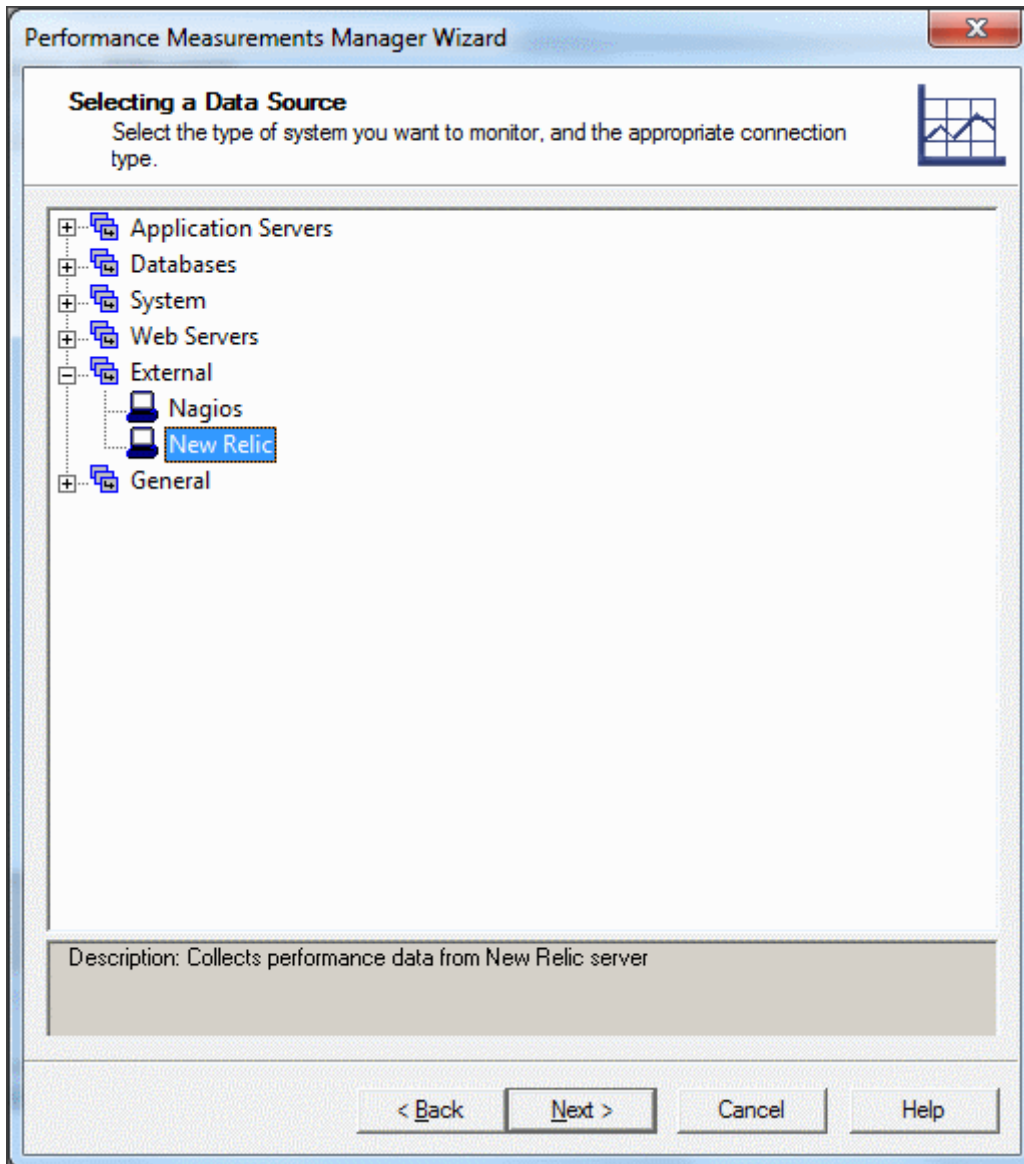


Figure 249: Selecting New Relic in Data Source Screen

3. In the Selecting a Host screen, enter the **API Key** you had recorded in *Get or Generate a New Relic API Key* (on page 433).
Note that you can change the New Relic **Root URL** if required; in most cases the default value is the correct one.

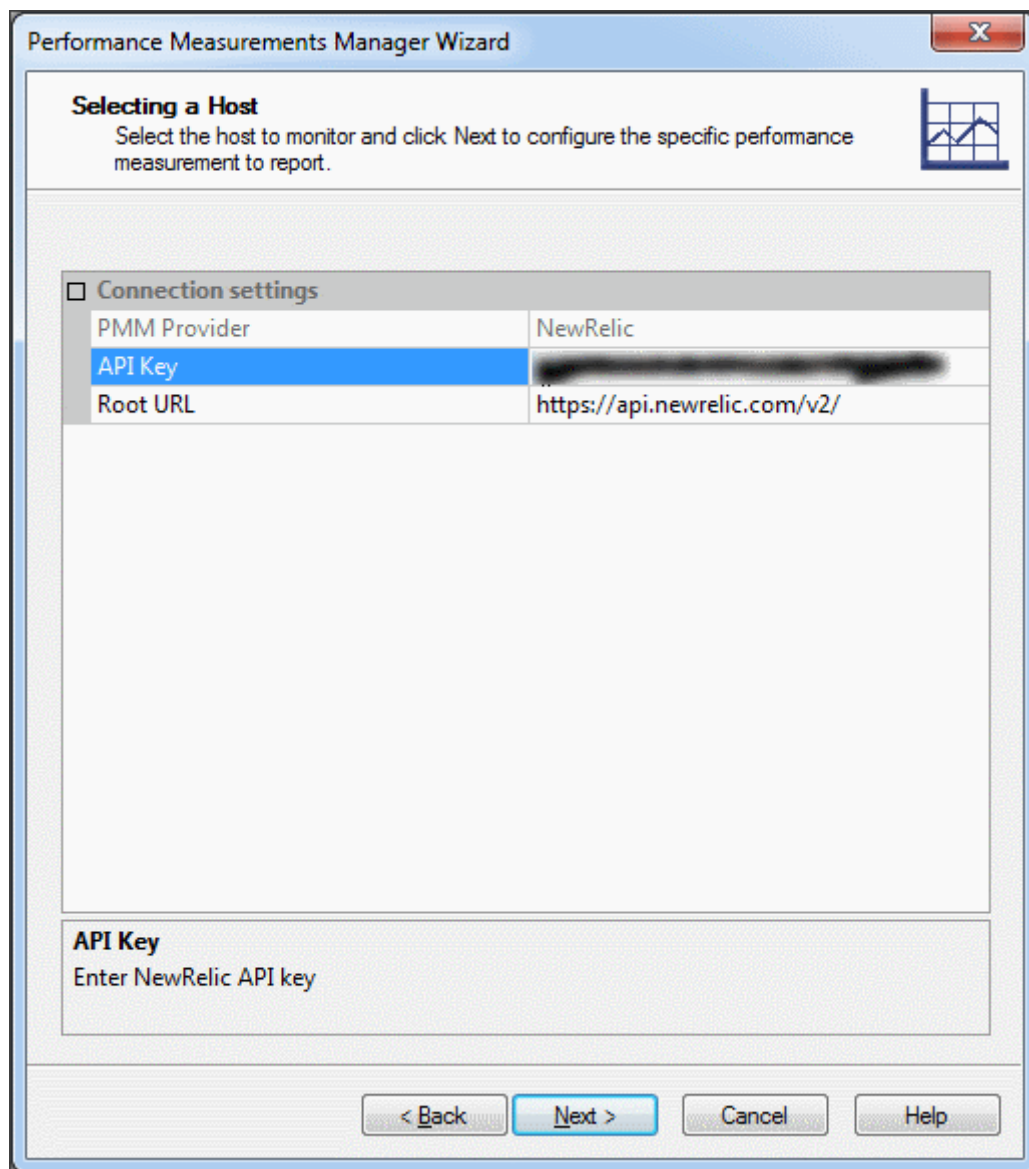


Figure 250: Specifying New Relic Server in Host Selection Screen

4. In the Selecting the Measurements to Monitor screen, do the following for each site/OS/host/device you wish to monitor
 - a. Expand a site/OS/host/device by clicking the + adjacent to the item.

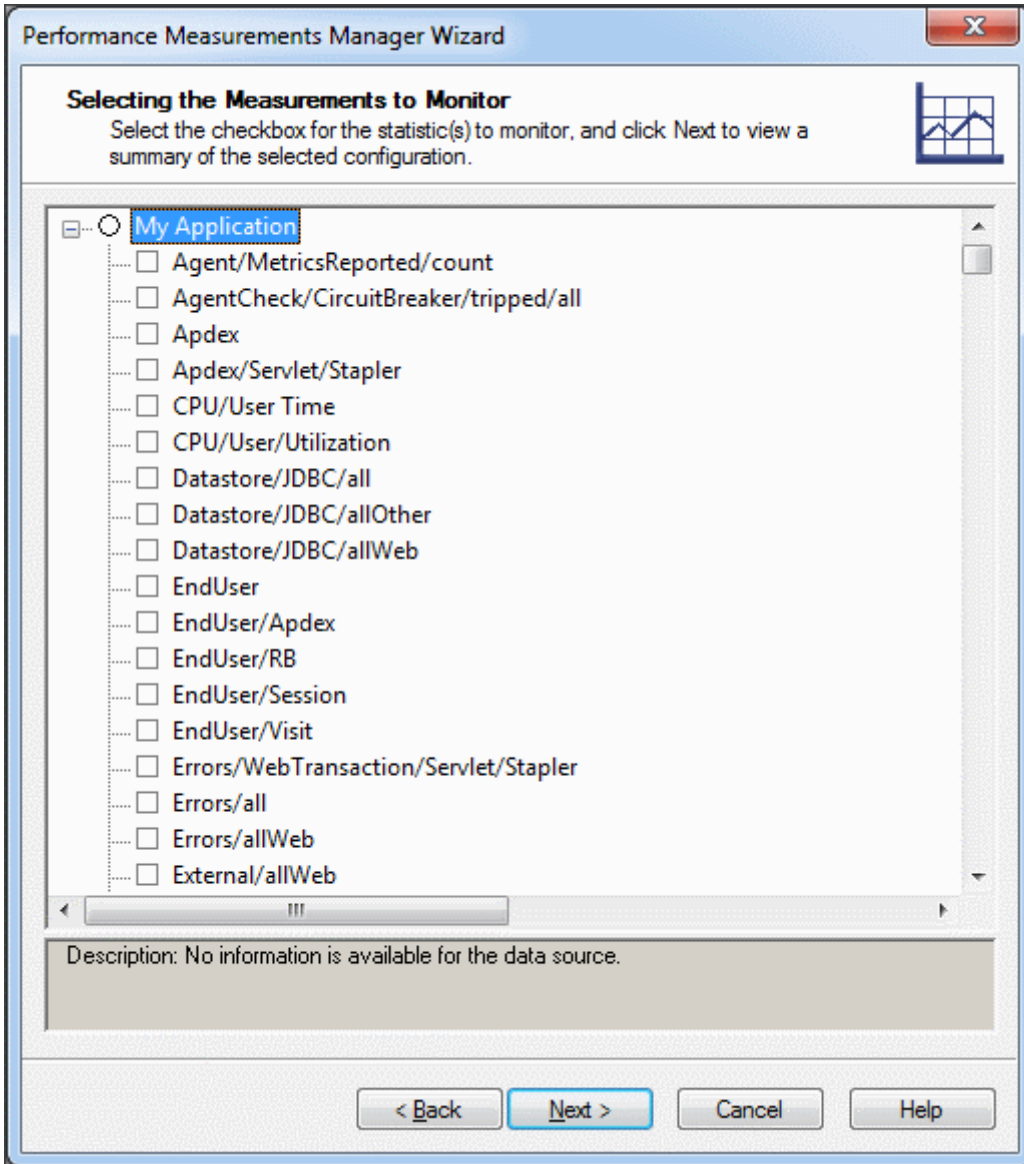


Figure 251: Selecting the Item to Monitor

- b. The sub components list all the statistics collected by New Relic for this item. Select the desired statistics.

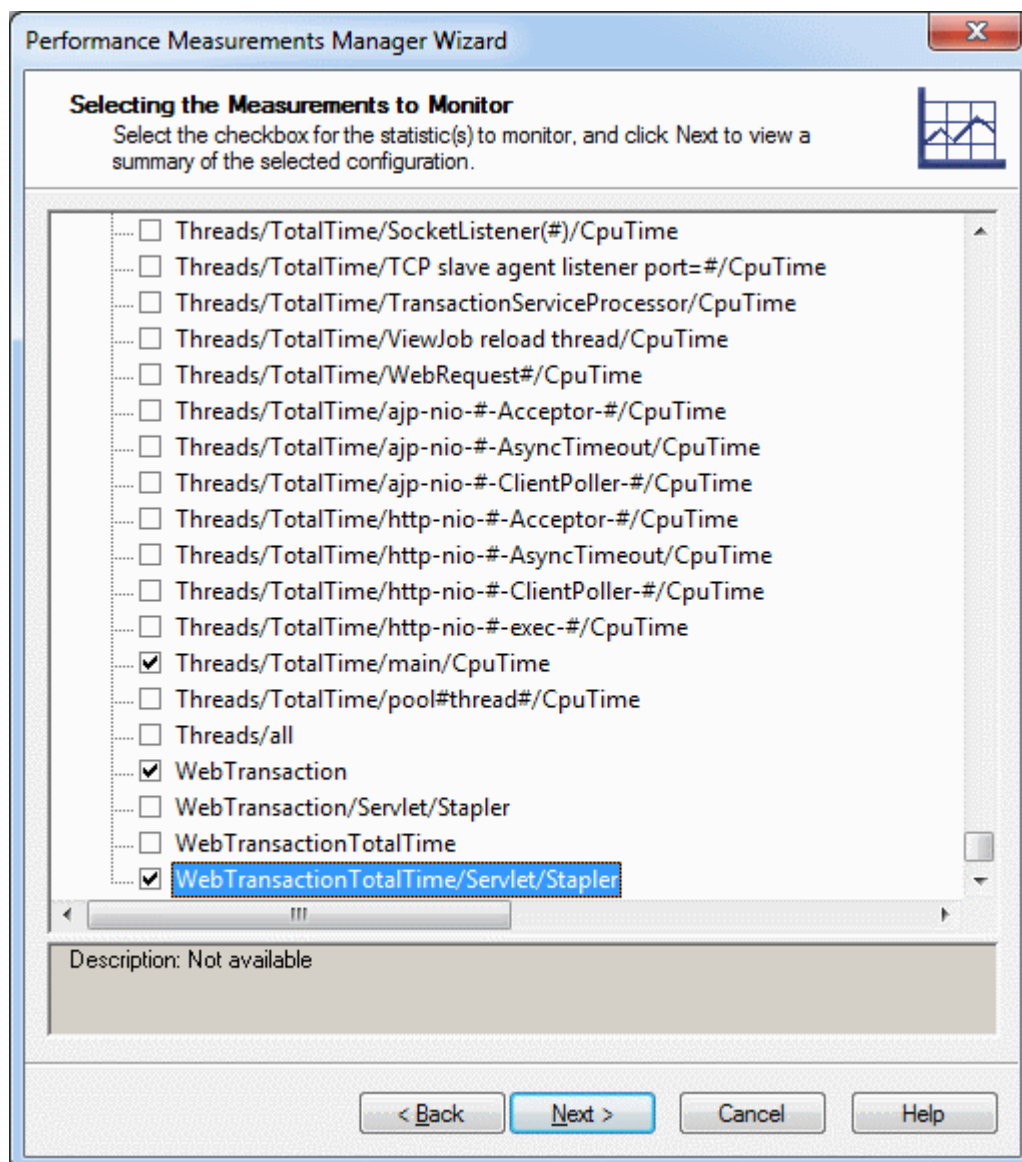


Figure 252: Selecting the Measurements to Monitor Screen

5. After all the desired items and statistics are selected, click **Next**. The Wizard displays a summary of the host, data source, and measurements configured for monitoring.
6. To accept the PMM configuration, click **Finish**.

The PMM Wizard closes and the selected configuration is added to the PMM main window.

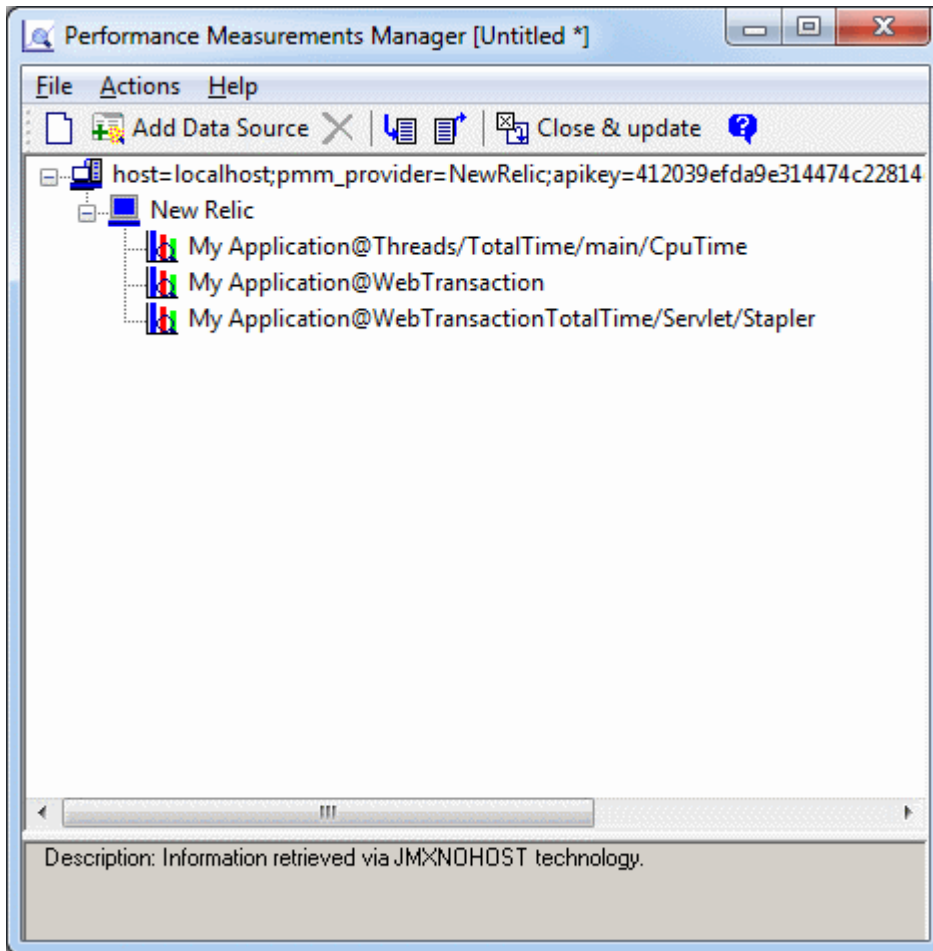


Figure 253: PMM Wizard Configuration Summary

Selecting the New Relic Statistics to Display in Reports

After integrating WebLOAD with New Relic, you can specify, while running a session, which New Relic statistics to view in the report view.

1. Open a report as described in *Opening Reports* (on page 298).
2. In the **PM@<New-Relic-host>** node, select the statistics you wish to display in the report.

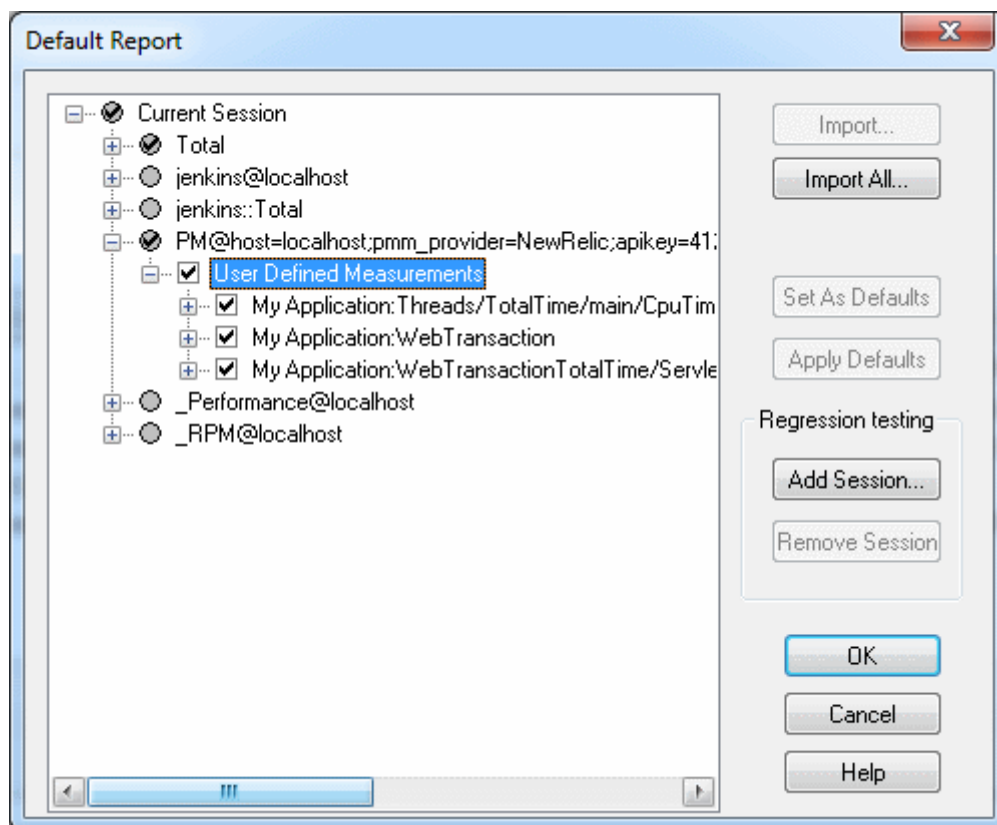


Figure 254: Selecting the Statistics to Display in a Report

The following figure shows a report in Report view, displaying the statistics collected by New Relic that were selected to appear in the report.

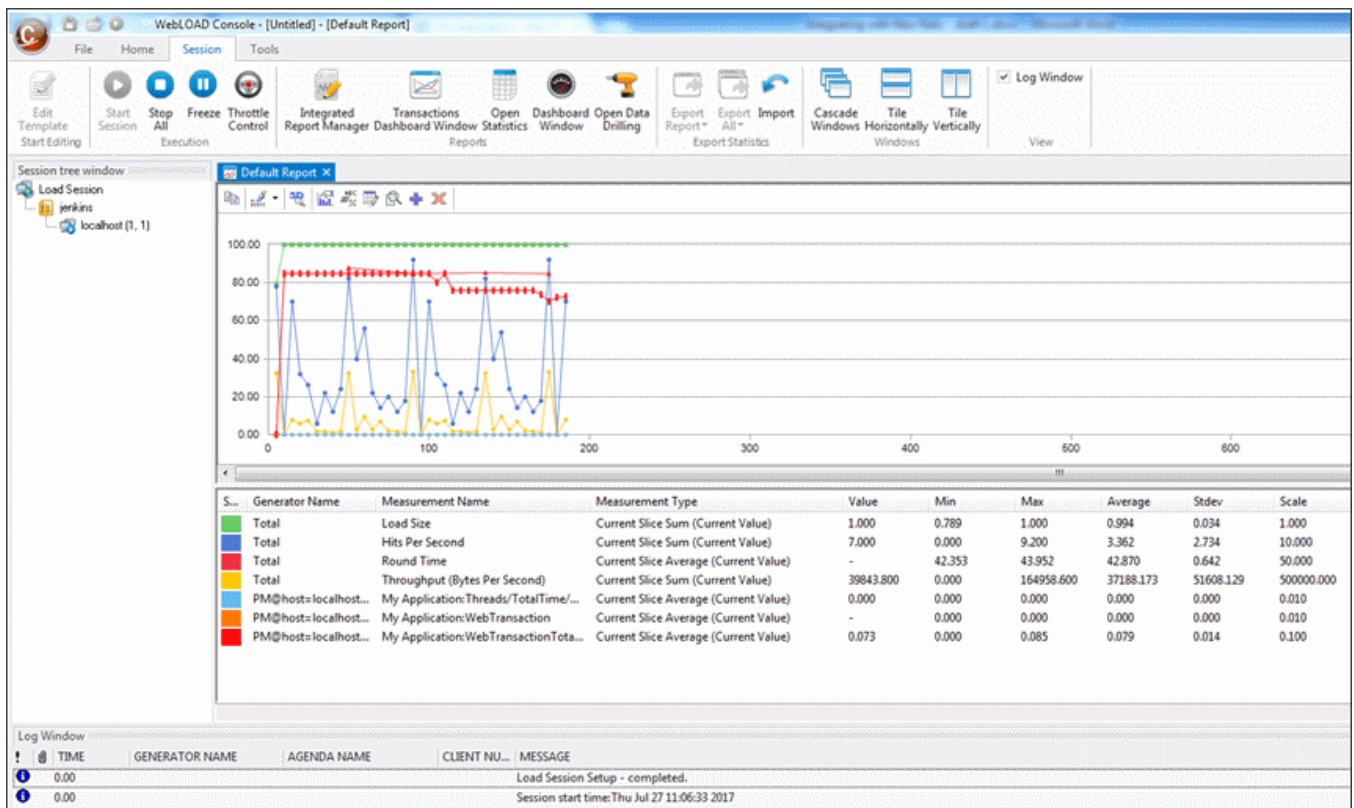


Figure 255: Viewing New Relic Statistics in Report View

WebLOAD Console Error Messages

This section describes the errors and log messages that can be generated by WebLOAD Console, as well as the JavaScript compiler and any human errors or user messages detected by WebLOAD Console at run-time. Log messages can also be added to the script by the user. The log messages are displayed in the Log Window.

Message Severity

Messages fall into four categories:

- Success / Info Message
- Minor Error
- Error
- Severe Error

Success / Info Message

Success / Info Messages are added to the script by the user. Success / Info Messages can be used in order to facilitate debugging. They do not stop WebLOAD Console from running.

Minor Error

Minor Errors do not stop WebLOAD Console from running. However, they do indicate suspicious conditions. The source of the problem may be generated by the SUT, the connection to the server, networking etc.

Error

Errors do not stop WebLOAD Console from running. However, they do stop the round and restart the script for the Virtual Client that encountered the error.

Severe Error

Severe Errors cause the Load Generator, on which the error was generated to stop immediately. A dialog box will be displayed.

WebLOAD Console Error Messages

Connection errors are usually due to an error in the connection definition. For example, an incorrect name or address specification, or an incorrect proxy designation can cause an error. Most such specification errors are the result of errors in data entry. Verify your configuration settings and addresses with your system administrator. Careful attention to detail will prevent most errors from ever occurring.

Errors are listed in alphabetical order.

Table 59: Errors

Error	Explanation	Severity
Analysis error	Data analysis error.	Error
Cannot resolve DNS entry	DNS address not found. Verify the URL in the JavaScript file.	Severe Error
Conditional GET error (HTTP code 304)	Document has not been modified, as expected by a conditional GET statement.	Minor Error
Conflict error (HTTP code 409)	Request could not be completed due to resource conflict. Check what other programs are currently running on your system.	Error
Connection failed	Connection failed for any reason other than the reasons specified in this table.	Severe Error
Connection refused	Connection refused by the server. Unable to access the server.	Minor Error
Connection timeout	Connection has timed out. Reason for failure not specified. If this connection is essential for WebLOAD Console to continue, this error will be considered a Severe Error and WebLOAD Console will stop. If the rest of the test session is able to continue, this error will be a Minor Error only.	Error / Severe Error
Data retrieval error	Data retrieval error. Cannot retrieve data from the server.	Severe Error

Error	Explanation	Severity
DeleteCookie error-Parameters	Parameters missing from DeleteCookie function.	Error
DeleteCookie error-not found	Error deleting cookie. Cookie not found.	Error
Error while reading	Error reading data from the server.	Error
Forbidden, reason not specified (HTTP code 403)	The specified URL address or server request is forbidden.	Error
Gateway error (HTTP code 502)	Invalid gateway response to WebLOAD Console request.	Error
Gateway timeout (HTTP code 504)	Gateway did not respond within specified timeout period. Reason for failure not specified.	Error
Illegal command	Illegal JavaScript command.	Severe Error
Illegal request (HTTP code 400)	Illegal server request.	Error
Internal server error (HTTP code 500)	The server went down.	Error
Length required (HTTP code 411)	More length is required for the current request than is presently available.	Severe Error
Load Generator disconnected	The Load Generator was disconnected from the Console. This usually indicates that the Load Generator stopped operating due to an unexpected problem.	Severe Error
Method not allowed (HTTP code 405)	Specified access or request method is not allowed.	Severe Error
No buffer space	No buffer space available. Unable to access the server.	Error
Not Acceptable (HTTP code 406)	Specified configuration, object description, parameters, authentication fields, or other submitted information is not acceptable.	Severe Error
Not implemented (HTTP code 501)	Selected feature is not currently implemented.	Severe Error
Open connection failed	Open connection has failed either due to an unspecified timeout or because the server refused the open connection.	Error
Other	Failure due to any other reason not specified in this list.	Severe Error

Error	Explanation	Severity
Precondition failed (HTTP code 412)	The current request failed due to failure in a necessary precondition to this request.	Severe Error
Proxy authentication required (HTTP code 407)	Access to the requested proxy must be authenticated.	Error
Proxy required (HTTP code 305)	You must use a proxy for the current test session.	Minor Error
Redirection error	New location not found.	Severe Error
Request entity too large (HTTP code 413)	Requested server object is too large to transfer successfully.	Error
Request timeout (HTTP code 408)	Request was not completed within specified timeout period. Reason for failure not specified. If this request is essential for WebLOAD Console to continue, this error will be considered a Severe Error and WebLOAD Console will stop. If the rest of the test session is able to continue, this error will be of Error level only.	Error / Severe Error
Request URI too long (HTTP code 414)	URI of requested object is too long to handle.	Error
Resource not available (HTTP code 410)	The requested resource is not currently available from the server.	Severe Error
Send failed	Request was not sent successfully to the server.	Error
Service temporarily unavailable (HTTP code 503)	The specified service is temporarily not available.	Error
SetCookie error-Parameters	Parameters missing from SetCookie function.	Error
SetCookie error	SetCookie failure for unspecified reason.	Error
Socket failed	You are unable to reach this URL. Stop the test session, and restart.	Severe Error
Socket not bound	Socket not bound to an address. Unable to access the server.	Error
Unauthorized request (HTTP code 401)	Request requires authorization and user authentication. For example, by specifying a username and password.	Error

Error	Explanation	Severity
Unsupported media type (HTTP code 415)	The specified media format is not supported by WebLOAD Console at this time (for example, rewriteable CD's).	Error
Unsupported protocol	The specified protocol is not supported by WebLOAD Console at this time (for example, ftp://...).	Severe Error
Unsupported version of HTTP (HTTP code 505)	The specified version of HTTP is not supported by WebLOAD Console at this time.	Error
URL not found (HTTP code 404)	URL address as specified is not found. Verify the URL in the JavaScript file.	Warning
URL temporarily not correct (HTTP code 302)	Requested resource has been temporarily moved to a different URL.	Minor Error



Reference Materials

WebLOAD Console Files

The following is an explanation of the files associated with a WebLOAD Console Load Session.

Table 60: WebLOAD Console Load Session Files

WebLOAD Extension	WebLOAD File Type
WebLOAD.INI	Configuration variables
MONITOR.INI	INI settings for the Console
HOSTS.LST	List of local hosts sites read from your system's LOCAL.HOSTS file
.JS FILES	Script files for all Clients
.TPL FILES	Load Template files
.LS FILES	Load Session files
.DAT FILES	Load Session data files
.ISD FILES	Index statistics database files
.SDB FILES	Statistics database files
.MIX FILES	Script Mix files
.MDB FILES	Data Drilling files
WebLOAD.TAR	UNIX installation files
.MDB Files	Saved Log Window files
.WLP Files	WebLOAD Recorder Project files
.WLS Files	WebLOAD Recorder Session files
.WLA Files	WebLOAD Recorder Actual Repository files
.WLE Files	WebLOAD Recorder Expected Repository files


Appendix G

Glossary

Glossary Term	Description
AAT	An older, obsolete WebLOAD utility that was used for recording web session activities as a JavaScript file. It is replaced by WebLOAD Recorder.
Aborted Rounds	The number of times the Virtual Clients started to run a script but did not complete the script, during the last reporting interval. This might be due to a session being stopped either automatically or manually by the user.
Script	Specification of the sequence of HTTP protocol calls sent by Virtual Clients to the SUT (System Under Test). Scripts are written in JavaScript. You can either write scripts as a text file or generate them automatically using the WebLOAD Recorder.
Application Being Tested (ABT)	See <i>SUT</i> .
Attempted Connections	The total number of times the Virtual Clients attempted to connect to the SUT during the last reporting interval.
Automatic Transaction counters	If you have Automatic Transactions enabled, WebLOAD creates three counters for each GET and POST statement in the script: <ul style="list-style-type: none"> The total number of times it occurred The number of times it succeeded The number of times it failed during the last reporting interval.
Average	For timers, average is the total amount of time counted by the timer (not the elapsed time) divided by the Count (that is, the total number of readings). For example, the average for Transaction Time is the amount of time it took to complete all the successful transactions, divided by the number of successful transactions (the Count).
Built-in Timer	A timer measures the time required to perform a given task. WebLOAD supports both programmed timers and built-in timers. ROUND TIME is a built-in timer. The ROUND TIME is the time needed for one complete execution of a script.

Glossary Term	Description
Connect Time	<p>The time it takes for a Virtual Client to connect to the System Under Test (the SUT), in seconds. In other words, the time it takes from the beginning of the HTTP request to the TCP/IP connection.</p> <p>The value posted in the Current Value column is the average time it took a Virtual Client to connect to the SUT during the last reporting interval.</p> <p>If the Persistent Connection option is enabled, there may not be a value for Connect Time because the HTTP connection remains open between successive HTTP requests.</p>
Connection Speed (Bits Per Second)	<p>The number of bits transmitted back and forth between the Virtual Clients and the System Under Test (SUT), divided by the time it took to transmit those bits, in seconds.</p> <p>You can set the Virtual Clients to emulate a particular connection speed during the test, either by using the Variable Connection Speed settings, or by coding the connection speed in the script.</p> <p>If a connection speed is specified for the test, WebLOAD reports it in the Statistics Report.</p> <p>The value posted in the Current Value column is the number (sum) of bits passed per second during the last reporting interval. It should match, very closely, the connection speed you specified for the test.</p>
Console	<p>The WebLOAD component that manages the test session.</p> <p>The Console performs the following:</p> <ul style="list-style-type: none"> • Configures Load Session hosts and scripts • Schedules Load Session scripts • Configures Goal-Oriented test sessions • Monitors the application's performance under the generated load • Manages the Load Session as it is running, allowing you to pause, stop, and continue Load Session components as needed • Displays the current performance of the SUT • Provides a final performance reports for Probing Clients and Virtual Clients • Manages exporting of performance reports

Glossary Term	Description
Count	(For timers only.) The total number of readings (the number of times the item being timed has occurred) for the timed statistic since the beginning of the test. For example, for Transaction Time, Count shows the number of transactions that have been completed.
Current Slice	The value posted for this reporting interval in the Statistics Report main window.
Current Slice Average	<p>For per time unit statistics and counters, average is the total of all of the current values for the last reporting interval, divided by the number of readings.</p> <p>For timers, average is the total amount of time counted by the timer (not the elapsed time), divided by the Count (that is, the total number of readings for the last reporting interval). For example, the average for Transaction Time is the amount of time it took to complete all the successful transactions in the last reporting interval, divided by the number of successful transactions (the Current Slice Count).</p>
Current Slice Count	(For timers only.) The total number of readings (the number of times the item being timed has occurred) for the timed statistic for the last reporting interval. For example, for Transaction Time, Current Slice Count shows the number of transactions that have been completed over the last reporting interval.
Current Slice Max	The highest value reported for this statistic over the last reporting interval.
Current Slice Min	The lowest value reported for this statistic over the last reporting interval.
Current Slice Standard Deviation	The average amount the measurement for this statistic varies from the average over the last reporting interval.
Current Slice Sum	The aggregate or total value for this statistic in this script over the last reporting interval.
DNS Lookup Time	The time it takes to resolve the host name and convert it to an IP address by calling the DNS server.
Failed Connections	<p>The total number of times the Virtual Clients tried to connect to the SUT but were unsuccessful, during the last reporting interval.</p> <p>This number is always less than or equal to the number of failed hits because hits can fail for reasons other than a failed connection.</p>

Glossary Term	Description
Failed Hits	The total number of times the Virtual Clients made an HTTP request but did not receive the correct HTTP response from the SUT during the last reporting interval. Note that each request for each gif, jpeg, html file, etc., is a single hit.
Failed Hits Per Second	<p>The number of times the Virtual Clients did not obtain the correct HTTP response, divided by the elapsed time, in seconds.</p> <p>The value posted in the Current Value column is the number (sum) of unsuccessful HTTP requests per second during the last reporting interval.</p>
Failed Pages Per Second	<p>The number of times the Virtual Clients did not obtain the correct response to an upper level request, divided by the elapsed time, in seconds.</p> <p>The value posted in the Current Value column is the number (sum) of unsuccessful requests per second during the last reporting interval.</p>
Failed Rounds	The total number of times the Virtual Clients started but did not complete the script during the last reporting interval.
Failed Rounds Per Second	The number of times the Virtual Clients started but did not complete an iteration of the script, divided by the elapsed time, in seconds. The value posted in the Current Value column is the number (sum) of failed iterations of the script per second during the last reporting interval.
First Byte	The time it takes a Virtual Client to receive the first byte of data.
Gallery	See <i>Templates Gallery</i> .
Goal-Oriented Test	<p>A WebLOAD component enabling you to define the performance goals required, and view the status of your application when it is operating under this performance goal. WebLOAD provides a Goal-Oriented Test Wizard for configuring these performance goals. WebLOAD automatically accelerates the number of Virtual Clients accessing your website until you meet your performance goal.</p> <div style="display: flex; align-items: center;">  <p>Note: The Goal-Oriented Test Wizard was previously called the Cruise Control Wizard.</p> </div>
Goal-Oriented Test Wizard	See <i>Goal-Oriented Test</i> .

Glossary Term	Description
Hit Time	<p>The time it takes to complete a successful HTTP request, in seconds. Each request for each gif, jpeg, html file, etc., is a single hit. The time of a hit is the sum of the Connect Time, Send Time, Response Time, and Process Time.</p> <p>The value posted in the Current Value column is the average time it took to make an HTTP request and process its response during the last reporting interval.</p>
Hits	<p>The total number of times the Virtual Clients made HTTP requests to the System Under Test (SUT) during the last reporting interval.</p> <p>For example, a Get statement for a URL retrieves a page. The page can include any number of graphics and contents files. Each request for each gif, jpeg, html file, etc., is a single hit.</p>
Hits Per Second	<p>The number of times the Virtual Clients made an HTTP request, divided by the elapsed time, in seconds. Each request for each gif, jpeg, html file, etc. is a single hit.</p> <p>The value posted in the Current Value column is the number (sum) of HTTP requests per second during the last reporting interval.</p>
Host	<p>A computer connected via a network, participating in a test session. Each Host in a test session has assigned tasks. A host can act as either a Load Machine or a Probing Client Machine. All hosts participating in a test session must be accessible to the Console over a network. Therefore they must run TestTalk, the network agent.</p>
HTTP Response Status	<p>WebLOAD creates a row in the Statistics Report for each kind of HTTP status code it receives as an HTTP response from the SUT (redirection codes, success codes, server error codes, or client error codes).</p> <p>The value posted is the number of times the Virtual Clients received that status code during the last reporting interval.</p>
Integrated Report	<p>A single configurable report that can integrate both standard performance data, and data from the NT Performance Monitor. This report gives you a more complete picture of the performance of your application. The data to be monitored and the data to be displayed in the report are both configurable in the Console.</p>
Internet Productivity Pack (IPP)	<p>Provides a set of protocol implementations enabling you to load-test your application using these protocols.</p>

Glossary Term	Description
Java and ActiveX counters	<p>You can add function calls to your scripts that enable you to instantiate and call methods and properties in Java and ActiveX components (see the <i>WebLOAD Scripting Guide</i>). If there are ActiveX or Java function calls in the script that you are running, WebLOAD reports three counters for them in the Statistics Report:</p> <ul style="list-style-type: none"> • The total number of times it occurred • The number of times it succeeded • The number of times it failed during the last reporting interval. <p>The row heading in the Statistics Report is the name of the function call.</p>
Java and ActiveX timers	<p>You can add function calls to your scripts that enable you to instantiate and call methods and properties in Java and ActiveX components (see the <i>WebLOAD Scripting Guide</i>). If there are ActiveX or Java function calls in the script you are running, WebLOAD reports timers for them in the Statistics Report.</p> <p>The timer value is the average amount of time it took to complete the function call, in seconds, during the last reporting interval.</p> <p>The row heading in the Statistics Report is the name of the function call.</p>
Load Generator	<p>The component of the Load Machine that generates Virtual Clients. Load Generators have the task of bombarding the System Under Test with HTTP protocol call requests as defined in the Script. WebLOAD assesses the application's performance by measuring the response time experienced by the Virtual Clients. The number of Virtual Clients at any given moment is determined by the user.</p>
Load Generator Machine	<p>See <i>Load Machine</i>.</p>
Load Machine	<p>A host that runs Load Generators. Load Generators bombard the application under test with a large load, to enable complete scalability and integrity testing.</p>

Glossary Term	Description
Load Session	<p>A Load Session includes both the complete Load Template and the results obtained while running that Load Session. A Load Template consists of information about the hosts and scripts participating in the current Load Session. The Load Template will also include scheduling information. The complete Load Template is illustrated in the Session Tree. Storing a Load Template saves you time when repeatedly running WebLOAD with the same, or even a similar network configuration, since you do not have to recreate your Load Template from scratch each time you want to start working. Storing Load Session results can be useful when you want to examine results from multiple test sessions or for analyzing test session results.</p>
Load Size	<p>The number of Virtual Clients running during the last reporting interval.</p>
Load Template	<p>A Load Template contains the complete Load Session definition, without the test results. A Load Template includes information about the participating hosts and the scripts used in the current Load Session. The definition also includes scheduling information and the configuration of the Server Monitor and Integrated Reports. The complete Load Template is illustrated in the Session Tree. Storing a Load Template saves you time when repeatedly running WebLOAD with the same, or even a similar network configuration, since you do not have to recreate your Load Template from scratch each time you rerun a test.</p>
Page Time	<p>The time it takes to complete a successful upper level request, in seconds. The Page Time is the sum of the Connection Time, Send Time, Response Time, and Process Time for all the hits on a page.</p> <p>The value posted in the Current Value column is the average time it took the Virtual Clients to make an upper level request and process its response during the last reporting interval.</p>
Pages	<p>The total number of times the Virtual Client made upper level requests, both successful and unsuccessful, during the last reporting interval.</p>
Pages Per Second	<p>The number of times the Virtual Clients made upper level requests divided by the elapsed time, in seconds.</p> <p>The value posted in the Current Value column is the number (sum) of requests per second during the last reporting interval.</p>

Glossary Term	Description
Per Time Unit statistics	Ratios that calculate an average value for an action or process. For example: Transactions Per Second, Rounds Per Second.
Portfolio	A Portfolio of reports enables you to generate a single, inclusive report that contains all the charts generated by the templates included in the portfolio.
Probing Client	A software program which "bombards" the SUT as a single Virtual Client, to further measure the performance of the SUT. WebLOAD generates exact values for Probing Client performance.
Probing Client Machines	Hosts running Probing Client software simulating one Virtual Client, and running at the same time as Load Machines.
Probing Client software	See <i>Probing Client</i> .
Process Time	<p>The time it takes WebLOAD to parse an HTTP response from the SUT and then populate the document-object model (DOM), in seconds.</p> <p>The value posted in the Current Value column is the average time it took WebLOAD to parse an HTTP response during the last reporting interval.</p>
Receive Time	The elapsed time between receiving the first byte and the last byte.
Report Portfolio	See <i>Portfolio</i> .
Resource Manager	<p>Distributes and circulates WebLOAD testing resources (Virtual Clients and Probing Clients) amongst users on a "need to use" basis. The Resource Manager is packaged with a maximum number of Virtual Clients, Probing Clients and Connected Workstation ports, as defined by the WebLOAD package.</p> <p>With the Resource Manager, every WebLOAD Console can operate in Standalone Workstation mode or Connected Workstation mode.</p>
Response Data Size	<p>The size, in bytes, of all the HTTP responses sent by the SUT during the last reporting interval.</p> <p>WebLOAD uses this value to calculate Throughput (bytes per second).</p>

Glossary Term	Description
Response Time	<p>The time it takes the SUT to send the object of an HTTP request back to a Virtual Client, in seconds. In other words, the time from the end of the HTTP request until the Virtual Client has received the complete item it requested.</p> <p>The value posted in the Current Value column is the average time it took the SUT to respond to an HTTP request during the last reporting interval.</p>
Responses	<p>The number of times the SUT responded to an HTTP request during the last reporting interval.</p> <p>This number should match the number of successful hits.</p>
Round Time	<p>The time it takes one Virtual Client to finish one complete iteration of a script, in seconds.</p> <p>The value posted in the Current Value column is the average time it took the Virtual Clients to finish one complete iteration of the script during the last reporting interval.</p>
Rounds	<p>The total number of times the Virtual Clients attempted to run the script during the last reporting interval.</p>
Rounds Per Second	<p>The number of times the Virtual Clients attempted to run the script, divided by the elapsed time, in seconds.</p> <p>The value posted in the Current Value column is the number (sum) of attempts (both successful and unsuccessful) per second during the last reporting interval.</p>
Send Time	<p>The time it takes the Virtual Client to write an HTTP request to the SUT, in seconds.</p> <p>The value posted in the Current Value column is the average time it took the Virtual Clients to write a request to the SUT during the last reporting interval.</p>
Server Performance Measurements	<p>If you selected Performance Monitor statistics for the report, WebLOAD creates a row for them and reports their values in the Statistics Report.</p> <p>For definitions of the statistics, see the Server Monitor Definition dialog box.</p> <p>Be selective when choosing server performance measurements , otherwise the system resources required to manage the data might affect the Console.</p>

Glossary Term	Description
Session Tree	A graphic representation of a Load Template and status. It illustrates the different components of a test session, including Load Machines and Probing Clients, the scripts that they execute, and their status.
Single Client	See <i>Probing Client</i> .
Standard Deviation	The average amount the measurement varies from the average since the beginning of the test.
Successful Connections	<p>The total number of times the Virtual Clients were able to successfully connect to the SUT during the last reporting interval.</p> <p>This number is always less than or equal to the number of successful hits because several hits might use the same HTTP connection if the Persistent Connection option is enabled.</p>
Successful Hits	The total number of times the Virtual Clients made an HTTP request and received the correct HTTP response from the SUT during the last reporting interval. Each request for each <code>gif</code> , <code>jpeg</code> , <code>html</code> file, etc., is a single hit.
Successful Hits Per Second	<p>The number of times the Virtual Clients obtained the correct HTTP response to their HTTP requests divided by the elapsed time, in seconds.</p> <p>The value posted in the Current Value column is the number (sum) of successful HTTP requests per second during the last reporting interval.</p>
Successful Pages Per Second	The value posted in the Current Value column is the number (sum) of successful requests per second during the last reporting interval.
Successful Rounds	The total number of times the Virtual Clients completed one iteration of the script during the last reporting interval.
Successful Rounds Per Second	<p>The number of times the Virtual Clients completed an entire iteration of the script, divided by the elapsed time, in seconds.</p> <p>The value posted in the Current Value column is the number (sum) of successful iterations of the script per second during the last reporting interval.</p>
SUT	The system running the Web application currently under test. The SUT (System Under Test) is accessed by clients through its URL address. The SUT can reside on any machine or on multiple machines, anywhere on the global Internet or your local intranet.
Template	See <i>Load Template</i> .

Glossary Term	Description
Templates Gallery	The Templates Gallery is a single entity that contains predefined templates, user-defined templates, and portfolios.
Test Program	See <i>Test Script</i> .
Test Script	The script. This defines the test scenario used in your Load Session. Scripts are written in JavaScript.
Test Template	See <i>Load Template</i> .
TestTalk	The network agent. This program enables communication between the Console and the host computers participating in the test.
Throttle Control	A WebLOAD component that enables you to dynamically change the Load Size while a test session is in progress.
Throughput (Bytes Per Second)	The average number of bytes per second transmitted from the SUT to the Virtual Clients running the script during the last reporting interval. In other words, this is the amount of the Response Data Size, divided by the number of seconds in the reporting interval.
Time to First Byte	The time that elapsed since a request was sent until the Virtual Client received the first byte of data.
User-defined Automatic Data Collection	<p>If you have Automatic Data Collection enabled, WebLOAD creates three counters for each GET and POST statement in the script:</p> <ul style="list-style-type: none"> • The total number of times the Get and Post statements occurred • The number of times the statements succeeded • The number of times the statements failed during the last reporting interval.
User-defined counters	<p>Your own counters that you can add to scripts using the <code>SendCounter ()</code> and the <code>SendMeasurement ()</code> functions (see the <i>WebLOAD Scripting Guide</i>). If there is a user-defined counter in the script that you are running, WebLOAD reports the counter's values in the Statistics Report.</p> <p>The row heading is the name (argument) of the counter. That is, the row heading is the string in parenthesis in the <code>SendCounter ()</code> or <code>SendMeasurement ()</code> function call.</p> <p>The value reported is the number of times the counter was incremented during the last reporting interval.</p>

Glossary Term	Description
User-defined timer	<p>Timers that you can add to scripts to keep track of the amount of time it takes to complete specific actions (see the <i>WebLOAD Scripting Guide</i>). If there are any timers in the scripts that you are running, WebLOAD reports their values in the Statistics Report.</p> <p>The row heading is the name (argument) of the timer. That is, the row heading is the string in parenthesis in the <code>SetTimer()</code> function call. The timer represents the time it takes to complete all the actions between the <code>SetTimer()</code> call and its corresponding <code>SendTimer()</code> call, in seconds.</p> <p>The value posted is the average time it took a Virtual Client to complete the actions between the pair of timer calls, in seconds, during the last reporting interval.</p>
User-defined Transaction counters	<p>Transaction functions that you can add to scripts for functional tests (see the <i>WebLOAD Scripting Guide</i>). If there is a user-defined transaction function in the script that you are running, WebLOAD reports three counters for it in the Statistics Report:</p> <ul style="list-style-type: none"> • The total number of times the transaction occurred • The number of times a transaction succeeded • The number of times a transaction failed during the last reporting interval. <p>The row heading is the name (argument) of the transaction. That is, the row heading is the string in parenthesis in the <code>BeginTransaction()</code> function call.</p>
User-defined Transactions timers	<p>A timer for user-defined transaction functions. If there is a user-defined transaction function in the script that you are running, WebLOAD reports a timer for it in the Statistics Report.</p> <p>The row heading is the name (argument) of the user-defined transaction. That is, the row heading is the string in parenthesis in the <code>BeginTransaction()</code> function call.</p> <p>The timer represents the average time it took to complete all the actions between the <code>BeginTransaction()</code> call and its corresponding <code>EndTransaction()</code> call, in seconds, during the last reporting interval.</p>

Glossary Term	Description
Virtual Client	Artificial entities run by Load Generators. Each such entity is a perfect simulation of a real client accessing the System Under Test (SUT) through a Web browser. Virtual Clients generate HTTP calls that access the SUT. The Load Generators that run Virtual Clients can reside anywhere on the Internet or on your local intranet. Scripts are executed by all the Virtual Clients in parallel, achieving simultaneous access to the SUT. The size of the load on your SUT is determined by the number of Virtual Clients being generated. You may define as many Virtual Clients as needed, up to the maximum supported by your WebLOAD “package.”
WebLOAD Analytics	WebLOAD Analytics enables you to analyze data, and create custom, informative reports after running a WebLOAD test session.
WebLOAD Console	See <i>Console</i> .
WebLOAD Recorder	An easy-to-use tool for recording, creating, and authoring protocol scripts for the WebLOAD environment.
WebLOAD Load Template	See <i>Load Template</i> .
WebLOAD Session	See <i>Load Session</i> .
WebLOAD Wizard	A WebLOAD Wizard that steps you through the configuration process. Each screen of the WebLOAD Wizard contains text explaining the configuration process. The WebLOAD Wizard enables you to create a basic Load Template. After using the demo, you can use the Console menus to add functionality not available through the WebLOAD Wizard.
WebRM	See <i>Resource Manager</i> .

Index

-

- .dat ▪ 96
- .fta ▪ 449
- .fte ▪ 449
- .fts ▪ 449
- .ls ▪ 96
 - open ▪ 98
 - save ▪ 96
- .mdb ▪ 267, 449
- .mix ▪ 105
- .pft ▪ 449
- .tpl ▪ 91, 272
 - open ▪ 92

A

- AAT ▪ 451
- Aborted Rounds ▪ 451
 - definition ▪ 320
- About button ▪ 76
- About Button ▪ 76
- About Script Options ▪ 211
- About the Goal Definition Dialog Box ▪ 151
- About the Performance Measurements
 - Manager ▪ 363
- Acknowledgments from a ping ▪ 26
- Actions Menu ▪ 367
 - PMM ▪ 367
- Activating Throttle Control ▪ 282
- Actual Repository Files
 - .fta ▪ 449
- Adding a Browser Version ▪ 223
- Adding a User-Defined Timer ▪ 305
- Adding Data Sources ▪ 375
- Adding Data Sources to a Configured Host ▪ 384
- Adding Host Computers ▪ 111
- Adding Host Computers to the Host
 - Computers List ▪ 147
- Adding Measurements to a Configured Data
 - Source ▪ 383
- Adding Performance Goals ▪ 153
- Additional Information
 - created by ▪ 93, 97

- custom ▪ 93, 97
- Load Session ▪ 96
- Load Template ▪ 93
- Load Template fields ▪ 93
- results description ▪ 97
- session comments ▪ 97
- set ▪ 93, 97
- subject ▪ 93, 97
- system under test ▪ 93, 97
- template comments ▪ 93, 97
- test description ▪ 93, 97
- title ▪ 93, 97

- After Completing the WebLOAD Wizard ▪ 132

- Analyzing Test Results ▪ 39

- Apache 2 Web Server Using SNMP ▪ 399

- Apache Web Server Using SNMP ▪ 399

- AppDynamics, integrating with ▪ 409

- configuring AppDynamics ▪ 410
- configuring AppDynamics Lite ▪ 410
- configuring AppDynamics Pro ▪ 412
- configuring WebLOAD ▪ 409
- viewing WebLOAD in AppDynamics ▪ 417

- Append

- configuration files ▪ 372
- PMM configuration files ▪ 372

- Application Being Tested (ABT) ▪ 451

- Application Server

- resource monitor ▪ 363

- Application Server Monitors ▪ 387

- Application Servers

- BEA WebLogic ▪ 375
- IBM WebSphere ▪ 375
- Microsoft Active Server Pages ▪ 375
- Microsoft ASP .Net ▪ 375
- MySAP.com ▪ 375

- Assigning a PMM Configuration to a
 - WebLOAD Console Template ▪ 373

- Attempted Connections ▪ 451

- definition ▪ 320

- Authentication

- set ▪ 232

- Authentication for Protocol Scripts

- Script options ▪ 232
- session options ▪ 232

- Authentication for Protocol Scripts Tab ▪ 232

- Authentication Settings

- configure ▪ 232

- Automatic Data Collection

- enable ▪ 246

- HTTP level ▪ 246
 - object level ▪ 246
 - pages ▪ 246
 - Script option ▪ 246
 - session option ▪ 246
 - Automatic Transactions
 - counters ▪ 451
 - data drilling ▪ 246
 - Autosave File Name
 - setting ▪ 179
 - Average ▪ 330, 451
- B**
- Bandwidth
 - configure ▪ 234
 - Script options ▪ 234
 - session options ▪ 234
 - set ▪ 234
 - set for load generator ▪ 234
 - set for probing client ▪ 234
 - Bandwidth Settings ▪ 234
 - BEA WebLogic
 - data source ▪ 375
 - Browser Emulation
 - add version ▪ 223
 - enable keep-alive ▪ 226
 - enable persistent connection ▪ 226
 - options ▪ 221
 - script options ▪ 221
 - session options ▪ 221
 - set browser type ▪ 223
 - set HTTP version ▪ 229
 - set load generator max connections ▪ 224
 - set probing client max connections ▪ 224
 - set redirection limits ▪ 225
 - set version ▪ 223
 - simulate DNS cache ▪ 225
 - simulate SSL cache ▪ 226
 - Browser Emulation Settings
 - configure ▪ 221
 - Browser Type
 - set ▪ 223
 - Browser version
 - add ▪ 223
 - Built-in Timer ▪ 452
- C**
- Case Inspections
 - syntax check ▪ 242
 - Changing the Length of the Reporting Interval ▪ 318
 - Changing the Reports Parameters ▪ 305
 - Changing the Scale of a Report for a Selected Measurement ▪ 312
 - Clearing the Log Window ▪ 293
 - Client Type Feature Summary ▪ 11
 - Cloud Accounts
 - add host computers from cloud accounts ▪ 111
 - create an Amazon EC2 account ▪ 108
 - create WebLOAD cloud accounts ▪ 109
 - Cloud machines
 - add as host computer ▪ 111
 - set global shut-down policy ▪ 110
 - shut down manually ▪ 280
 - Command Line
 - example ▪ 274
 - launch WebLOAD Console ▪ 272
 - parameters ▪ 273
 - syntax ▪ 273
 - Commit - Git action ▪ 285
 - Commit dir - Git action ▪ 286
 - Communications
 - errors ▪ 275
 - establishing ▪ 275
 - pinging a host ▪ 26
 - testing connections ▪ 26
 - troubleshooting ▪ 27
 - Completing the WebLOAD Wizard ▪ 131
 - Completing Your Manual Load Template Configuration ▪ 175
 - Components ▪ 45
 - Console System button ▪ 45
 - Quick access toolbar button ▪ 47
 - results window ▪ 60
 - ribbon ▪ 47
 - right-click menus ▪ 48
 - session tree ▪ 48
 - status bar ▪ 48
 - title bar ▪ 45
 - CONFIG.encap (Peer Encapsulator File) ▪ 397
 - CONFIG.Master (Peer Master Agent File) ▪ 397
 - Configuration Example ▪ 20
 - Configuration Files
 - append ▪ 372
 - attach to WebLOAD Console templates ▪ 371, 373
 - create new ▪ 371
 - export ▪ 373

- import ▪ 372
- new ▪ 371
- open ▪ 372
- replace ▪ 372
- save ▪ 371, 373, 374
- use ▪ 371
- Configuration Pane ▪ 368
 - PMM ▪ 368
- Configuration Tree
 - PMM ▪ 368
- Configuration Tree Icons ▪ 369
- Configuration Tree Structure ▪ 369
 - PMM ▪ 368
- Configuring a Goal - An Example ▪ 156
- Configuring a Load Template Manually ▪ 165
- Configuring Administrator Privileges for Perfmon (Windows) ▪ 385
- Configuring an Integrated Report ▪ 303
- Configuring an Integrated Report to Include Data from Previously Saved Load Sessions ▪ 309
- Configuring Browser Parameter Settings ▪ 221
- Configuring Java Options ▪ 252
- Configuring Load Template Using WebLOAD Wizard ▪ 34
- Configuring Reset Mechanism Settings ▪ 250
- Configuring the Data Sources, Hosts, and Measurements to Monitor ▪ 375
- Configuring the SNMP Master Agent ▪ 397
- Connect Time ▪ 452
 - definition ▪ 320
- Connected Workstation ▪ 87
- Connecting to WebRM ▪ 87
- Connection Speed
 - definition ▪ 320
- Connection Speed (Bits Per Second) ▪ 452
- Connection Tab ▪ 234
- Connections
 - required ▪ 25
 - testing ▪ 26
- Console ▪ 453
 - About button ▪ 76
 - components ▪ 45
 - configuration dialog boxes ▪ 165
 - console system button ▪ 45
 - dashboard ▪ 63
 - data drilling ▪ 65
 - definition ▪ 9
 - edit mode ▪ 45, 68
 - features ▪ 43
 - file tab ▪ 70
 - Help drop-down ▪ 76
 - home tab ▪ 71
 - Load Session diagram ▪ 9
 - log window ▪ 66
 - modes ▪ 45
 - open ▪ 44, 84
 - Quick access toolbar ▪ 47
 - results window ▪ 60
 - ribbon ▪ 47, 70
 - right-click menus ▪ 48
 - run mode ▪ 45, 69
 - session mode ▪ 277
 - session tab ▪ 72
 - session tree ▪ 48
 - Startup dialog box ▪ 67
 - status bar ▪ 48
 - title bar ▪ 45
 - tools tab ▪ 74
 - transaction dashboard ▪ 64
- Console Elements ▪ 45
- Console System button ▪ 45
- Context Menus ▪ 48
- Control Buttons ▪ 153
- Correcting the All Hosts are Unreachable or Stopped Error ▪ 275
- Correcting the Load Session Setup Not Completed Error ▪ 276
- Correcting the Load Session Setup Timed Out Error ▪ 276
- Count ▪ 330, 453
- Counter Statistics
 - current value calculation ▪ 319
 - defined ▪ 317
- Created By
 - additional information ▪ 93
- Creating
 - integrated reports ▪ 40
 - Script ▪ 29
- Creating a Load Schedule Manually ▪ 171
- Creating a Load Schedule with the Load Profiler ▪ 173
- Creating a Mix ▪ 95, 105, 142
- Creating a New Configuration File ▪ 371
- Creating a New Load Template ▪ 92
- Creating an Equal Load Schedule Manually ▪ 115

- Creating an Equal Load Schedule with the Load Profiler ▪ 116
 - Creating Cloud Accounts ▪ 109
 - Creating Load Templates ▪ 77, 91
 - Creating Load Templates with One or More Single Scripts ▪ 167
 - Creating Load Templates with the Goal-Oriented Test ▪ 135
 - Creating Load Templates with the WebLOAD Wizard ▪ 99
 - Creating the Test Scripts ▪ 82
 - Current Session
 - Script options ▪ 211
 - Current Session Options ▪ 256
 - functional testing ▪ 256
 - Current Slice ▪ 330, 453
 - average ▪ 453
 - count ▪ 453
 - max ▪ 453
 - min ▪ 454
 - standard deviation ▪ 454
 - sum ▪ 454
 - Current Value ▪ 330
 - calculation ▪ 319
 - defined ▪ 317
 - Current Value Computation ▪ 319
 - Custom
 - additional information ▪ 93
 - Customizing the Export File ▪ 360
- D**
- Dashboard ▪ 63, 295
 - calculated hits per day ▪ 295
 - calculated pages per day ▪ 295
 - calculated throughput per day ▪ 295
 - failed hits ▪ 295
 - hits per second ▪ 295
 - pages per second ▪ 295
 - rounds executed ▪ 295
 - rounds failed ▪ 295
 - running Virtual Clients ▪ 295
 - throughput per second ▪ 295
 - time ▪ 295
 - Data Drilling ▪ 65, 267, 343
 - automatic transactions ▪ 246
 - collect instances ▪ 189
 - instance grid ▪ 346
 - open report ▪ 344
 - transaction instance tree ▪ 347
 - transaction reason failure grid ▪ 345
 - transactions grid ▪ 344
 - Data Drilling Reports ▪ 65, 343
 - Data Sources
 - add ▪ 375
 - add measurements to ▪ 383
 - add to hosts ▪ 384
 - application servers ▪ 375
 - configure ▪ 375
 - databases ▪ 375
 - delete ▪ 384
 - other ▪ 375
 - PMM ▪ 369
 - PMM Wizard ▪ 375
 - system ▪ 375
 - Web servers ▪ 375
 - Database Monitors ▪ 395
 - Databases
 - Microsoft SQL Server 2000 ▪ 375
 - Microsoft SQL Server using SNMP ▪ 375
 - monitoring ▪ 363
 - Oracle ▪ 375
 - resource monitor ▪ 363
 - Deactivating Throttle Control ▪ 282
 - Default Options ▪ 256
 - functional testing ▪ 256
 - Script ▪ 211
 - Default Time in Integrated Report
 - set ▪ 180
 - Default Viewer
 - open when export ▪ 185
 - Defect Tracking
 - set ▪ 198
 - Defects
 - submit ▪ 199
 - Defining a Mix ▪ 142
 - Defining Host Computers ▪ 169
 - Defining Load Machines ▪ 170
 - Defining Performance Goals ▪ 147, 151
 - Defining Probing Client Machines ▪ 170
 - Defining the Acceleration Rate ▪ 158
 - Defining WebLOAD Console's Action Once the Goals are Reached ▪ 160
 - Deleting a Data Source ▪ 384
 - Deleting a Goal ▪ 157
 - Deleting a Host Computer ▪ 113, 149, 171
 - Deleting a Measurement ▪ 385
 - Deleting a Monitored Host ▪ 384
 - Deleting a Report ▪ 303

- Deleting a script from a Mix ▪ 106, 144
 - Description
 - log window ▪ 265
 - Description Pane ▪ 368
 - PMM ▪ 368
 - Designating a Host Computer as a Load Machine ▪ 112, 148, 170
 - Designating a Host Computer as a Probing Client Machine ▪ 112, 149, 170
 - Designating all Host Computers as Load Machines ▪ 112, 149
 - Designating All Host Computers as Load Machines ▪ 170
 - Diagnostic
 - case inspections ▪ 242
 - enable ▪ 240
 - enhanced RadView support diagnostic ▪ 244
 - log report diagnostics ▪ 243
 - scope inspections ▪ 242
 - Script options ▪ 240
 - session options ▪ 240
 - syntax check ▪ 242
 - type inspections ▪ 242
 - value inspections ▪ 242
 - Diagnostic Options ▪ 240
 - Diagnostic Settings ▪ 240
 - Diagnostic Tab ▪ 240
 - Distributing the Load Equally ▪ 115
 - DNS Cache
 - simulate ▪ 225
 - DNS Lookup Time ▪ 454
 - definition ▪ 320
 - Dynatrace options ▪ 197
 - Dynatrace, integrating with ▪ 419
 - Monitoring a WebLOAD Load Session in Dynatrace ▪ 419
 - Specifying the Dynatrace System Profile ▪ 422
 - Viewing in Dynatrace a WebLOAD Point of Interest ▪ 421
 - Viewing in Dynatrace the Transactions Related to WebLOAD Errors ▪ 423
- E**
- Edit Mode ▪ 68
 - Console ▪ 45
 - options ▪ 54
 - right-click menu options ▪ 54
 - session tree ▪ 53, 68
 - Edit Mode and Run Modes of Operation ▪ 68
 - Editing a Report ▪ 301
 - Editing the Load Template ▪ 132
 - Enabling enhanced RadView Support
 - Diagnostic ▪ 244
 - Enabling Persistent Connection (Keep-Alive) ▪ 226
 - Enabling Show Performance Warnings ▪ 244
 - Enabling Syntax Checking ▪ 242
 - Error ▪ 293
 - Error Handling ▪ 265
 - Error Messages ▪ 261, 443
 - alphabetical listing of errors ▪ 444
 - logging ▪ 289
 - message status ▪ 265, 443
 - print ▪ 291
 - save ▪ 292
 - sort ▪ 291
 - Errors in Communication ▪ 275
 - Errors, full view ▪ 191
 - Establishing Communication ▪ 275
 - Event Viewer ▪ 66
 - expected vs actual results ▪ 266
 - Examining Your Test Resources ▪ 84
 - Example of Test Configuration ▪ 20
 - Examples ▪ 274
 - Expected Repository Files
 - .fta ▪ 449
 - Expected vs. Actual Results ▪ 266
 - Explanation of Creation and Analysis of Performance Reports ▪ 269
 - Export
 - configuration files ▪ 373
 - PMM configuration files ▪ 373
 - Export options ▪ 183
 - Export Options ▪ 184, 185, 186
 - Export Resolution
 - set ▪ 184
 - Export to HTML Settings ▪ 186
 - Exporting All Statistical Data ▪ 357
 - Exporting All Statistics to a Tab File ▪ 359
 - Exporting All Statistics to Excel ▪ 358
 - Exporting All Statistics to HTML ▪ 359
 - Exporting Performance Reports ▪ 351
 - Exporting Performance Reports to a Tab File ▪ 355
 - Exporting Performance Reports to Excel ▪ 353
 - Exporting Performance Reports to HTML ▪ 356

Exporting PMM Configuration Files ▪ 373
 Exporting Selective WebLOAD Report Data ▪ 352

F

Failed Connection
 definition ▪ 320
 Failed Connections ▪ 454
 Failed Hits ▪ 454
 definition ▪ 320
 Failed Hits Per Second ▪ 454
 definition ▪ 320
 Failed Pages Per Second ▪ 454
 Failed Rounds ▪ 454
 definition ▪ 320
 Failed Rounds Per Second ▪ 454
 definition ▪ 320
 Failure
 setting severity level for ▪ 261
 Features ▪ 12
 File menu
 PMM ▪ 366
 File Menu ▪ 366
 File tab ▪ 70
 new ▪ 70
 open Load Session ▪ 70
 open Load Template ▪ 70
 print ▪ 70
 save Load Session ▪ 70
 save Load Template ▪ 70
 Finishing the Goal-Oriented Test Wizard ▪ 161
 Firewall ▪ 22
 First Byte ▪ 454
 Form
 Script level verification ▪ 256
 Forming a Test Plan ▪ 81
 Frame
 Script level verification ▪ 256
 Full view of HTTP errors
 setting ▪ 189
 Full View of HTTP Errors ▪ 191
 Functional Testing ▪ 245
 automatic data collection ▪ 246
 forms ▪ 256
 frames ▪ 256
 images ▪ 256
 links ▪ 256
 page time verifications ▪ 247

 report default values ▪ 256
 reporting events ▪ 262
 select tests ▪ 258
 tables ▪ 256
 titles ▪ 256
 track transaction failures ▪ 262
 user-defined transactions ▪ 259
 verification function return values ▪ 261
 verify text ▪ 256
 view errors ▪ 264
 view results ▪ 264

Functional Testing Options ▪ 245
 Functional Testing Tab ▪ 245
 open ▪ 256
 Script options ▪ 245
 session options ▪ 245
 Functional Verification Testing ▪ 255

G

Gallery ▪ *See* Templates Gallery
 General Options
 autosave file name setting ▪ 179
 statistic view set ▪ 183
 Getting Started ▪ 7
 Git, using ▪ 283
 access to repositories ▪ 284
 Commit action ▪ 285
 Commit dir action ▪ 286
 Git UI, launching ▪ 288
 prerequisites ▪ 283
 Pull action ▪ 287
 Push action ▪ 286
 repositories location, specifying ▪ 284
 Global Options ▪ 177
 acceleration point measurements set ▪ 181
 advanced ▪ 193
 autosave file name setting ▪ 179
 data drilling ▪ 189
 database ▪ 194
 default integrated report time range set ▪ 180
 defect tracking ▪ 198
 Dynatrace ▪ 197
 export ▪ 183, 184, 185, 186
 file locations ▪ 186
 general tab ▪ 179
 goal reached delay setting ▪ 181
 load generator statistics set ▪ 180
 notification mail ▪ 196
 open ▪ 177

- set ▪ 177
 - statistic view set ▪ 183
 - statistics reporting resolution set ▪ 180
 - Goal Defaults
 - set ▪ 157
 - use ▪ 157
 - Goal Definition
 - about ▪ 151
 - add goal ▪ 153
 - control buttons ▪ 153
 - delete goal ▪ 153
 - measurement ▪ 152
 - operator ▪ 152, 153
 - report ▪ 152
 - set as default ▪ 153
 - set defaults ▪ 153
 - source to track ▪ 152
 - value ▪ 152
 - Goal Table
 - about ▪ 152
 - Goal-Oriented Test
 - acceleration point measurements set ▪ 181
 - goals reached delay set ▪ 181
 - set options ▪ 181
 - Goal-Oriented Test ▪ 455
 - Goal-Oriented Test Wizard ▪ 67, 135
 - acceleration rate ▪ 158
 - acceleration rate dialog box ▪ 158
 - add goal ▪ 153
 - default report ▪ 162
 - define a Mix ▪ 142
 - define a performance goal ▪ 151
 - define acceleration rate ▪ 158
 - delete goal ▪ 157
 - delete Script from Mix ▪ 144
 - example of configuring a goal ▪ 156
 - finish dialog box ▪ 161
 - goal definition ▪ 151
 - goal definition dialog box ▪ 151
 - goal definition example ▪ 156
 - goal definition measurement ▪ 152
 - goal definition source ▪ 152
 - goal table ▪ 152
 - goals achieved dialog box ▪ 160
 - goals definition control buttons ▪ 153
 - goals definition operator ▪ 152
 - goals definition report ▪ 152
 - goals definition value ▪ 152
 - how it works ▪ 136
 - Mix ▪ 139
 - Mix Script workflow ▪ 141
 - open ▪ 138
 - operator ▪ 153
 - performance measurements manager ▪ 145
 - PMM ▪ 145
 - run a Mix of Scripts ▪ 141
 - Script selection ▪ 140
 - select a script ▪ 141
 - select a Script ▪ 139
 - select host computers ▪ 146
 - set action after reaching goal ▪ 160
 - set and use goal defaults ▪ 157
 - set defaults ▪ 157
 - set performance monitor ▪ 141
 - set Script options ▪ 140
 - single Script ▪ 139
 - view Script ▪ 140
 - view test results ▪ 162
 - welcome dialog box ▪ 138
 - Goal-Oriented Test Wizard ▪ 455
- ## H
- Help drop-down ▪ 76
 - Help menu
 - PMM ▪ 367
 - Help Menu ▪ 367
 - Hit Time ▪ 455
 - definition ▪ 320
 - Hits ▪ 455
 - calculated per day ▪ 295
 - definition ▪ 319
 - failed ▪ 295
 - Hits Per Second ▪ 295, 455
 - definition ▪ 320
 - Home tab ▪ 71
 - Add Remote PMM ▪ 71
 - create Mix ▪ 71
 - Delete ▪ 71
 - goal-oriented test wizard ▪ 71
 - Load Machine selection ▪ 71
 - Mix creation ▪ 71
 - Performance Measurement Manager ▪ 71
 - Schedule ▪ 71
 - Scripts selection ▪ 71
 - SLA manager ▪ 71
 - Update floating license resources ▪ 71
 - WebLOAD wizard ▪ 71
 - Host ▪ 455
 - communication ▪ 24

- define host computers ▪ 169
 - pinging ▪ 26
 - selection window ▪ 168
 - Host Computer
 - add ▪ 147
 - define ▪ 147, 169
 - delete ▪ 149, 171
 - designate all load machines as ▪ 149
 - designate load machine as ▪ 148
 - designate probing client as ▪ 149
 - Goal-Oriented Test ▪ 146
 - select ▪ 168
 - select in Goal-Oriented Test Wizard ▪ 146
 - Host Computers
 - add cloud host computer ▪ 111
 - add local host computer ▪ 111
 - define ▪ 111
 - manually shut down cloud host computer ▪ 280
 - select ▪ 107
 - Host Machine
 - selecting ▪ 35
 - Host Selection Dialog Box
 - open ▪ 168
 - Hosts
 - add ▪ 375
 - add data sources ▪ 384
 - configure ▪ 375
 - delete ▪ 384
 - PMM ▪ 369
 - How are Tests Defined? ▪ 18
 - How does the Goal-Oriented Test Work? ▪ 136
 - How Does WebLOAD Work? ▪ 8
 - How Does WebRM Work ▪ 88
 - How to Set Global Options ▪ 177
 - HTML Export Settings ▪ 186
 - HTML file
 - export settings ▪ 186
 - HTTP Client ▪ 11
 - HTTP errors, full view ▪ 191
 - HTTP Response Status ▪ 456
 - definition ▪ 321
 - HTTP Version
 - set ▪ 229
- I**
- IBM WebSphere
 - data source ▪ 375
 - Icon Conventions ▪ 3
 - Icons
 - PMM ▪ 369
 - status ▪ 51
 - Image
 - Script level verification ▪ 256
 - Import
 - configuration files ▪ 372
 - PMM configuration files ▪ 372
 - Importing a Complete Report Configuration ▪ 306
 - Importing a Report Configuration for a Selected Component ▪ 306
 - Importing External Statistics ▪ 407
 - Importing Report Parameters from a Saved Load Template ▪ 306
 - Including a User-Defined Transaction in your script ▪ 259
 - Incrementing Intervals ▪ 123
 - Incrementing Intervals (Time Calculate) ▪ 124
 - Incrementing Intervals (Time Calculate) Model load profiler ▪ 124
 - Incrementing Intervals (Time Calculate) Model base load size ▪ 124
 - Incrementing Intervals (Time Calculate) Model concluding load size ▪ 124
 - Incrementing Intervals (Time Calculate) Model parameters ▪ 124
 - Incrementing Intervals (Time Calculate) Model time between each interval T1 ▪ 124
 - Incrementing Intervals (Time Calculate) Model time of each interval T2 ▪ 124
 - Incrementing Intervals (Time Calculate) Model time to increase each interval ▪ 124
 - Incrementing Intervals Model
 - base load size ▪ 123
 - load profiler ▪ 123
 - load to increase each interval ▪ 123
 - parameters ▪ 123
 - time between each interval T1 ▪ 123
 - time of each interval T2 ▪ 123
 - total time in minutes ▪ 123
 - Individually Scheduling the Load to Generate on Each Load Machine ▪ 118
 - Inserting a Text Verification Test ▪ 248
 - Instance Grid
 - duration ▪ 346
 - load generator ▪ 346
 - reason ▪ 346
 - round number ▪ 346
 - severity ▪ 346

- thread number ▪ 346
- time ▪ 346
- total load size ▪ 346

Integrated Report ▪ 61, 295

- add a saved Load Session ▪ 309
- add a user-defined timer ▪ 305
- change report parameters ▪ 305
- change the scale value ▪ 312
- chart time window width ▪ 180
- configure ▪ 303
- deleting a report ▪ 303
- import report parameters ▪ 306
- include data from saved Load Sessions ▪ 309
- Load Generator statistics ▪ 307
- new report ▪ 298, 430, 440
- open an existing report ▪ 301
- regression testing ▪ 308
- remove a saved Load Session ▪ 309
- remove data from saved Load Sessions ▪ 309
- select measurements ▪ 304
- view true value vs. relative value ▪ 313

Integrated Report Toolbar ▪ 313

Integrated Report Viewing ▪ 310

Integrated Reports ▪ 456

Integrated Results

- creating ▪ 40

Integrating with AppDynamics ▪ 409

Integrating with Dynatrace ▪ 419

Integrating with Nagios ▪ 425

Integrating with New Relic ▪ 433

Introducing WebLOAD ▪ 5

Introduction ▪ 1

J

Java and ActiveX counters ▪ 456

Java and ActiveX Counters ▪ 321

Java and ActiveX timers ▪ 456

Java and ActiveX Timers ▪ 321

JUnit building blocks ▪ 284

K

Keep-Alive

- enable ▪ 226

Keyboard

- navigating PMM with ▪ 370
- navigating session tree with ▪ 51
- shortcut keys for navigating PMM ▪ 370
- shortcut keys for navigating session tree ▪ 51

Keyboard Shortcuts

- for navigating PMM ▪ 370
- for navigating session tree ▪ 51

L

Launching WebLOAD Console Through the Command Line Interface ▪ 272

License information, viewing ▪ 76

Limiting a Run by Number of Rounds ▪ 116

Linear Model

- concluding load size ▪ 121
- load profiler ▪ 121
- parameters ▪ 121
- starting load size ▪ 121
- total time in minutes ▪ 121

Linear Scheduling Model ▪ 121

Link

- Script level verification ▪ 256

Load Generator ▪ 457

- bandwidth ▪ 234
- set max connection properties ▪ 224
- sharing data ▪ 193

Load Generator Machine ▪ 457

Load Generator Software

- Load Session diagram ▪ 9

Load Generator Spawning

- setting ▪ 193

Load Generator statistics ▪ 307

Load Generator Statistics

- set ▪ 183

Load Machine

- about ▪ 146
- define ▪ 170
- definition ▪ 9
- delete ▪ 113
- designate ▪ 112
- designate all as host computers ▪ 149
- designate as host computer ▪ 148
- icon ▪ 50
- Load Session diagram ▪ 9
- replace ▪ 113
- schedule equally in WebLOAD wizard ▪ 115

Load Machines

- select ▪ 168

Load Profiler

- incrementing intervals (time calculate) model ▪ 124
- incrementing intervals model ▪ 123
- linear model ▪ 121

- ramp up model ▪ 127
- random model ▪ 122
- schedule equal load manually ▪ 116, 119
- schedule load manually ▪ 173
- scheduling models ▪ 121
- step increment (time calculate) model ▪ 126
- step increment model ▪ 125
- user defined model ▪ 128
- Load Session ▪ 67, 457
 - .ls ▪ 96
 - available resources ▪ 84
 - configure ▪ 79
 - diagram ▪ 9
 - establishing communication ▪ 275
 - goals of the test ▪ 81
 - icon ▪ 50
 - launch through command line ▪ 272
 - manage ▪ 96
 - open ▪ 98
 - operating commands ▪ 278
 - pause command ▪ 279
 - plan ▪ 79, 81
 - resume command ▪ 279
 - run ▪ 79
 - run from system memory ▪ 271
 - running ▪ 271
 - save ▪ 96
 - save additional information ▪ 96
 - start ▪ 84, 271
 - start command ▪ 279
 - stop all command ▪ 280
 - stop command ▪ 280
 - view performance reports ▪ 79
 - workflow ▪ 79
- Load Session Components Software ▪ 20
- Load Session Workflow ▪ 79
- Load Size ▪ 457
 - change dynamically during runtime ▪ 281
 - definition ▪ 319
- Load Template ▪ 67, 91, 457
 - .tpl ▪ 91
 - configuring ▪ 34
 - define manually ▪ 166
 - edit ▪ 132
 - manual definition workflow ▪ 165
 - new ▪ 92
 - open ▪ 92
 - run ▪ 79
 - save ▪ 92
 - save additional information ▪ 93
 - Script selection ▪ 166
- Load Template Configuration
 - add all load machines ▪ 170
 - add load machine ▪ 170
 - add probing client ▪ 170
 - add probing clients ▪ 170
 - add Script ▪ 166, 167
 - delete host computer ▪ 171
 - host computer definition ▪ 168, 169
 - schedule load machines ▪ 171
 - schedule load machines with load profiler ▪ 173
- Load Template Defintion
 - set Script options ▪ 167
 - view Script ▪ 167
- Log Files
 - .mdb ▪ 449
- Log Message
 - options ▪ 182
 - view ▪ 290
- Log Report
 - diagnostics ▪ 243
- Log Report Diagnostics ▪ 243
 - enable ▪ 243
- Log Window ▪ 66
 - attachments ▪ 265
 - Clearing the Log Window ▪ 293
 - customize ▪ 289
 - description ▪ 265
 - Error ▪ 293
 - log error messages ▪ 289
 - message attachments ▪ 265
 - message source ▪ 265
 - message status ▪ 265
 - Minor Error ▪ 292
 - options ▪ 289
 - print contents ▪ 267
 - print error messages ▪ 291
 - save contents ▪ 267
 - save error messages ▪ 292
 - Severe Error ▪ 293
 - sort error messages ▪ 291
 - Success / Info Message ▪ 292
 - test summary ▪ 265
 - view ▪ 290
 - view message status ▪ 292
 - viewresults ▪ 265

M

- Main Window
 - PMM ▪ 365
- Managing Load Sessions ▪ 96
- Managing Load Templates ▪ 91
- Managing Load Templates and Load Sessions ▪ 91
- Managing the Communication Between the Systems in your Load Session ▪ 24
- Manual Test Configuration
 - workflow ▪ 165
- Manual Verification Tests
 - text ▪ 258
- Max ▪ 330
- Max Time
 - page time verifications ▪ 247
- Measurements
 - add ▪ 375
 - add to data sources ▪ 383
 - configure ▪ 375
 - delete ▪ 385
 - PMM ▪ 369
- Menu Bar ▪ 365
- Message Severity ▪ 443
- Message Source ▪ 265
- Message Status ▪ 265, 292, 443
 - Error ▪ 265, 293, 443
 - Minor Error ▪ 265, 292, 443
 - Severe Error ▪ 293, 443
 - Success / Info Message ▪ 265, 292, 443
- Microsoft ASP .Net
 - data source ▪ 375
- Microsoft Active Server Pages
 - data source ▪ 375
- Microsoft Internet Information Server (IIS)
 - data source ▪ 375
- Microsoft Media Server
 - data source ▪ 375
- Microsoft SQL Server 2000
 - data source ▪ 375
- Microsoft SQL Server using SNMP
 - data source ▪ 375
- Min ▪ 330
- Minor Error ▪ 292
- mix
 - WebLOAD wizard ▪ 104
- Mix ▪ 102
 - add Script ▪ 142
 - create ▪ 105, 142
 - define ▪ 105, 142
 - delete a script ▪ 106
 - delete a script ▪ 144
 - Goal-Oriented Test wizard ▪ 139
 - icon ▪ 50
 - new ▪ 95
 - run a Mix of Scripts ▪ 141
 - select ▪ 141, 166
 - select in Goal-Oriented Test Wizard ▪ 141
 - select in WebLOAD wizard ▪ 102
 - set Script options ▪ 142
 - view a script ▪ 107, 144
 - weight ▪ 142
- Modes of Operation
 - edit mode ▪ 68
 - run mode ▪ 68, 69
 - session mode ▪ 277
- Modifying Script Options ▪ 213
- Modifying the Host Selection ▪ 133
- Modifying the Load Schedule ▪ 133
- Monitor
 - SNMP data sources ▪ 403
- Monitoring
 - application server resources ▪ 363
 - database resources ▪ 363
 - databases ▪ 363
 - performance ▪ 363
 - PMM ▪ 363
 - real-time ▪ 363
 - server-side applications ▪ 363
 - stream technology ▪ 363
 - stream technology resources ▪ 363
 - system resources ▪ 363
 - web server resources ▪ 363
- Monitoring Generic SNMP Data Sources ▪ 403
- MySAP.com
 - data source ▪ 375

N

- Nagios, integrating with ▪ 425
- Named Transactions ▪ 246
- Navigating the PMM Configuration Tree ▪ 370
- Navigating the Session Tree ▪ 51
- Navigation Block ▪ 261
- Netscape LDAP
 - data source ▪ 375
- Network Configuration Parameters ▪ 24
- New Relic, integrating with ▪ 433

O

- On line Help modules, viewing ▪ 76
- Online Help ▪ 4
- Open
 - configuration files ▪ 372
 - PMM configuration files ▪ 372
- Open Viewer When Exporting HTML Files ▪ 186
- Open Viewer When Exporting Tab Files ▪ 185
- Opening / Importing Saved PMM
 - Configuration Files ▪ 372
- Opening a Data Drilling Report ▪ 344
- Opening a Saved Load Template ▪ 92
- Opening an Existing Report ▪ 301
- Opening Reports ▪ 298, 430, 440
- Opening Saved Load Sessions Files ▪ 98
- Opening the Console ▪ 44
- Opening the Functional Testing Tab ▪ 256
- Opening the Goal-Oriented Test Wizard ▪ 138
- Opening the Host Selection Dialog Box ▪ 168
- Opening the Log Window ▪ 290
- Opening the Log Window Menu ▪ 291
- Opening the Performance Measurements Manager ▪ 364
- Opening the WebLOAD Wizard ▪ 34, 100
- Opening Throttle Control ▪ 281
- Operating Commands ▪ 278
 - pause ▪ 279
 - resume ▪ 279
 - start ▪ 279
 - stop ▪ 280
 - stop all ▪ 280
- Operator ▪ 153
- Options
 - global ▪ 177
 - set for current session ▪ 211
 - set for Script ▪ 211
- Oracle
 - configure on Windows ▪ 395, 397
 - configure SNMP master agent ▪ 397
 - data source ▪ 375
 - start SNMP agents ▪ 398
 - Windows configuration tasks ▪ 395, 397
- Oracle Monitor Configuration Tasks on UNIX ▪ 397
- Oracle Monitor Configuration Tasks on Windows ▪ 395
- Other

- general performance monitor ▪ 375
- general SNMP ▪ 375
- general UNIX ▪ 375

P

- Page Time ▪ 458
 - definition ▪ 319
- Page Time Verifications
 - enable ▪ 247
 - max time ▪ 247
 - Script option ▪ 247
 - session option ▪ 247
 - severity level ▪ 247
 - verification ▪ 247
- Pages ▪ 458
 - calculated per day ▪ 295
 - definition ▪ 319
 - per second ▪ 295
- Pages Per Second ▪ 458
 - definition ▪ 319
- Parameters ▪ 273
- Pass / Fail Definition
 - Script options ▪ 249
 - session options ▪ 249
 - set absolute ▪ 249
 - set relative ▪ 249
- Pass / Fail Definition Settings ▪ 249
- Pause
 - test session ▪ 279
- Pausing a Test Session ▪ 279
- Per Time Unit statistics ▪ 458
- Per Time-Unit Statistics
 - current value calculation ▪ 319
 - defined ▪ 317
- Performance Measurements
 - select ▪ 130
- Performance Measurements Manager ▪ 363
- Performance Measurements Managersee PMM ▪ 363
- Performance Monitor
 - data source ▪ 375
- Performance Statistic
 - selecting ▪ 36
- Persistent Connection
 - enable ▪ 226
- Pinging a Host ▪ 26
- Planning a Test Program ▪ 82
- Planning a Test Session ▪ 79

PMM

- about ▪ 363
- Actions Menu ▪ 367
- activate monitors ▪ 363
- add data sources ▪ 375
- add data sources to hosts ▪ 384
- add hosts ▪ 375
- add measurements ▪ 375
- add measurements to data sources ▪ 383
- adding data source ▪ 367
- application server monitors ▪ 387
- application servers data sources ▪ 375
- clear all ▪ 367
- close and update template ▪ 366
- collapse tree ▪ 367
- configuration files ▪ 371
- configuration pane ▪ 368
- configuration tree ▪ 368
- configuration tree icons ▪ 369
- configuration tree structure ▪ 368
- configurations ▪ 363
- configure administrator privileges for
 - Perfmon (Windows) ▪ 385
- create new configuration files ▪ 371
- data sources ▪ 363, 369, 375
- database monitors ▪ 395
- databases data sources ▪ 375
- delete ▪ 367
- delete data sources ▪ 384
- delete hosts ▪ 384
- delete measurements ▪ 385
- description pane ▪ 368
- exit ▪ 366
- expand tree ▪ 367
- export configuration files ▪ 373
- exporting configuration files ▪ 366
- extend the MIB filter ▪ 403
- file menu ▪ 366
- Help menu ▪ 367
- hosts ▪ 363, 369, 375
- import configuration files ▪ 372
- importing configuration files ▪ 366
- interated report ▪ 363
- keyboard shortcut keys for navigating ▪ 370
- main window ▪ 365
- measurements ▪ 363, 369, 375
- menu bar ▪ 365
- monitor databases ▪ 363
- monitor server side applications ▪ 363
- monitor SNMP data sources ▪ 403

- monitor stream technology ▪ 363
- monitor Web server measurements ▪ 363
- monitoring performance ▪ 363
- new ▪ 366
- new configuration ▪ 366
- open ▪ 364, 366
- open configuration files ▪ 372
- open PMM Wizard ▪ 375
- other data sources ▪ 375
- real-time monitor ▪ 363
- save configuration files ▪ 373, 374
- saving configuration files ▪ 366
- standalone tool ▪ 363
- system data sources ▪ 375
- template ▪ 364
- web server monitors ▪ 399
- Web servers data sources ▪ 375
- wizard ▪ 363

PMM Actions Menu

- add data source ▪ 367
- clear all ▪ 367
- collapse tree ▪ 367
- delete ▪ 367
- expand tree ▪ 367

PMM Configuration Files

- append ▪ 372
- attach to WebLOAD Console templates ▪ 371, 373
- create new ▪ 371
- export ▪ 373
- import ▪ 372
- new ▪ 371
- open ▪ 372
- replace ▪ 372
- save ▪ 371, 373, 374
- use ▪ 371

PMM Configuration Tree

- icons ▪ 369

PMM File Menu

- close and update template ▪ 366
- exit ▪ 366
- export ▪ 366
- import ▪ 366
- new ▪ 366
- new configuration ▪ 366
- open ▪ 366
- save ▪ 366

PMM Wizard

- data source screen ▪ 375
- finish screen ▪ 375

RADVIEW

- host selection screen ▪ 375
- measurements selection screen ▪ 375
- open ▪ 375
- welcome screen ▪ 375
- Portfolio ▪ 458
- Printing Error Messages Recorded in the Log Window ▪ 291
- Printing the Contents of the Log Window ▪ 267
- Probing Client ▪ 458
 - about ▪ 146
 - bandwidth ▪ 234
 - define ▪ 112, 170
 - designate ▪ 112
 - designate as host computer ▪ 149
 - select ▪ 168
 - set max connections ▪ 224
- Probing Client Machine ▪ 458
 - icon ▪ 50
 - Load Session diagram ▪ 9
- Probing Client software ▪ 458
- Probing Client Software
 - Load Session diagram ▪ 9
- Probing Clients ▪ 87
- Process Time ▪ 458
 - definition ▪ 320
- Project Files
 - .pft ▪ 449
- Pull - Git action ▪ 287
- Push - Git action ▪ 286

Q

- Quick access toolbar ▪ 47
- Quick Start Steps ▪ 29

R

- RadView Support
 - diagnostics ▪ 244
- RadView Support Diagnostic
 - enable ▪ 244
- Ramp Up ▪ 127
 - base load size ▪ 127
 - load profiler ▪ 127
 - max load size ▪ 127
 - parameters ▪ 127
 - ramp down time ▪ 127
 - ramp up time ▪ 127
 - time to run max load size ▪ 127
- Random ▪ 122
- Random Model
 - load profiler ▪ 122
 - max load size ▪ 122
 - min load size ▪ 122
 - parameters ▪ 122
 - start load size ▪ 122
 - total time in minutes ▪ 122
- Real-Time Monitor
 - PMM ▪ 363
- Receive Time ▪ 458
 - definition ▪ 320
- Redirection
 - set limits ▪ 225
- Reference Materials ▪ 449
- Regression Testing ▪ 308
- Removing a Previously Saved Load Sessions from the Integrated Report Configuration ▪ 309
- Renaming a Report ▪ 302
- Replace
 - configuration files ▪ 372
 - PMM configuration files ▪ 372
- Replacing a Load Machine or Probing Client Machine ▪ 113
- Report
 - events ▪ 262
- Report Default Values
 - Script level verification ▪ 256
- Report Export Options ▪ 351
- Report Portfolio ▪ 458
- Reported Value ▪ 319
- ReportEvent
 - description ▪ 262
 - event-name ▪ 262
 - syntax ▪ 262
- Reporting Events ▪ 262
- Reporting Interval ▪ 318
- Reports
 - dashboard ▪ 63, 295
 - data drilling ▪ 65, 343
 - export ▪ 351
 - export all statistical data ▪ 357
 - export all statistics to a HTML file ▪ 359
 - export all statistics to a tab file ▪ 359
 - export all statistics to Excel ▪ 358
 - export options ▪ 351
 - export selective data ▪ 352
 - export to a HTML ▪ 356
 - export to a tab file ▪ 355

- export to Excel ▪ 353
 - Integrated ▪ 295
 - set ▪ 180
 - set options ▪ 180
 - statistics ▪ 317
 - transactions dashboard ▪ 64, 295
 - Requesting Resources from WebRM ▪ 88
 - Required Connections ▪ 25
 - Required software ▪ 20
 - Resource Manager ▪ 87, 459
 - Resource Manager WebRM ▪ 87
 - Response Data Size ▪ 459
 - definition ▪ 320
 - Response Time ▪ 459
 - definition ▪ 320
 - Responses ▪ 459
 - definition ▪ 320
 - Results
 - view in log window ▪ 265
 - Results Description
 - additional information ▪ 97
 - Results of Test ▪ 39
 - Results Window ▪ 60
 - dashboard ▪ 63
 - data drilling ▪ 65
 - integrated report ▪ 61
 - transactions dashboard ▪ 64
 - Resume
 - test session ▪ 279
 - Resuming a Test Session ▪ 279
 - Return Values
 - Error ▪ 261
 - MinorError ▪ 261
 - SevereError ▪ 261
 - Success ▪ 261
 - Ribbon ▪ 47
 - file tab ▪ 70
 - home tab ▪ 71
 - session tab ▪ 72
 - tools tab ▪ 74
 - Ribbon Options ▪ 70
 - Right-click Menus ▪ 48
 - Right-Click Options in Edit Mode ▪ 54
 - Right-Click Options in Run Mode ▪ 58
 - Round termination ▪ 213, 217
 - Round Time ▪ 459
 - definition ▪ 320
 - Round timeout ▪ 213, 217
 - Rounds ▪ 459
 - definition ▪ 320
 - executed ▪ 295
 - failed ▪ 295
 - Rounds Per Second ▪ 459
 - definition ▪ 320
 - Run All Tests ▪ 256
 - Run Mode ▪ 68
 - Console ▪ 45
 - options ▪ 58
 - right-click menu options ▪ 58
 - session tree ▪ 56, 69
 - status icons ▪ 51
 - view the session tree ▪ 56
 - Running a Load Session ▪ 271
 - Running a Load Session from the System
 - Memory ▪ 271
 - Running a Mix of Scripts ▪ 141
 - Running Single Scripts ▪ 140
 - Running Tests and Analyzing Test Results ▪ 269
 - Running the Test ▪ 38
- ## S
- Save
 - configuration files ▪ 373, 374
 - PMM configuration files ▪ 373, 374
 - Save Flag ▪ 259
 - Saving a Load Session ▪ 96
 - Saving a Load Template ▪ 92
 - Saving Additional Information for the Load
 - Session ▪ 96
 - Saving Additional Information with the Load
 - Template ▪ 93
 - Saving and Exporting PMM Configurations ▪ 373
 - Saving Error Messages Recorded in the Log
 - Window ▪ 292
 - Saving PMM Configuration Files ▪ 374
 - Saving the Contents of the Log Window ▪ 267
 - Saving the Current Load Template ▪ 272
 - Schedule
 - individual ▪ 118
 - WebLOAD wizard ▪ 114
 - Scheduling
 - manually ▪ 171
 - Scheduling a Session Manually with the Load
 - Profiler ▪ 119
 - Scheduling Model
 - incrementing intervals ▪ 123

- incrementing intervals(time calculate) ▪ 124
 - linear ▪ 121
 - load profiler ▪ 121
 - ramp up ▪ 127
 - random ▪ 122
 - step increments ▪ 125
 - step increments (time calculate) ▪ 126
 - user defined ▪ 128
- Scheduling Models ▪ 121
- Scheduling the Load Session ▪ 114
- Scheduling the Session Manually ▪ 118
- Scheduling the Test ▪ 35
- Scheduling the Test Session ▪ 171
- Scope Inspections
 - syntax check ▪ 242
- script
 - current session options ▪ 211
 - default options ▪ 211
 - options ▪ 211
- Script ▪ 102, 451
 - copy ▪ 21
 - creating ▪ 29
 - delete from Mix ▪ 106, 144
 - icon ▪ 50
 - Mix ▪ 139, 166
 - options ▪ 211
 - run a Mix of Scripts ▪ 141
 - run in Goal-Oriented Test Wizard ▪ 141
 - run test with single ▪ 167
 - select ▪ 166
 - select in Goal-Oriented Test wizard ▪ 139
 - select in WebLOAD wizard ▪ 102, 103
 - selecting ▪ 34
 - share ▪ 21
 - shared disk ▪ 21
 - single ▪ 139, 166
 - syntax check ▪ 242
 - verify ▪ 256
 - view from Mix ▪ 107, 144
- Script Options ▪ 211
 - authentication tab ▪ 232
 - automatic data collection ▪ 246
 - bandwidth ▪ 234
 - browser emulation tab ▪ 221
 - browser type ▪ 223
 - connection tab ▪ 234
 - current session ▪ 211
 - default ▪ 211
 - diagnostic tab ▪ 240
 - DNS cache ▪ 225
 - enhanced RadView support diagnostic ▪ 244
 - functional testing tab ▪ 245
 - HTTP version ▪ 229
 - keep-alive ▪ 226
 - load generator max connections ▪ 224
 - log report diagnostics ▪ 243
 - max connection ▪ 224
 - modify ▪ 213
 - page time verifications ▪ 247
 - pass / fail definition tab ▪ 249
 - persistent connection ▪ 226
 - probing client max connections ▪ 224
 - redirection limits ▪ 225
 - Show Performance Warnings ▪ 244
 - sleep time control tab ▪ 215
 - SSL cache ▪ 226
 - syntax check ▪ 242
 - version ▪ 223
- Script Verification
 - compare forms ▪ 256
 - compare images ▪ 256
 - compare links ▪ 256
 - compare number of frames ▪ 256
 - compare tables ▪ 256
 - compare title ▪ 256
 - report default values ▪ 256
 - run all tests ▪ 256
 - text search ▪ 256
- Search Paths
 - set default ▪ 186
- Selecting
 - host machine ▪ 35
 - performance statistic to monitor ▪ 36
 - Script ▪ 34
- Selecting a Mix ▪ 104
- Selecting a script ▪ 166
- Selecting a Script ▪ 103
- Selecting a script or Mix ▪ 102
- Selecting a Script or Mix ▪ 139
- Selecting Host Computers ▪ 107, 146
- Selecting Hosts ▪ 168
- Selecting Measurements for the Integrated Report ▪ 304
- Selecting Performance Measurements ▪ 145
- Selecting the Performance Measurements ▪ 130
- Selecting the Tests to Perform ▪ 258
- Send Time ▪ 460
 - definition ▪ 320

- Server Performance Measurements ▪ 460
 - definition ▪ 321
- Server Performance Measurements ▪ 321
- Server-Side Applications
 - monitoring ▪ 363
- Session Comments
 - additional information ▪ 97
- Session Files
 - .fts ▪ 449
- Session Mode
 - Console ▪ 277
 - viewing the session tree ▪ 278
- Session Options
 - authentication for protocol scripts ▪ 232
 - automatic data collection ▪ 246
 - bandwidth ▪ 234
 - browser emulation tab ▪ 221
 - diagnostic tab ▪ 240
 - functional testing tab ▪ 245
 - setting ▪ 38
 - sleep time control tab ▪ 215
- Session tab ▪ 72
 - Dashboard window ▪ 72
 - Edit template ▪ 72
 - export all ▪ 72
 - export report ▪ 72
 - Import external statistics ▪ 72
 - Integrated report manager ▪ 72
 - Log window ▪ 72
 - Open data drilling ▪ 72
 - Open statistics ▪ 72
 - Start session ▪ 72
 - Stop all ▪ 72
 - Throttle control ▪ 72
 - Transactions dashboard window ▪ 72
- Session Tree ▪ 48, 460
 - edit mode ▪ 53, 68
 - editing configuration ▪ 68
 - icons ▪ 50
 - navigate with keyboard shortcuts ▪ 51
 - right mouse button functions ▪ 68, 69
 - run mode ▪ 56, 69
 - session mode ▪ 278
 - status icons ▪ 51
 - viewing configuration ▪ 69
- Session Tree Icons ▪ 50
- SetFailureReason
 - syntax ▪ 262
- Setting Additional Information Properties ▪ 93, 97
- Setting and Using Defaults ▪ 157
- Setting Authentication ▪ 232
- Setting Automatic Data Collection ▪ 246
- Setting Defaults ▪ 157
- Setting Global Options ▪ 177
- Setting Goal-Oriented Test Options ▪ 181
- Setting Load Generator and Probing Client Max Connections ▪ 224
- Setting Page Time Verifications ▪ 247
- Setting Redirection Limits ▪ 225
- Setting Script Options ▪ 211
- Setting Session Options ▪ 38
- Setting Sleep Time Playback Settings ▪ 215
- Setting the Autosave File Name Default Location ▪ 179
- Setting the Browser Cache ▪ 219
- Setting the Browser Type and Version ▪ 223
- Setting the Export to Tab File Settings ▪ 185
- Setting the HTTP Parameters ▪ 229
- Setting the Log Message Options ▪ 182
- Setting the Log Window Options ▪ 289
- Setting the Reports Options ▪ 180
- Setting the SSL Options ▪ 213
- Setting the Statistic View Settings ▪ 183
- Setting the Statistics Resolution to Export ▪ 184
- Setting the Throttle Control Max Value ▪ 182
- Setting up the SNMP Agent and Configuring the Sun One Server as a Proxy for the SNMP Agent ▪ 400
- Setting WebLOAD Console Timeout Values ▪ 181
- Severe Error ▪ 293
- Severity Level ▪ 261
 - page time verifications ▪ 247
 - Script verification ▪ 256
- Sharing a script ▪ 21
- Show Performance Warnings ▪ 244
- Simulating DNS Cache ▪ 225
- Simulating SSL Cache ▪ 226
- Single Client ▪ 460
- SLA Mail
 - manager ▪ 204
- SLA Manager ▪ 204
- Sleep Time
 - as recorded ▪ 215
 - ignore ▪ 215

- set ▪ 215
- set deviation ▪ 215
- set random ▪ 215
- Sleep Time Control
 - Script options ▪ 215
 - session options ▪ 215
- SNMP
 - configure master agent on UNIX ▪ 397
 - data source ▪ 375
 - start agents on UNIX ▪ 398
- SNMP Agent
 - Sun One ▪ 400
- SNMP Data Sources
 - monitor ▪ 403
- snmpd.config File ▪ 398
- Software for Load Session Components ▪ 20
- Sorting Error Messages ▪ 291
- SSL Bit Limit
 - set ▪ 213
- SSL Cache
 - simulate ▪ 226
- Standard Deviation ▪ 330, 460
- Start
 - test session ▪ 279
- start_peer Script ▪ 398
- Starting a Load Session ▪ 271
- Starting a New Template Manually ▪ 166
- Starting a Test Session ▪ 279
- Starting the SNMP Agents ▪ 398
- Starting WebLOAD Console ▪ 84
- Statistic View
 - set ▪ 183
- Statistical Measurements and Their Values ▪ 317
- Statistics
 - current value ▪ 317
 - defined ▪ 317
 - differences between probing and virtual clients ▪ 330
 - list of ▪ 319
 - report resolution ▪ 318
- Statistics Definitions ▪ 319
- Statistics Exporting Resolution
 - set ▪ 184
- Statistics for Virtual Clients and Probing Clients
 - 330
- Statistics report ▪ 317
- Statistics Report ▪ 62
 - history by measurement and Script ▪ 339
 - history by measurement and total ▪ 340
 - main window ▪ 317
 - summary by measurement ▪ 337
 - summary by Script ▪ 333
 - summary by total ▪ 335
 - summary reports ▪ 330
 - use ▪ 332
- Statistics Report-History by Measurement and Script ▪ 339
- Statistics Report-History by Measurement and Total ▪ 340
- Statistics Reporting Resolution
 - set ▪ 180
- Statistics Report-Summary by Measurement ▪ 337
- Statistics Report-Summary by Script ▪ 333
- Statistics Report-Summary by Total ▪ 335
- Statistics Window ▪ 62
- Status Bar ▪ 48
- Status Icons ▪ 51
 - currently running ▪ 51
 - error ▪ 51
 - initialization complete ▪ 51
 - no current activity ▪ 51
 - stopped ▪ 51
 - temporarily paused ▪ 51
- Step Increment (Time Calculate) Model
 - load profiler ▪ 126
- Step Increment Model
 - load profiler ▪ 125
- Step Increments ▪ 125
 - load to increase each interval ▪ 125
 - parameters ▪ 125
 - starting load size ▪ 125
 - time of each interval ▪ 125
 - total time in minutes ▪ 125
- Step Increments (Time Calculate) ▪ 126
 - concluding load size ▪ 126
 - load to increase each interval ▪ 126
 - parameters ▪ 126
 - starting load size ▪ 126
 - time of each interval ▪ 126
- Stop
 - test session ▪ 280
- Stop All
 - test session ▪ 280
- Stopping a Load Session Component ▪ 280
- Stopping a Test Session ▪ 280
- Storage Locations
 - set default ▪ 186

- Subject
 - additional information ▪ 93, 97
 - Success / Info Message ▪ 292
 - Successful Connections ▪ 460
 - definition ▪ 320
 - Successful Hits ▪ 460
 - definition ▪ 319
 - Successful Hits Per Second ▪ 460
 - definition ▪ 320
 - Successful Pages Per Seconds ▪ 460
 - Successful Rounds ▪ 461
 - definition ▪ 320
 - Successful Rounds Per Second ▪ 461
 - definition ▪ 320
 - Sum ▪ 330
 - Summary Report ▪ 330
 - values ▪ 330
 - Summary Reports ▪ 330
 - Sun One
 - configuration tasks ▪ 400
 - configure ▪ 400
 - data source ▪ 375
 - set as proxy for SNMP agent ▪ 400
 - Sun One (iPlanet) Configuration Tasks ▪ 400
 - SUT ▪ 451, 461
 - Syntax Check
 - case inspections ▪ 242
 - enable ▪ 242
 - scope inspections ▪ 242
 - type inspections ▪ 242
 - value inspections ▪ 242
 - System
 - resource monitor ▪ 363
 - UNIX platforms network data using SNMP ▪ 375
 - UNIX platforms using RSTATD ▪ 375
 - UNIX UC-Davis ▪ 375
 - Windows .Net framework ▪ 375
 - Windows NT/2000/XP ▪ 375
 - System Under Test
 - additional information ▪ 93, 97
- T**
- Tab File
 - export settings ▪ 185
 - Tab File Export Settings ▪ 185
 - Table
 - Script level verification ▪ 256
 - Technical Support ▪ 4
 - Technical Support Website ▪ 4
 - Template ▪ 461
 - Load Template ▪ 67
 - Template Comments
 - additional information ▪ 93, 97
 - Templates
 - attach PMM configuration files ▪ 371, 373
 - Templates Gallery
 - definition ▪ 461
 - Test
 - analyzing results ▪ 39
 - running ▪ 38
 - schedule ▪ 171
 - schedule manually ▪ 171
 - schedule with load profiler ▪ 173
 - scheduling ▪ 35
 - text ▪ 258
 - viewing results ▪ 40
 - viewing results with WebLOAD Analytics ▪ 41
 - Test Description
 - additional information ▪ 93, 97
 - Test Parameters
 - edit ▪ 68
 - Test Program ▪ 461
 - Test Script ▪ 461
 - Test Session
 - diagram ▪ 9
 - Test Summary ▪ 265
 - Test Template ▪ 461
 - Testing Connection ▪ 26
 - TestTalk ▪ 24, 461
 - definition ▪ 9
 - open ▪ 84
 - Text
 - search ▪ 258
 - test ▪ 258
 - verification of ▪ 258
 - Text Not To Find ▪ 258
 - Text Search
 - Script level verification ▪ 256
 - Text To Find ▪ 258
 - The Advanced Tab ▪ 193
 - The Console in Edit Mode ▪ 68
 - The Console in Run Mode ▪ 69
 - The Console Ribbon Options ▪ 70
 - The Console Screen in Session Mode ▪ 277
 - The Data Drilling Tab ▪ 189

- The Database Tab ▪ 194
 - The Defect Tracking Tab ▪ 198
 - The Dynatrace Tab ▪ 197
 - The Export Tab ▪ 183
 - The File Locations Tab ▪ 186
 - The Functional Testing Tab ▪ 255
 - The General Tab ▪ 179
 - The Goal Table ▪ 152
 - The Goal-Oriented Test Wizard Workflow ▪ 137
 - The Manual Test Configuration Workflow ▪ 165
 - The Notification Mail Tab ▪ 196
 - The Performance Measurements Manager Main Window ▪ 365
 - The PMM Configuration Tree Structure ▪ 368
 - The Results Window ▪ 60
 - The Session Tree ▪ 48
 - The Session Tree in Edit Mode ▪ 53
 - The Session Tree in Run Mode ▪ 56
 - The Session Tree Structure ▪ 50
 - The SLA Manager ▪ 204
 - The WebLOAD Console ▪ 43
 - The WebLOAD Console Startup Dialog Box ▪ 67
 - The WebLOAD Wizard Workflow ▪ 100
 - Throttle Control ▪ 281, 461
 - activate ▪ 282
 - de-activating ▪ 282
 - max value set ▪ 182
 - maximum number of virtual clients ▪ 182
 - open ▪ 281
 - Throttle Control - Changing the Load Size Dynamically During Runtime ▪ 281
 - Throughput
 - calculated per day ▪ 295
 - definition ▪ 319
 - per second ▪ 295
 - Throughput (Bytes Per Second) ▪ 461
 - Time ▪ 295
 - Time to First Byte ▪ 461
 - definition ▪ 320
 - Timeout
 - set options ▪ 181
 - start session timeout set ▪ 181
 - terminate session timeout set ▪ 181
 - Timer Statistics
 - current value calculation ▪ 319
 - defined ▪ 317
 - Title
 - additional information ▪ 93, 97
 - Script level verification ▪ 256
 - Title bar ▪ 45
 - Tools
 - current session options ▪ 256
 - default options ▪ 256
 - Tools tab ▪ 74
 - cloud options ▪ 74
 - current session options ▪ 74
 - default options ▪ 74
 - global options ▪ 74
 - Script options ▪ 74
 - terminate cloud machines ▪ 74
 - WebLOAD Analytics ▪ 74
 - WebLOAD Recorder ▪ 74
 - Tracking Defects ▪ 199
 - Tracking Reasons for Transaction Failure ▪ 262
 - Transaction Failure
 - track ▪ 262
 - Transaction Name ▪ 259
 - Transaction Reason Failure Grid ▪ 345
 - Transactions
 - named ▪ 246
 - user-defined ▪ 259
 - Transactions Dashboard ▪ 64, 297
 - Transactions Grid
 - failed transactions ▪ 344
 - marked count ▪ 344
 - successful count ▪ 344
 - total count ▪ 344
 - transaction name ▪ 344
 - Troubleshooting Communications ▪ 27
 - Type Inspections
 - syntax check ▪ 242
 - Typographical Conventions ▪ 3
- ## U
- UNIX
 - data source ▪ 375
 - UNIX platforms network data using SNMP
 - data source ▪ 375
 - UNIX platforms using RSTATD
 - data source ▪ 375
 - UNIX UC-Davis
 - data source ▪ 375
 - Updating the Number of Requested Resources ▪ 89
 - User Defined ▪ 128

- data file name ▪ 128
 - delimiter ▪ 128
 - load profiler ▪ 128
 - max load size ▪ 128
 - min load size ▪ 128
 - parameters ▪ 128
 - total time in minutes ▪ 128
 - User-Defined
 - Automatic Data Collection ▪ 321
 - Counters ▪ 321
 - Timer ▪ 321
 - Total TPS ▪ 321
 - Transaction Counters ▪ 321
 - Transaction Timers ▪ 321
 - transactions ▪ 259
 - User-defined Automatic Data Collection ▪ 462
 - User-defined counters ▪ 462
 - User-defined timer ▪ 462
 - User-defined Transaction counters ▪ 463
 - User-Defined Transaction Example ▪ 260
 - User-Defined Transactions
 - syntax ▪ 259
 - User-Defined Transactions and Verification ▪ 259
 - User-Defined Transactions and Verification Syntax ▪ 259
 - User-defined Transactions timers ▪ 463
 - UserMIBFilter.ini
 - extend ▪ 403
 - Using Defaults ▪ 157
 - Using PMM Configuration Files ▪ 371
 - Using the Goal-Oriented Test ▪ 135
 - Using the Performance Measurements Manager ▪ 363
 - Using the Statistics Report ▪ 332
 - Using the WebLOAD Wizard to create Load Templates ▪ 99
- V**
- Value Inspections
 - syntax check ▪ 242
 - Verification
 - page time verifications ▪ 247
 - Verification Function ▪ 259
 - return values ▪ 261
 - Verification Function Return Values ▪ 261
 - Verification Tests
 - for entire Script ▪ 256
 - form ▪ 256
 - frame ▪ 256
 - image ▪ 256
 - link ▪ 256
 - perform ▪ 258
 - report default values ▪ 256
 - table ▪ 256
 - text ▪ 256, 258
 - title ▪ 256
 - view errors ▪ 264
 - view results ▪ 264
 - Verifying Text ▪ 258
 - Version
 - set ▪ 223
 - Viewing a script from a Mix ▪ 107, 144
 - Viewing Error Information in the Event Viewer ▪ 266
 - Viewing Error Information in the Log Window ▪ 265
 - Viewing Integrated Reports ▪ 310
 - Viewing Test Results
 - real time ▪ 40
 - WebLOAD Analytics ▪ 41
 - Viewing Test Results on the WebLOAD Dashboard ▪ 295
 - Viewing Test Results on the WebLOAD Transactions Dashboard ▪ 297
 - Viewing the Child Transaction Instance Tree ▪ 348
 - Viewing the Entire Log Message ▪ 290
 - Viewing the Goal-Oriented Test Default Report ▪ 162
 - Viewing the Instance Grid ▪ 346
 - Viewing the Log Window ▪ 289
 - Viewing the Message Status ▪ 292
 - Viewing the Network Configuration ▪ 24
 - Viewing the Parent Transaction Instance Tree ▪ 347
 - Viewing the Session Tree in Session Mode ▪ 278
 - Viewing the Transaction Grid ▪ 344
 - Viewing the Transaction Reason Failure Grid ▪ 345
 - Viewing the True Value vs. Relative Value of a Measurement ▪ 313
 - Viewing the WebLOAD Wizard Default Report ▪ 132
 - Viewing Verification Test Results ▪ 264
 - Virtual Client ▪ 463

Virtual Clients ▪ 87
 running ▪ 295

W

Web Server

resource monitor ▪ 363

Web Server Monitors ▪ 399

Web servers

Microsoft Internet Information Server (IIS) ▪
 375

Netscape LDAP ▪ 375

Sun One ▪ 375

WebLOAD

architecture ▪ 9

dashboard ▪ 295

description ▪ 7

features ▪ 12

transactions dashboard ▪ 297

WebLOAD Analytics ▪ 463

viewing test results ▪ 41

WebLOAD Architecture ▪ 9

WebLOAD Console ▪ 463

dashboard ▪ 295

files used by WebLOAD Console ▪ 449

launch through command line interface ▪ 272

open ▪ 84

start ▪ 84

transactions dashboard ▪ 295

WebLOAD Console Error Messages ▪ 261, 443,
 444

WebLOAD Console Features ▪ 43

WebLOAD Console Files ▪ 449

WebLOAD Console Operating Commands ▪
 278

WebLOAD Console Options for Creating Load
 Templates: ▪ 77

WebLOAD Console Performance Reports ▪ 295

WebLOAD Console Templates

attach PMM configuration files ▪ 371, 373

WebLOAD Console.exe Command ▪ 272

WebLOAD Documentation ▪ 1

WebLOAD Features ▪ 12

WebLOAD Load Template ▪ 464

WebLOAD Offers Two Client Type Modes ▪ 11

WebLOAD Quick Start ▪ 29

WebLOAD Recorder ▪ 463

WebLOAD Resource Manager WebRM ▪ 87

WebLOAD Session ▪ 464

WebLOAD Statistics Reports ▪ 317

WebLOAD Wizard ▪ 67, 464

configuring Load Template ▪ 34

create Mix ▪ 105

create Mix ▪ 104

default report ▪ 132

define a Mix ▪ 105

delete load machine ▪ 113

delete Script from Mix ▪ 106

designate load machine ▪ 112

designate probing clients ▪ 112

finish dialog box ▪ 131

host selection dialog box ▪ 107

individual schedule ▪ 118

limiting a run by number of rounds ▪ 116

modify host selection ▪ 133

modify load machine ▪ 113

modify load schedule ▪ 133

open ▪ 100

opening ▪ 34

Performance Measurements Manage dialog
 box ▪ 130

replace load machine ▪ 113

schedule equal load ▪ 115

schedule equal load manually ▪ 115

schedule equal load with load profiler ▪ 116,
 119

schedule session ▪ 114

Script selection ▪ 103

Script source dialog box ▪ 100

Script/Mix selection ▪ 102

select host computers ▪ 107

select mix ▪ 104

select performance measurements ▪ 130

view Script from Mix ▪ 107, 144

view test results ▪ 132

WebRM ▪ 87

Welcome to WebLOAD ▪ 7

What is the Product of Your WebLOAD
 Testing? ▪ 10

Where to Get More Information ▪ 4

Windows

data source ▪ 375

Windows .Net framework

data source ▪ 375