

Oracle® Load Testing

Load Testing User's Guide

Release 13.2.0.1

E15484-19

October 2017

Oracle Load Testing Load Testing User's Guide, Release 13.2.0.1

E15484-19

Copyright © 1997, 2017, Oracle and/or its affiliates. All rights reserved.

Primary Author: Rick Santos

Contributing Author: Mary Anna Brown

Contributor:

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface	xi
Audience	xi
Documentation Accessibility	xii
Related Documents	xii
Conventions	xiii
1 Introduction	
1.1 Feature Highlights	1-1
1.2 Advantages of Oracle Load Testing	1-2
1.2.1 Web-Based Interface	1-2
1.2.2 Infrastructure Performance Monitors	1-2
1.2.3 Oracle Load Testing Helps You Ensure Application Performance	1-2
2 Planning for Load Testing	
2.1 Goals of Scalability Testing	2-1
2.2 Phases of Scalability Testing	2-2
2.3 Criteria for Accurate Scalability Testing	2-2
2.4 Determine Additional Tools Required to Perform Testing and Diagnosis	2-3
2.5 Determining the Hardware Needed to Execute the Tests	2-4
2.6 Who Should be Responsible for Load Testing?	2-5
2.7 What to Avoid When Testing for Scalability	2-5
2.8 Performing Scalability Testing	2-5
2.8.1 Define the Process	2-6
2.8.2 Define the Criteria	2-7
2.8.3 Planning the Scalability Tests	2-8
2.8.4 Planning the Load Test Scenarios	2-8
2.8.5 Create and Verify the Test Scripts	2-9
2.8.6 Create and Verify the Load Test Scenarios	2-9
2.8.7 Execute the Tests	2-10
2.8.8 Evaluate the Results	2-11
2.8.9 Generate Analysis Reports	2-11
2.9 Summary	2-11
3 Basics	
3.1 Installing Oracle Load Testing	3-1

3.1.1	Installing Oracle Load Testing.....	3-1
3.1.2	Preconditions for Using Functional Testing Scripts	3-2
3.1.3	Popup Blockers	3-3
3.1.4	Port Configuration.....	3-3
3.1.5	Installing Oracle Load Testing Agent.....	3-4
3.1.5.1	Verify Network Access to Agent Systems	3-5
3.1.5.2	Configuring Oracle Load Testing Agents	3-5
3.2	Installing the Linux Agent.....	3-6
3.3	Moving an Existing Installation to a New Machine	3-6
3.3.1	Transfer Database Schemas.....	3-7
3.3.2	Install Oracle Application Testing Suite and Configure Database Connections.....	3-7
3.3.3	Copy Repositories and Files	3-9
3.3.4	Restart the Service and Verify the Setup	3-9
3.4	Adding Repositories.....	3-9
3.5	Changing the Web Server Port.....	3-10
3.6	Using SSL	3-11
3.7	Changing the OLT Controller Heap Settings.....	3-11
3.7.1	Basic Guidelines for the Controller Heap Settings	3-12
3.7.2	Modifying the JVM Heap Settings On Windows Machines.....	3-12
3.7.3	Modifying the JVM Heap Settings On Linux Machines	3-12
3.7.4	Limitations.....	3-13
3.8	Oracle Application Testing Suite Tools Menu.....	3-13
3.9	Administrator	3-14
3.9.1	Menu Options.....	3-14
3.9.1.1	Help Menu.....	3-14
3.9.1.2	Logout	3-14
3.9.2	Users Tab.....	3-15
3.9.3	Usage Audit Tab	3-15
3.9.4	Adding Users.....	3-15
3.9.5	Editing Users	3-16
3.9.6	Deleting Users	3-16
3.9.7	Restoring Users	3-16
3.9.8	Auditing Usage	3-17
3.10	Main Window Features.....	3-17
3.10.1	Overview of the Menu Options	3-17
3.10.2	Toolbar	3-17
3.11	Composer Tab	3-17
3.11.1	Profiles Section	3-18
3.11.1.1	Scripts Panel	3-18
3.11.1.2	Profile Panel	3-18
3.11.1.3	Properties Panel	3-20
3.11.2	Ramp Section.....	3-20
3.11.3	Goals Section	3-20
3.11.4	Monitoring Section	3-21
3.12	Active Session Tab	3-22
3.13	Reports Tab	3-22
3.14	Schedule Tab.....	3-22

4 Managing Systems

4.1	Systems Manager	4-1
4.2	Defining Systems	4-2
4.2.1	Adding New VU Agent Systems.....	4-2
4.2.2	Adding New System Groups.....	4-3
4.2.3	Adding Systems to Groups	4-4
4.2.4	Adding New ServerStats Data Collectors	4-4
4.2.5	Adding New Monitored Systems.....	4-5
4.2.5.1	Configuring JMX Monitors	4-5
4.2.5.2	Adding a Monitored System.....	4-7
4.2.6	Renaming Systems.....	4-10
4.2.7	Editing Systems.....	4-11
4.2.8	Deleting Systems.....	4-11
4.3	Estimating Hardware	4-11
4.3.1	Estimate for an Oracle Load Testing Scenario.....	4-11

5 Creating Scenario Profiles

5.1	Adding Profiles to a Scenario.....	5-1
5.1.1	Adding Profiles to a Scenario.....	5-1
5.1.2	Removing Profiles From a Scenario	5-2
5.1.3	Specifying Scenario Profile Properties.....	5-2
5.1.3.1	Load Test Script Settings	5-3
5.1.3.2	Functional Test Script Settings	5-12
5.1.4	Determining the Number of Virtual Users	5-18
5.1.5	Using the Data Bank Control	5-19
5.1.6	Using Synchronization Points	5-21
5.1.7	Using IP Spoofing	5-22
5.2	Importing and Exporting Script Files	5-23

6 Defining VU Rampup

6.1	Specifying Virtual User Rampup.....	6-1
6.1.1	Specifying How Many Virtual Users to Start at a Time.....	6-1
6.1.1.1	Examples	6-2
6.1.2	Using the Advanced Scenario Ramp Options	6-2

7 Setting Test Goals

7.1	Setting Test Goals.....	7-1
-----	-------------------------	-----

8 Defining ServerStats Monitors

8.1	About ServerStats	8-1
8.2	Creating Monitors Overview	8-2
8.3	Using Configuration Files.....	8-2
8.3.1	Creating a New Configuration File.....	8-3
8.3.2	Editing Configurations	8-3
8.3.3	Deleting Configurations	8-4

8.4	Adding Data Source Monitors	8-4
8.4.1	Configuring for Monitoring Disk Performance	8-5
8.4.2	Adding Monitors	8-5
8.4.3	ComPlus Metric Properties	8-6
8.4.4	Database Metric Properties	8-6
8.4.5	JMX Metric Properties.....	8-6
8.4.6	Perfmon (Windows Performance Monitor) Metric Properties	8-6
8.4.7	Ping Metric Properties	8-7
8.4.8	URL Metric Properties	8-7
8.4.9	Virtual Agent Metric Properties	8-8
8.4.10	IBM Websphere PMI Metric Properties	8-9
8.5	Using Metric Profiles to Create Monitors.....	8-9
8.6	Using Metrics to Create Monitors	8-9
8.7	Editing Monitors In a Configuration	8-9
8.8	Working With Metric Profiles.....	8-10
8.8.1	Adding Metric Profiles.....	8-10
8.8.2	Editing Metric Profiles	8-11
8.8.3	Cloning Metric Profiles	8-11
8.8.4	Deleting Metric Profiles	8-12
8.8.5	Testing Metric Profiles	8-12
8.9	Working With Metrics.....	8-12
8.9.1	Adding Metrics	8-12
8.9.1.1	Adding ComPlus Metrics.....	8-13
8.9.1.2	Adding Database Metrics.....	8-13
8.9.1.3	Adding JMX Metrics	8-14
8.9.1.4	Adding Perfmon (Windows Performance Monitor) Metrics.....	8-14
8.9.1.5	Adding Ping Metrics.....	8-15
8.9.1.6	Adding URL Metrics.....	8-15
8.9.1.7	Adding Virtual Agent Metrics.....	8-17
8.9.1.8	Adding IBM Websphere PMI Metrics.....	8-18
8.9.2	Editing Metrics.....	8-18
8.9.3	Cloning Metrics.....	8-18
8.9.4	Deleting Metrics.....	8-19
8.9.5	Testing Metrics.....	8-19
8.9.6	Viewing Metrics	8-19
8.10	Using Diskperf.....	8-19

9 Running and Saving Scenarios

9.1	Working with Scenario Files	9-1
9.1.1	Saving Scenarios	9-1
9.1.2	Opening Existing Scenarios.....	9-2
9.1.3	Renaming Scenarios	9-2
9.1.4	Deleting Scenarios	9-2
9.2	Running Scenario Sessions	9-3
9.2.1	Start Scenario Session Immediately	9-3
9.2.2	Schedule without Starting the Scenario Session	9-3
9.3	Manually Starting and Stopping Sessions.....	9-3

9.3.1	Starting the Session.....	9-3
9.3.2	Stopping Ramping New Virtual Users.....	9-4
9.3.3	Stopping All Virtual Users	9-4
9.3.4	Managing Sessions	9-4
9.4	Running Scenarios from the Command Line	9-5

10 Viewing Active Sessions

10.1	Active Session Tab Overview.....	10-1
10.1.1	Session Menu.....	10-1
10.2	Active Session Overview	10-2
10.3	Virtual User Status.....	10-2
10.3.1	Overview of the Virtual Users Grid.....	10-2
10.3.2	Controlling Individual Virtual Users.....	10-3
10.3.3	Modifying Run Attributes.....	10-3
10.4	Viewing Runtime Graphs	10-4
10.4.1	Using Default Runtime Graphs	10-4
10.4.1.1	Viewing Runtime Graphs.....	10-4
10.4.1.2	Performance Vs. Users Graph.....	10-5
10.4.1.3	Errors Vs. Users Graph.....	10-5
10.4.1.4	Performance Vs. Time Graph	10-6
10.4.1.5	Errors Vs. Time Graph.....	10-6
10.4.1.6	Users Vs. Time Graph.....	10-6
10.4.1.7	Users Vs. Time Graph	10-6
10.4.2	Custom Graphs	10-6
10.5	Viewing Virtual User Logs	10-9
10.6	Viewing Sync Point Status.....	10-9
10.6.1	Releasing Synchronization Points	10-10
10.6.2	Releasing Individual Virtual Users.....	10-10
10.7	Attaching to a Session	10-10
10.8	Detaching from a Session.....	10-10

11 Using Graphs and Reports

11.1	Reports Tab Overview	11-1
11.2	Session Overview	11-1
11.3	Graphs	11-3
11.4	Virtual User Logs	11-6
11.5	Setting the Reporting Options.....	11-6
11.6	Creating Reports	11-7
11.7	Exporting Graphs and Data.....	11-8
11.7.1	Exporting CSV Data	11-8
11.7.2	Exporting for Excel	11-9
11.7.3	Exporting to PNG	11-9
11.8	Working With Graph Queries.....	11-9
11.8.1	Saving Graph Queries.....	11-9
11.8.2	Running a Saved Query.....	11-10
11.8.3	Editing Queries	11-10

11.8.4	Deleting Queries	11-10
11.9	Using Time Range Filters.....	11-10
11.10	Using Raw Data.....	11-11
11.10.1	Enabling Raw Data Collection.....	11-11
11.10.2	Viewing Raw Data in Reports.....	11-11
11.10.3	Raw Data Database Tables and Counters	11-12
11.10.4	Example Database Queries for Raw Data	11-13

12 Working With Sessions

12.1	Managing Sessions.....	12-1
12.2	Exporting Session Data	12-2
12.2.1	File Format.....	12-2
12.3	Importing Session Data	12-3

13 Managing Files and Databases

13.1	Database Management.....	13-1
13.1.1	Adding an Oracle Database	13-1
13.1.2	Selecting the Database to Use	13-2
13.1.3	Removing Databases	13-2
13.1.4	Updating Databases	13-3
13.1.5	Exporting and Importing Databases.....	13-3
13.1.5.1	Exporting Databases Using Export Utility.....	13-3
13.1.5.2	Importing Databases Using Import Utility.....	13-4
13.1.5.3	Importing Databases Using Data Pump Import Utility.....	13-4
13.2	Importing and Exporting Data Bank Files	13-5
13.3	Importing and Exporting Data Files	13-5
13.4	Using the Reset Password Utility	13-6

14 Setting Options

14.1	Setting General Options.....	14-1
14.2	Setting System Options	14-1
14.3	Setting Manage Options.....	14-1
14.3.1	Repository Options.....	14-1
14.3.2	Session Options.....	14-2
14.3.3	Scenario Options.....	14-2
14.3.4	Graph Options.....	14-2
14.3.5	Scripts Options.....	14-2
14.3.6	Data Bank Options.....	14-2
14.3.7	Data File Options	14-3
14.4	Setting Profiles Options	14-3
14.4.1	Setting Browser Emulation Options	14-3
14.4.2	Setting Scenario Default Properties	14-3
14.4.3	Setting Ramp Options	14-14
14.5	Setting ServerStats Options	14-14
14.6	Setting Session Options.....	14-14
14.6.1	Setting Session Start and Stop Options.....	14-14

A Internet Status Codes

A.1	Server Status Codes	A-1
A.1.1	Information Codes	A-1
A.1.2	Success Codes	A-1
A.1.3	Redirection Codes	A-2
A.1.4	Failure Codes	A-2
A.1.5	Server Error Codes	A-4
A.2	Internet Codes	A-4
A.2.1	Internet Error Codes	A-4
A.2.2	FTP API Error Codes	A-8
A.2.3	Gopher API Error Codes	A-8
A.2.4	HTTP API Error Codes	A-9
A.2.5	Additional Information	A-10

B Error Messages

B.1	Viewing Error Codes	B-1
B.2	Error Messages	B-1
B.3	RSWISERV Errors	B-3
B.4	Obtaining Support Files	B-4
B.5	Oracle Application Testing Suite Application Service Messages	B-4
B.5.1	Agents are Orphaned	B-4
B.5.2	Server is Not Initialized	B-4
B.5.3	Unable to Start Session	B-5
B.6	Problems During Load Testing	B-6
B.6.1	Cannot Open Scenario	B-6
B.6.2	Unable to Start Agent Session	B-6
B.6.3	Performance Problems During the Load Test	B-7
B.6.4	Unexpected Agent Exit in the VU Grid	B-8
B.7	Data Collector Startup Errors	B-9

C Command Line Reference

C.1	Specifying Command Line Settings	C-1
C.1.1	Error Handling	C-2
C.2	Supported Command Line Settings	C-2
C.2.1	Commands	C-2
C.2.2	General Settings	C-5
C.2.3	AddAgent Settings	C-5
C.2.4	AddAgentGroup Settings	C-6
C.2.5	Export and Import Settings	C-6
C.2.6	Report Settings	C-7
C.2.7	Run Settings	C-7
C.2.8	Status Settings	C-11
C.2.9	Stop Settings	C-12

D ServerStats Metric Descriptions

D.1	AIX 4.3 Virtual Agent.....	D-1
D.2	Apache Web Server	D-1
D.3	Cisco Network Devices	D-1
D.4	Foundry Load Balancer.....	D-2
D.5	General Web Server	D-2
D.6	Host.....	D-2
D.7	Host Resources MIB.....	D-2
D.8	IIS Web Server	D-3
D.9	Linux Virtual Agent	D-3
D.10	Memory	D-3
D.11	MIB-II Network Devices	D-4
D.12	Netscreen Firewall	D-4
D.13	Network.....	D-5
D.14	Oracle Database.....	D-5
D.15	Solaris Virtual Agent	D-6
D.16	Solaris (multi-CPU).....	D-6
D.17	SQL Server Databases.....	D-6
D.18	WebLogic.....	D-7

Preface

Welcome to the Oracle Load Testing User's Guide. This guide explains how to get started using the features and options of Oracle Load Testing for load testing of Web-based applications.

Audience

This guide is intended for Web test engineers who will be conducting load and scalability testing of a Web site or application. The guide does require an understanding of software or Web application testing concepts. Test engineers using Oracle Load Testing should be familiar with Oracle OpenScript scripts and the concepts of Web application testing, load testing, scalability testing, and reporting.

Prerequisites

Oracle Load Testing does not require any programming experience to develop load scenarios. However, test engineers performing distributed load testing in a network environment should be familiar with networking concepts and how to verify communication between workstations in a network.

Using This Guide

This guide is organized as follows:

[Chapter 1, "Introduction"](#) provides an overview of the major features of Oracle Load Testing.

[Chapter 2, "Planning for Load Testing"](#) provides a basic methodology for testing scalability and performance of Web applications throughout the life cycle.

[Chapter 3, "Basics"](#) provides descriptions of the Oracle Load Testing main window features and menu options.

[Chapter 4, "Managing Systems"](#) explains how to define the systems that will be used to run Virtual Users for creating load on the application-under-test.

[Chapter 5, "Creating Scenario Profiles"](#) explains how to define scenario profiles based upon OpenScript scripts and Oracle Load testing profile properties.

[Chapter 6, "Defining VU Rampup"](#) explains how to specify the Virtual User ramp rate for the current scenario.

[Chapter 7, "Setting Test Goals"](#) explains how to test goals for the current scenario.

[Chapter 8, "Defining ServerStats Monitors"](#) explains how to use the ServerStats features of Oracle Load Testing to monitor systems and metrics during a load test of the application-under-test.

[Chapter 9, "Running and Saving Scenarios"](#) explains how to use the Ramp settings to specify start and stop times for scenario profiles and the rampup specifications for each virtual user.

[Chapter 10, "Viewing Active Sessions"](#) explains how to view runtime session data and graphs for currently running scenarios.

[Chapter 11, "Using Graphs and Reports"](#) explains how to view real-time performance, statistics, and error information in graphs and reports.

[Chapter 12, "Working With Sessions"](#) explains how to attach and detach from running sessions.

[Chapter 13, "Managing Files and Databases"](#) explains how to manage databases for use in Oracle Load Testing and how to import and export files to and from the Oracle Load Testing Server.

[Chapter 14, "Setting Options"](#) explains how to specify default options used by Oracle Load Testing.

[Appendix A, "Internet Status Codes"](#) lists the status codes and Internet error codes that can be returned from a server.

[Appendix B, "Error Messages"](#) provides a reference for error messages and codes generated by Oracle Load Testing.

[Appendix C, "Command Line Reference"](#) provides a reference for the parameters for running Oracle Load Testing scenarios from the command line.

[Appendix D, "ServerStats Metric Descriptions"](#) describes the default Oracle Load Testing ServerStats metrics. Use the Perfmon data source to add metrics that are not listed.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Related Documents

For more information, see the following documents in the Oracle Application Testing Suite documentation set:

- *Oracle Application Testing Suite Release Notes*
- *Oracle Application Testing Suite Installation Guide*
- *Oracle Application Testing Suite Security Guide*
- *Oracle Application Testing Suite Getting Started Guide*
- *Oracle Functional Testing OpenScript User's Guide*
- *Oracle Functional Testing OpenScript Programmer's Reference*

- *Oracle Load Testing Load Testing User's Guide*
- *Oracle Test Manager Test Manager User's Guide*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Introduction

Oracle Load Testing allows you to easily and accurately test the performance and scalability of your Web applications, Oracle packaged applications and Web services. Oracle Load Testing not only stresses your application to simulate the impact of end-user workloads, but also enables rigorous validation that protocol-based legacy client server testing tools cannot provide. Its integrated scripting platform cuts scripting time in half, eliminating weeks from a project's testing schedule. Oracle Load Testing is a component of Oracle Application Testing Suite, the centerpiece of the Oracle Enterprise Manager solution for comprehensive testing of packaged, Web and service-oriented architecture-based applications.

Oracle Load Testing is the easiest way to validate the performance and scalability of your Web applications, Oracle packaged applications and Web services. It can simulate thousands of virtual users accessing the application simultaneously and measures the effect of the load on application performance, without requiring a substantial hardware investment.

The realistic usage scenarios that you can configure in Oracle Load Testing can handle even the most complex applications. By enabling virtual users to simulate many different end user parameters (including configurable browser types, connection speeds, and think times), testers can stress their Web application just like real users will to understand exactly how the application will scale under peak load conditions. Oracle Load Testing's virtual users can generate multithreaded browser requests while performing rigorous functional validation under load conditions—validation that protocol-based, legacy client server testing tools cannot provide.

Oracle Load Testing can also be used to test the performance of Web service interfaces by simulating thousands of concurrent clients accessing SOA-based applications, through its integrated load testing accelerator for Web Services. In addition, integrated load testing accelerators for Oracle E-Business Suite, Siebel and ADF applications provide more efficient and optimized performance testing for your Oracle applications.

1.1 Feature Highlights

Oracle Load Testing offers the following advantages for load testing Web-based applications:

- Automates testing of the most complex Web applications, Oracle packaged applications and Web services with robust test scripts.
- Simulates hundreds to tens of thousands of users while minimizing the test hardware required.

- Gathers critical application and infrastructure performance metrics to identify bottlenecks.
- Simplifies accessibility with an intuitive Web based user interface.
- Allows distributed users to share testing results during live testing.
- Scalable enterprise architecture built on WebLogic Server and Oracle Database.

1.2 Advantages of Oracle Load Testing

Oracle Load Testing offers several advantages over current load testing tools for Web-based applications, as follows:

- Maximizes application performance by allowing developers to test and tune the application under peak load conditions.
- Improves application response times by quickly identifying and addressing bottlenecks.
- Pinpoints hard-to-find bottlenecks in the back-end application infrastructure.
- Reduces testing time by enabling viewing and sharing of real-time test results via the Web.

1.2.1 Web-Based Interface

Oracle Load Testing is deployed on the Oracle WebLogic Server and its intuitive Web-based interface allows you to configure your load test scenarios and provides graphs and reports that enable testers to analyze application performance during a load test. From any Web browser, users can easily configure load tests, set up server monitors, run tests, and view real-time and post-run results. Oracle Load Testing helps distributed teams reduce the time and complexity of live load tests by enabling collaborative testing, so distributed users can view and analyze results from the same running load test in their own browser. It promotes team interactions for more productive analysis, diagnostics, and tuning during the entire live test process.

1.2.2 Infrastructure Performance Monitors

Oracle Load Testing also offers a comprehensive set of infrastructure performance monitors that record in-depth performance metrics of Web servers, application servers, databases, and other infrastructure components during the load test. When this information is combined with the performance results gathered by the virtual users, developers have the information needed to analyze and ensure optimal application performance during and after test execution.

Oracle Load Testing allows you to create customized post-run reports on the results of your load tests. These historical reports let you compare the results of multiple load test sessions and correlate virtual user response times with performance statistics collected from the various tiers of the application infrastructure, to identify and diagnose performance bottlenecks.

1.2.3 Oracle Load Testing Helps You Ensure Application Performance

Oracle Load Testing enables you to make critical decisions about system architecture, tuning, and hosting alternatives. It leverages Application Testing Suite's OpenScript integrated scripting platform for creating load test scripts that automate complex business transaction. This integrated scripting platform provides a unique combination of ease-of-use and flexibility through its intuitive graphical scripting

interface and powerful Java IDE for extending scripts at the code level. It also provides custom capabilities for testing SOA and Oracle packaged applications through its integrated testing accelerators. Load test scripts can also be generated in OpenScript from Oracle's Real User Experience Insight (RUEI) product and these scripts are based on actual live user sessions with the application captured by RUEI.

Oracle Load Testing pinpoints bottlenecks that could limit performance and cause application slow-downs. It provides a fully Web-based user interface for configuring and running load tests and integrated performance diagnostics for monitoring application infrastructure during a load test to identify bottlenecks. Oracle Load Testing also enables multi-user collaboration by allowing testers to view and share real-time results during load test execution through their browser. With the ability to tune your application under peak load conditions prior to deployment, you can ensure the health of critical business applications that drive your revenue.

Planning for Load Testing

This chapter provides a basic methodology for testing the scalability and performance of Web applications throughout the life cycle. It outlines the process for selecting the appropriate tools and the recommended steps to perform effective scalability testing. This chapter is broadly divided into the following sections:

- **Goals and Requirements of Scalability Testing:** What should you aim to accomplish as a result of scalability testing for each phase of your Web application development.
- **Methodology:** The process and the steps that are required to ensure performance and scalability throughout the application life cycle.
- **Test Planning and Execution:** How you should plan and execute scalability testing during each phase of development.

2.1 Goals of Scalability Testing

The primary goals of a load test are as follows:

1. Determine the user limit for the Web application.
 - The user limit is the maximum number of concurrent users that the system can support while remaining stable and providing reasonable response time to users as they perform a variety of typical business transactions.
 - The user limit should be higher than the required number of concurrent users that the application must support when it is deployed.
2. Determine client-side degradation and end user experience under load.
 - Can users get to the Web application in a timely manner?
 - Are users able to conduct business or perform a transaction within an acceptable time?
 - How does the time of day, number of concurrent users, transactions and usage affect the performance of the Web application?
 - Is the degradation "graceful?" Under heavy loading conditions, does the application behave correctly in "slow motion," or do components crash or send erroneous/incomplete pages to the client?
 - What is the failure rate that users observe? Is it within acceptable limits? Under heavy loading conditions do most users continue to complete their business transactions or do a large number of users receive error messages?
3. Determine server-side robustness and degradation.

- Does my Web server crash under heavy load?
- Does my application server crash under heavy load?
- Do other middle-tier servers crash or slow down under heavy load?
- Does my database server crash under heavy load?
- Does my system load require balancing, or if a load balancing system is in place, is it functioning correctly?
- Can my current architecture be fine-tuned to extract better performance?
- Should hardware changes be made for improved performance?
- Are there any resource deadlocks in my system?

2.2 Phases of Scalability Testing

The following are the different phases of load and scalability testing for a Web application:

Architecture Validation - tests the scalability of the architecture early in the development of a Web application, presumably after a prototype of the application has been created that can generate transactions to touch all tiers of the application. This allows the engineering organization to determine the viability of the architectural framework selected to build the Web application.

Performance Benchmarking - sets and creates the benchmark tests for the initial version of the application for all business transactions and gives the engineering and the quality assurance groups a set of metrics to quantify the scalability of the application. Based on the requirements specified, the development group will either maintain this scalability or improve upon it through the subsequent milestones.

Performance Regression - is the phase where the Web application is tested with the established benchmarks to ensure that the changes made to the application do not result in degradation of scalability. These tests are executed when key milestones have been reached or architectural modifications have been made during the development of the application. It is also common that the benchmark tests and the metrics originally set for the application be replaced or augmented with additional tests and newer metrics to reflect the improvements made to the application.

Acceptance and Scalability Fine Tuning - is the final load testing phase prior to the official launch of the Web application where all the different pieces of the Web application - including the hardware, load balancing components and all software components - are integrated and the scalability is validated. Different scenarios of real-life usage are emulated and the scalability of the final configuration is validated. These different scenarios are also used to configure the hardware and software components to yield optimal performance.

24x7 Performance Monitoring - after the application is deployed, it is essential to monitor the performance of the system under the real load generated by actual users so that crashes or slow-downs can be spotted before they become problematic. In this phase, data pertaining to real life usage can be collected to help refine future scalability tests for accurate emulation of load.

2.3 Criteria for Accurate Scalability Testing

In order to emulate a realistic load that will correlate with real-life usage of the application, a load-testing tool must:

- produce load that stresses all tiers of a multi-tier application;
- allow for the simulation of a realistic mix of different groups performing different types of business activities on the site during peak periods;
- emulate page and resource request patterns produced by popular browsers such as Internet Explorer and Firefox;
- validate the responses coming back from the Web server for each of the thousands of concurrent users to ensure that the correct pages are being returned by the Web application under stress;
- allow for easy maintenance of the scripts as the application changes so that scalability can be re-verified each time that the system is changed.

In addition, the following criteria are also important:

- Dynamic Dial-up of users - this capability allows you to add new users to the load test without stopping the current test. For example, if you are running a 100 user load test and you will dynamically add another 100 users, you don't have to stop the load test and restart a new test with 200 users;
- Real-Time Virtual User Debugger - the load testing tool should have some capability to allow you to visually monitor the progress of a user at any given point in time when the load test is in progress;
- Real-Time Graphs that allow you to understand the scalability characteristics of the application as the load test is in progress;
- allow for distributing load tests from a number of machines on the LAN/WAN with a central point of control;
- allow the load tests to be executed with recorded think times, random think times (following some kind of statistical distribution), and with no think times;
- measure response times of entire business transactions in addition to individual objects on pages such as sub-frames and images;
- allow for simulation of different types of caching behaviors;
- run data-driven tests to allow for unique concurrent users on the system;
- allow for complex scheduling to allow for different scenarios of starting, stopping, and ramp-up;
- provide reports and a performance database to allow for post-run analysis and comparison with previously established benchmarks.

2.4 Determine Additional Tools Required to Perform Testing and Diagnosis

Scalability testing requires several types of software tools for the various levels of testing and reporting/analysis of results. Before you can perform the testing, you need to familiarize yourself with the following software tools that will be used:

Oracle OpenScript - used to create the various scripts that the virtual users will run. The scripts specify the actual steps used in performing the business transactions in the application. Thorough testing of an application may require many different scripts that exercise different areas (i.e. the business transactions). scripts are easily maintained over time as the application changes.

Oracle Load Testing - used to define virtual user profiles and scenarios and to perform the load test. Oracle Load Testing runs the scripts as multiple virtual users that test the

scalability of the application. Oracle Load Testing defines the number and types of virtual users and which script(s) different virtual users run. It also provides real-time reports and graphs for evaluating the progress of the tests and post-analysis reports and graphs for post-run analysis of test results.

Oracle Load Testing ServerStats - provides real time system monitoring for a variety of data sources for monitoring the impact of the load test on the individual servers (Web server, database server, application server, and system counters). For example, the load test many require monitoring software for a Web server, an application server, and a mainframe data base server.

In addition to Oracle Application Testing Suite, you may need additional software tools for other specialized monitoring or reporting.

Other System Monitoring Tools - You should determine what other software tools may be necessary for monitoring the load test.

Logging Tools - You should determine what software tools will be used for logging transaction and performance data or logging errors.

In addition, you will also need to gather data from other tools that monitor the state of different components in the application architecture. This will allow you to correlate the client-side degradation noticed during the scalability tests to one or more scalability problems with specific components of the application.

2.5 Determining the Hardware Needed to Execute the Tests

To execute a scalability test effectively, the appropriate hardware needed to run the test tools must be procured and configured.

In order to generate the load on the Web application using thousands of concurrent users you must consider the following:

Load Distribution Capability - Does the load test tool allow for load to be generated from multiple machines and controlled from a central point?

Operating System - What operating systems does the load test master and the load generation agents run under?

Processor - What type of CPU is required for the master and virtual user agent?

Memory - How much memory is required for the master and virtual user agent?

To insure that you have the appropriate hardware to execute scalability tests, ask your load testing tool vendor to provide you with the hardware requirements for the load test master and agent machines.

General Rules of Thumb:

- Windows NT 4.0/2000/2003 are better suited to run load test virtual users than Windows 98/XP, as they are more scalable and more stable operating systems.
- If the CPU utilization of any workstation in the load test - be it the load master or the load agents running concurrent users - is higher than 70-80%, or the memory consumption is over 85%, the processes running on that workstation will experience operating system resource conflicts and the performance results from that test station will be skewed.
- You should consider running the load master on a separate machine if possible. The machine that serves as the master typically needs a high performance CPU. The virtual users can run on one or more machines. These machines need to have sufficient memory to run a large number of virtual users.

- To determine the number of virtual users that can be run on a machine, you can estimate based on the amount of memory each virtual user would consume. If the virtual users are running as threads within a process, on an average they consume 300-500 KB of memory. If the virtual users are running as separate processes within a process, on an average they consume 1024 - 2048 KB of memory.
- A hardware configuration document is usually available from each test vendor that explains the hardware requirements for a particular load testing setup.

2.6 Who Should be Responsible for Load Testing?

The following groups should have active participation in the load test:

Development Engineers and Architecture Groups - design and perform architecture validation tests and benchmarking criteria for the Web application. They work with the quality assurance groups to fine-tune the application and deployment architecture to perform optimally under load.

In some large software organizations, dedicated performance architecture groups exist that are charged with building and maintaining scalable frameworks.

Quality Assurance Organizations - design and execute development tests to verify correct operation of the application and acceptable performance.

Integration and Acceptance Organizations - design and perform integration tests that ensure all tiers and hardware operates together correctly before acceptance and deployment of the application. In most organizations, the quality assurance groups are charged with this responsibility as well.

Monitoring and Operations Groups - design and perform monitoring tests to ensure that the deployed application is available 24x7 and is not degrading under conditions of load or over long periods of regular usage.

2.7 What to Avoid When Testing for Scalability

Organizations performing scalability testing should avoid common pitfalls that guarantee incorrect results and potential failure. These include the following:

- performing load tests on applications that are changing even as the tests are being performed;
- performing load tests with applications that are not functionally tested so that even basic capabilities are not operational;
- performing load tests on certain parts of the application that work and extrapolating the results to the entire application;
- performing load tests with a smaller number of concurrent users and extrapolating the result for larger numbers.

2.8 Performing Scalability Testing

The general process for performing scalability testing on a Web application is as follows:

1. Define a process that is repeatable for executing scalability tests throughout the application life-cycle.
2. Define the criteria for scalability.
3. Determine the software tools required to run the load test.

4. Determine and configure the hardware and environment needed to execute the scalability tests.
5. Plan the scalability tests.
6. Plan the test scenarios.
7. Create and verify the scripts.
8. Create and verify the load test scenarios.
9. Execute the tests.
10. Evaluate the results against the defined criteria.
11. Generate required reports.

The details for the above steps are explained in the following sections.

2.8.1 Define the Process

Once the requirements for a load testing effort are defined, test planners need to define the process. In defining the process, test planners should consider the following issues and questions:

Required Applications - What application(s) will the load testing be performed against?

Scheduling - When will the testing be performed? What are the dates, times, build availability, and testing milestones that need to be met?

Personnel - Who will perform the analysis, planning, test development, test execution, and evaluation? Which internal department personnel (for example, business analysts, network specialists, quality assurance engineers, and developers) will be involved? Will any third-party personnel (for example, tools vendor, Internet Service Provider, or testing lab) be required?

Location - Where will the testing be performed? Will testing be performed internally or at an external location such as at an Internet Service Provider or testing lab?

Testing Environment- What SW/HW environment will the load tests be run against? When specifying the testing environment, you should look for and avoid the following common pitfalls:

- **Application stability** - Make sure that the application is not being changed even as the load test is being undertaken. Quite frequently the entire application or parts of it are changed even as the application is being load tested.
- **Deployment environment** - Make sure that the environment under which the application is operating when the load test is being performed is very close to the real deployment environment, if not exactly the same. For example, if your requirement states that the load test has to be performed against an HTTPS server that is configured with enough horsepower to sustain heavy loads, you should not run it against a smaller server that is used by the development group.
- **Acceptance environment** - As part of the acceptance tests that are run prior to shipping the product, you must make sure that the environment used to perform the load test is exactly the same as the live production environment (as defined in a specification document).

Hardware Allocation - Is the required hardware (network, master load-test computer, agent computers, etc.) allocated and available for use? Testing vendors should be able to help determine the necessary hardware based on some of the following information:

- number of virtual users or the desired throughput for the application as a whole (Transactions per second);
- maximum or acceptable duration for each business transaction;
- maximum or acceptable duration for delay between business transactions.

2.8.2 Define the Criteria

Before you can begin planning for load testing, you need to define the criteria that indicate whether or not the application will be accepted and ready for live deployment. When defining the criteria, you should specify the following:

Load to be Simulated - What number of virtual users need to be emulated? This indicates the number of concurrent users on the Web server.

Number of Business Transactions to Simulate - How many business transactions are to be simulated for the load test? This is determined by the analysis of the application during requirements planning and may be specified as transactions-per-second (TPS), transactions-per-hour, or simultaneous user sessions.

Types of Business Transactions to be Simulated - What are the business transactions that need to be simulated (for example, read an account balance, make an account transaction, check account details, check contributions, etc.)?

Criteria for Each Business Transactions - For each business transaction you should determine the following:

- **Acceptable response time under various loads** - What is an acceptable response time under various conditions of load. For example, what is the acceptable response time when running 100 virtual users? 200 virtual users? Also, what is the acceptable response time when running the maximum limit of virtual users.
- **Acceptable failure rate** - What is the acceptable failure rate for all of the transactions and for each business transactions when under load? For example, zero failures allowed for up to 100 virtual users, 5% failures for 200 virtual users, etc.
- **Categories of users** - What are the categories of users to simulate in the various transactions? Are they first time users or are they repeat users? First time users have a higher overhead on the Web server since all the images must be downloaded. You can design and develop tests for both types of users and run combinations of load tests under differing conditions.
- **SSL and HTTP** - Does testing require a combination of SSL and plain HTTP, only SSL, or only HTTP?
- **Browsers to simulate** - What browsers will be simulated in the load test? Will testing simulate Internet Explorer or Firefox (or both)?
- **Pacing mode** - What is the virtual user pacing that will be used for the load test? Will the testing be performed using recorded "think times" (that is, running with delays between pages that correspond to the same natural pauses that occurred while recording the script)? Should you try a worst-case stress test with no delays between pages? Or alternatively, should you use a random distribution of delays representing a range of user-speeds from expert users on T3 connections to novice users on slow modems?
- **Delay Between Business Transaction Runs** - What delay time will be included between business transaction tests, if any?

- **With or without images** - Will virtual users run with images or without images? Images constitute an additional load on the Web server. In many cases, you may want to perform load testing both with and without images for comparison.

Overall Transactions-Per-Second Throughput Required - What is the overall transactions per second (TPS) throughput required for the load test? This can be computed based on the number of simultaneous business transactions and the duration of typical transactions.

Type of Error Handling - What type of error handling is required when executing the load test? Does the load test need to be stopped on encountering certain types of error or just log the error and continue? What types of error logging do we need to enable for each concurrent user and for the different components in the application architecture?

Type of Transaction and Performance Data Logging - What type of transaction and performance data needs to be logged for the various scripts?

2.8.3 Planning the Scalability Tests

Developing detailed test plans before you actually create the tests is an important step in making sure the tests conform to the business analysis of the application and the defined criteria.

For each test that will perform a business transaction you need to plan and define the following information:

Steps for Scripts - Each script should have a detailed sequence of steps that define the exact actions a user would perform. Multiple scripts can be used. For example, you can define a specific script that performs user login, several scripts that perform specific business transactions, and another script that logs users off. For each script, you should define the expected results. Oracle OpenScript lets you quickly and easily record scripts that emulate a user's actions.

Run-Time Data - The test plan should specify any run-time data that is needed to interact with the application, for example, login user IDs, passwords, and other run-time data specific to the application.

Data Driven Tests - If the scripts require varying data at run-time, you'll need to have an understanding of all the fields that require this data. You also need to define the data sources and any tool(s) needed to either create fictitious data or extract real data from existing databases.

Oracle OpenScript Data Bank Wizard lets you specify and connect external data sources to the scripts.

2.8.4 Planning the Load Test Scenarios

In addition to the business transaction details for each script, the test plan should also specify the different user groups and test scenarios that will be required for load testing. For each test scenario you need to plan and define the following information:

Type of User - Is this user a first-time user of the application or a repeat user? This is important if the application responds differently for a first-time user than it does for a repeat user and places more stress on the server. Oracle Load Testing scenarios can specify either a first-time user or a repeat user.

Transactions to Perform - Which business transaction(s) will this user perform? In what sequence? If the application requires a first-time user to perform some type of

registration, then the user profile for first-time users should include a registration script.

Number of Users - How many virtual users with this user profile will run over the same time interval? Oracle Load Testing lets you specify the number of virtual users for each test scenario.

Which System - Which specific computer(s) will be used to generate the load for this user group? Oracle Load Testing can run virtual users on a single system or on multiple, distributed systems running Oracle Load Testing agents. Oracle Load Testing can specify which virtual user scenarios run on which workstations.

Which Browser - Which browser will this user group emulate? Oracle Load Testing can specify virtual user scenarios emulate either Internet Explorer or Firefox.

Pacing mode - What pacing mode will be used for the user group? Will the testing be performed using recorded think times, a range of times, or as fast as possible? Oracle Load Testing virtual user scenarios let you specify recorded, random, or no pacing.

Delay Between Business Transaction Runs - What delay time will be included between business transactions, if any? Oracle Load Testing lets you specify the amount of delay time between transaction runs.

With or Without Images - Will the user group run with images or without images? You may want to create different user groups that perform load testing both with and without images for comparison. Oracle Load Testing provides this capability.

2.8.5 Create and Verify the Test Scripts

After planning the scripts, you will use Oracle OpenScript to create and verify each script.

Create the Scripts - This process is defined by Oracle OpenScript (recording user actions) and the individual test plans for each script. When creating the script, you specify the following information as defined in the test plan:

- user actions to perform
- timers
- tests to perform
- data sources

Verify the Scripts - Once each script is created, you should verify that the script performs as expected and produces the desired result. Each script should be verified independent of any other scripts and in a controlled manner to simplify script debugging.

2.8.6 Create and Verify the Load Test Scenarios

Once the individual scripts have been created and verified, you can create and verify the load test scenarios. It will save you a lot of time and aggravation if you perform a number of simple verification steps before your full-blown load test.

Verify scripts with Multiple Virtual Users - Before combining multiple scripts into a single load test scenario, you should verify that you can successfully run a single script as multiple virtual users. Each script should perform as expected as defined by the criteria for the application.

Oracle Load Testing lets you run multiple scenarios with different virtual user characteristics.

Verify distributed test execution on multiple machines - You should verify the load test tool's ability to execute the individual scripts properly in a distributed environment if you plan to use multiple CPU's for load generation. This usually involves a master system controlling the virtual user execution on multiple workstations on the network. This can help you isolate any installation or networking-related issues.

Verify real-life scenarios that include one of each user group - Before executing the full load test you should create and verify a scenario that includes one virtual user of each user group you wish to run at the same time. That is, before you run a test with 20 VU's of group A, and 20 VU's of group B, and 60 VU's of group C, you should first run one VU of group A, one VU of group B, and one VU of group C, and make sure that the results are as expected.

Create real-life scenarios - This process should be defined in the test plans for each scenario. When creating the individual scenarios, you specify the following information as defined in the test plan:

- type of user
- pacing mode
- navigation/transactions to perform
- delay between transaction runs
- number of users of each type
- with or without images
- system used for load generation
- error log settings
- browser emulation

2.8.7 Execute the Tests

Once you have created and verified the basic load test scenarios above, you can begin to run the load test scenarios with many virtual users, and expect that the test results will be valid.

Run basic tests to ensure scaling - run tests with a minimal amount of virtual users to ensure that the system scales up correctly.

- **Run individual business transactions** - Run each of the different business transactions starting with 10 virtual users scaling up to 25 - 50 virtual users.
- **Run combinations of business transactions** - Run a combination of different business transaction scripts starting with 5 virtual users scaling up to 25 virtual users.

If the above two scenarios execute without any problems, the next step is to execute the full load test with the full number of virtual users of each user-group type.

Run the real-life scenarios - Run each of the real scenarios as outlined in the previous steps:

- Increase the scenarios up to the required number of simultaneous virtual users;
- Monitor for any errors in the system.

Re-Run these scenarios with a real user - While the load test is running, a real person should access the system through a standard browser and report performance observations:

- Observe the degradation times for a real user;
- Observe any errors if they are being reported back in the browser.

2.8.8 Evaluate the Results

For each of the load test scenarios, examine the following performance data and validate the results against the expected criteria:

- Response times for groups of users at the different numbers of virtual users;
- System throughput at various numbers of virtual users;
- Any errors that may have occurred.

View Run graph options let you evaluate performance in real-time.

Save the erroneous HTML when problems occur to help the development group debug the errors.

2.8.9 Generate Analysis Reports

Document the performance by generating the various reports that may be required for acceptance and deployment of the application. The following are some examples of the types of reports that can be generated from a load test:

- Performance vs. Time
- Statistics vs. Time
- Users vs. Time
- Errors vs. Users
- Statistics vs. Users
- Errors vs. Time
- Any other error reports that may be required for the development group to debug and fix any problems that may have occurred.

Oracle Load Testing graphs in the Reports tab let you view performance and error data from the load test in multiple formats.

2.9 Summary

Load testing throughout the development cycle has become an essential part of the process of designing scalable, reliable Web applications. Developers and QA professionals now rely on load testing tools as a means to validate system architectures, tune applications for maximum performance, and assess the impact of hardware upgrades. Consequently, it is critical that the load test results can be used with confidence as the basis for key decisions about application readiness and potential changes to the system's hardware and software. Using the methodology embodied in this guide along with accurate load testing tools such as Oracle Load Testing, you now have a systematic approach to ensure the performance of your Web applications. With load testing established as a routine part of the application lifecycle you can be sure to avoid costly "scalability surprises" when your application goes live for the first time or after any subsequent release.

This chapter explains how to get started using Oracle Load Testing. It explains how to install and start the program, and the features of the main window.

3.1 Installing Oracle Load Testing

The installation for Oracle Load Testing depends upon how you plan the virtual user test configuration. At a minimum, you need to install Oracle Load Testing on a single system that can access the Web application. If you are performing distributed testing in a networked environment, you need to install at least one station with Oracle Load Testing to use as a "Master" station and install the Oracle Load Testing Agent (or another seat of Oracle Load Testing) on the client machines to use as virtual user Agent stations.

The following sections explain the procedures for installing Oracle Load Testing and the Oracle Load Testing Agent.

3.1.1 Installing Oracle Load Testing

The Oracle Load Testing setup procedure installs both Oracle Load Testing Server and Oracle Load Testing Agent. You should not install Oracle Load Testing Agent separately on the same system. To install Oracle Load Testing:

1. Go to:
<http://www.oracle.com/technetwork/oem/downloads/index-084446.html>.
2. Download the Oracle Application Testing Suite product from the Oracle Web site and save it to a temporary directory on your hard disk. See the *Oracle Application Testing Suite Installation Guide* for additional information about the product zip files.
3. Unzip the download file and then run setup.bat.
4. Follow the setup instructions to install the Oracle Application Testing Suite.

Note: A product establishes its Default Repository in \$installDir/OFT, where \$installDir is the directory where Oracle Application Testing Suite is installed or, if Oracle Application Testing Suite is not installed, where OpenScript is installed.

During the Oracle Application Testing Suite installation, you will be required to enter a master password to be used with Oracle Application Testing Suite products. *Remember this password.* It will be required to log in to the Administrator, Oracle Load Testing, and Oracle Test Manager.

5. For Windows installations, select **Oracle Load Testing** from the Oracle Application Testing Suite **Start** menu or enter `http://<machine>:8088/olt` or `http://localhost:8088/olt` in your browser where *<machine>* is the name of the machine where the Oracle Application Testing Suite is installed.

For Linux installations, enter `http://<machine>:8088/olt` or `http://localhost:8088/olt` in your browser where *<machine>* is the name of the machine where the Oracle Application Testing Suite is installed.

Two default administrator accounts are created at installation. The usernames for the default accounts are **administrator** and **default**. The default password for both the **administrator** and **default** accounts is the master password specified during the Oracle Application Testing Suite installation process. Use the Oracle Application Testing Suite Administrator to customize the Oracle Load Testing database by creating user accounts, assigning user names and passwords, and assigning the type of access that they have in Oracle Load Testing. See [Section 3.9, "Administrator"](#) for additional information about using the Oracle Application Testing Suite Administrator.

Use the Oracle Application Testing Suite Database Configuration utility to configure database connections to Oracle Load Testing databases. On Windows machines, you can access the Database Configuration utility from the **Tools** sub menu of the Oracle Application Testing Suite **Start** menu. On Linux machines, you can access the Database Configuration utility from `<oats_install>/bin/DbConfig.sh`.

Oracle Load Testing starts in an existing browser if one is available. To set Oracle Load Testing to always start in a new browser window, change the **Reuse windows for launching shortcuts** setting in Internet Explorer to be deselected. In IE, select **Internet Options** from the **Tools** menu then click the **Advanced** tab to change the setting under **Browsing** from the Advanced tab to access this setting.

3.1.2 Preconditions for Using Functional Testing Scripts

The follow are the preconditions for running OpenScript functional test scripts with Oracle Load Testing:

Installation Requirements:

- Linux: Must have installed 'vnc server'(4.2) and 'firefox'(ESR 31, 38, 45).
- Windows: Must enable and start remote desktop service. Windows 2008 R2 is the minimum supported system. Windows 7 is not supported as Windows 7 only allows the single primary user of the licensed computer to access a session of the computer.

Privilege & License Requirements:

- Linux: IPtables shall have rules accepting vnc server listen ports (5901-59**).
- Windows: The agent system must have enough licenses to support the required number of the concurrent RDP sessions.
- Windows: Configure 'Oracle ATS Agent' service to "Allow service to interact with desktop".
 1. Go to Windows Services (Select **Start**, then **Run**, and enter `services.msc` or Go to Control Panel, Administrative Tools, then Services).
 2. Select 'Oracle ATS Agent' service then right-click and select **Properties**.

3. Select the **Log On** tab, then select **Local System account** and select the **Allow service to interact with desktop** checkbox.
4. Click **OK**.
5. Click 'Restart the service' to restart the service.

Enabling the **Allow service to interact with desktop** checkbox is a one time requirement. After enabling this option and playing back Functional Test scripts one time on the agent, this setting can be disabled if you do not want to see the RDP session (which contains the browser playback within the session).

- Windows: The agent system must have the required privileges to create windows accounts with 'remote desktop user group' privileges.
- Windows: The User Account Control (UAC) level that controls Windows notifications on the agent system must be set to "Never Notify" - the least secure setting. This is *not* the default setting for Windows.
- Windows: Internet Explorer Enhanced Security Configuration on the agent system must be disabled for the Administrators group (which includes all of the user accounts for the OLT functional test support). This is *not* the default setting for Windows.

The Oracle Application Testing Suite setup, installation, configuration and other scripts do not make these changes and security settings are not modified as part of the account creation for playback. These requirements must be manually configured by functional test users.

Encrypted Script Requirements:

You will need the password(s) for Oracle OpenScript scripts that use password protected encryption. When you add a password protected script to configure parameters in the **Build Scenarios** tab, the **Import Script Password** dialog box opens for specifying the script password.

3.1.3 Popup Blockers

Popup blockers must be turned off for Oracle Load Testing to operate. To turn off popup blockers:

Firefox - Select **Options** from the **Tools** menu. Uncheck Block Popup Windows.

Internet Explorer - Select **Pop-up Blocker** from the **Tools** menu. If the pop-up blocker is on, select **Turn Off Pop-up Blocker**.

3.1.4 Port Configuration

If you are using a firewall between the Oracle Load Testing Server machine and the agent machines, port 9001 must be open in your firewall software itself and on the agent machines.

To change this port on a Windows agent machine, change the `oracle.oats.agentmanager.port` setting in the `<installdir>\agentmanager\agentmanager.properties` file.

To change this port on a Linux agent machine, do one of the following:

1. `<installdir>/agentmanager/bin/agentmanager.sh --verbose --port=<new port>`
start

2. `AM_PORT=<#portnum>` in `/etc/init.d/OracleATSAgent` and then restart the ATS agent.

In addition, change the port in the system configuration in Oracle Load Testing by selecting **Options** from the **User** menu then selecting **VU Agent Systems** in the **Systems** group. Select the agent machine that is running on a different port, click **Edit**, and change the default port 9001 to the port that you set in the `agentmanager.properties` file.

To run Oracle Load Testing, it is recommended that you have the following general communication ports open before starting the Oracle Load Testing Application Service.

```
http=8088
agent=9001
dataCollector=9001
webservice=8083
https=8089
```

To change the ports:

1. Go to `http://localhost:8088/console` to start the WebLogic Console.
2. Log in as username "oats" and the password you defined during the Oracle Application Testing suite installation procedure.
3. In the Domain Structure, select Environment under "oats" and then select Servers.
4. Select AdminServer(Admin).
5. Change the port and release the config.
6. In `<oats-home>/config/oats-config.xml` update http, admin, and cluster URLs with appropriate port specifications.
7. Restart the ATS service and login to Oracle Load Testing or Oracle Test Manager.

See the Oracle WebLogic Server documentation for additional information about using the Console application.

The following ports are used between the Oracle Load Testing Server and the Agent Machine itself:

- 9001
- 8088

3.1.5 Installing Oracle Load Testing Agent

The Oracle Load Testing Agent is a subset of the full Oracle Load Testing installation. You should not install the Oracle Load Testing Agent if you have Oracle Load Testing installed on the system. You install Oracle Load Testing Agent on the agent systems when you want to run distributed load tests in a network environment and you want the agent systems to have more system resources available for running virtual users.

To install the Oracle Load Testing Agent:

1. Run the downloaded `setup.bat` installation program from the download zip file.
2. Follow the setup procedure to the Select Components and Installation Directory screen.
3. Clear all check boxes except the **Remote Agent** check box.
4. Click **Next** as necessary to complete the installation.

5. Verify network access from Oracle Load Testing system to the Agent systems and configure the Agent Systems as explained in the next sections.
6. Configure the agent system.
7. Define the system in Oracle Load Testing by selecting **Options** from the **User** menu menu and selecting **Systems**.

Once the Oracle Load Testing Agent is installed and the Agent systems are configured, you do not need to start or run an application. When you define your Oracle Load Testing scenarios, you specify which Agent machine to use to run the virtual users in the System field of the Build Scenarios tab. Oracle Load Testing automatically accesses and starts the Agent when you start a Session.

3.1.5.1 Verify Network Access to Agent Systems

Once you have the Oracle Load Testing and Agent software installed on the individual systems, you should verify network access between the Oracle Load Testing system and each Agent system. This section provides basic tips and techniques to make sure that Oracle Load Testing can successfully communicate with each Agent system.

1. Make sure that you have the Oracle Load Testing Agent software loaded on the Agent system(s) and that it is the same version as the Oracle Application Testing Suite software that is loaded on the Oracle Load Testing system. The systems you plan to use as agents must have either the Oracle Load Testing Agent software or the full Oracle Application Testing Suite installed to work as agents. It is not recommended to use the local Oracle Load Testing Agent and the Oracle Application Testing Suite server software on the same system. Otherwise, resource conflicts between the two programs may occur.
2. Make sure you can successfully Ping all of the Agent systems from the Oracle Load Testing system. The names you use to Ping the systems are the same names that you will specify for the Agent systems in the Oracle Load Testing system. If you cannot successfully Ping the Agent systems, contact your network administrator to resolve the issue. If you cannot Ping the agent systems from the Oracle Load Testing system, you will not be able to run the agents from the Oracle Load Testing system.
3. In the Oracle Load Testing system, add a script to the Configure Parameters of the Scenario list. Enter the machine name or IP address of the Agent system where you want to run the script into the System field on the Build Scenarios tab of Oracle Load Testing.

3.1.5.2 Configuring Oracle Load Testing Agents

If you need to change the account that the agent service, and thus the agent itself uses, you must specify the login information. You may need to do this:

- When testing with client side certificates. They are typically installed on agent machines and may only be accessible when logged in as that user.
- When the system account has a proxy server configuration setup for use by the machine's local Internet Explorer.

To configure Oracle Load Testing agents:

1. Open **Administrative Tools** in the **Control Panel** then open **Services**. The Services dialog box displays.
2. Select **Oracle Load Testing Agent Service**.
3. Select **Properties** from the **Action** menu.

4. Click the **Log On** tab.
5. Select **This account** to specify the login information:
 - **This Account** - specify the account or use the Browse button to navigate to the account.
 - **Password** - specify the login password.
 - **Confirm Password** - confirm the login password.
6. Click **OK**.

If the password for this account changes, you must change the password using this procedure.

3.2 Installing the Linux Agent

Oracle Application Testing Suite server components (Oracle Load Test/Oracle Test Manager) and Agent components (Oracle Load Test agent/Data Collector) can be installed on Linux via a separate installer; however Oracle OpenScript is Windows only.

See the instructions in the *Oracle Application Testing Suite Installation Guide* for details about installing the applications and agent on Linux machines.

Once the Linux agent is installed, define the remote agent system in the Oracle Load Testing **Options**.

When you define your Oracle Load Testing scenarios, you specify which Agent machine to use to run the virtual users in the System field of the Build Scenarios tab. Oracle Load Testing automatically accesses and starts the Agent when you start a Session.

3.3 Moving an Existing Installation to a New Machine

In some cases, it may be necessary to move an existing Oracle Application Testing Suite installation from one machine to another. This could be either for an Oracle Application Testing Suite software upgrade or a hardware upgrade.

There are several steps involved in moving an existing installation to another machine.

- Transfer database schemas
- Install Oracle Application Testing Suite and configure database connections
- Copy repositories, workspaces, scripts, and files not stored in a repository:
 - Oracle Load Testing scenarios,
 - Oracle Load Testing ServerStats metrics,
 - customized files,
 - data files.
- Restart the Oracle Application Testing Suite service and verify the setup on the new machine.

The following sections describe the steps.

3.3.1 Transfer Database Schemas

Transfer the OATS, OTM, and OLT database schemas using the expdp and impdp database utilities, as follows:

1. Log into SQLPlus as system user and do the following:

```
CREATE DIRECTORY bkp_dir AS 'C:\backup';
```

2. Execute the following from the server\BIN directory of Oracle database (e.g. C:\OracleATS\oxe\app\oracle\product\1x.x.0\server\BIN) where x.x is the version number of Oracle Application Testing Suite installation:

```
expdp system/<passwd>@xe schemas=OATS directory=bpk_dir
dumpfile=oatsbcp_x.x.dmp logfile=oatsbcpexp_x.x.log
```

```
expdp system/<passwd>@xe schemas=OLT directory=bpk_dir
dumpfile=oltbcp_x.x.dmp logfile=oltbcpexp_x.x.log
```

```
expdp system/<passwd>@xe schemas=OTM directory=bpk_dir
dumpfile=otmbcp_x.x.dmp logfile=otmbcpexp_x.x.log
```

3. In the new machine, install your database.
4. Log into SQLPlus as system user and do the following:

```
CREATE DIRECTORY bkp_dir AS 'C:\backup';
```

5. Copy the backup dump files to the new machine.
6. Execute the following from the server\BIN directory of Oracle database (e.g. C:\OracleATS\oxe\app\oracle\product\10.2.0\server\BIN) where x.x is the same version number used for step 2:

```
impdp system/<passwd>@xe schemas=OATS directory=bpk_dir
dumpfile=oatsbcp_x.x.dmp logfile=oatsbcpimp_x.x.log
```

```
impdp system/<passwd>@xe schemas=OLT directory=bpk_dir
dumpfile=oltbcp_x.x.dmp logfile=oltbcpimp_x.x.log
```

```
impdp system/<passwd>@xe schemas=OTM directory=bpk_dir
dumpfile=otmbcp_x.x.dmp logfile=otmbcpimp_x.x.log
```

3.3.2 Install Oracle Application Testing Suite and Configure Database Connections

This section explains the installation and database configuration steps for using an existing database.

1. Install the Oracle Application Testing Suite on the new machine. When the Oracle Database installation screen appears, select "Configure an existing Oracle XE or EE Database" and manually enter system password which was used for DB installation.

During database configuration, the installation will prompt the following message:

```
"Warning: The database you have configured already contains the OLT and OTM
schemas. You will need to run the Database Configuration Utility to
re-configure these schemas after installation completes"
```

2. After the installation completes, select **Oracle Application Testing Database Configuration** from the **Tools** submenu of the Oracle Application Testing Suite **Start** menu.

3. Select **Oracle Load Testing**.
4. Click the **New** icon on the Database Connections toolbar.
5. Select **Use existing schema** and specify the Connection details for the existing database.
 - Name - OLT (can be anything).
 - Description - enter any description for the existing database.
 - Hostname, port, service of database instance.
 - Username - olt (must be "olt").
 - Password - the olt user password (the password specified during the Oracle Application Testing Suite installation).
6. Optionally, select **Insert sample data**.
7. Click **Save**.
8. In the new dialog box that opens, enter the password for the "Administrator" and "Default" accounts.
9. Set the new OLT database as current by clicking the **Set Current** (check mark) icon on the toolbar. Click **OK** in the confirmation message.
10. Select **Oracle Test Manager**.
11. Click the **New** icon on the Database Connections toolbar.
12. Select **Use existing schema** and specify the Connection details for the existing database.
 - Name - OTM (can be anything).
 - Description - enter any description for the existing database.
 - Hostname, port, service of database instance.
 - Username - otm (must be "otm").
 - Password - the otm user password.
13. Optionally, select **Insert sample data**.
14. Click **Save**.
15. Select **Configuration schema**.
16. Click the **New** icon on the Database Connections toolbar.
17. Select **Use existing schema** and specify the Connection details for the existing database.
 - Name - OATS (can be anything).
 - Description - enter any description for the existing database.
 - Hostname, port, service of database instance.
 - Username - oats (must be "oats").
 - Password - the otm user password.
18. Click **Save**.

3.3.3 Copy Repositories and Files

Any files not stored in a repository folder will need to be copied manually from the old machine to the new machine. Close all Oracle Application Testing Suite applications before copying files.

Copy the following file types from the old machine to the new machine.

- If your repositories were stored on the local machine, copy Repository folders and scripts from the old machine to the new machine. Make sure the folder names and directory structure are all the exact same. The default repository is <installdir>/OFT.
- databank files (*.csv, *.txt saved to your custom directories).
- OLT Scenario files (*.scn and *.scnzip files not stored in a repository).
- OLT Session data files (*.osd saved to your custom directories).
- "custom" ServerStats Configurations (*.config files stored in <installdir>/config/serverstats).
- "custom" ServerStats Metrics (*.metric files stored in <installdir>/config/serverstats).
- "custom" ServerStats Metric Profiles (*.hwm or *.zip files stored in <installdir>/config/serverstats).
- "custom" MIBS added to ServerStats (*.mib files stored in <installdir>/config/serverstats/mibs).
- "custom" Report Template files (*.rtf files saved to your custom directory).
- "customized" template.properties file in the <installdir>/data/olt/reports folder.
- "custom" data files (*.dat files saved to your custom directory).
- any saved or exported report files.
- any other custom files used with Oracle Application Testing Suite stored on the old machine.

3.3.4 Restart the Service and Verify the Setup

To restart the Oracle Application Testing Suite service:

1. Open Control Panel.
2. Open Administrative Tools.
3. Open Services.
4. Select Oracle ATS Server and click **Restart** the service.
5. After the service restarts, open each Oracle Application Testing Suite application and verify the setup matches the old installation.

3.4 Adding Repositories

Repositories specify the location to use to store scripts and related asset files.

Repositories also provide a way to share files between OpenScript and Oracle Load Testing. Oracle Load Testing requires that all assets live inside of a named Repository. Oracle Load Testing will not be able to find an asset located in the local file system outside of a repository. Any shared directory can be used as a repository. However, all repositories shared between Oracle Load Testing, Oracle Test Manager, OpenScript,

and team members must share the same repository name. For example, if one member of a team calls a shared repository `SharedRepo1`, but another member of a team calls the same shared repository `Shared_Repository_1`, it is possible that some script assets may not be found when the team members share scripts.

To reduce the chance of local repository name conflicts, it is recommended that you create a new local repository named something unique to the user, such as `<machineName>.<windowsUserName>.MyRepository`. Store in this folder all scripts that are not intended to be shared among team members.

Best Practices:

- Always store scripts and assets (i.e. databanks, .jar files, etc.) inside named repositories.
- Avoid selecting the **Save path relative to current script** option in OpenScript when saving scripts.
- Establish a consistent repository naming scheme across all Oracle Load Testing, Oracle Test Manger, and OpenScript installations.
- Avoid using the repository named "Default" for storing local scripts. Use `"machineName.Default"` instead.

To add a repository:

1. Select **Options** from the **User** menu.
2. Select **Repositories** in the **Manage** group of the left pane.
3. Click **New**. A new entry is made in the table.
4. Enter the name of the repository.

Note: When using OpenScript scripts with Oracle Load Testing, the repository names you specify should match the repository name specified in OpenScript (including case).

5. Enter the location of the repository.

3.5 Changing the Web Server Port

The default Oracle Load Testing web server port is 8088. You can change this to another port. The port number must be changed in the WebLogic Console and the Oracle Application Testing Suite configuration.

To change the port in the WebLogic Console:

1. Go to `http://localhost:8088/console` to start the Oracle WebLogic Server Administration Console.
2. Log in as an administrator (the default username is "oats") using the password you defined during the Oracle Application Testing Suite installation procedure.
3. In the Domain Structure, select Environment under "oats" and then select Servers.
4. Select AdminServer(Admin).
5. Change the port and release the configuration.

See the Oracle WebLogic Server Administration Console documentation for additional information about using the Console application.

6. In `<oats-home>/config/oats-config.xml` update http, admin, and cluster URLs with appropriate port specifications.
7. Restart the ATS service and login to Oracle Load Testing or Oracle Test Manager.

To change the port in the Oracle Application Testing Suite configuration:

1. Open the file `<oats-home>/config/oats-config.xml` in a text editor.
2. Change the port number from 8088 to the new value in all property keys where the port number is used.
3. Save the file.
4. Open the Registry Editor (regedit) and navigate to:


```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\OracleATSServer\Parameters
```
5. Change the `Port` value to the new Port number and close the Registry Editor.
6. Open the Control Panel and then open Services in the Administrative Tools.
7. Restart the "Oracle Application Testing Suite Application Service" service.
8. Login to Oracle Load Testing and select **Options** from the **User** menu, and then select **ServerStats Data Collectors** in the **Systems** group.
9. Select your Oracle Load Testing server system, click **Edit**, enter the changed port number, and click **Save**.

If you are using remote data-collectors (non-localhost), Steps 6 and 7 should not be performed.

3.6 Using SSL

You can set up Oracle Load Testing to use SSL (Secure Sockets Layer). The procedure is comprised of the following steps:

1. Go to `http://localhost:8088/console` to start the WebLogic Console.
2. Log in as "oats" using the password you defined during the Oracle Application Testing suite installation procedure.
3. In the Domain Structure, select Environment under "oats" and then select Servers.
4. Select AdminServer(Admin).
5. Select the SSL tab.

See the Oracle WebLogic Server documentation for additional information about using the Console application.

3.7 Changing the OLT Controller Heap Settings

The default JVM Heap size is sufficient for most common load testing scenarios. However, it may not be suitable to operate efficiently when executing large, long-running load tests or load tests that generate verbose logging, such as when logging of all VU Log request/response details is enabled. Tuning the JVM Heap settings that are specific to your system becomes critical in such cases. This section describes how to modify the maximum Heap size for the Oracle Load Testing controller and specifies the values to enter that are specific to your system.

For additional information, see the *Java Platform, Standard Edition Tools Reference* documentation:

Windows:

<http://docs.oracle.com/javase/8/docs/technotes/tools/windows/java.html>

Linux:

<http://docs.oracle.com/javase/8/docs/technotes/tools/unix/java.html>

3.7.1 Basic Guidelines for the Controller Heap Settings

The following are the basic guidelines for the settings to use for the maximum JVM heap size (recommended heap size will depend on type of scripts and scale of load test):

On Linux machines:

- 32bit - Set the maximum JVM Heap size to -Xmx2g
- 64bit - Set the maximum JVM Heap size to -Xmx3g

On Windows machines:

- 32bit - User can not set more than 1 GB heap size
- 64bit - Set the maximum JVM Heap size to -Xmx3g

Where:

- -Xms = Starting heap size (default is 256m)
- -Xmx = Maximum Heap size (default is 512m)

3.7.2 Modifying the JVM Heap Settings On Windows Machines

To modify the heap setting on Windows machines:

1. Stop the Oracle ATS Server service. Use **Services** in the Control Panel or `net stop OracleATSServer` in a command window.
2. Open the Windows Registry Editor (`Regedit.exe`).
3. Navigate to `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\OracleATSServer\Parameters`.
4. Select the key named `CmdLine`.
5. Select **Modify** from the **Edit** menu of the Registry Editor.
6. In the **Value Data** field, modify the startup value for the Heap from `-Xms256m` to desired value. For example, to update the value to use a 1GB Heap, modify the value to `-Xms1g`.
7. In the **Value Data** field, modify the maximum value for the Heap from `-Xmx512m` to desired value. For example, to update the value to use a 2GB Heap, modify the value to `-Xmx2g`.
8. Click **OK**.
9. Close the Registry Editor
10. Start the Oracle ATS Server service. Use **Services** in the Control Panel or `net start OracleATSServer` in a command window.

3.7.3 Modifying the JVM Heap Settings On Linux Machines

To modify the heap setting on Linux machines:

1. Stop the Oracle ATS Server daemon (service). For example,
`/etc/init.d/OracleATSServer stop`.
2. Open the daemon script for editing using a text editor of your choice. For example,
`nano -w /etc/init.d/OracleATSServer`.
3. Locate the line: "export USER_MEM_ARGS="-Xms256m -Xmx512m".
4. Modify the startup value for the Heap from -Xms256m to desired value. For example, to update the value to use a 1GB Heap, modify the value to -Xms1g.
5. Modify the maximum value for the Heap from -Xmx512m to desired value. For example, to update the value to use a 2GB Heap, modify the value to -Xmx2g.
6. Close the editor.
7. Start Oracle ATS Server daemon (service). For example,
`/etc/init.d/OracleATSServer start`.

3.7.4 Limitations

Users of 32-bit Windows Systems should leave the current default settings unchanged. However, you can enable the 4GT feature in your 32-bit edition of Windows to request a larger heap from the Operating system. Once you have enabled this feature you can then change the maximum Heap size setting to 2GB(-Xmx2g).

The Oracle Load Testing Controller also requires the Windows Weblogic patch in order to use the 4GT tuning feature. If you are running large load tests using Oracle Application Testing Suite version 9.31 on *Windows 32-bit systems*, you should upgrade to version 9.31.044 or later and set Windows 4GT tuning. Only version 9.31.044 and higher will allow the heap to be set to the recommended -Xmx2g, and only when the Windows 32-bit system has 4GT tuning applied.

On 32-bit Linux Systems, change the maximum Heap size setting to 2GB (-Xmx2g).

On 64-bit Systems, the recommend maximum Heap size setting is 3GB(-Xmx3g).

3.8 Oracle Application Testing Suite Tools Menu

The Oracle Application Testing Suite tools menu has options for viewing version information, restarting and stopping the Oracle Application Testing Suite Application service, and creating support files for troubleshooting. Select **Tools** from the Oracle Application Testing Suite start menu. This menu has the following options:

About Oracle Application Testing Suite - displays the About Oracle Application Testing Suite dialog box that shows copyright and version information. It also has information about your system.

Create Support Package - for troubleshooting purposes, creates the OATSSupport.zip file and places it on your desktop. From there you can email it to your support representative. This file contains the log files used for troubleshooting. On Linux machines, use `<oats_install>/bin/oats_support.sh` to create support packages.

Oracle Application Testing Database Configuration - opens the database configuration utility for adding or removing database connections for Oracle Load Testing and Oracle Test Manger. On Linux machines, use `<oats_install>/bin/DbConfig.sh` to start the database configuration utility.

Oracle Load Testing Agent Authentication Manager - opens the Agent Authentication Manager for defining authentication profiles for multiple load testing agent machines. On Linux machines, use `<oats_install>/jdk/jre/bin/java -jar`

`<oats_install>/agentmanager/AMAuthManager.jar` to start the Agent Authentication Manager.

Restart Oracle Application Testing Suite Application Service - stops and restarts the Oracle Application Testing Suite Application service. On Linux machines, use `<oats_install>/bin/restartSvc.sh [OracleATSServer|OracleATSAgent]` to restart the service.

Stop Oracle Application Testing Suite Application Service - stops the Oracle Application Testing Suite Application Service. On Linux machines, use `<oats_install>/bin/stopSvc.sh [OracleATSServer|OracleATSAgent]` to stop the service.

3.9 Administrator

The Administrator allows you to create user accounts, assign them user names and passwords, and assign the type of access that they have in Oracle Load Testing, none, full control, or view only. The Administrator also lets you optionally enable authentication for Oracle Load Testing. When Oracle Load Testing login is enabled, users must login to access Oracle Load Testing.

Two default administrator accounts are created at installation. The usernames for the default accounts are **administrator** and **default**. The default password for both the **administrator** and **default** accounts is the master password specified during the Oracle Application Testing Suite installation process. You can change the password after logging in to the Administrator. It is recommended that you change the default password as soon as you log in.

To start the Administrator:

1. For Windows installations, select **Administrator** from the Oracle Application Testing Suite **Start** menu or enter `http://<machine>:8088/admin` or `http://localhost:8088/admin` in your browser where `<machine>` is the name of the machine where the Oracle Application Testing Suite is installed.

For Linux installations, enter `http://<machine>:8088/admin` or `http://localhost:8088/admin` in your browser where `<machine>` is the name of the machine where the Oracle Application Testing Suite is installed.

2. Enter your password.
3. Select the load testing database that you want to access.
4. If you want to change the administrator password, click the User tab, select the Administrator user, and click **Edit**.
5. Enter the new password, verify it, and click **OK**.

3.9.1 Menu Options

This section describes the menus and options available in the Administrator.

3.9.1.1 Help Menu

Contents - displays the online help table of contents.

About Admin - displays version and copyright information.

3.9.1.2 Logout

Exits the Administrator.

3.9.2 Users Tab

The Users tab is where you add, edit, and delete users and specify what Oracle Load Testing features they can access.

Add - displays the Add User dialog box for adding new users.

Edit - displays the Edit User dialog box for the selected user.

Delete - deletes the selected user.

Restore - displays the Restore Previously Deleted User dialog box for restoring a previously deleted user to the users list.

Username - displays the user name for logging in to Oracle Load Testing.

First Name - displays the user's first name.

Last Name - displays the user's last name.

E-Mail - displays the user's e-mail address.

Access - displays the type of access the user has in Oracle Load Testing.

Administrator Access - displays whether the user can access the Administrator.

3.9.3 Usage Audit Tab

The Usage Audit tab is where you review and audit the load testing sessions stored in the Oracle Load Testing database.

OLT Databases - lists the installed Oracle Load Testing Databases available for auditing.

User Name - shows the name of the user who ran the load test. "Anonymous" indicates the login feature was disabled for the instance of Oracle Load Testing that ran the test and there is no username associated with the test. "Command Line" indicates the load test ran from the command line interface.

Session Name - shows the name of the load testing session.

Start Time - shows the start date and time for the load testing session.

End Time - shows the end date and time for the load testing session.

Duration (HH:MM:SS) - shows the duration of the load testing session in hours, minutes, and seconds.

Machine Name - shows the name of the machine on which the load testing session was run.

Max VU Count - shows the maximum count of Virtual Users that were run for the load testing session.

3.9.4 Adding Users

To add a user:

1. Click **Add**.

First Name - enter the user's first name.

Last Name - enter the user's last name.

E-Mail - enter the user's e-mail address.

Username - enter the user's username.

Password - enter the user's password.

Confirm Password - re-enter the user's password.

Access - sets the type of access the user has in Oracle Load Testing.

- **Full Control** - the user has full control in Oracle Load Testing.
- **View Only** - the user cannot run tests but can view reports.

Enable Administrator Access - gives this user the ability to log on to the Oracle Load Testing Administrator for managing the database.

2. Enter the user's information.
3. Select or clear the E-Mail notification and Administrator access options.
4. Click **OK**.

3.9.5 Editing Users

To edit a user:

1. Select the user whose information you want to change.
2. Click **Edit**.

First Name - enter the user's first name.

Last Name - enter the user's last name.

E-Mail - enter the user's e-mail address.

Username - enter the user's username.

Password - enter the user's password.

Confirm Password - re-enter the user's password.

Access - sets the type of access the user has in Oracle Load Testing.

- **Full Control** - the user has full control in Oracle Load Testing.
- **View Only** - the user cannot run tests but can view reports.

Enable Administrator Access - gives this user the ability to log on to the Oracle Load Testing Administrator for managing the database.

3. Make any changes.
4. Click **OK**.

3.9.6 Deleting Users

To delete a user:

1. Select the user you want to delete.
2. Click **Delete**.
3. Click **Yes** when asked to confirm the deletion.

3.9.7 Restoring Users

You can restore a previously deleted user to the user list. To restore users:

1. Click **Restore**.
2. Enter the Username of the user to restore.

3. Click OK.

3.9.8 Auditing Usage

To audit usage:

1. Click the **Usage Audit** tab.
2. Select the database. The session information appears in the right pane of the Usage Audit tab. See [Section 3.9.3, "Usage Audit Tab"](#) for additional information.

3.10 Main Window Features

The Oracle Load Testing main window is where you perform the majority of your load/performance testing activities. The main window consists of the menu bar, toolbar, and three main tabs. Each tab contains sub-tabs for configuring scenarios, viewing active session data and generating post-session reports.

3.10.1 Overview of the Menu Options

The Oracle Load Testing **User** menu has the following options:

- Account Info - opens account information.
- Options - opens the options dialog for specifying Oracle Load Testing options.
- Help - opens the help contents.
- Log Off - logs out of Oracle Load Testing.

3.10.2 Toolbar

The toolbar has the following buttons:

New Scenario - Creates a new Oracle Load Testing scenario.

Open Scenario - Opens an existing Oracle Load Testing scenario to run or modify.

Save Scenario - Saves any changes to the currently open Oracle Load Testing scenario. If the scenario has not been saved before, Oracle Load Testing asks for a filename.

Run Now - starts the current scenario immediately.

Stop All Virtual Users - Stops all virtual users that are running in the current scenario.

Abort All Virtual Users - Aborts all virtual users that are running in the current scenario.

3.11 Composer Tab

The Composer tab is where you specify which scripts to include in the scenario, the VU ramp options, testing goals, and ServerStats monitoring options.

Scenario name menu - shows the current scenario name. The menu has the following options:

- **New** - creates a new scenario.
- **Open** - opens an existing saved scenario.
- **Save** - saves the current scenario.
- **Save As** - saves the current scenario to a new name.

- **Import** - imports an existing saved scenario.
- **Export** - exports the current scenario.

The Composer tab has the following main toolbar buttons:

- **New Scenario** - creates a new scenario.
- **Load Scenario** - opens an existing saved scenario.
- **Save Scenario** - saves the current scenario.

The Composer tab has the following sub-sections:

- Profiles
- Ramp
- Goals
- Monitoring

3.11.1 Profiles Section

The Profiles section is where you specify the scripts to include in the scenario and the profile options for each script. The **Profiles** section has the following panels:

- Scripts
- Profiles
- Properties

3.11.1.1 Scripts Panel

The **Scripts** panel includes the script tree and toolbar.

The **Available Scripts** list shows the Oracle OpenScript scripts in the current repository/workspace.

The scripts toolbar has the following options:

- **Add to Scenario** - adds the currently selected script to the **Profiles** list.
- **Path** - shows the directory path of the selected repository. The default repository is the OFT folder in your installation directory. New repositories can be created by selecting **Options** from the **User** menu then selecting **Repositories** in the **Manage** group.
- **<Script list>** - a list of Oracle OpenScript scripts that are available to include in virtual user scenarios. Scripts can be both load testing-type scripts and functional testing-type scripts. See the preconditions for functional testing-type scripts in [Section 3.1.2, "Preconditions for Using Functional Testing Scripts"](#).

You can click and drag scripts to the **Profiles** panel or select scripts and click the **Add to Scenario** button.

3.11.1.2 Profile Panel

The **Profile** panel shows the scripts selected for the current Oracle Load Testing scenario. You can configure each using the options provided here or in the **Properties** panel for the selected script.

You can change the fields that are displayed and the default values for each field by selecting **Options** from the **User** menu then selecting **Default Properties** in the **Profiles** group and checking or unchecking the **Table View** option.

Show/Hide Scripts tree - opens the Available Scripts panel if it is hidden or closes the Available Scripts panel if it is visible.

Virtual User Scale - specifies the maximum number of virtual users to which to scale the load test. Editing this value will distribute virtual users among all scripts in the scenario.

VU Distribution - opens a graph showing the Virtual User distribution.

Table View - Shows the profile list in table view. Properties for selected profiles can be set in the Profile table and the Properties panel.

Icon View - Shows the profile list in Icon view. Properties for selected profiles can only be set in the Properties panel.

Profile - a list of scripts selected to be in the load scenario. In Table view, the fields displayed here can be customized by selecting **Options** from the **User** menu then selecting **Default Properties** in the **Profiles** group. Select the fields you want to display by checking the field's corresponding checkbox in the Table View column.

Max # VUs - specifies the number of virtual users to run for the selected profile. For each virtual user, Oracle Load Testing runs a separate instance of the script(s) specified in the virtual user profile. For functional testing-type scripts, a warning indicator appears if you specify more VUs than the maximum allowed for the Oracle Load Testing Agent machine.

Agent System - specifies the machine on which the virtual users will run. When running virtual users across systems on a LAN/WAN, select the system name of a system running either Oracle Load Testing Server or Oracle Load Testing Agent from the option dropdown. Systems are defined using the VU Agent Systems option in the System Manager. Initially, you must define the machine names or IP addresses of the system(s). Once the name(s) or IP addresses have been specified, you can select the system name from the drop-down list for future load tests.

When determining the number of virtual users to run per process or system, you need to include the Client overhead in the resource allocation. Each VU requires approximately 350 KB-500 KB of memory to run. When calculating the available memory to run VUs on an agent system, you must account for a 20-30% client system overhead. Therefore, you only have 70-80% of the physical memory (RAM) available to run VUs. For functional testing-type scripts, the number of VUs may be limited by the number of accounts available on the Oracle Load Testing Agent machine (Remote Desktop for Windows, VNC for Linux).

Iteration Delay - specifies the amount of time (in seconds) to wait between iterations of virtual user runs. You specify the number of iterations using the Ramp options.

VU Pacing - specifies the script playback delay between pages for each virtual user. This is the amount of time the user looks at a page before making the next request and is commonly referred to as "think time." There are four options:

- **Recorded** - uses the delay times that were recorded in the Oracle OpenScript script. You can set minimum and maximum delay times (in seconds) that override the script delay times in the Minimum and Maximum edit boxes.
- **Recorded/Random** - uses random delay times based upon the recorded user delay. Oracle Load Testing sets the low end of the random range as the actual recorded user delay minus the Lower percentage setting. Oracle Load Testing sets the high end of the random range as the actual recorded user delay plus the Upper percentage setting. For example, if the actual recorded delay time was 100 seconds and the Lower and Upper settings are 10% and 25% respectively, Oracle Load Testing uses random delay times between 90 and 125 seconds.

- **Random** - uses random times for Virtual User pacing. You can set minimum and maximum delay times for random delay in the Minimum and Maximum fields.
- **No Delay** - plays back the scripts at the fastest possible speed with no time between page requests.

Delete - deletes the selected profile from the scenario.

For information about the Selected VU Profile settings see [Chapter 5, "Creating Scenario Profiles"](#).

3.11.1.3 Properties Panel

Displays the Properties for configuring the parameters of the script selected in the Profile panel. Select the script name in the Profiles pane and set the settings.

The Profile Settings are listed in the Profile Settings. Note that the settings differ between load test-type (protocol based) scripts and functional test-type (DOM based) scripts.

3.11.2 Ramp Section

The Ramp section lets you configure the settings for the Virtual User Ramp rate and stop conditions.

Ramp Preview - shows a simulated ramp graph based on the specified ramp rate. Click the Expand/Collapse Ramp Preview button to show or hide the preview panel.

Ramp Rate - specifies the rate at which Virtual Users start.

Stop Condition - specifies when the Virtual Users should stop running.

Advanced - specifies advanced ramp conditions for each Virtual User individually.

- **Handle ramping of profiles individually** - when selected, the advanced ramp conditions will used.
- **Profile** - lists the names of the Virtual User profiles in the scenario. Select individual profiles and click the Ramp Rate, Start condition, and Stop Condition to change the default settings.
- **Ramp Rate** - specifies the rate at which Virtual Users start.
- **Start Condition** - specifies when the Virtual Users should stop running.
- **Stop Condition** - specifies when the Virtual Users should stop running.

3.11.3 Goals Section

The Goals section lets you define specific test goals that specify when to stop ramping up Virtual Users.

Goals - specifies the goals and conditions to use as test goals.

- **Add Goal** - Adds a new goal to the set of goals.
- **Stop ramping VUs when** - specifies which goal condition stops the ramping of virtual users.
 - **any of these goals are met** - the ramping of virtual users stops if any of the specified goal conditions are met.
 - **all of these goals are met** - the ramping of virtual users stops only when all of the specified goal conditions are met.

- **Goal 1 (through X)** - specifies the goal criteria for the load test. Click the (+) button to add additional goals and conditions. Click the (X) button to remove goals and conditions.
 - **Add Condition** - add additional conditions to the current goal.
 - **Goal is met when** - specifies the goal condition to achieve:
 - **any of these conditions apply** - when selected, the test goal is achieved when any of the specified goal conditions are met. Specify one or more goal conditions.
 - **all of these conditions apply** - when selected, the test goal is achieved only when all of the specified goal conditions are met. Specify one or more goal conditions.
 - Click the (X) button to remove conditions.
 - **Select Counter** - opens a list of available script counters to use as the goal condition criteria. For each script counter, the following conditions can be specified for the counter value of the test:
 - is greater than [value]
 - is less than [value]
 - equals [value]
 - does not equal [value]
- Results** - specifies the test results to set if the goals are met or not met.
- **When goals are met, mark load session** - specifies the test result to set if the goals are met: No result, Passed, Failed, or custom result.
 - **When goals are not met, mark load session** - specifies the test result to set if the goals are not met: No result, Passed, Failed, or custom result.
 - **Edit results values** - opens options for editing the results values and creating custom values.

3.11.4 Monitoring Section

The Monitoring section lets you define specific target systems and metrics to monitor during the load test.

Monitoring Targets - specifies the metrics for monitoring.

New Configuration - clears the current configuration for creating anew configuration.

Open Configuration - opens a dialog box for selecting a saved configuration to open.

Save Configuration - opens a dialog box for saving the current configuration to a new configuration file.

Add Target - opens a menu for selecting target systems for monitoring.

- **Database** - opens a dialog box for specifying a database system to monitor.
- **Weblogic** - opens a dialog box for specifying a Weblogic system to monitor.
- **Linux** - opens a dialog box for specifying a Linux system to monitor.
- **Windows** - opens a dialog box for specifying a Windows system to monitor.
- **Generic Target** - opens a dialog box for specifying any system to monitor.

- **Existing** - opens a list of existing target configurations that have already been created.

Test Configuration - checks to see whether the Oracle Load Testing server can contact the system and displays an informational message indicating if the system is available.

3.12 Active Session Tab

The Active Session tab is where you can view runtime graphs and reports. These graphs are only available for the running load test session. Use the **Reports** tab to view reports and graphs after the load test has finished running.

To stop the display from being refreshed click the **Pause** button.

To resume refreshing the display, click the **Resume** button. Note that exiting the tab and returning to the tab will also resume refreshing the display.

The Session menu provides options selecting active sessions to attach or detach.

The Session Overview tab shows a thumbnail view of each graph and statistics for the active session. Click on a thumbnail to see a full view of that graphs or reports.

The Virtual User Status tab shows the virtual users grid.

The Runtime Graphs tab is used to create a custom runtime graphs. The Runtime Graphs sub-tab is refreshed according to what is set in the **Graph refresh interval** setting in the reporting options (select **Options** from the **User** menu).

The Virtual User Logs tab shows the virtual user information saved to the log file.

The ServerStats tab shows server statistics for metrics being monitored.

The Sync Point Status tab shows sync point status.

For information about reports and graphs, see [Chapter 11, "Using Graphs and Reports"](#).

3.13 Reports Tab

The Reports tab is where you can view reports and graphs for sessions for which you have saved data for reporting.

For information about reports and graphs, see [Chapter 11, "Using Graphs and Reports"](#).

3.14 Schedule Tab

The Schedule tab is where you add scenarios to the calendar without starting and specify the start and stop times for running the scenario session.

For information about reports and graphs, see [Section 9.2.2, "Schedule without Starting the Scenario Session"](#)..

Managing Systems

This chapter explains how to define the systems that will be used to run Virtual Users for creating load on the application-under -test.

4.1 Systems Manager

The System Manager, accessed by selecting **Options** from the **User** menu, lets you add and remove Systems and create System groups.

The Systems Manager lets you configure four types of systems and groups and has the following options for each:

VU Agent Systems - these are systems that you want to use as remote virtual user agents for running virtual users during a load test. These systems appear in the **Agent Systems** option in the Profiles section of the Composer tab.

- **New System** - displays the Add System dialog box for adding a new VU Agent system.
- **Edit System** - displays the Edit System dialog box for configuring the system name, IP address Start parameters.
- **Delete System** - deletes the selected systems. To select more than one system, hold down the CTRL key.
- **Name** - lists the specified name(s) of the available systems.
- **Host / IP** - lists the specified Host name or IP address of the agent system.

VU Agent System Groups - system groups let you distribute virtual users across multiple VU Agent systems that have been grouped.

- **New Group** - displays the New Group dialog box for configuring a new system group.
- **Edit Group** - displays the Edit Group dialog box for adding or deleting systems from the group.
- **Delete Group** - deletes the selected system groups. To select more than one system group, hold down the CTRL key.
- **Name** - lists the system groups that are available.

ServerStats Data Collectors - these are systems that you want to use as remote data collectors for gathering ServerStats data.

- **New System** - displays the Add System dialog box for adding a new ServerStats data collector.

- **Edit System** - displays the Edit System dialog box for editing the selected data collector.
- **Delete System** - deletes the selected data collectors. To select more than one data collector, hold down the CTRL key.
- **Name** - lists the available data collectors.

Monitored Systems - these are systems that you want to monitor with ServerStats.

- **New System** - displays the Add System dialog box for adding a new system that you want to monitor using ServerStats.
- **Edit System** - displays the Edit System dialog box for editing the selected system.
- **Delete System** - deletes the selected systems. To select more than one system, hold down the CTRL key.
- **Name** - lists the available systems.
- **Host / IP** - lists the specified Host name or IP address of the system.

For more information about using the System Manager, see [Section 4.2, "Defining Systems"](#).

4.2 Defining Systems

Before you can select systems in Oracle Load Testing Scenarios, you must define the machines that are Oracle Load Testing agent systems. Select **Options** from the **User** menu then expand the **Systems** group to define systems that the Oracle Load Testing scenarios can use as agents. Once systems are defined you can add them to System Groups.

You can also add target Monitored Systems in the Monitoring subsection of the **Composer** tab.

Note: See the installation section in the Basics chapter for more information about installing the Oracle Load Testing Agent software on each system and verifying network access between the Oracle Load Testing system and each agent system.

In addition, if you are using Oracle Load Testing ServerStats you must define the data collector systems and systems being monitored.

4.2.1 Adding New VU Agent Systems

To add a new VU Agent system:

1. Select **Options** from the **User** menu to display the Options Manager.
2. Expand **Systems** and select **VU Agent System**.
3. Click **New** to display the Add VU Agent System dialog box.

General - enter the system information.

- **Name** - enter the system name.
- **Host / IP** - enter the host name or IP address of the system.

Start - enter the system information.

- **Port** - enter the port number to use.

- **Username** - enter the user name for agent authentication. The Username is the user name specified for the agent Authentication Profile in the Oracle Load Testing Agent Authentication Manager. The username for the default agent Authentication Profile is JMSAdmin. To view other defined agent Authentication Profile Usernames, select **Oracle Application Testing Suite** from the **Start** menu, then select **Oracle Load Testing Agent Authentication Manager** from the **Tools** submenu. Select an agent Authentication Profile to view the details. On Linux machines, use:

```
<instdir>/jdk/jre/bin/java -jar <instdir>/agentmanager/AMAuthManager.jar
```

to start the **Oracle Load Testing Agent Authentication Manager**.

- **Password** - enter the password for agent authentication. The Password is the password specified for the agent Authentication Profile in the Oracle Load Testing Agent Authentication Manager. The password for the default agent Authentication Profile is blank. To change the password for a defined agent Authentication Profile Usernames, start the **Oracle Load Testing Agent Authentication Manager** and select an agent Authentication Profile to view the details and enter a new password.

Test - checks to see whether the Oracle Load Testing server can contact the system and displays an informational message indicating if the system is available.

Note: If a firewall is running on the agent system, you must make sure Port 7 is open. The controller uses Port 7 to ping the agent system to determine if the system can be accessed or not while adding the agent system using the user interface.

4. Enter the name of the system in the Name field, enter the name or IP address of the system in the Host / IP field, and specify the operating system type.
5. Enter the port and authentication settings.
6. Click **OK**.

Note: Systems and system groups appear in the Systems list on the Build Scenarios tab.

4.2.2 Adding New System Groups

To add a new system group:

1. Select **Options** from the **User** menu.
2. Expand **Systems** and select **VU Agent System Groups**.
3. Either select the group you want to change and click **Edit** or click **New** to create a new group.

Name - enter the name of the system group.

VU Agent Systems - lists the systems that are available to add to the group. Select the systems that you want to add and deselect the systems you want to remove.

4. For new groups, enter a name for the group.
5. Select the systems you want to add from the **VU Agent Systems** list.
6. Click **OK**.

Note: Systems and system groups appear in the Systems list on the Build Scenarios tab.

4.2.3 Adding Systems to Groups

To add systems to groups:

1. Select **Options** from the **User** menu.
2. Expand **Systems** and select **VU Agent System Groups**.
3. Either select the group you want to change and click **Edit Group** or click **New Group** to create a new group.
4. Select the systems you want to add from the Systems list.
5. Click **OK**.

Note: Systems and system groups appear in the Systems list on the Build Scenarios tab.

4.2.4 Adding New ServerStats Data Collectors

To add a new ServerStats Data Collector:

1. Select **Options** from the **User** menu to display the Options Manager.
2. Expand **Systems** and select **ServerStats Data Collector**.
3. Click **New System** to display the Add ServerStats Data Collector dialog box.

General - enter the system information.

- **Name** - enter the name of the data collector.
- **Host / IP** - enter the host name or IP address of the data collector.

Remote Data Collector - enter the port information.

- **Port** - enter the port number to use.
- **Username** - enter the user name for the data collector. The Username is the user name specified for the agent Authentication Profile in the Oracle Load Testing Agent Authentication Manager. The username for the default agent Authentication Profile is JMSAdmin. To view other defined agent Authentication Profile Usernames, select **Oracle Application Testing Suite** from the **Programs** Start menu, then select **Oracle Load Testing Agent Authentication Manager** from the **Tools** submenu. Select an agent Authentication Profile to view the details. On Linux machines, use:

```
<instdir>/jdk/jre/bin/java -jar <instdir>/agentmanager/AMAuthManager.jar
```

to start the **Oracle Load Testing Agent Authentication Manager**.

- **Password** - enter the password for the data collector. The Password is the password specified for the agent Authentication Profile in the Oracle Load Testing Agent Authentication Manager. The password for the default agent Authentication Profile is blank. To change the password for a defined agent Authentication Profile Usernames, start the **Oracle Load Testing Agent Authentication Manager** and select an agent Authentication Profile to view the details and enter a new password.

4. Enter the name of the system in the Name field, and enter the name or IP address of the system in the Host / IP field.
5. Click **OK**.

Note: Systems and system groups appear in the Agent Systems list on the Composer tab.

4.2.5 Adding New Monitored Systems

Monitored systems are those systems that will be monitored using ServerStats. In addition to defining the system, you can have Oracle Load Testing discover the components that are available for monitoring, manually add new component types and components, and configure the data sources to use. Information that is configured here will be available when you configure ServerStats monitoring in the Monitoring section of the **Composer** tab.

You can use the **Add Target** option in the Monitoring section of the **Composer** tab to quickly define a Monitored System with a set of metrics in a configuration. You can also use the Systems Manager in the User Options to define Monitored Systems.

4.2.5.1 Configuring JMX Monitors

A data collector can only monitor one type of JMX monitor at a time. To monitor more than one type of JMX monitor at the same time, you must use a separate data collector for each. All JMX monitors require some set up. Following are the broad steps followed by the specific procedures:

WebLogic 9.0

1. Copy configuration jar files to the data collector machines

WebSphere 6.0, 5.1, 5.0

1. Copy configuration jar files to the data collector machines
2. Update the properties file

WebSphere 6.1, 7.0

1. Copy configuration jar files to the data collector machines
2. Copy keystore/trust store files

The following are the default values for the monitored system:

Port: 8880 (default)

Username: admin (check with the JMX system administrator for changes to the username)

Password: password (check with the JMX system administrator for changes to the password)

Trust Store File: C:\keys\DummyClientTrustFile.jks

Trust Store Password: WebAS

Key Store File: C:\keys\DummyClientKeyFile.jks

Key Store Password: WebAS

WebLogic 8.0, 8.1

1. Copy configuration jar files to the data collector machines

2. Create jar files
3. For WebLogic 8.1, update the properties file

Copying the JMX Server Installation Jar Files

Before these server types can be used, the JMX agent needs one or more specific configuration jar files to be copied from the JMX server installation to the <install_dir>\DataCollector\classes directory on all machines that will be used as data collectors for sampling from that server. The default <install_dir> is C:\OracleATS. The file(s) can be obtained from your application server installation and copied to the appropriate directory as listed in the following table.

Note: Versions prior to 9.20 included the version number in the directory names for the Jar files. If you are upgrading from a previous version of Oracle Application Testing Suite and have JMX monitors configured, you will need to move the Jar files to the directories specified below. If the Jar files are in the previous version locations, you may receive an error message similar to the following message:
 Error loading websphereXX JMX classes--check classpath setting in data collectors OSDC.properties.

Application Name	Files to Copy	Default Directory
Redhat JBoss 4.x	jbossall-client.jar	<install_dir>\DataCollector\classes\jboss
Redhat JBoss 5.x	all *.jar files under <JBoss5 install_dir>/jboss-as/client/	<install_dir>\DataCollector\classes\jboss
Redhat JBoss 6.x	all *.jar files under <JBoss6 install_dir>/client/	<install_dir>\DataCollector\classes\jboss Execute C:\<JBoss6 install_dir>\bin\run.bat -b hostname.com, to start the JBoss server (hostname.com is the FQDN of your machine). Note the -b argument above. JBoss server does not know which hostname to bind (i.e localhost or hostname or hostname.domain) so it must be specified as described above.
Oracle WebLogic 9.0	weblogic.jar, webservices.jar	<install_dir>\DataCollector\classes\weblogic
Oracle WebLogic 9.1	weblogic.jar, webservices.jar	<install_dir>\DataCollector\classes\weblogic
Oracle WebLogic 10.x	wlfullclient.jar (This file must be created using the WebLogic JarBuilder tool. On the WebLogic sever, switch to server/lib and use the command java -jar wljarbuilder.jar to create wlfullclient.jar in the server/lib directory, then copy the file.)	<install_dir>\DataCollector\classes\weblogic

Application Name	Files to Copy	Default Directory
IBM WebSphere 6.0	admin.jar, bootstrap.jar, bsf.jar, classloader.jar, client.jar, commons-el.jar, configmanager.jar, db2j.jar, deployutils.jar, emf.jar, ffcd.jar, filetransfer.jar, ibmcertpathprovider.jar, ibmjceprovider.jar, ibmjsse.jar, idl.jar, iwsorb.jar, j2ee.jar, jacl.jar, js.jar, jspcore.jar, jspruntime.jar, jsptranslation.jar, jspvisitor.jar, mail-impl.jar, mail.jar, management.jar, pluginconfig.jar, ras.jar, runtime.jar, runtimefw.jar, sas.jar, security.jar, soap.jar, tcljava.jar, uddi4j.jar, utils.jar, validationmgr.jar, wasjmx.jar, wasproduct.jar, wccm_base.jar, webcontainer.jar, webservices.jar, wjmxapp.jar, wlmserver.jar, workspace.jar, wsdl4j.jar, wsexception.jar, wsprofile.jar, wssec.jar	<installdir>\DataCollector\classes\websphere
IBM WebSphere 6.1, 7.0	Both versions java\jre\lib*.jar java\jre\lib\ext*.jar WebSphere 6.1 runtimes\com.ibm.ws.admin.client_6.1.0.jar plugins\com.ibm.ws.security.crypto_6.1.0.jar WebSphere 7.0 runtimes\com.ibm.ws.admin.client_7.0.0.jar plugins\com.ibm.ws.security.crypto.jar Trust Files DummyClientKeyFile.jks DummyClientTrustFile.jks	<installdir>\DataCollector\classes\websphere

4.2.5.2 Adding a Monitored System

To add a new monitored system with a ServerStats configuration in the **Monitoring** section of the **Composer** tab:

1. Open the **Composer** tab.
2. Click the Monitoring step.
3. Select **Add Target**.
4. Select the system type or Generic Target.
5. Specify the system information.
6. If you want to use the default ServerStats metric profile for the system type, select the **Apply [type] metric profile automatically** option.
7. Click **OK**.

8. Use the **Add Counters** menu to add additional counters to the configuration.
9. Click **Test** to verify the metrics can access the target monitored system and retrieve metric data.
10. Optionally, save the configuration.

To add a new monitored system using the Systems Manager:

1. Select **Options** from the **User** menu to display the Options Manager.
2. Expand **Systems** and select **Monitored System**.
3. Click **New System** to display the New Monitored Target dialog box.

This dialog box lets you configure systems that are going to be monitored using ServerStats. You can manually add system components and component types, discover components, and configure data sources.

General

- **Name** - enter the name of the monitored system.
- **Host / IP** - enter the host name or IP address of the monitored system.

Data Sources

Following are the options for each type of monitored system. You only need to specify the settings for the type of monitored system you are adding. For example, if the monitored system is a database, then you need to specify the database settings. You can also configure this information from ServerStats when you configure the monitor.

Database - refer to the JDBC-ODBC documentation for information on configuring your database data source.

Oracle Load Testing uses a JDBC driver to connect to your database. When you select any driver other than Custom, the appropriate settings for that driver are automatically provided. Use these guidelines to select and configure the appropriate driver for your database.

For monitoring enterprise level databases, the native drivers (Oracle Thin JDBC driver) is recommended over the JDBC:ODBC Bridge option. The following are the driver options:

- **Oracle Thin JDBC Driver** - This driver option applies to Oracle databases. This driver is installed automatically as part of Oracle Load Testing Data collectors.
- **Sun JDBC:ODBC Bridge Driver** - This driver option is available as an option for SQL and Oracle databases and any other database for which you have an ODBC driver. This bridge driver is installed automatically as part of Oracle Load Testing.
 - **SQL Database** - The SQL Server ODBC driver is installed with MSDE and Microsoft SQL Server. If you do not have either of these on the Oracle Load Testing server and you are using a remotely installed SQL database for Oracle Load Testing, you need to install the SQL Server ODBC driver on the Oracle Load Testing machine and set up an ODBC DSN. The ODBC driver is included with the SQL Server Client utilities.
 - **Oracle Database** - You must set up an Oracle ODBC on the Oracle Load Testing machine in order to use this driver.

- **Driver** - Select a driver type from the list: Oracle Thin JDBC driver, Sun JDBC:ODBC Bridge, or Custom. You must have the appropriate driver installed on the Oracle Load Testing machine to set up a Database monitor.
- **Driver String** - This information will vary depending on the type of database that you are monitoring. If you selected any option other than Custom, the appropriate string is automatically displayed. For example, this is the string for the Oracle Thin JDBC driver:

```
oracle.jdbc.driver.OracleDriver
```

If you selected a Custom driver type, you can type in the Driver String yourself.

- **Connect String** - For most drivers, this string is constructed from the information you supplied in the previous fields. The structure of the Connect String is different for each driver type, but Oracle Load Testing builds this string for all driver types except a Custom driver type. For a Custom driver setting, type in the Connect String.
- **Database Name** - For the Oracle Thin JDBC driver, provide the database or server ID.
- **Port** - Oracle Load Testing displays the default port for the driver you selected. For example, the default port for an Oracle Thin JDBC driver is 1521. Modify the port number if necessary. This is not required for a JDBC:ODBC or Custom driver setting.
- **Instance** - Specify the SQL server named instance that you want to use. If nothing is specified, Oracle Load Testing uses the default instance as set up on your server. Refer to your database administrator for details.
- **Database Name or Database SID** - For the Oracle Thin JDBC driver, provide the database or server ID.
- **Username** - enter the username for connecting to the database, if required.
- **Password** - enter the password for connecting to the database, if required.

WebSphere

- **Port** - enter the port number for the connection.

JMX

- **Server Type** - select the JMX server type you are using. The following is a list of supported types. Other types may be supportable. Contact support for more information.
 - Oracle WebLogic
 - IBM WebSphere
 - Redhat JBoss
- **Port** - enter the port number for the connection.
- **Username** - enter the username for logging on to the server.
- **Password** - enter the password for logging on to the server.
- **Trust Store File Name** - enter the client-side trust store path and file name.
- **Trust Store Password** - enter the trust store file password.
- **Key Store File Name** - enter the client-side key store path and file name.

- **Key Store Password** - enter the key store file password.

Note: Refer to your system administrator for information on configuring your server.

Perfmon (Windows Performance Monitor) - authentication can be left blank if the system being monitored has a data collector running on it. Authentication is required when the system being monitored is remote to the data collector.

- **Username** - enter the username for logging on to the system.
- **Password** - enter the password for logging on to the system.
- **Domain Name** - enter the domain or machine name of the user name account.

Url

- **System Homepage** - enter the URL of the page you want to monitor.

Virtual Agent

- **Connection Protocol** - Specifies the protocol to use to execute the command, Local Machine, Telnet, or SSH. You can use the Local protocol to monitor a remote machine if you have a Data Collector installed on that machine. Plink must be installed in the datacollector\bin directory on the machine on which you are running the Data Collector. This is only required if you intend to use the SSH connection method.
 - **Remote Port** - Specifies the port number. The default port for Telnet is 23, and the default port for SSH is 22.
 - **Remote Username** - Specify a user name to log into an account on the host system.
 - **Remote Password** - Specify the password required to log into the User Name account on the system.
 - **Command Prompt** - Specify the prompt for the host machine. If you do not specify a prompt, Oracle Load Testing will attempt to infer the prompt by parsing the screen output. The default command prompt for the root user is #. For other users that have not configured a custom prompt, the default is \$.
 - **Operating System** - Specifies the operating system of the host machine.
4. Enter the name of the system in the Name field, and enter the name or IP address of the system in the Host / IP field.
 5. Enter data source information for the type of system you are adding.
 6. Click **OK**.

Note: Systems, system groups, and cloud service configurations appear in the Agent Systems list in the Profiles pane of the Composer tab.

4.2.6 Renaming Systems

To rename a system:

1. Select **Options** from the **User** menu.
2. Expand **Systems** and click the type of system that you are renaming.

3. Select the system you want to rename.
4. Click **Edit**.
5. Enter the new name in the Name field. All instances of the system (within groups) are also renamed automatically.
6. Click **OK**.

4.2.7 Editing Systems

To edit a system:

1. Select **Options** from the **User** menu.
2. Expand **Systems** and select the system you want to edit.
3. Click **Edit** to display the Edit System dialog box for that type of system.
4. Make any changes.
5. Click **OK**.

4.2.8 Deleting Systems

To delete a system:

1. Select **Options** from the **User** menu.
2. Expand **Systems** and click the type of system you want to delete.
3. Select the systems you want to delete. To select more than one system, hold down the CTRL key.
4. Click **Delete**. All instances of the system (within groups) are also deleted automatically.
5. Click **OK**.

4.3 Estimating Hardware

The Hardware Estimation tools provide ways for you to get an estimate of the amount of hardware that will be required to run an Oracle Load Test session for a given scenario.

4.3.1 Estimate for an Oracle Load Testing Scenario

The quick estimate is a simple, quick and quite accurate solution for estimating the hardware requirements of an OpenScript script that will be used as a Virtual User in an Oracle Load Test scenario.

The estimate process consists of the following:

- Running a script in no delay mode and collecting hardware consumption data (CPU/Memory/Network) for only one VU for a short period of time.
- Measuring the CPU and Memory consumption data of a VU.
- Deducing hundreds or thousands of VUs running concurrently with simulated think time.
- Estimating the hardware requirements of the specified number of VUs based on the deduction result.

To run a hardware estimate:

1. Add scripts to the **Composer** tab and save the scenario.

Note: The Hardware Estimation feature does not support Block Scenario scripts.

2. Click the **Hardware Estimation** toolbar button in the Profiles pane..
3. Click **Generate Now** to start a hardware estimation session for the selected scenario. A background session and local agent start and run one Virtual User for the selected scenario.
4. When the estimation finishes, click **View Report** to view the results.

Creating Scenario Profiles

This chapter explains how to define scenario profiles based upon OpenScript scripts and Oracle Load testing profile properties.

5.1 Adding Profiles to a Scenario

Virtual user profiles are Oracle OpenScript scripts plus the specified Oracle Load Testing attributes. Virtual user scenarios specify the following attributes for each profile:

- number of virtual users to simulate
- the system on which the virtual users will run
- which browser to emulate
- the delay time between virtual user iterations
- pacing of the scripts
- type of user to simulate
- object downloading
- use of Data Banks
- use of synchronization points
- error handling for virtual users
- setting up virtual users for viewing in the virtual user logs component

The following sections explain how to specify virtual user scenarios.

5.1.1 Adding Profiles to a Scenario

Once you have a set of OpenScript scripts, you can define Oracle Load Testing profiles to create scenarios to run performance and load testing on your Web pages or application.

To add script profiles to the Scenario list:

1. If necessary, open a saved Scenario file.
2. Select the **Composer** tab.
3. Select the repository and workspace containing the profile you want to add.
4. Select the script you want to add as a Virtual User profile in the **Available Scripts** list.

Note: For databank settings to work properly, the name of the profiles added to the scenario should be unique.

5. Click the **Add** button or drag the script name in the **Available Scripts** list to the Profile section.
6. If you add a password protected script to configure parameters in the **Composer** tab, the **Import Script Password** dialog box opens for specifying the script password. Enter the script password and click **Validate** to verify the password is correct.
7. Repeat steps 5 and 6 to add additional scripts to the Virtual User profile.
8. Once you select the scripts to include in the scenario, you can specify the Profile Properties for each one.
9. When finished adding scripts, click **Ramp** to specify the ramp-up, and stop settings for the scenario.
10. Click **Goals** to specify the test goals for the scenario.
11. Click **Monitoring** to specify ServerStats monitors for the scenario.
12. .Optionally, save the scenario.

5.1.2 Removing Profiles From a Scenario

To remove profiles from the Scenario Profiles list:

1. If necessary, open a saved Scenario file.
2. Select the **Composer** tab.
3. Select the profile in the **Profiles** list.
4. Click the **Delete** button.
5. Save the scenario by selecting **Save** from the **Scenario** menu.

5.1.3 Specifying Scenario Profile Properties

Once you select the scripts to include in the scenario, you can specify the Scenario properties for each one.

To set the properties for a specific script:

1. If necessary, open a saved Scenario file.
2. Select the Composer tab.
3. Select the scripts to include in the Scenario.
4. Edit the properties in the **Properties** pane. The tabs in the **Properties** pane shows the settings for the script selected in the **Profile** pane. Select the script name in the **Profile** pane and set the properties. Note that the properties differ between load test-type (protocol based) scripts and functional test-type (DOM based) scripts.

Note: A subset of the parameters in this dialog box can be specified directly in the Profile pane of the Composer tab. To specify which parameters are displayed in the Profile pane, select **Options** from the **User** menu then select **Default Properties** in the **Profiles** group and select the fields you want to display by selecting the **Table View** checkbox for each field.

5. Select the script or user defined profile that you want to edit from the list. See the preconditions for functional testing-type scripts in [Section 3.1.2, "Preconditions for Using Functional Testing Scripts"](#).
6. Set the properties.
7. Repeat steps 5 and 6 for each script and user defined profile or use the copy and paste functions on the right click menu to copy settings from one script and user defined profile to another.

5.1.3.1 Load Test Script Settings

Right Click Menu - lets you copy and paste settings from one script or user defined profile to another. Right click on the script or user defined profile whose settings you want to copy and click **Copy Settings**. Right click on the script or user defined profile that you want to copy the settings to and click **Paste Settings**.

The following are the settings for Load test-type profiles:

General - the main settings are as follows:

- **Max. # VUs** - specifies the number of virtual users to run for the selected profile. For each virtual user, Oracle Load Testing runs a separate instance of the script(s) specified in the virtual user profile.
- **Agent System** - specifies the machine on which the virtual users will run. When running virtual users across systems on a LAN/WAN, enter the machine name of a system running either Oracle Load Testing or Oracle Load Testing Agent. Systems are defined using the Systems Manager. Initially, you must define the machine names or IP addresses of the system(s) in the Systems Manager. Once the name(s) or IP addresses have been specified, you can select the system name from the drop-down list for future load tests.

When determining the number of virtual users to run per process or system, you need to include the Client overhead in the resource allocation. Each VU requires approximately 350 KB-500 KB of memory to run. When calculating the available memory to run VUs on an agent system, you must account for a 20-30% client system overhead. Therefore, you only have 70-80% of the physical memory (RAM) available to run VUs.

- **Iteration Delay** - specifies the amount of time (in seconds) to wait between iterations of virtual user runs.
- **VU Pacing (Think Time)** - specifies the script playback delay for each virtual user. There are four options:
 - **Recorded** - uses the delay times that were recorded in the Oracle OpenScript script. You can set minimum and maximum delay times (in seconds) that override the script delay times in the Minimum and Maximum edit boxes.
 - **Recorded/Random** - uses random delay times based upon the recorded user delay. Oracle Load Testing sets the low end of the random range as the actual user delay minus the Lower percentage setting. Oracle Load Testing sets the

high end of the random range as the actual user delay plus the Upper percentage setting. For example, if the actual recorded delay time was 100 seconds and the Lower and Upper settings are 10% and 25% respectively, Oracle Load Testing uses random delay times between 90 and 125 seconds.

- **Random** - uses random times for Virtual User pacing. You can set minimum and maximum delay times for random delay in the Minimum and Maximum edit boxes.
- **No Delay** - plays back the scripts at the fastest possible speed.

Note: For OpenScript scripts, the VU Pacing overrides the times specified in `think()` and `beginStep()` methods.

Data Bank - the Data Bank load testing playback settings are as follows:

- **Use Data Bank** - when true, scripts that have Oracle OpenScript Data Banks will use the Data Banks as part of the virtual user playback. When false, scripts playback using the recorded data rather than the Data Bank.
- **Setup Timeout:** Specifies how much time to spend preparing a databank for use before timing out. The value is in seconds. This setting includes the total time to do all of the following activities:

If using a Database-backed databank:

- Connect to the database
- Query
- Read records, write into the file
- Create the index simultaneously
- Disconnect

If using a CSV-backed databank:

- Time required to parse the CSV file and create the index

If using Random Unique:

- Time to shuffle the index

- **Read Timeout** - specifies the amount of time to wait for a databank read or get operation for a script at run-time before timing out.

Sync Points Settings- the Sync Point settings are as follows:

- **Name** - specifies the name of the Synchronization point. This is the same name as specified for the sync point in the OpenScript script.
- **Group type** - specifies how to determine when to release virtual users based on the following options:
 - **All users in scenario** - calculates when to release virtual users based on all virtual users whether or not they are running.
 - **Running users** - calculates when to release virtual users based on the number of running users only.
- **Release trigger** - specifies the percentage of virtual users that must be waiting before they are released. For example, if ten virtual users are running and the release trigger is set to fifty percent, then the waiting virtual users are released

when five of them are waiting. Setting the value greater than one hundred percent causes the virtual users to wait until they are manually released.

- **Open gate time** - specifies the amount of time to keep the release open after it has been triggered.
- **Maximum Wait** - specifies, per individual virtual user, the maximum amount of time, in seconds, that the synchronization point will wait before releasing the virtual user.

Browser Settings- the browser settings are as follows:

- **Browser Emulation** - specifies the type of browser to emulate.
- **Connection Speed Emulation** - specifies the line speed to simulate for the virtual user's Internet connection. Set the speed to a specific number if you want the virtual user to simulate a dial-up connection using a modem, DSL, or other speed. Set the speed to True Line Speed if you want the virtual user to run using the actual connection speed.
- **Cache Download Pages** - when true, downloaded pages are stored in a local cache and caching options are enabled. Caching places less of a load on the server as only newer pages are requested and brought down from the Web server. When false, caching is not used. No caching places more of a load on the Web server because pages and images are brought down from the Web server for every request.
 - **Clear Cache After Iteration** - when true, the cache is cleared after each iteration of the script.
 - **When Page Out of Date** - when true, the cache is cleared automatically when the page is out of date.
 - **Every Visit** - when true, the cache is cleared on every visit to the page.
 - **Max. Cache Size (MB)** - specifies the maximum cache size.
- **Use IP Spoofing** - when true, Oracle Load Testing uses different IP addresses for Virtual User agents. Each virtual user must get a defined IP address. You must define the IP addresses available for use by Oracle Load Testing Agents in the TCP/IP network protocols of the system. All IP addresses must be added to each Agent system. See [Section 5.1.7, "Using IP Spoofing"](#) for additional information.
- **Enable Cookies** - when true, the virtual user profiles will use cookies. Use this setting if your Web application uses cookies to manage session and other context information.

Extensibility - the extensibility settings are as follows:

- **Execute User Defined Tests** - when true, Oracle Load Testing runs Oracle OpenScript Text Matching and Server Response tests.

Virtual User Logs - the VU Logs settings are as follows:

- **Enable Logging** - turns VU logging on and off. The default is on. When On, the **Message Delivery** and **Logged Messages** settings are also enabled.
- **Message Delivery** - specifies when messages are delivered to the Virtual User log, as follows:

On Error - enables delivery of messages only when an error occurs. Using this a user can debug what happened on a particular step or transaction when an error occurred. All messages, as specified by the Logged Messages settings, for steps or transactions are cached an error occurs.

Always - all messages generated by Virtual Users will be logged.

- **Logged Messages** - specifies the type of logged messages, as follows:

Standard - The standard messages consist of basic level messages which provide an overview of the chronological flow of a Virtual User. The types of messages included in this are as follows:

- BeginPage -Logs the step-group (page) name, when VU starts a page.
- FoundResource - Logs the resources' urls when download manager is turned on and discovers resources from pages.
- ScriptError [Without stack trace] - Logs the script exception type and messages, when an OATS defined exception happens. It does not matter if the 'Error Recovery' settings handles it as 'warn' or 'ignore'. The name of the exception class is appended to "ScriptError" as a whole message type, for example, ScriptError<SolveException>
- CachedData - Logs the cached resources' urls, when a VU requests on a cached resource (304 NOT MODIFIED or Found In Cache).
- ThinkTime - Logs a message with the think time in seconds when a VU is in iteration delay, step delay, or manual delay.
- SyncPoint - Logs whether a VU is suspended by a Sync Point or continues from a Sync Point.
- Action - Logs the details of an action when a VU is navigating to a page (http), or executing a sql statement (util).

Extended - The extended messages consist of all the message types included in Standard plus selective inclusion of extended message types, which can have a substantial overhead. Selecting this option enables the selection of the previously excluded message types. All these message types or their groups are turned off by default. The extended message types or their groups are as follows:

- Server Communication Content - Enables logging of all contents that are communicated with the server. For example, for an HTTP script it will consist of RequestHeader, ResponseHeader and ResponseContent.
- Parameter Substitution - Enables logging of the variables name/value being substituted when parameters are transformed (messages of type ParameterSubstitution).
- Error Stack Trace -Enables logging of messages of type ScriptError to be reported with the stack trace in the content.
- User Defined Messages - Enables logging of the messages if API 'info()' , 'warn()', 'fail()', 'reportFailure()' methods are used (messages of type CustomizedLog).
- Verification Notifications - Enables logging of the test type, test name, and test result of all types of verifications/tests (messages of type Verification).

Reporting- the Reporting settings are as follows:

- **Generate Timers For Pages** - when true, Oracle Load Testing automatically adds timers for each script page for reporting. The timers are used in Oracle Load Testing to provide performance monitoring and timing information for each page of the script played back by a scenario.
- **Generate Timers For Step Groups** - when true, Oracle Load Testing automatically adds timers for each OpenScript Step Group for reporting. The timers are used in

Oracle Load Testing to provide performance monitoring and timing information for each Step Group the script(s) played back by a scenario.

- **Generate Timers For Resources** - when true, Oracle Load Testing automatically adds timers for all resources for monitoring and reporting purposes. Resources include images and other objects downloaded from the server as specified by the **OpenScript Download Manager** section of the Scenario Defaults.

Error Handling - the Error Handling settings are as follows:

- **On Error Stop Virtual User** - when true, a virtual user is stopped if an error is encountered.
- **Stop Remaining Iterations on Failure** - when true, all remaining iterations for a virtual user are stopped if an error is encountered.
- **Socket Timeout** - specifies the maximum amount of time a virtual user waits for a socket connection before timing out.
- **Request Timeout** - specifies the maximum amount of time a virtual user waits to access a page before timing out.
- **Connection Idle Timeout** - specifies the socket 'idle timeout' and uses a new connection when reusing an idle-timeout socket. This is used to specify the timeout for a socket that gets closed by the server side after a long idle period.

Advanced - the Error Handling settings are as follows:

- **Maximu Users Per Process** - sets the maximum number of virtual users per single agent process. When running virtual users as threads in a single process, **Max Users Per Process** sets the maximum number of virtual user threads in a single process. Oracle Load Testing spawns new processes if the number of virtual users exceeds the maximum number in any single process and runs the additional virtual uses as threads in the new process.

The default setting is unlimited virtual users per agent process.

- **Max HTTP Connections Per User** - specifies the maximum number of server connections per process per server. Each VU makes multiple connections to request additional resources for images and additional frames for example. Setting this option specifies a limit on the total number of connections that the VU s can make to the server. The default setting is "Default," which means use the default connection limits as configured on the agent machine. (See Microsoft KBase article Q183110 for more information.)
- **Ignore HTTP Proxy Settings** - specifies whether to ignore the agent machine's default proxy setting as defined in Internet Explorer.
- **Debug Users** - specifies whether to enable debug features.

Agent Preferences - When a setting is set to the default value, this means that the value that will be used is what is set in the OracleATS\OFT\jagent\JavaAgent.properties file, unless a value is not set in the JavaAgent.properties file. In this case, the Java Agent uses the internal default value.

- **Persist Raw Data** - when true, Oracle Load Testing saves every single measured data point to the database.

See [Section 11.10, "Using Raw Data"](#) for additional information about how to use the raw data.

- **Report Counters** - when true, Oracle Load Testing counters are reported.

- **Report Sender Interval** - when you select other, enter the time in milliseconds for how frequently the agent reports its status and accrued counters. The default in the JavaAgent.properties file is 5000.
- **Maximum JVM Heap Size (MB)** - specifies the maximum size of the JVM heap. This value cannot be more than 90% of the total memory size.
- **Proxy Host** - select other to enter the proxy host and override the system-specified proxy host.
- **Proxy Port** - select other to enter the proxy port and override the system-specified proxy port.
- **Non Proxy Hosts** - select other to enter non-proxy hosts. Delimit multiple hosts with a bar (|).
- **Enable GZIP** - when true, support for gzip compression is enabled. The browser Request includes the Accept-Encoding: gzip header indicating a gzip compressed page response will be accepted. If the server uses gzip compression, the response includes the Content-Encoding: gzip header indicating the returned page is in gzip compressed format. The browser unzips the compressed file before rendering the HTML page. Gzip compression is typically used to provide faster transfer of large HTML pages between the browser and the server.
- **Enable Deflate** - when true, support for deflate compression is enabled. The browser Request includes the Accept-Encoding: deflate header indicating a deflate compressed page response will be accepted. If the server uses deflate compression, the response includes the Content-Encoding: deflate header indicating the returned page is in deflate compressed format. The browser inflates the compressed file before rendering the HTML page. Deflate compression is typically used to provide faster transfer of large HTML pages between the browser and the server.
- **Language** - when you select other, enter the language to override the Accept-Language header. The default is the locale assigned by the JVM.
- **HTTP Version** - select the HTTP protocol version to specify in the GET or POST request/response between client and server. The HTTP/1.0 protocol is an early implementation of the Hypertext Transfer Protocol. HTTP/1.1 is a standards-based enhancement to the HTTP/1.0 protocol. See the Key Differences between HTTP/1.0 and HTTP/1.1 at <http://www8.org/w8-papers/5c-protocols/key/key.html>
- **Accept String** - this setting controls what the Accept: HTTP header value looks like. When you select other, enter the string. The default in the JavaAgent.properties file is: text/html, image/gif, image/jpeg, */*. If you modify a navigation in a script by adding a custom Accept: header, the custom header value from the script is used instead.
- **Enable Keep Alive** - when true, the connection request header is set to Connection: Keep-Alive. For HTTP/1.0, the socket connection is kept open until either the client or the server drops the connection. For HTTP/1.1 all connections are kept alive unless a Connection: close header is specified.
- **Preserve Connections Between Iterations** - used to preserve connections between Virtual User agents and the browser between successive iterations of the script. Set to True if the browser should attempt to reuse any open browser connections if possible between iterations. Each virtual user maintains its own set of connections that it never shares with other virtual users. The default value is True, preserve connections between iterations.

- **Preserve Variables Between Iterations** - used to preserve or automatically clear variables added in the **Run** section of OpenScript scripts between successive iterations of the **Run** section.
- **Preserve Cookies Between Iterations** - used to preserve or automatically clear cookies added in the **Run** section of OpenScript scripts between successive iterations of the **Run** section.
- **Max Keep-Alive Requests** - select other to specify the maximum number of requests to make on a keep alive connection before closing it.
- **Download Local Files** - when true, the Java Agent retrieves the requested local file contents.
- **Max Content Download Size** - specifies the maximum size for downloads. You can specify Unlimited or Other. If you select Other, specify the maximum size in kilobytes.
- **Secure Protocol** - select the Secure Socket Layer version to use for the proxy server. When recording a secure site in the browser, the user only sees the Proxy Recorder's certificate not the secure web site's certificate. The Browser, Proxy Recorder, and Secure Server each have their own private and public keys which are used to encrypt/decrypt data.
 - **SSL:** Use Secure Socket Layer protocol with the proxy server. OpenScript supports SSLv1.0, SSLv2.0, SSLv3.0 and TLSv1.0.
 - **TLSv1.2:** Use Secure Socket Layer Transport Layer Security version 1.2.
- **Do Not Request URLs ending in** - ignores requests in the script that end with the extensions specified in this field. Specify the file extensions, separated by commas, that should not be requested. For example, `.png, .ico, .gif, .jpg, .jpeg, .css, .js`. This setting only applies to only to OpenScript load test scripts.
- **Additional Arguments** - specifies custom OpenScript script.java code arguments. You can create your own settings in OpenScript scripts. For example, you can create custom settings in OpenScript script.java code, as follows:


```
if (getSettings().get("MyCustomSetting").equals("abc")) {
    info("We're running in ABC mode.");
}
```

You can then set the additional arguments in the **Additional Arguments** field as follows:

```
-MyCustomSetting abc
```
- **Global Headers** - specifies any custom "Global Headers" string to use in the Request header for script playback. The format is in the form: `name1:value1;name2:value2;name3:value3`. For example: `x-oracle-slm-message-id: bcn=<beacon_name>; svc=<service_name>`.
- **Replace URLs** - specifies the URL replacement string in the form: `originalURL1=replacementURL1, originalURL2=replacementURL2, [...]`. During playback, anytime the agent makes a request to a URL starting with a segment, `originalURL`, the agent replaces the original URL segment with `replacementURL`. This feature is only supported for Load Test scripts.
 - `originalURL` - Specify the starting segment of the `URL:port` that appears in the script that should be replaced. This value is case-sensitive.

- *replacementURL* - Specify the new starting segment *URL:port* that the agent requests instead of *originalURL*.

For both parameters, if the protocol is omitted, HTTP protocol is assumed. If no port is specified after the host, port 80 is assumed for HTTP protocol, and port 443 is assumed for HTTPS protocol. URLs are replaced after all correlations are applied. One or more URL replacement pairs may be specified, separating each replacement pair with a comma. The following examples show the format of **Replace URLs** strings:

```
test_server:7789=production_server:7789
```

```
test:7789=prod:7789,https://stage.oracle.com/main=https://prod.oracle.com/home
```

Error Recovery - The OpenScript Error Recovery categories lets you specify error recovery actions for exceptions that occur during playback. You can set the error recovery action for individual playback exceptions. You can set the action as Fail, Warn, Ignore, or Report as follows:

- **Fail:** Report the error as failure and stop script execution.
- **Warn:** Report the error as a warning and continue script execution.
- **Ignore:** Ignore the error and continue script execution.
- **Report Error and Continue:** Report the error and continue script execution.

Error Recovery playback preferences specified in the OpenScript Preferences are stored on the local machine and only apply when the script is played back from inside OpenScript on that machine. If you import your script to Oracle Load Testing on another server and your script depends on an error recovery setting being a certain way in order for it to work, then you can set the error recovery setting in the OpenScript script Java code.

In OpenScript scripts, error settings can be turned on and off at any time, overriding the default Oracle Load testing and OpenScript Preferences using script Java code. For example:

```
getSettings().setErrorRecovery("http.zeroLengthDownloads", "IGNORE");
// user code executed in script, such as http.get(), http.post(), ...
getSettings().setErrorRecovery("http.zeroLengthDownloads", "FAIL");
```

See the OpenScript documentation for a list of error recovery settings.

Error Recovery - General - the General Error Recovery settings are as follows:

- **File Not Found** - specifies the error recovery action if a file is not found.
- **Segment Parser Error** - specifies the error recovery action if the XPath Segment Parser cannot verify the correctness of an XPath.
- **Create Variable Fail** - specifies the error recovery action if a script fails to create a variable.
- **Encryption Service not Initialized** - specifies the error recovery action when the password encryption service was not initialized.
- **Binary Decoding Error** - specifies the error recovery action if a binary post data parameter error occurs.
- **Variable Not Found** - specifies the error recovery action if a variable cannot be found when parsing transformed strings.

- **Unexpected Script Error** - specifies the error recovery action if any unexpected script error occurs.
- **Child Script Failed** - specifies the error recovery action if an error occurs in a script that is a child of another script.
- **Call Function Failed** - specifies the error recovery action if an error occurs in a script that calls a function of another script.
- **Encrypting/Decrypting Failed** - specifies the error recovery action if an error occurs encrypting or decrypting a script.

Error Recovery - HTTP - the HTTP Module Error Recovery settings are as follows:

- **HTML Parsing Error** - specifies the error recovery action if an HTML parsing error occurs.
- **Text Match Fail** - specifies the error recovery action if a text matching test fails.
- **Solve Variable Fail** - specifies the error recovery action if the value of any variable cannot be solved.
- **Response Time Error** - specifies the error recovery action if a Server Response Time test fails.
- **Invalid HTTP Response** - specifies the error recovery action if the sever returns an invalid HTTP response.
- **Invalid URL** - specifies the error recovery action if the server returns an Invalid URL response code.
- **Zero Downloads Fatal** - specifies the error recovery action if a server response indicates zero bytes length.
- **Client Certificate Keystore Error** - specifies the error recovery action if the Client Certificate Keystore indicates an error.

Error Recovery - Oracle Forms Load - the Oracle Forms Load Test Module Error Recovery settings are as follows:

- **Forms Connect Error** - specifies the error recovery action if a server connection error occurs.
- **Forms I/O Communication Error** - specifies the error recovery action if a read/write or communication error occurs with an Oracle Forms message.
- **Forms Playback Error** - specifies the error recovery action if an error occurs during forms playback.
- **Forms Component not Found** - specifies the error recovery action if a component of a form is not found.
- **Forms Content Match Failed** - specifies the error recovery action if a content matching test fails.

Download Manager - the OpenScript Download Manager settings are as follows:

- **Use OpenScript Download Manager** - when true, the Download Manager is enabled during playback. When false, the Download Manager is not enabled during playback.
- **CSS Resource** - when true, css resources in <Link> tags are downloaded during playback. When false, css resources are not downloaded during playback.
- **Image Resource** - when true, image resources in tags, in the "background" attribute of a tag, or in <style> tags with "background:url" patterns are

downloaded during playback. When false, image resources are not downloaded during playback.

- **Embedded Object Resource** - when true, object resources in <Embed> tags or in <Object> tags are downloaded during playback. When false, object resources are not downloaded during playback.
- **Script Resource** - when true, script resources in <Script> tags are downloaded during playback. When false, script resources are not downloaded during playback.
- **Applet Resource** - when true, applet resources in <Applet> tags are downloaded during playback. When false, applet resources are not downloaded during playback.

Forms LT Playback - the Oracle EBS/Forms load testing playback settings are as follows:

- **Capture Message Details:** Specifies if forms message details are captured during playback. When selected, OpenScript captures and stores Forms message requests, responses, and information about all loaded Forms components during playback. This information is useful to have when debugging the script.

OpenScript displays captured details in the "Messages" and "Object Details" tabs of the Details view. Oracle Load Testing displays this information in the virtual user logs based on the "virtual user logs" settings.

Capturing message details is a memory-intensive operation. During heavy load testing, it is recommended to clear this setting to reduce the amount of heap space required by the agent.

- **Heart Beat Interval (sec):** Specifies how often to check the connection to the EBS/Forms server.

5.1.3.2 Functional Test Script Settings

The following are the settings for Functional test-type profiles:

General - the general settings are as follows:

- **Max # VUs** - specifies the number of virtual users to run for the selected profile. For each virtual user, Oracle Load Testing runs a separate instance of the script(s) specified in the virtual user profile.
- **Agent System** - specifies the machine on which the virtual users will run. When running virtual users across systems on a LAN/WAN, enter the machine name of a system running either Oracle Load Testing or Oracle Load Testing Agent. Systems are defined using the Systems Manager. Initially, you must define the machine names or IP addresses of the system(s) in the Systems Manager. Once the name(s) or IP addresses have been specified, you can select the system name from the drop-down list for future load tests.

When determining the number of virtual users to run per process or system, you need to include the Client overhead in the resource allocation. Each VU requires approximately 350 KB-500 KB of memory to run. When calculating the available memory to run VUs on an agent system, you must account for a 20-30% client system overhead. Therefore, you only have 70-80% of the physical memory (RAM) available to run VUs.

- **Iteration Delay** - specifies the amount of time (in seconds) to wait between iterations of virtual user runs.

- **VU Pacing (Think Time)** - specifies the script playback delay for each virtual user. There are four options:
 - **Recorded** - uses the delay times that were recorded in the Oracle OpenScript script. You can set minimum and maximum delay times (in seconds) that override the script delay times in the Minimum and Maximum edit boxes.
 - **Recorded/Random** - uses random delay times based upon the recorded user delay. Oracle Load Testing sets the low end of the random range as the actual user delay minus the Lower percentage setting. Oracle Load Testing sets the high end of the random range as the actual user delay plus the Upper percentage setting. For example, if the actual recorded delay time was 100 seconds and the Lower and Upper settings are 10% and 25% respectively, Oracle Load Testing uses random delay times between 90 and 125 seconds.
 - **Random** - uses random times for Virtual User pacing. You can set minimum and maximum delay times for random delay in the Minimum and Maximum edit boxes.
 - **No Delay** - plays back the scripts at the fastest possible speed.

Note: For OpenScript scripts, the VU Pacing overrides the times specified in `think()` and `beginStep()` methods.

- **Use Data Bank** - when true, scripts that have Oracle OpenScript Data Banks will use the Data Banks as part of the virtual user playback. When false, scripts playback using the recorded data rather than the Data Bank.

Browser Settings- the browser settings are as follows:

- **Browser Type** - specifies the type of browser to use for functional test scripts: Internet Explorer or Firefox. The default is Internet Explorer on Windows. The default is Firefox on Linux. This setting is specific to functional test scripts.
- **Browser Path Override** - specifies an alternative path to use when launching the specified browser type. Explorer and Firefox browser processes physically exist in the file system. In case the path to one of these browsers is incorrect, specify an alternative path to use when launching the specified browser type. This setting is not intended to be used to specify the path to an unsupported browser. This setting is specific to functional test scripts.
- **Browser Additional Arguments** - specifies any additional startup arguments that should be used when launching the browser process on playback. The default is no additional arguments other than what may be required internally. This setting is specific to functional test scripts.
- **Resolution Size** - specifies the screen resolution of the playback agent machine. Functional test scripts with mouse-click actions and screenshot capturing are dependent upon this setting. This setting is specific to functional test scripts.
- **Dismiss Alerts Dialogs** - specifies if JavaScript alert dialog boxes are automatically dismissed if they appear during playback. The default is false; do not automatically dismiss alert dialogs.

Virtual User Logs - the VU Logs settings are as follows:

- **Enable Logging** - turns VU logging on and off. The default is on. When On, the **Message Delivery** and **Logged Messages** settings are also enabled.

- **Message Delivery** - specifies when messages are delivered to the Virtual User log, as follows:

On Error - enables delivery of messages only when an error occurs. Using this a user can debug what happened on a particular step or transaction when an error occurred. All messages, as specified by the Logged Messages settings, for steps or transactions are cached an error occurs.

Always - all messages generated by Virtual Users will be logged.

- **Logged Messages** - specifies the type of logged messages, as follows:

Standard - The standard messages consist of basic level messages which provide an overview of the chronological flow of a Virtual User. The types of messages included in this are as follows:

- BeginPage -Logs the step-group (page) name, when VU starts a page.
- FoundResource - Logs the resources' urls when download manager is turned on and discovers resources from pages.
- ScriptError [Without stack trace] - Logs the script exception type and messages, when an OATS defined exception happens. It does not matter if the 'Error Recovery' settings handles it as 'warn' or 'ignore'. The name of the exception class is appended to "ScriptError" as a whole message type, for example, ScriptError<SolveException>
- CachedData - Logs the cached resources' urls, when a VU requests on a cached resource (304 NOT MODIFIED or Found In Cache).
- ThinkTime - Logs a message with the think time in seconds when a VU is in iteration delay, step delay, or manual delay.
- SyncPoint - Logs whether a VU is suspended by a Sync Point or continues from a Sync Point.
- Action - Logs the details of an action when a VU is navigating to a page (http), or executing a sql statement (util).

Extended - The extended messages consist of all the message types included in Standard plus selective inclusion of extended message types, which can have a substantial overhead. Selecting this option enables the selection of the previously excluded message types. All these message types or their groups are turned off by default. The extended message types or their groups are as follows:

- Server Communication Content - Enables logging of all contents that are communicated with the server. For example, for an HTTP script it will consist of RequestHeader, ResponseHeader and ResponseContent.
- Parameter Substitution - Enables logging of the variables name/value being substituted when parameters are transformed (messages of type ParameterSubstitution).
- Error Stack Trace -Enables logging of messages of type ScriptError to be reported with the stack trace in the content.
- User Defined Messages - Enables logging of the messages if API 'info()' , 'warn()', 'fail()', 'reportFailure()' methods are used (messages of type CustomizedLog).
- Verification Notifications - Enables logging of the test type, test name, and test result of all types of verifications/tests (messages of type Verification).

- Table Test - turns Table Test logging on and off. The default is off. This setting is specific to functional test scripts.
- Object Test - turns XML Test logging on and off. The default is off. This setting is specific to functional test scripts.
- Screenshot - turns Screenshot logging on and off. The default is off. This setting is specific to functional test scripts.
- XML Test - turns XML Test logging on and off. The default is off. This setting is specific to functional test scripts.

Reporting- the Reporting settings are as follows:

- **Generate Timers For Step Groups** - when true, Oracle Load Testing automatically adds timers for each OpenScript Step Group for reporting. The timers are used in Oracle Load Testing to provide performance monitoring and timing information for each Step Group the script(s) played back by a scenario.
- **Generate Timers For Resources** - when true, Oracle Load Testing automatically adds timers for all resources for monitoring and reporting purposes. Resources include images and other objects downloaded from the server as specified by the **OpenScript Download Manager** section of the Scenario Defaults.

Error Handling - the Error Handling settings are as follows:

- **On Error Stop Virtual User** - when true, a virtual user is stopped if an error is encountered.
- **Stop Remaining Iterations on Failure** - when true, all remaining iterations for a virtual user are stopped if an error is encountered.

Advanced - the Error Handling settings are as follows:

- **Object Timeout** - specifies the maximum time to wait for an object to download before timing out.

Agent Preferences - When a setting is set to the default value, this means that the value that will be used is what is set in the OracleATS\OFT\jagent\JavaAgent.properties file, unless a value is not set in the JavaAgent.properties file. In this case, the Java Agent uses the internal default value.

- **Persist Raw Data** - when true, Oracle Load Testing saves every single measured data point to the database.
See [Section 11.10, "Using Raw Data"](#) for additional information about how to use the raw data.
- **Report Counters** - when true, Oracle Load Testing counters are reported.
- **Report Sender Interval** - when you select other, enter the time in milliseconds for how frequently the agent reports its status and accrued counters. The default in the JavaAgent.properties file is 5000.
- **Maximum JVM Heap Size (MB)** - specifies the maximum size of the JVM heap. This value cannot be more than 90% of the total memory size.
- **Preserve Variables Between Iterations** - used to preserve or automatically clear variables added in the **Run** section of OpenScript scripts between successive iterations of the **Run** section.
- **Replace URLs** - specifies the URL replacement string in the form:
originalURL1=replacementURL1, originalURL2=replacementURL2, [. . .]. During playback, anytime the agent makes a request to a URL starting with a segment,

originalURL, the agent replaces the original URL segment with *replacementURL*. This feature is only supported for Load Test scripts.

- *originalURL* - Specify the starting segment of the *URL:port* that appears in the script that should be replaced. This value is case-sensitive.
- *replacementURL* - Specify the new starting segment *URL:port* that the agent requests instead of *originalURL*.

For both parameters, if the protocol is omitted, HTTP protocol is assumed. If no port is specified after the host, port 80 is assumed for HTTP protocol, and port 443 is assumed for HTTPS protocol. URLs are replaced after all correlations are applied. One or more URL replacement pairs may be specified, separating each replacement pair with a comma. The following examples show the format of **Replace URLs** strings:

```
test_server:7789=production_server:7789
```

```
test:7789=prod:7789,https://stage.oracle.com/main=https://prod.oracle.com/home
```

- **Additional Arguments** - specifies custom OpenScript script.java code arguments. You can create your own settings in OpenScript scripts. For example, you can create custom settings in OpenScript script.java code, as follows:

```
if (getSettings().get("MyCustomSetting").equals("abc")) {
    info("We're running in ABC mode.");
}
```

You can then set the additional arguments in the **Additional Arguments** field as follows:

```
-MyCustomSetting abc
```

OpenScript Error Recovery - General - the General Error Recovery settings are as follows:

- **File Not Found** - specifies the error recovery action if a file is not found.
- **Segment Parser Error** - specifies the error recovery action if the XPath Segment Parser cannot verify the correctness of an XPath.
- **Create Variable Fail** - specifies the error recovery action if a script fails to create a variable.
- **Encryption Service not Initialized** - specifies the error recovery action when the password encryption service was not initialized.
- **Binary Decoding Error** - specifies the error recovery action if a binary post data parameter error occurs.
- **Variable Not Found** - specifies the error recovery action if a variable cannot be found when parsing transformed strings.
- **Unexpected Script Error** - specifies the error recovery action if any unexpected script error occurs.
- **Child Script Failed** - specifies the error recovery action if an error occurs in a script that is a child of another script.
- **Call Function Failed** - specifies the error recovery action if an error occurs in a script that calls a function of another script.
- **Encrypting/Decrypting Failed** - specifies the error recovery action if an error occurs encrypting or decrypting a script.

OpenScript Error Recovery - Functional Test - the Functional Test Module Error Recovery settings are as follows:

- **Text Matching Failed** - specifies the error recovery action if a text matching test error occurs.
- **Object Test Failed** - specifies the error recovery action if an Object test error occurs.
- **Table Test Failed** - specifies the error recovery action if a Table test error occurs.
- **XML Test Failed** - specifies the error recovery action if an XML test error occurs.

OpenScript Error Recovery - Web Functional Test - the Web Functional Test Module Error Recovery settings are as follows:

- **Response Time Error** - specifies the error recovery action if a Response time error occurs.
- **Solve Variable Failed** - specifies the error recovery action if a Solve Variable error occurs.
- **Wait for Page Timeout** - specifies the error recovery action if a Wait for Page Timeout occurs.
- **Object Not Found** - specifies the error recovery action if an Object Not Found error occurs.
- **Playback Failed** - specifies the error recovery action if a Playback Failed error occurs.
- **Title Test Failed** - specifies the error recovery action if a Title Test Failed error occurs.
- **HTML Test Failed** - specifies the error recovery action if a Title Test Failed error occurs.

OpenScript Error Recovery - Oracle Forms Functional Test - the Oracle Forms Functional Test Module Error Recovery settings are as follows:

- **Oracle Forms Error** - specifies the error recovery action if an Oracle Forms error occurs.
- **Status Bar Test Error** - specifies the error recovery action if a Status Bar Test error occurs.

Databank Configuration - the Databank Configuration load testing playback settings are as follows:

- **Databank Setup Timeout:** Specifies how much time to spend preparing a databank for use before timing out. The value is in seconds. This setting includes the total time to do all of the following activities:

If using a Database-backed databank:

- Connect to the database
- Query
- Read records, write into the file
- Create the index simultaneously
- Disconnect

If using a CSV-backed databank:

- Time required to parse the CSV file and create the index

If using Random Unique:

- Time to shuffle the index

- **Read Timeout** - specifies the amount of time to wait for a databank read or get operation for a script at run-time before timing out.

Cache and Cookies - the Cache and Cookies settings are as follows:

- **Clear Cache Before Playback** - specifies if the browser cache is cleared before script playback begins.
- **Clear Cache Between Iterations** - specifies if the browser cache is cleared between each playback iteration.
- **Clear Session Cookies Before Playback** - specifies if the session cookies are cleared before script playback begins.
- **Clear Session Cookies Between Iterations** - specifies if the session cookies are cleared between each playback iteration.
- **Clear Persistent Cookies Before Playback** - specifies if the persistent cookies are cleared before script playback begins.
- **Clear Persistent Cookies Between Iterations** - specifies if the persistent cookies are cleared between each playback iteration.

Event Timeout - the Event Timeout settings for EBS/Forms are as follows:

- **Forms Startup Timeout** - specifies the maximum number of seconds playback should wait for a form to appear before considering the form not found. This is the default timeout when waiting for a form to appear before invoking an action against it. This is also the default timeout when waiting for a form to appear before continuing the script.
- **Forms Action Timeout** - specifies the maximum number of seconds Oracle Load Testing should wait for forms action playback until success.
- **Forms Response Timeout** - specifies the maximum number of seconds playback should wait for forms response before timing out.

5.1.4 Determining the Number of Virtual Users

The number of virtual users that can run reliably and provide accurate performance data for your application depends upon several factors:

- Resources required by the application;
- Number of machines being used for the load test;
- Amount of memory on each machine;
- Operating system being used;
- CPU utilization.

The general rules of thumb for determining the number of virtual users are as follows:

- Keep CPU usage below 90% for each system in the load test, including the Oracle Load Testing system and each agent system.
- Keep memory consumption for each workstation in the load test below 70-75% of the physical amount of memory in each workstation. That is, keep the number of virtual users below where page swapping of memory takes place.

- If you need to run more virtual users for your load test than CPU usage or memory limits provide for, you should add more physical memory to a system, or add additional agent systems.

You can use system performance monitoring tools, such as Performance Monitor or Task Manager in Windows, to determine system resource usage. See [Section 4.3, "Estimating Hardware"](#) for additional information about using the Oracle Load Testing Hardware Estimation tool.

5.1.5 Using the Data Bank Control

Oracle Load Testing virtual users can play back scripts that access a Data Bank. You enable Data Bank access by selecting the **Use Data Bank** option in the Data Bank properties tab for the selected virtual user profile. The virtual user must use a script that is configured in OpenScript with a Data Bank asset.

Note: If you have added the same profile to the scenario multiple times, the databank settings for the first instance of the profile in the scenario will apply to all instances of the profile in the scenario.

To use the Data Bank control:

1. Select the **Composer** tab.
2. Add a script that uses a Data Bank to the **Profile** list.
3. Make sure the script or profile is selected in the Profiles panel.
4. In the **Data Bank** tab of the Properties pane, select the **Use Data Bank** option.
5. Select the Data Bank name to show the properties.
6. Specify the Range, Starting Record, and Databank Settings.

The Data Bank names are listed on the left. Select which Data Bank to view.

Preview - shows the Data Bank variables, values, and types, as follows:

Source - shows the path, name, type and number of records of the selected databank file.

Iterate Over Range - when selected, the virtual user uses the specified subset of records in the databank. Specify the first and last records to use for the range as the minimum and maximum values. The range includes both the starting and ending record in the specified range.

- **Start with Record** - specifies which databank record to use first. The first record in a databank is 1. The starting record must be within the specified range of records. For example if you select **Iterate Over Range** and set the range to 5:10, the starting record must be at least 5, but not more than 10.

Advance to Next Record - specifies when the virtual user should advance to the next databank record during script playback. The following options are available:

- **When Script Requests Next Record** - the databank record advances every time a script explicitly requests a record during script playback. A record request corresponds to the script Java code calling the `getDatabank(alias).getNextRecord()` method. This is the default behavior.
- **Each Iteration of Script** - the databank record advances before a script containing the databank starts another playback iteration.

- **Each Occurrence** - the databank record advances when a script refers to a databank column (i.e. databank field) in the script. A record request corresponds to the script Java code evaluating a parameterized value such as `{{db.fmstocks_data.ticker}}`. For example, if you have an employee databank with a `firstName` field and the `firstName` column is specified as the **Column** value in the OpenScript script, the databank record advances only when the `{{db.employees.firstName}}` value in the script Java code is evaluated on script playback.
- **Keep the First Record Assigned** - the databank record does not advance once a virtual user gets a record from the databank, it uses that record forever and never requests another record. This option applies to scenarios running more than one virtual user. Virtual users may still request an individual record using `getRecord(n)`, `getLastRecord()`, or `getFirstRecord()`.

Select Next Record - specifies how a new record is selected from the databank when the databank record advances. The following options are available:

- **Sequentially** - the databank records increment one by one in sequential order from the start of the specified range. When multiple virtual users are running, records are distributed in sequential order across all virtual users.
- **Random** - the databank records are selected at random from the databank. The same record may be used multiple times before all records are exhausted. Random record selection is only provided for databanks that can be indexed. When configuring databank settings, if the databank file is too large to index, the **Random** or **Shuffle** options may not be available. The **When Out of Records** and **Each User Iterates Independently** settings do not apply when Random is selected.
- **Shuffle** - the databank records are selected at random from the databank ensuring that once a record is selected, it is never selected again. The setting works similar to selecting a random card from a deck until no cards are left. Shuffle mode only supports databanks containing fewer than 200,000 records. For databanks containing more than 200,000 records, you can shuffle the values in the actual data file or you should use the Random mode.
- **Use Seed** - specifies a randomization seed value to use when using the **Random** or **Shuffle** modes. Use the same seed across multiple tests to create the same sequence of random numbers for all tests. If 0 or not specified, a seed is generated automatically based on the current time.

When Out of Records - specifies the action the virtual user takes if all databank records in the specified range have been used and a new record is requested. The following options are available:

- **Loop Over Range** - loops back to the first record in the range after all records in the range are used and continues distributing records. Use the **Maximum Iterations** settings to prevent the virtual user from running forever.
- **Stop User** - the virtual user immediately stops running the next time a record is requested from the databank after all records in the range are used. The virtual user will stop regardless of how many iterations are specified by the **Maximum Iterations** settings.
- **Keep Same Record** - continues to use the last record requested after all records in the range are used. No additional records are requested from the databank. Any calls in the Java code to `getNextDatabankRecord()` are ignored after all records are used. Custom Java code may be used in the script to have

Virtual users request an individual record using `getRecord(n)`, `getLastRecord()`, or `getFirstRecord()`.

Each User Iterates Independently - when selected, each Virtual User cursors through records independently. For example, VU 1 gets records 1, 2, 3, [...] and VU 2 gets records 1, 2, 3, [...]. This option only applies for scenarios running more than one virtual user.

This option only applies for databanks that can be indexed. When configuring databank settings, if the databank file is too large to index, the `Let Each User Iterate Over Records Independently` option may not be available.

7. Use the arrow buttons to change the record data.
8. Select other Scenario Profile options as needed to configure how each Oracle Load Testing agent uses Data Bank records.

5.1.6 Using Synchronization Points

A sync point allows multiple virtual users to synchronize their actions and interactions with the application under test. Sync points provide the ability to create realistic multi-user situations that may expose resource conflicts such as deadlocks. When you specify a sync point, multiple virtual users executing the script will reach this sync point at various times depending on a number of factors (for example, the speed of the machine).

Sync points cause each virtual user to wait until all virtual users have reached that sync point. Each of the virtual users notifies the master upon reaching the sync point. The master waits for all of the virtual users to notify it and then issues the go-ahead for all the virtual users to continue past that sync point.

Sync points are added to individual scripts (parent or child scripts) when they are created in OpenScript. The execution parameters for sync points are defined in the Oracle Load Testing application.

To add a sync point to an OpenScript script:

1. Create or open a script in OpenScript.
2. Select the script node where you want to add the sync point.
3. Select **Add** from the **Script** menu then select **Other**.
4. Select the Synchronization Point node and click **OK**.
5. Enter a name for the synchronization point and click **OK**.
6. Save the script in OpenScript.

To define the sync point execution parameters in Oracle Load Testing:

1. Start Oracle Load Testing.
2. Select the script with the sync point and add it to the scenario.
3. Click the Configure Sync Point icon.
4. Expand the script name, select the name of the sync point in the left pane, and specify the execution parameters for the sync point in the right pane, as follows:

Name - specifies the name of the Synchronization point. This is the same name as specified for the sync point in the OpenScript script.

Group type - specifies how to determine when to release virtual users based on the following options:

- **All users in scenario** - calculates when to release virtual users based on all virtual users whether or not they are running.
- **Running users** - calculates when to release virtual users based on the number of running users only.

Release trigger - specifies the percentage of virtual users that must be waiting before they are released. For example, if ten virtual users are running and the release trigger is set to fifty percent, then the waiting virtual users are released when five of them are waiting. Setting the value greater than one hundred percent causes the virtual users to wait until they are manually released.

Open gate time - specifies the amount of time to keep the release open after it has been triggered.

Maximum Wait - specifies, per individual virtual user, the maximum amount of time, in seconds, that the synchronization point will wait before releasing the virtual user.

5. Repeat these steps to specify the execution parameters for other sync points in the same or other scripts.
6. Click the **OK** button when you have finished specifying sync point execution parameters. The sync point execution parameters will be saved as part of the virtual user scenario.

To check the sync point status or release sync points for running virtual users:

1. Start a virtual user scenario with one or more scripts containing sync points.
2. Click the **Active Session** tab and select the **Sync Point Status** sub-tab.

-or-

Click the **Virtual User Status** sub- tab and right-click on a running virtual user.

5.1.7 Using IP Spoofing

Oracle Load Testing can use IP spoofing to assign different IP addresses to virtual users. Each virtual user must get a defined IP address when using IP spoofing.

Before virtual users can use IP spoofing, you must define the IP addresses available for use by Oracle Load Testing in the TCP/IP network protocols of the workstation.

You define the IP addresses using the Advanced IP Addressing options of the TCP/IP properties of the Network Protocols. On Windows NT systems, the Network Protocols are accessed using the Control Panel, as follows:

The general procedure is as follows:

1. Open the TCP/IP Network Protocols.
2. In IP Address, select Specify an IP Address.
3. In the Advanced IP Address settings, add the IP Address and Subnet Mask.
4. Enter as many IP addresses/Subnet Masks as you have available for use by Oracle Load Testing virtual users.
5. Repeat the above steps on each Oracle Load Testing Agent system.
6. In the Profile Properties on the **Composer** tab, set **Use IP Spoofing** to True in the Browser Settings section.
7. Start the scenario.

The virtual users on each Agent machine will use the defined IP addresses in the order they were defined in the TCP/IP network protocols. For example the first virtual user uses the first IP address (index value 0), the second virtual user uses the second IP address (index value 1), and so on. If there are more virtual users than available IP addresses, Oracle Load Testing loops back to the first IP address and goes through the list of IP addresses repeatedly until all virtual have been assigned an IP address.

Note: Oracle Load Testing uses layering on the TCP/IP stack to perform IP spoofing with virtual users. While virtual users are running, the Oracle Load Testing layer is at the top of the TCP/IP stack. If an unusual event (for example, a restart) occurs while the system is running Oracle Load Testing virtual users with IP spoofing, other applications that use the TCP/IP stack may be affected. If this occurs, use `regsvr32.exe` to unregister the file `sporder.dll`.

5.2 Importing and Exporting Script Files

You can use the Script Options to import and export scripts that have been exported from OpenScript as .zip files. This can be useful if you are accessing Oracle Load Testing on a remote machine via the Web UI and want to upload scripts from your local Oracle Functional Testing.

To import script .zip files:

1. Select **Options** from the **User** menu.
2. Select **Scripts** in the **Manage** group.

This dialog box lets you import and export script files from the local machine to the Oracle Load Testing server.

Import - opens the Choose File dialog box for browsing and selecting a file.

Export - opens the file dialog box for browsing and selecting a file location.

3. Click **Import**.
4. Enter the file name or click **Browse** to locate the file.
5. Click **Open**.
6. Click **OK**.
7. Click **Upload**.

To export script .zip files:

1. Select **Options** from the **User** menu.
2. Select **Scripts** in the **Manage** group.
3. Select the script you want to export.
4. Click **Export**.
5. Specify the file location and click **OK**.

Defining VU Rampup

This chapter explains how to specify the Virtual User ramp rate for the current scenario.

6.1 Specifying Virtual User Rampup

The **Ramp** section on **Composer** tab is where you specify the rate at which the Controller starts running virtual users.

The **Ramp Rate** and **Stop Conditions** settings work together to specify the rate at which virtual users ramp up. Use the **Run Now** button to start the scenario immediately. Use the Schedule to specify a start time if not using the **Run Now** button.

Use the **Advanced options** to specify start, stop, and ramp-up conditions on a per-script basis. When the **Handle ramping of profiles individually** is set in the Advanced pane of the Ramp section, the Ramp Rate and Stop Conditions are disabled. Clear the **Handle ramping of profiles individually** option in the Advanced pane to re-enable the Ramp Rate and Stop Conditions settings. See [Section 6.1.2, "Using the Advanced Scenario Ramp Options"](#) for additional information about the Advanced options.

6.1.1 Specifying How Many Virtual Users to Start at a Time

The **Ramp Rate** pane of the **Ramp** section of the **Composer** tab lets you specify the Virtual User ramp settings for the current Scenario.

To specify how many Virtual Users to start at a time:

1. Add scripts in the Profiles section of the **Composer** tab.
2. Click the **Ramp** section.

The **Add** settings in the **Ramp Rate** pane specifies the start rate for virtual users.

- **users** - specifies the number of the virtual users to start at a time based upon the **After** setting. Selecting this option enters a default number of virtual users, which is the total number, in the associated editbox. Change the number as required.
- **percent** - specifies a percentage of the virtual users to start at a time based upon the **After** setting. Selecting this option enters a default percentage in the associated editbox. Change the percentage as required.

The **every** settings specifies how often the specified number or percentage of virtual users start running and has the following options:

- **seconds** - specifies the number of seconds to wait before starting the specified number or percentage of virtual users. Selecting this option enters a default number of seconds. Change the number as required.
 - **iterations** - specifies the number of scenario iterations to wait before starting the specified number or percentage of virtual users. Selecting this option enters a default number of iterations. Change the number as required.
3. Set the **Add** options in the **Ramp Rate** pane.
 4. Set the **every** options in the **Ramp Rate** pane.
 5. Set the **Stop Condition** option in the **Stop Condition** pane.

The **Stop Conditions** pane is where you specify when the Scenario profiles should stop running.

 - **Run until user stops test** - repeats running of the virtual user scenario continuously until you manually stop the virtual users.
 - **Run for specified duration** - allows you to specify how long to continue running the virtual user scenario. Selecting this option opens an editbox in the format: **dd:hh:mm:ss**. Change the time as necessary. The Session plays back the scenario profiles until the specified time and then automatically stops running the scenario.
 - **Run for specified iterations** - allows you to specify the number of times the virtual user profiles iterate while the session is running. Selecting this option opens an editbox for specifying the number of iterations. Change the iteration count as necessary. The virtual users run the scenario profiles the specified number of times and then stops automatically after the count is reached.
 6. Click the **Run Now** button to start the session immediately or click the **Schedule** button to schedule the session to run at a later time.

Once the session starts running, the specified number or percentage of virtual users start based on the **Add/every** settings.

6.1.1.1 Examples

Example: If you specify the percentage as 10% and the **After** setting is set to delay of 10 seconds, the Scenario starts running ten percent of the virtual users every ten seconds until all virtual users are running.

Example: If you specify the number as 5 and the **After** setting is set to 1 iteration, five virtual users start after each run of the scenario profiles until all virtual users are running.

Example: If you specify the iteration count as 1 and the **Add** setting is set to 5 users at a time, five virtual users start first. After the first five complete one iteration of the profile, another five virtual users start. Five additional virtual users start after each iteration until all virtual users are running.

Example: If you specify the delay as 120 seconds and the **Add** setting is set to 10% percent seconds, ten percent of the virtual users start every 120 seconds until all virtual users are running.

6.1.2 Using the Advanced Scenario Ramp Options

The Advanced Scenario Ramp Options let you specify start, stop, and ramp-up conditions on a per-script basis. These settings override the start, stop, and ramp-up settings specified in the Ramp Rate and Stop Conditions panes.

To specify advanced ramp-up settings:

1. Add scripts in the Profiles section of the **Composer** tab.
2. Click the **Ramp** section in the **Composer** tab.
3. Select the **Handle ramping of profiles individually** option in the **Advanced** pane. The advanced ramp settings are as follows:

Ramp Rate - specifies the rate at which the virtual users for the selected profile should start. The available ramp-up settings are as follows:

- **Add** specifies the start rate (how many) for virtual users.
 - **users** - specifies the number of the virtual users to start at a time based upon the **After** setting. Selecting this option enters a default number of virtual users, which is the total number, in the associated editbox. Change the number as required.
 - **percent** - specifies a percentage of the virtual users to start at a time based upon the **After** setting. Selecting this option enters a default percentage in the associated editbox. Change the percentage as required.
- **every** specifies how often the specified number or percentage of virtual users start running and has the following options:
 - **seconds** - specifies the number of seconds to wait before starting the specified number or percentage of virtual users. Selecting this option enters a default number of seconds. Change the number as required.
 - **iterations** - specifies the number of scenario iterations to wait before starting the specified number or percentage of virtual users. Selecting this option enters a default number of iterations. Change the number as required.

Start Condition - specifies the start conditions for the selected profile. The start settings are as follows:

- **When load test starts** - (default) when selected, the script starts based upon the scenario start criteria specified in the Schedule or when the **Run Now** button is clicked.
- **After a delay of (hh:mm:ss)** - when selected, the script starts the specified amount of delay time after the scenario starts.
- **When this profile runs [script name]** - when selected, the script selected in the left pane starts after the script specified by [script name] reaches the specified number of iterations or the specified delay time.
- **When this profile stops [script name]** - when selected, the script selected in the left pane starts after the script specified by [script name] stops. The start of the script selected in the left pane will be delayed by the specified number of seconds.
- **When this counter [counter name]** - when selected, the script selected in the left pane starts when the value of the counter specified by [counter name] reaches the criteria specified by the operator and threshold value. The start of the script selected in the left pane will be delayed by the specified number of seconds.

The relational operators used to measure the counter data against the threshold value are as follows:

- is greater than. The script selected in the left pane starts when the specified counter value is greater than the specified threshold value.
- is less than. The script selected in the left pane starts when the specified counter value is less than the specified threshold value.
- equals. The script selected in the left pane starts when the specified counter value equals the specified threshold value.
- does not equal. The script selected in the left pane starts when the specified counter value is not equal to the specified threshold value.

Stop Condition - specifies the stop conditions for the selected profile. The stop settings are as follows:

- **When load test stops** - (default) when selected, the script stops based upon the scenario stop criteria specified in the Schedule or when the Stop or Abort button is clicked.
- **After a delay of (hh:mm:ss)** - when selected, the script stops after the specified amount of delay time.
- **After each user iterates [X] times** - when selected, the script stops after the specified number of iterations.
- **When this profile runs [script name]** - when selected, the script selected in the left pane stops after the script specified by [script name] reaches the specified number of iterations or the specified delay time.
- **When this profile stops [script name]** - when selected, the script selected in the left pane stops when the script specified by [script name] stops. The stopping of the script selected in the left pane will be delayed by the specified number of seconds.
- **When this counter [counter name]** - when selected, the script selected in the left pane stops when the value of the counter specified by [counter name] reaches the criteria specified by the operator and threshold value. The stopping of the script selected in the left pane will be delayed by the specified number of seconds.

The relational operators used to measure the counter data against the threshold value are as follows:

- is greater than. The script selected in the left pane stops when the specified counter value is greater than the specified threshold value.
- is less than. The script selected in the left pane stops when the specified counter value is less than the specified threshold value.
- equals. The script selected in the left pane stops when the specified counter value equals the specified threshold value.
- does not equal. The script selected in the left pane stops when the specified counter value is not equal to the specified threshold value.

4. In the Advanced pane, specify the **Ramp Rate**, **Start Condition**, and **Stop Condition**, for each profile.

Setting Test Goals

This chapter explains how to test goals for the current scenario.

7.1 Setting Test Goals

You can set test goals that specify conditions for specific counter and script parameters.

To set test goals for a scenario:

1. Click the **Composer** tab.
2. Add a Script to a scenario in the Profiles pane.
3. Click the **Goals** section on the **Composer** tab.

This dialog box specifies the test goals for the current scenario.

Add Goal - adds a new goal condition.

stop ramping VUs when:

- **any of these goals are met** - when selected, the load test scenario continues ramping up virtual users until any of the specified goal(s) are met. Specify one or more goals for the scenario below.
- **all of these goals are met** - when selected, the load test scenario continues ramping up virtual users until the all of the specified goal(s) are met. Specify one or more goals for the scenario below.
 - **Goal 1** (through X) - specifies the goal criteria for the load test. Click the **Add Goal** button to add additional goals and conditions.

Add Condition - adds a new goal condition to the current goal.

Goal is met when:

any of the following conditions are met - when selected, the test goal is achieved when any of the specified goal conditions are met. Specify one or more goal conditions.

all of the following conditions are met - when selected, the test goal is achieved only when all of the specified goal conditions are met. Specify one or more goal conditions.

Select Counter - opens a list of counters that can be used as the goal condition.

For each goal condition, the following script counters can be used as a condition of the test. Click the **Add Condition** button to add additional conditions to the current goal.

- Total - Average Performance (sec)
- Total - Average Server Time (sec)
- Total - Hits/sec
- Total - Kb Rcvd/sec
- Total - Pages Rcvd/sec
- Total - Trans Failed/sec
- Total - Trans/sec
- [Script name] - Average Performance (sec)
- [Script name] - Average Server Time (sec)
- [Script name] - Hits/sec
- [Script name] - Kb Rcvd/sec
- [Script name] - Pages Rcvd/sec
- [Script name] - Trans Failed/sec
- [Script name] - Trans/sec
- - Number of VUs

For each script counter, the following conditions can be specified for the counter value of the test:

- is greater than [value]
- is less than [value]
- equals [value]
- does not equal [value]

When goals are met, mark load session

- No result
- Passed
- Failed
- User-defined value [value]

When goals are not met, mark load session

- No result
- Passed
- Failed
- User-defined value [value]

4. Select the **stop ramping VU when** option.
5. Click **Add Goal**.
6. For each goal, select the **Goal is met when** option.

7. For each goal condition, select the counter, select the condition, and specify the value. Click the **Add Condition** button to add additional counters and conditions to the goal.
8. Click the **Add Goal** button to add additional goals and conditions.
9. Repeat steps 5 and 6 for each goal and condition.
10. Set the **When goals are met, mark load session** and **When goals are not met, mark load session** options.
11. Start the load test.

Defining ServerStats Monitors

This chapter explains how to use the ServerStats features of Oracle Load Testing to monitor systems and metrics during a load test of the application-under-test.

8.1 About ServerStats

ServerStats lets you monitor a variety of server-side application, database, system, and Web server statistics. You can configure ServerStats to display real-time performance statistics for the various hosts and services available from the server, such as percentage of CPU usage, memory usage, Web server statistics, etc.

The ServerStats configurations contain one or more configured monitors for monitoring performance of application infrastructure during a load test. Once saved, you can add monitors into the configuration and apply them to the system to be monitored.

The Metric Profiles and Metrics options in the ServerStats options are used to create generic metric profiles and metrics that can be saved for use in multiple configurations. These generic definitions are available when you add a monitor to the configuration. Metric definitions monitor individual data points. Metric profiles monitor a group of data points.

Metric Profiles and Metrics are shipped with a default set of metrics. Additional metrics not included in the default metrics can be accessed by creating custom profiles or metrics.

Oracle Load Testing supports the following metric profiles:

Application Servers

- Oracle WebLogic Server

Databases

- Oracle
- SQL Server

Network Devices

- Cisco
- Foundry Load Balancer
- Host-Resources-MIB
- MIB-II
- Netscreen Firewall

Systems

- AIX 4.3
- HP-UX 11.0
- Linux
- Solaris
- Windows

WebServers

- Apache
- General
- IIS

Refer to Appendix E for a description of the metrics included with each profile.

Remote data collectors can be configured to run ServerStats monitors from remote locations.

8.2 Creating Monitors Overview

To create a ServerStats monitor, the following broad tasks must be completed:

- Configure the target systems you want to monitor and the data collectors you are using either in the Options or directly from the ServerStats Monitoring configuration.
- Create generic metric profiles and metrics if you want to use them in more than one configuration, otherwise metric profiles and metrics can be added directly from the configuration.
- Create and save a configuration.
- Add monitors to the configuration.

Refer to Appendix E for examples.

8.3 Using Configuration Files

ServerStats lets you maintain multiple configurations of servers, systems, and counters in different configuration files that can be selected in the Monitoring section of the Composer tab to be run with a load test. Configuration files store a particular set of ServerStats settings for monitoring server performance under load. The files store the following:

- Configured data sources
- Selected counters from each data source
- Selected metric profiles and metrics
- Data source and counter properties

Configurations are listed in the Monitoring section of the Composer tab.

Note: ServerStats configurations are only initiated and run as part of a load test. If you need to add additional metrics during a load test, start a new session and compare data later using absolute time on the reports.

8.3.1 Creating a New Configuration File

To create a new configuration file:

1. Select **Monitoring** on the **Composer** tab.
2. Click **New** to clear any existing configuration.
3. Click **Add Target** to list the system types.
4. Select the system type:
 - Database - opens a dialog box for specifying a database system to monitor.
 - Weblogic - opens a dialog box for specifying a Weblogic system to monitor.
 - Linux - opens a dialog box for specifying a Linux system to monitor.
 - Windows - opens a dialog box for specifying a Windows system to monitor.
 - Generic Target - opens a dialog box for specifying any system to monitor.
 - Existing - opens a list of existing target configurations that have already been created.
5. Specify the system connection information for the target system type.
6. If you want to use the default profile counters for the system type, make sure the **Apply [system type] profile automatically** option is selected.
7. Click **OK**.
8. If you do not use the **Apply [system type] profile automatically** option or you want to add additional counters to the configuration, click **Add Counters** and select where you want to retrieve the counter from and add the counter.
9. Click **Test** to make sure the monitors are configured correctly. Oracle Load Testing tries to start the monitors and collect data from the systems the monitors are configured against.
10. Click **Save Configuration**.
11. Enter a name for the configuration file and click **OK**.

8.3.2 Editing Configurations

To rename a configuration:

1. Select **Options** from the **User** menu.
2. Select **Configurations** in the **ServerStats** group to open the Configurations options.
3. Select the configuration you want to rename.
4. Click **Edit** above the list of configurations to display the Edit Configuration dialog box for changing the name of the configuration.

Name - displays the name of the configuration.
5. Edit the name.

6. Click **OK** to exit the Configurations dialog box.

To edit a saved configuration file:

1. Select **Monitoring** on the **Composer** tab.
2. Click **Load configuration**.
3. Select the configuration file to load:
4. Use the Add Counter menu to add counters to the configuration.
5. Use the Edit option to change the properties of an existing monitor in the configuration.
6. Use the Delete option to remove a counter from the configuration.
7. Click **Test** to make sure the monitors are configured correctly. Oracle Load Testing tries to start the monitors and collect data from the systems the monitors are configured against.
8. Click Save Configuration.

8.3.3 Deleting Configurations

To delete a configuration:

1. Select **Options** from the **User** menu.
2. Select **Configurations** in the **ServerStats** group to open the Configurations options.
3. Select the configurations that you want to delete and click **Delete**.
4. Click **Yes** to confirm the deletion.
5. Click **Close**.

8.4 Adding Data Source Monitors

You can add the following data source monitors:

COM+ - The ComPlus Monitor retrieves data for counters provided by ComPlus objects on Windows 2000 or later operating systems. The counters that are available are determined by the ComPlus applications that are installed on your system.

Database - You can set up a monitor that performs a database query for an individual machine.

JMX - The JMX monitor lets you determine if a Java application is performing properly. The JMX monitor checks the specified JMX MBean Attribute values to confirm that the Java application is performing as expected. The JMX monitor also provides an easy way to confirm whether or not the JMX application is running.

Perfmon - A Perfmon monitor checks on the performance of a specific Perfmon metric on a single machine. The monitor must have appropriate privileges to access the Perfmon statistics on the monitored machine.

Ping - Ping is an easy way to confirm that a host is responding. The Ping monitor sends an echo request to a specified host and waits for a reply. The monitor returns the round trip time.

URL - A URL monitor lets you determine if a URL that is critical to your web site is functioning as it should. In addition, the URL monitor can be used to parse data from the Apache server status page using a regular expression.

Virtual Agent - You can set up Virtual Agent monitors to run an executable program from the command line of the system. You specify a Perl expression to return metrics from the command. Virtual Agent is an easy way to execute operating system or custom programs on a host and return specific metrics. The Virtual Agent used to be called Remote Agent.

IBM Websphere PMI - IBM Websphere PMI is an easy way to confirm that a host is responding.

8.4.1 Configuring for Monitoring Disk Performance

To monitor disk performance on a Windows machine, `Diskperf -yv` must be enabled on that machine. See "Using Diskperf" later in this chapter for more information.

8.4.2 Adding Monitors

To add data source monitors:

1. Select the **Options** section on the **User** menu.
2. Select **Metrics** in the **ServerStats** group.
3. Select **New** and select the data source for the type of monitor you want to add.
4. Enter the data required to configure the monitor. All monitors include the following properties:

Name - enter a name for the monitor.

Description - enter a description for the monitor.

Category - displays the metric's category.

Data Source - displays the metric's Data Source type.

Translation Type - Translation Type controls how a metric's data is displayed in Reports.

- **No Translation** - data is displayed as it is provided by the metric's data source.
- **Rate (minutes)** - data is presented as a number, per minute. For example, if the time interval between two samples was 10 minutes, and 50 errors were found, the Rate (minutes) would be five (5) errors per minute.

Note: Rate (minutes) translates data sources that return their values as an endlessly increasing counter. If the data source does not return data this way, use No Translation.

SampleFrequency - specifies how often to sample data.

Sample Multiplier - enter the multiplier you want to be applied to the data that is collected. Setting the sample multiplier to 1 means that the data will be reported as is. If the metric measurement is expressed in kilobytes, setting the sample multiplier to 1024 translates it to bytes. If the metric measurement is expressed in milliseconds, setting the sample multiplier to .001 translates it to seconds.

5. Click **OK**.

The following sections list the additional properties required for specific data source types.

8.4.3 ComPlus Metric Properties

This section describes the additional fields required for the COM+ data source monitor.

Object ID - displays the object ID.

Object Name - displays the object name.

Counter Name - displays the counter's name.

Counter ID - displays the counter's ID.

8.4.4 Database Metric Properties

This section describes the additional fields required for the Database data source monitor.

Query - Enter a valid database statement. For example, you can enter the following Oracle SELECT statement:

```
select value from v$sysstat where statistic# = 1
```

If your database server supports returning a result set from a stored procedure you can call the procedure. For example, the following SQL Server stored procedure, present in the database, returns a result set (not an output parameter) showing the configured sample rate (samples/minute):

```
p_computeSampleRate
```

Report Type

- **Measure Query Time** - Select this option to see how long the query takes.
- **Read Result Set** - Select this option to match a value in the result.

Result Pattern - Enter the Perl expression for the value that should be returned for the specified query.

8.4.5 JMX Metric Properties

This section describes the additional fields required for the JMX data source monitor.

MBean Name - the fully qualified MBean name.

Attribute Name - the MBean attribute name to request.

Sub Attribute Name - the MBean sub attribute name to request.

Description - enter a description for the MBean.

Filter MBeans By - select a filter to use.

Discovery MBeans - lists the available MBeans.

8.4.6 Perfmon (Windows Performance Monitor) Metric Properties

This section describes the additional fields required for the Perfmon (Windows Performance Monitor) data source monitor.

Object - specifies the performance object name.

Counter - specifies the counter name.

Instance - specifies the instance of the counter to monitor.

8.4.7 Ping Metric Properties

This section describes the additional fields required for the Ping data source monitor.

Measurement - Specifies the metric to use for the Ping monitor. Select the metric to use: Average Round Trip Time (PingAvgRTT) or Percentage of Success (PingPercentSuccess).

Length - Specifies how many bytes should be included in each echo request packet.

Time to Live - Specifies the maximum number of router hops permitted. If the echo request takes more than the maximum number of hops, it is thrown away and the Ping attempt is treated as a failure.

8.4.8 URL Metric Properties

This section describes the additional fields required for the URL data source monitor.

URL Usage - specifies whether to use the URL specified in the **System Homepage** field on the Add/Edit Monitored System dialog box.

- **Use System Homepage** - use the URL specified in the **System Homepage** field on the Add/Edit Monitored System dialog box.
- **Specify** - enables the URL field for specifying the URL of the page you want to monitor.

URL - Type the URL for the web page you want to check. The URL must be specified with the full protocol designation of http or https. For example:

<http://www.company.com>.

If you use Get as the HTTP Method (the default), simply include any additional parameters you want to include as part of the URL string, for example:

<http://search.yahoo.com/bin/search?p=electronics>

URL Method

- **POST** - the Post method, usually associated with a form, allows a much larger amount of information to be associated with the request.
- **GET** - include any additional query strings as part of the URL address.

Note: Some servers limit the length of URL and query strings to about 240 characters.

Post Data - If you select Post as the URL Method, you can add a series of name-value pairs. For example, a URL generated after completing a form may appear as follows:

```
POST http://www.company.com/cgi-bin/guestbook.pl
... (additional headers)
name=Harold&nickname=Harry
```

To reference this page, enter name=Harold&nickname=Harry in this field.

User Agent - you can specify a particular user agent when you monitor a URL. When you configure a User Agent, the browser and operating system you specify is added to the header of the URL request. This allows you to test pages designed for a specific browser URL confirm that a page designed for multiple browsers works appropriately for each of them.

Enable Cookies - specify if cookies are enabled.

- **False** - select this option if you want the URL test to prevent the use of cookies when accessing the page.
- **True** - select this option if you want to allow the URL monitor to support cookies as part of the URL request. A fresh cookie is retrieved each time the test is run.

Custom Cookie - type in a cookie value for the page you are monitoring. If you supply a particular value, remember that the cookie is saved and can become out of date. Refresh cookie requests are ignored.

Metric Type

- **Download Time** - returns the download time.
- **Download Size in Bytes** - returns the size of the file.
- **Result of Perl Expression** - specifies a Perl expression used to retrieve data.

Include image download times - calculates the URL availability and access time with retrieval of embedded images (such as .gif and .jpeg files and references within tags).

Include object download times - calculates URL availability and access time with retrieval of embedded objects (such as ActiveX objects and Java applets). Oracle Load Testing measures the time taken to retrieve data used by objects, but does not include the time needed to retrieve any executable file required in order to interpret the data. Content types other than HTML and images are also classified as objects.

Result of Perl Expression - enter the Perl expression to use if you have selected Result of Perl Expression as the metric type. For example, you could use the following Perl expression to extract requests per second from the Apache Server status page:

```
/[0-9]+\.[0-9] requests\/sec/
```

Username - if the URL specified requires a name and password for access, enter the name in this field.

Password - if the URL specified requires a name and password for access, enter the password in this field.

Use HTTP Proxy - select **True** if your network makes use of a proxy server.

Proxy Host - specify the name or IP address of the proxy server.

Proxy Port - the port of the proxy server.

Proxy Username - the proxy server login user name.

Proxy Password - the proxy server login password.

8.4.9 Virtual Agent Metric Properties

This section describes the additional fields required for the Virtual Agent data source monitor.

Command Line - specifies the command to execute on the host machine.

Matching Regexp - specifies a Perl expression to use to report the data returned from the executable program.

Key of value in Regexp - specify the key of value to return from the Regular Expression specified in **Matching Regexp**. Key of value 0 returns the first match, 1 returns the 2nd match, etc. The match is of the Regular Expression inside the parenthesis (). For example, if the Command Line is `free -m` and the Matching Regexp

is `/Mem: .+?(\d+) .+?(\d+)/`, a Key of Value 0 returns the value in the first `(\d+)` of the Regular Expression.

Instance Name - specify the instance name.

8.4.10 IBM Websphere PMI Metric Properties

This section describes the additional fields required for the IBM Websphere PMI data source monitor.

Document - specify the WebSphere document.

Server - specify the server instance name that you want to connect with.

Node - specify the hostname/Cluster Node Name in uppercase.

Counter - specify the metric to monitor.

8.5 Using Metric Profiles to Create Monitors

Metric profiles are collections of metric definitions for monitoring general system types such as WebSphere, Oracle, etc. A profile can then be used to monitor a specific server when it is added to a ServerStats configuration.

1. Select **Options** from the **User** menu.
2. Select **Metric Profiles** in the **ServerStats** group.
3. Click **New**.

This dialog box is where you select the metric to include in the metric profile you want to add. Expand the appropriate node and select the data source, metric, or metric profile that you want to add.

4. Expand the **Available Counters** node, select the metric you want to add and click **Add**.
5. Repeat step 4 for each metric you want to add to the Metric Profile.
6. Click **OK** when finished

8.6 Using Metrics to Create Monitors

To create monitors using metrics:

1. Select **Options** from the **User** menu.
2. Select **Metrics** in the **ServerStats** group.
3. Click **New**.
4. Select the Data Source Type..
5. Specify the Metric properties.
6. Click **OK** when finished.

8.7 Editing Monitors In a Configuration

To edit a monitor in a configuration:

1. Select **Monitoring** on the **Composer** tab.
2. Select **Load Configuration** if you have a save configuration file.

3. Select the configuration you want to edit and click **OK**.
4. Select the monitor that you want to edit and click **Edit** .
5. Make any changes.
6. Click **OK** when you are done.

8.8 Working With Metric Profiles

This section explains how to work with metric profiles. Metric Profiles are a predefined group of metrics that can be added to a monitoring configuration by adding the Metric Profile rather than the individual metrics.

8.8.1 Adding Metric Profiles

You can add a user defined metric profile that contains any combination of default and user defined metrics.

1. Select **Options** from the **User** menu.
2. Select **Metric Profiles** in the **ServerStats** group.
 - New** - displays the Add Profile dialog box for adding a user defined metric profile.
 - Edit** - displays Edit Profile dialog box. This option is only available for user defined metric profiles. Default profiles cannot be edited.
 - Clone** - displays the Clone Profile dialog box for copying a default profile to create a user defined metric profile that can be edited.
 - Delete** - deletes the selected user defined metric profile. Default profiles cannot be deleted.
 - Import** - opens a dialog box for selecting the Metric Profile .zip file to import.
 - Export** - opens a dialog box for selecting the Metric Profile .zip file to import.
 - Metric Profile List** - lists the default profiles and user defined metric profiles.
3. Click **New**. The Add Metric Profile dialog box is displayed.
 - Name** - the name of the profile.
 - Description** - the profile's description.
 - Category** - displays the category to which the profile belongs.
 - Available Counters** - displays the list of available metrics.
 - Profile Metrics** - lists the selected metrics.
 - Add** - adds the selected metric to the **Profile Metrics** list.
 - Remove** - removes the selected metric from the **Profile Metrics** list.
4. Enter a name and description for the profile.
5. Add metrics by selecting them in the Available Counters list and clicking **Add**.
6. Click **OK** when finished. The new profile is listed in the User Defined category in the list of metric profiles.

8.8.2 Editing Metric Profiles

You can edit user defined metric profiles. Default profiles cannot be changed. Changing a metric profile does not change it in any configurations using the metric profile.

1. Select **Options** from the **User** menu.
2. Select **Metric Profiles** in the **ServerStats** group.
3. Select the user defined metric profile that you want to edit and click **Edit**. The Edit Profile dialog box is displayed.

Name - the name of the profile.

Description - the profile's description.

Category - displays the category to which the profile belongs.

Available Counters - displays the list of available metrics.

Profile Metrics - lists the selected metrics.

Add - adds the selected metric to the **Profile Metrics** list.

Remove - removes the selected metric from the **Profile Metrics** list.

4. Add metrics by selecting them in the Available Counters list and clicking **Add**.
5. Remove metrics by selecting them in the Profile Metrics list and clicking **Remove**.
6. Click **OK** when finished.

8.8.3 Cloning Metric Profiles

Cloning a profile lets you start with a default profile and create a user defined metric profile that you can customize for your needs by adding or deleting metrics.

1. Select **Options** from the **User** menu.
2. Select **Metric Profiles** in the **ServerStats** group.
3. Select the profile that you want to clone and click **Clone**. The Clone Metric Profile dialog box is displayed.

Name - the name of the profile.

Description - the profile's description.

Category - displays the category to which the profile belongs.

Available Counters - displays the list of available metrics.

Profile Metrics - lists the selected metrics. Click the "x" in front of a metric to delete it from the list.

Add - adds the selected metric to the **Profile Metrics** list.

Remove - removes the selected metric from the **Profile Metrics** list.

4. Enter a name for the profile.
5. All of the metrics for the selected profile are automatically added to the **Profile Metrics** list.
6. Add additional metrics by selecting them in the Available Counters list and clicking **Add**.
7. Remove metrics by selecting them in the Profile Metrics list and clicking **Remove**.

8. Click **OK** when you are done. The new profile is listed in the User Defined category in the list of metric profiles.

8.8.4 Deleting Metric Profiles

Only user-defined metric profiles can be deleted. To delete a user-defined metric profile:

1. Select **Options** from the **User** menu.
2. Select **Metric Profiles** in the **ServerStats** group.
3. Select the user defined metric profile you want to delete and click **Delete**.
4. Click **Yes** when asked to confirm the deletion.

8.8.5 Testing Metric Profiles

To test metric profiles:

1. Select **Monitoring** on the **Composer** tab.
2. If you have a saved configuration with a Metric Profile select **Load Configuration** and open the configuration file.
3. If you do not have a saved configuration, click **Add Target** and select the system type or existing target.
4. Click **Add Counters**.
5. Select **From Library**.
6. Select the **Metric Profiles** tab.
7. Select the Metric Profile and click **Add**.
8. Click **OK**.
9. Click **Test**.

8.9 Working With Metrics

This section explains how to work with metrics.

8.9.1 Adding Metrics

You can add metrics that can be used in any configuration. Metrics that you add are listed in the User Defined node of the Metrics tree. The Metrics dialog box has the following options:

1. Select the **Options** section on the **User** menu
2. Select **Metrics** in the **ServerStats** group.
3. Select **New** and select the data source for the type of metric you want to add.
4. Enter the data required to configure the metric. All monitor include the following properties:

Name - enter a name for the monitor.

Description - enter a description for the monitor.

Category - displays the metric's category.

Data Source - displays the metric's Data Source type.

Translation Type - Translation Type controls how a metric's data is displayed in Reports.

- **No Translation** - data is displayed as it is provided by the metric's data source.
- **Rate (minutes)** - data is presented as a number, per minute. For example, if the time interval between two samples was 10 minutes, and 50 errors were found, the Rate (minutes) would be five (5) errors per minute.

Note: Rate (minutes) translates data sources that return their values as an endlessly increasing counter. If the data source does not return data this way, use No Translation.

SampleFrequency - specifies how often to sample data.

Sample Multiplier - enter the multiplier you want to be applied to the data that is collected. Setting the sample multiplier to 1 means that the data will be reported as is. If the metric measurement is expressed in kilobytes, setting the sample multiplier to 1024 translates it to bytes. If the metric measurement is expressed in milliseconds, setting the sample multiplier to .001 translates it to seconds.

The following sections explain how to add specific metric types:

8.9.1.1 Adding ComPlus Metrics

To add a ComPlus metric:

1. Select **Options** from the **User** menu.
2. Select the **Metrics** in the **ServerStats** group.
3. Click **New**.
4. Select **ComPlus** and specify the metric properties. ComPlus metrics require the following properties.

Object Name - enter the object name of the counter.

Object ID - enter the object identifier.

Counter Name - enter the counter name.

Counter ID - enter the counter ID.

5. Specify the metric properties and click **OK**.

8.9.1.2 Adding Database Metrics

To add a database metric:

1. Select **Options** from the **User** menu.
2. Select **Metrics** in the **ServerStats** group.
3. Click **New**.
4. Select **Database** and specify the metric properties. Database metrics require the following properties.

Query - Enter a valid database statement. For example, you can enter the following Oracle SELECT statement:

```
select value from v$sysstat where statistic# = 1
```

If your database server supports returning a result set from a stored procedure you can call the procedure. For example, the following SQL Server stored procedure, present in the database, returns a result set (not an output parameter) showing the configured sample rate (samples/minute):

```
p_computeSampleRate
```

Report Type

- **Measure Query Time?** - Select true to see how long the query takes. Select false to match a value in the result.
- **Read Result Set?** - Select true to return a result set if your database supports it. Select false to return an output parameter.

Result Pattern - enter the Perl expression for the value that should be returned for the specified query.

5. Enter the Metric properties.
6. Click **OK**.

8.9.1.3 Adding JMX Metrics

To add a JMX metric:

1. Select **Options** from the **User** menu.
2. Select **Metrics** in the **ServerStats** group.
3. Click **New**.
4. Select **JMX** and specify the metric properties. JMX metrics require the following properties:

MBean Name - specify the fully qualified MBean name.

Attribute Name - specify the MBean attribute name.

Sub Attribute Name - specify the MBean sub attribute name.

Description - enter a description of the metric.

Filter MBeans By - select the server type by which you want to filter MBeans or select All Counters to return all supported MBeans.

Discovery MBeans - select the type of MBeans to discover or select All Counters to discover all types of MBeans.

5. Enter the metric properties and click **OK**.

8.9.1.4 Adding Perfmon (Windows Performance Monitor) Metrics

To add a Perfmon metric:

1. Select **Options** from the **User** menu.
2. Select **Metrics** in the **ServerStats** group.
3. Click **New** and select Perfmon.
4. Select **Perfmon** and specify the metric properties. Perfmon metrics require the following properties.

Object Name - specify the object name. For example, Processor or Memory. You must enter the object name exactly as it is displayed in the Perfmon editor. Be careful to check for spaces and case in the Object name.

Counter Name - specify the counter name. For example, % Processor Time. You must enter the counter name exactly as it is displayed in the Perfmon editor. Be careful to check for spaces and case in the Counter name.

Instance Name - if the metric has an instance associated with it, specify it here.

5. Enter the Metric properties.
6. Click OK.

8.9.1.5 Adding Ping Metrics

To add a Ping metric:

1. Select **Options** from the **User** menu.
2. Select **Metrics** in the **ServerStats** group.
3. Click **New**.
4. Select **Ping** and specify the metric properties. Ping metrics require the following properties.

Measurement - Specifies the metric to use for the Ping monitor. Select the metric to use: Average Round Trip Time (PingAvgRTT) or Percentage of Success (PingPercentSuccess).

Length - specify how many bytes should be included in each echo request packet.

Length - specify how many bytes should be included in each echo request packet.

5. Enter the Metric properties.
6. Click OK.

8.9.1.6 Adding URL Metrics

To add a URL metric:

1. Select **Options** from the **User** menu.
2. Select **Metrics** in the **ServerStats** group.
3. Click **New**.
4. Select **URL** and specify the metric properties. URL metrics require the following properties.

URL Usage

- **Specify** - select this option if you want to specify a URL. Enter the URL in the **URL** field.
- **Use System Homepage** - (default) select this option to automatically use the home page defined for the system.
- **URL** - If **URL Usage** is set to **Specify**, type the URL for the web page you want to check. The URL must be specified with the full protocol designation of http or https. For example: http://www.company.com.

If you use Get as the HTTP Method (the default), simply include any additional parameters you want to include as part of the URL string, for example:

<http://search.yahoo.com/bin/search?p=electronics>

URL - If **URL Usage** is set to **Specify**, type the URL for the web page you want to check. The URL must be specified with the full protocol designation of http or https. For example: <http://www.company.com>.

If you use Get as the HTTP Method (the default), simply include any additional parameters you want to include as part of the URL string, for example:

<http://search.yahoo.com/bin/search?p=electronics>

URL Method

- **POST** - the Post method, usually associated with a form, allows a much larger amount of information to be associated with the request.
- **GET** - include any additional query strings as part of the URL address.

Note: Some servers limit the length of URL and query strings to about 240 characters.

Post Data - If you select Post as the URL Method, you can add a series of name-value pairs. For example, a URL generated after completing a form may appear as follows:

```
POST http://www.company.com/cgi-bin/guestbook.pl
... (additional headers)
name=Harold&nickname=Harry
```

To reference this page, enter `name=Harold&nickname=Harry` in this field.

User Agent - you can specify a particular user agent when you monitor a URL. When you configure a User Agent, the browser and operating system you specify is added to the header of the URL request. This allows you to test pages designed for a specific browser URL and confirm that a page designed for multiple browsers works appropriately for each of them.

Enable Cookies

- **False** - select this option if you want the URL test to prevent the use of cookies when accessing the page.
- **True** - select this option if you want to allow the URL monitor to support cookies as part of the URL request. A fresh cookie is retrieved each time the test is run.

Custom Cookie - type in a cookie value for the page you are monitoring. If you supply a particular value, remember that the cookie is saved and can become out of date. Refresh cookie requests are ignored.

Metric Type

- **Download Time** - returns the download time.
- **Download Size in Bytes** - returns the size of the file.
- **Result of Perl Expression** - specifies a Perl Expression used to retrieve the file size.

Types other than HTML and images are also classified as objects.

Include image download times - calculates the URL availability and access time with retrieval of embedded images (such as .gif and .jpeg files and references within tags).

Include object download times - calculates URL availability and access time with retrieval of embedded objects (such as ActiveX objects and Java applets). Oracle Load Testing measures the time taken to retrieve data used by objects, but does not include the time needed to retrieve any executable file required in order to interpret the data.

Include object download times - calculates URL availability and access time with retrieval of embedded objects (such as ActiveX objects and Java applets). Oracle Load Testing measures the time taken to retrieve data used by objects, but does not include the time needed to retrieve any executable file required in order to interpret the data.

Perl Expression - enter the Perl expression to use if you have selected Result of Perl Expression as the metric type.

Username - if the URL specified requires a name and password for access, enter the name in this field.

Password - if the URL specified requires a name and password for access, enter the password in this field.

Enable Proxy- select this checkbox if your network makes use of a proxy server.

Proxy Host - specify the name or IP address of the proxy server.

Proxy Port - the port of the proxy server.

Proxy Password - the proxy server login password.

Proxy Username - the proxy server login user name.

5. Enter the Metric properties.
6. Click OK.

8.9.1.7 Adding Virtual Agent Metrics

To add a Virtual Agent metric:

1. Select **Options** from the **User** menu.
2. Select **Metrics** in the **ServerStats** group.
3. Click **New**.
4. Select **Virtual Agent** and specify the metric properties. Virtual Agent metrics require the following properties.

Command Line - specifies the command to execute on the host machine.

Matching Regexp - specifies a Perl expression to use to report the data returned from the executable program.

Key of value in Regexp - specify the key of value to return from the Regular Expression specified in **Matching Regexp**. Key of value 0 returns the first match, 1 returns the 2nd match, etc. The match is of the Regular Expression inside the parenthesis (). For example, if the Command Line is `free -m` and the Matching Regexp is `/Mem: .+? (\d+) .+? (\d+) /`, a Key of Value 0 returns the value in the first `(\d+)` of the Regular Expression.

Instance Name - specify the instance name.

5. Enter the Metric properties.
6. Click **OK**.

8.9.1.8 Adding IBM Websphere PMI Metrics

To add an IBM Websphere PMI metric:

1. Select **Options** from the **User** menu.
2. Select **Metrics** in the **ServerStats** group.
3. Click **New**.
4. Select **IBM Websphere PMI** and specify the metric properties. WebSphere metrics require the following properties.

Document - specify the URL to use as the root for the queries to identify counters. For example, wasPerfTool/servlet/perfservlet.

Server- specify the server instance that you want to talk to.

Node - specify the hostname in uppercase.

Counter - specify the metric to monitor.

5. Enter the Metric properties.
6. Click **OK**.

8.9.2 Editing Metrics

Only user-defined metrics can be edited. Default metrics cannot be changed. Changing a metric changes it in any configuration that uses it.

To edit user-defined metrics:

1. Select **Options** from the **User** menu.
2. Select **Metrics** in the **ServerStats** group.
3. Select the user defined metric you want to edit and click **Edit**. The Edit Metric dialog is displayed showing the properties pertinent to the selected metric.
4. Make any necessary changes.
5. You can test the metric by clicking **Test** and specifying the test parameters.
6. Click **OK**.

8.9.3 Cloning Metrics

When you clone a default metric you create a user defined metric that can be edited, saved, and used in configurations.

To clone a metric:

1. Select **Options** from the **User** menu.
2. Select **Metrics** in the **ServerStats** group.
3. Select the metric you want to clone and click **Clone**. The Clone Metric dialog box is displayed. This dialog box displays name and description details about the metric as well as metric discovery and property information when applicable.
4. Make any changes and click **OK**. The metric is listed in the user defined node of the metric tree.

8.9.4 Deleting Metrics

User defined metrics can be deleted. Default metrics cannot be deleted.

To delete a metric:

1. Select **Options** from the **User** menu.
2. Select **Metrics** in the **ServerStats** group.
3. Select the user defined metric that you want to delete and click **Delete**.
4. Click **Yes** when asked to confirm the deletion.

8.9.5 Testing Metrics

To test a metric:

1. Select **Monitoring** on the **Composer** tab.
2. If you have a saved database configuration file select **Load Configuration** and open the configuration file.
3. If you do not have a saved configuration, click **Add Target** and select the system type. Enter the target system properties, click **OK** and add counters to the configuration.
4. Click **Test**.
5. When the test finishes, click **OK** to exit the Test Monitoring Configuration window.

8.9.6 Viewing Metrics

You can view metric descriptions and configuration details from **Monitoring** section of the **Composer** tab.

To view metrics details:

1. Select **Monitoring** on the **Composer** tab.
2. If you have a saved database configuration file select **Load Configuration** and open the configuration file.
3. If you do not have a saved configuration, click **Add Target** and select the system type. Enter the target system properties, click **OK** and add counters to the configuration.
4. Click **Edit** next to the metric. The Edit Monitor dialog box show details about the metric and the Data Collector system.
5. Click **OK** to exit the Edit Monitor dialog box.

8.10 Using Diskperf

Diskperf is a command that enables the Windows Perfmon counter that measures computer disk performance.

You will probably have to enable Diskperf on Windows machines when you have monitors that use Perfmon. These counters are may not be enabled by default on a Windows machine.

To execute the Diskperf command:

1. Log on with administrative rights to the NT or Windows 2000 machine whose disk performance you want to monitor.
2. At the command line, type one of the following:

Non-RAID or hardware RAID PCs: `diskperf -y`

Software RAID PCs: `diskperf -y e`

Windows 2000 PCs: `diskperf -yv`

Note: The complete list of options for either system can be found by running `diskperf -?`

3. Press Enter.
4. Restart the computer.

Running and Saving Scenarios

This chapter explains how to use the Ramp settings to specify start and stop times for scenario profiles and the rampup specifications for each virtual user. Default values can be set by selecting **Options** from the **User** menu then selecting **Ramp Defaults** in the Profiles group.

This chapter also explains how to use the Advanced Ramp-up settings for specifying start, stop, and ramp-up conditions on a per-script basis.

Note: Changing the Ramp defaults does not affect the current scenario. The changed Ramp defaults go into effect when you start a new scenario.

Before you can run scenarios, you must first define virtual user profiles and ramp options. [Chapter 5, "Creating Scenario Profiles"](#) explains how to define virtual user profiles.

9.1 Working with Scenario Files

Once you select virtual user profiles and set the properties, you can save the Scenario to a file for future use.

9.1.1 Saving Scenarios

To save a Scenario:

1. Select **Save** or **Save As** from the **Scenario** menu or click the toolbar button.

Repository - a list of repositories. Select the repository in which you want to save the scenario.

<**Scenario list**> - lists the scenarios in the selected workspace.

Name - enter the name for the scenario.

2. If the Save As dialog box appears, specify a file name in the Name field.
3. Select the repository and workspace where you want to save the scenario.
4. Click **OK**.

Note: Oracle Load Testing saves the current Profile settings as part of a saved scenario.

9.1.2 Opening Existing Scenarios

If you have previously defined a scenario and saved it to a file, you can open the scenario for use in Oracle Load Testing.

To open an existing Scenario:

1. Select **Open** from the **Scenario** menu. Oracle Load Testing opens a dialog for selecting the scenario file.

Repository - a list of repositories.

<Scenario list> - lists the scenarios in the selected workspace.

2. Select the Repository and Workspace containing the scenario you want to open.
3. Select the scenario and click **OK**.

9.1.3 Renaming Scenarios

To rename a scenario:

1. Select **Options** from the **User** menu.

2. Select **Scenarios** in the **Manage** group.

Edit - opens the Edit Scenario Dialog Box for editing the selected scenario.

Delete - deletes the selected scenario.

Import - opens a dialog for selecting the scenario file to import.

Export - opens options for saving the selected scenario to a file.

Name - lists the scenarios in the selected workspace.

3. Select the scenario you want to edit.

4. Click **Edit**.

Name - enter a name for the scenario.

5. Enter a new name for the scenario.

6. Click **OK**.

7. Click **Close**.

9.1.4 Deleting Scenarios

To delete a scenario:

1. Select **Options** from the **User** menu

2. Select **Scenarios** in the **Manage** group.

Edit - opens the Edit Scenario Dialog Box for editing the selected scenario.

Delete - deletes the selected scenarios. To select more than one scenario, hold down the CTRL key.

Import - opens a dialog for selecting the scenario file to import.

Export - opens options for saving the selected scenario to a file.

Name - lists the scenarios in the selected workspace.

3. Select the scenarios you want to delete. To select more than one scenario, hold down the CTRL key.

4. Click **Delete**.
5. Click **Yes** to confirm the deletion.
6. Click **Close** to exit the Scenario Manager.

9.2 Running Scenario Sessions

Once you have defined a scenario and set the profile attributes, you can run the scenario session immediately or schedule the session to run at a later time.

9.2.1 Start Scenario Session Immediately

If you want to use the default settings and start the scenario session, you can automatically start the session in one step.

To start a scenario session immediately, click the **Run Now** button.

Oracle Load Testing automatically opens the Active Session tab with the scenario loaded and starts the virtual user run.

9.2.2 Schedule without Starting the Scenario Session

Adding the scenario to the calendar without starting allows you to specify the start and stop times for running the scenario session.

To submit a Scenario without starting the session click the **Schedule** tab.

Click on the day to schedule a new task for the session run.

Enter the task name, start and end time, and optional description and click **OK**.

9.3 Manually Starting and Stopping Sessions

Once you have created scenario profiles, you can manually start and stop the session using the following:

- Start/Stop/Abort buttons
- Calendar

You need to manually stop the active session to submit more scenario profiles or stop any new virtual users from being started.

9.3.1 Starting the Session

To start the session manually:

1. Click the **Run Now** button.
2. If the Deferred Start options appear, allow the delay to continue until the start time.

or:

Click **Run Now** to start running the scenario immediately.

or

Click **Stop** from the **Session** menu to cancel the start.

9.3.2 Stopping Ramping New Virtual Users

To stop ramping new Virtual Users, select **Pause Ramp** from the Session menu on the Active Session tab.

9.3.3 Stopping All Virtual Users

To stop all running Virtual Users after they have finished, click the **Stop** button. The virtual users will complete the current iteration of the script and then stop.

To immediately stop all running Virtual Users click the **Abort** button.

9.3.4 Managing Sessions

Sessions specify the scope for Oracle Load Testing data collection and reporting. The data collected while the session is running virtual users is shown in the virtual user status grid, Oracle Load Testing ServerStats runtime performance statistics and load graphs, and can be saved to a database for post-testing analysis.

You can specify default settings for how sessions start and end data collection using **Options** from the **User** menu then selecting **Session Start/Stop**. Selecting this option opens the Session Start/Stop options:

Session Naming - defines how to name sessions.

- **Default Session prefix** - specifies a fixed name to add before the session name. Enter a name to use. Oracle Load Testing adds the increment number to the name you define. The default name is "Session" (for example, Session0001).
- **Prompt for session name before running** - when selected, the session name dialog box prompts for a session name when starting a session. When cleared, the session name will be automatically created based on the prefix.

Session End - defines how a session ends. When a session ends, Oracle Load Testing stops updating runtime data in the performance statistics, load graphs, and the database. Subsequent Autopilot runs start a new session for data collection and reporting purposes.

- **Stop session on last VU completion** - when selected, the session ends when the last virtual user has finished the run.
- **Stop session when browser is closed** - when there is an attached session, stops the session when the session timeout is reached after the browser is closed. When not checked and the browser is closed, the session continues to run until it's normal stopping time or until you reattach to the session and stop it manually. The default session timeout is ten minutes.
- **Terminate all agents at end of session** - when selected, all Oracle Load Testing Agents automatically close when a session ends.

Agent error handling - defines how to handle agents that encounter errors.

- **Stop ramp-up on agent error** - when selected, the autopilot stops submitting new virtual users if any virtual users fail to complete the initialization process. This may happen due to complications when starting the agent process or failures during script pre-run verification. This does not stop the test and previously running users continue to execute until the end of the session.
- **Drop failed agents from session** - when selected, the autopilot stops submitting new users to the agent machine that had the failure if a virtual users fails to start or is set to orphaned for any reason.

9.4 Running Scenarios from the Command Line

You can run and stop scenario files using the command line Java file `OLTCommandLine.jar` located in the `\lib` directory under the installation directory. See [Appendix C, "Command Line Reference"](#) for additional information.

Viewing Active Sessions

This chapter explains how to view runtime session data and graphs for currently running scenarios.

10.1 Active Session Tab Overview

The Active Session tab is where you can view runtime graphs. These graphs are only available for the running load test session. Use the **Reports** tab to view reports and graphs after the load test has finished running.

The Session menu provides options for selecting active sessions to attach or detach.

The Session Overview sub-tab shows an overview of the running virtual users, transactions, performance by profile, and any monitors configured for the scenario.

The Virtual User Status sub-tab shows the virtual users grid.

The Runtime Graphs sub-tab is where you can view runtime reports. These graphs are only available while the load test is running. Use the Reports tab to view reports and graphs after the load test has finished running.

The Virtual User Logs sub-tab shows the virtual user information saved to the log file.

The ServerStats sub-tab shows server statistics for metrics being monitored.

The Sync Point Status sub-tab shows the current status of Sync Points configured in the scenario profiles.

The Active Session tab is refreshed according to what is set in the **Graph Refresh Interval** setting in the reporting options.

10.1.1 Session Menu

The session menu includes the following options for working with sessions:

- **Add** - adds Virtual Users to the session.
- **Pause Ramp** - pauses ramping of additional Virtual Users.
- **Attach To** - attaches to a running session in full control mode.
- **Watch** - attaches to a running session in view only mode.
- **Detach** - detaches from the current session.
- **Stop** - stops the current session.
- **Terminate Idle Agents** - stops all idle agent processes running on your agent systems.

10.2 Active Session Overview

The Session Overview sub-tab shows an overview of the running virtual users, transactions, performance by profile, and any monitors configured for the scenario.

Virtual Users graph - shows the ramp rate of the running virtual users.

Transactions graph - shows the average transactions per second.

Overview pane - shows the min, max, average, and standard deviation for the listed performance measures.

Totals pane - shows the totals for the listed performance measures.

Performance by Profile pane - shows the min, max, average, passed, warning, and failed totals for each profile in the scenario.

Monitors pane - shows any ServerStats performance monitors configured for the scenario.

10.3 Virtual User Status

The Virtual User Status sub-tab shows the Virtual User grid for the running Virtual Users.

Once you start running virtual users, the Virtual User Status sub-tab on the **Active Sessions** tab lists the running virtual users and shows the run details for each.

10.3.1 Overview of the Virtual Users Grid

The Virtual User Status sub-tab on the **Active Session** tab show the virtual users grid. The virtual user grid has the following columns:

Results Per Page - select number of VUs you want to view.

arrows - use the arrows to go the first page, previous page, next page, and last page in the virtual user grid.

VU-Id - the ID number for the virtual user.

Profile - shows the name of the user profile being run by the virtual user.

Status - shows the current status for the virtual user.

Iterations - shows the number of times the virtual user has *attempted* to play back the run section of the user profile/script. The Iterations count will be incremented for each new iteration attempted. The iteration count is updated when the iteration ends. An iteration could end if the script fails, if the user calls "return" from inside the master script's `run()` method, or if the iteration completes all the way through successfully, for example.

When databanking with **Stop After Last Record Used**, the script will not stop immediately after the last record is requested. It is not known how many requests in the future would rely on the values of the last requested record. Therefore, the user does not abort until after the user attempts to request a record *after* the last record was already requested.

Failed - shows the number of failures that occurred for the virtual user.

Last Run Time - shows the average elapsed time it took for the virtual user to execute the scripts in the run section of the profile.

Current Step - shows the name of the OpenScript step group currently being accessed by the virtual user.

System - shows the name of the system on which the virtual user is running.

Data Bank - shows the record of data from the Oracle OpenScript Data Bank, if any, used by the script during playback.

Current Error - provides a description of the error encountered on the most current completed iteration by the virtual user while executing the scripts. If the next iteration passes, this column is cleared.

Previous Error - shows a description of the previous error that occurred while the Virtual User was running.

10.3.2 Controlling Individual Virtual Users

You can control the running of specific running virtual users using the virtual users grid.

1. Define virtual user profiles in the **Composer** tab.
2. Click the **Run Now** button.
3. Click the **Active Session** tab.
4. Click the **Virtual User Status** sub-tab.
5. Select the virtual user row in the **Virtual User Status** sub-tab.
6. Click the right mouse-button to open the popup menu.
7. Select options, as follows:
 - **Modify Run Attributes** - opens a dialog box for changing the profile attributes for the selected virtual user.
 - **Stop** - stops running the selected virtual user. Stopping allows the virtual user to finish executing the script, complete the run section but not run another iteration.
 - **Abort** - aborts running the selected virtual user. Abort stops execution after the current page in the script and terminates execution altogether.
 - **Release Sync** - releases the virtual user from any synchronization point that may have been reached during the execution.

10.3.3 Modifying Run Attributes

You can modify the profile run attributes for specific virtual users running in the virtual user grid.

1. Select the virtual user row in the virtual user grid.
2. Click the right mouse button to open the popup menu.
3. Select **Modify Run Attributes**. Oracle Load Testing opens a dialog box for modifying the attributes.
4. Select options, as follows:
 - **<check box>** - select the checkbox in front of the field that you want to change to make the field available to change.
 - **Enable Logging** - select when you want to enable or disable logging of messages.
 - **Message Delivery** - select when you want to log all virtual user messages, on error or always.

- **Logged Messages** - select when you want to change the type of messages that are logged, Standard or Extended. See the Logged Messages in [Section 5.1.3, "Specifying Scenario Profile Properties"](#) or [Section 14.4.2, "Setting Scenario Default Properties"](#) for details about Standard and Extended log messages.

10.4 Viewing Runtime Graphs

Oracle Load Testing performance graphs can be viewed using the **Runtime Graphs** sub-tab of the **Active Session** tab. The Runtime Graphs are available only when an Oracle Load Testing session is running.

You can view the graphs while the virtual users are running or after they have been stopped or have finished. The performance data is reported at the conclusion of each iteration. When the running virtual users complete the first iteration, the first set of performance data is captured. The graphs and performance statistics that you see are updated in real time based on the **Graph refresh interval** setting in Reports Options to give you a snapshot of the load test results. However, there is a small amount of processing time required to generate the graphs/statistics and print them to the screen. The default refresh is 30 seconds. If you run long or over night tests, you should increase the refresh interval to 60 seconds or longer.

If you run a very quick session, the Active Session tab may not have enough time to update the screen with the real time data. However, the data is stored in the database.

In addition to real-time data shown, Oracle Load Testing can generate a summary report and save data to the database for post-testing analysis. The Latest Load Session reports let you generate a summary of performance and error statistics data for the virtual user session. You can use the Reports tab to generate a variety of reports and graphs from the data stored in the database.

10.4.1 Using Default Runtime Graphs

The Runtime graphs lets you view real-time performance, statistical, and error information about the virtual users being run by the session. The default Oracle Load Testing graphs automatically start and consist of the following types of runtime graphs:

- Performance Vs. Users
- Errors Vs. Users
- Performance Vs. Time
- Errors Vs. Time
- Users Vs. Time
- Statistics Vs. Time

10.4.1.1 Viewing Runtime Graphs

To view runtime statistics:

1. Create a Session and click **Run Now** to start a session.
2. Click the **Runtime Graphs** sub-tab on the **Active Session** tab to display a small view of the default graphs. This is useful for getting an at-a-glance view of all of the run graphs.

3. You can switch between Oracle load Testing default graphs and custom graphs using the **View** selector list on the **Runtime Graphs** sub-tab.

To create a custom graph:

1. Click **Custom Graph** option on the View selector on the **Runtime Graphs** sub-tab.
2. Click **Add Graph**.
3. In the [session counters tree] section, expand the tree to locate the counter data you want to view. If you do not see the data series you are looking for, click **Filter Counters** and select Show All Counters to display all available data series.
4. Double-click counter(s) in the session counter tree to add the counter to the graph.
5. Click the **Toggle Legend** button to toggle the legend on and off. The legend shows the names and plot line color of each data series in the graph.
6. Click the **Toggle Stats** button to toggle the chart statistics table view on and off. The chart statistics show the Data Series and values for each series, as follows:
 - **Minimum** - shows the maximum value for the data series.
 - **Maximum** - shows the minimum value for the data series.
 - **Average** - shows the average value for the data series.
7. Click the **Options** button to customize the current graph. The options are the same as graphs created on the Reports tab.
8. Enter a name for the graph.
9. Select how to plot the graph in the **X-Axis** and **Y-Axis** tabs.
10. Click **Apply**. Each custom report is displayed under a separate tab so that you can select the one you want to view. Click **X** on the graph tab to delete the graph and its tab.

10.4.1.2 Performance Vs. Users Graph

This graph plots the average run time for the number of active virtual users.

If you have multiple profiles in the running Scenario the Performance Vs. Users graph shows separate plot lines for each scenario profile running.

The Average Performance values are a graphical representation of the Average Value data for the Elapsed time entries shown in the Performance Statistics view. The graph adds plot points based upon the number of running virtual users as the Autopilot ramps up to the total number of virtual users for all scenarios.

The plot points are added in the same manner as the Elapsed time entries in the Performance Statistics report.

10.4.1.3 Errors Vs. Users Graph

This graph plots the number of failed transactions-per-second for each Virtual User profile.

If you have multiple profiles in the running Scenario the Errors Vs. Users graph shows separate plot lines for each scenario profile running.

The failed transactions-per-second depends upon the number of virtual users being run by individual profiles and the number server requests (page, image, frame, etc.) that fail during the Oracle Load Testing session.

10.4.1.4 Performance Vs. Time Graph

This graph plots the average run time for virtual users over a period of time.

If you have multiple profiles in the running Scenario the Performance Vs. Time graph shows separate plot lines for each scenario profile running.

The Average Run Time values are a graphical representation of the Average Value data for the Elapsed time entries shown in the Performance Statistics report that are plotted over time.

10.4.1.5 Errors Vs. Time Graph

This graph plots the number of failed transactions per second for virtual users profiles over time.

If you have multiple profiles in the running Scenario the Errors Vs. Time graph shows separate plot lines for each scenario profile running.

The failed transactions-per-second depends upon the number of virtual users being run by individual profiles and the number server requests (page, image, frame, etc.) that fail during the Oracle Load Testing session.

10.4.1.6 Users Vs. Time Graph

This graph plots the number of active virtual users over a period of time.

If you have multiple profiles in the running Scenario the Users Vs. Time graph shows separate plot lines for each scenario profile running.

10.4.1.7 Users Vs. Time Graph

This graph plots the average hits, transactions and kilobytes per second over a period of time.

The Average Statistic values are a graphical representation of the Average Value data from the Performance Statistics tab over time. The plot points are updated while the Autopilot runs the virtual users.

10.4.2 Custom Graphs

The Custom graphs view of the Runtime Grahs sub-tab has the following options:

View - displays options for creating a new graph and viewing default graphs.

- **Add Graph** - displays a new tab with options for customizing the graph. The tab label is the name you enter in the Graph Title field on the Graph tab of the graph options.
- **Graph <#>** - displays custom graphs that you have created using **New Graph**.
- **Edit Graph**- the pencil button on the graph opens the selected graph for editing.
- **Remove Graph** - the (X) button on the tab deletes the associated graph tab.

Custom graphs have the following options:

- **Session** - displays the session that is running.
- **Filter Counters** - selects the filter to use for the session data counters.
 - **Show Average Performance Only** - when selected, the counters tree shows only average performance counters.
 - **Show All Counters** - when selected, the counters tree shows all counters.

- **[session counters tree]** - shows a list of session data generated by Oracle Load Testing and ServerStats. Click the arrow to expand the tree. The data series listed is a subset of all of the available data series. Click **Filter Counters** and select Show All Counters to display all available data series.
- **Default Graphs** - double-click a graph name in the session counters tree to view a default graph.
- **Oracle Database Diagnostics** - groups a set of Oracle Database profile metrics on the same graph. These metrics appear only if an Oracle Database ServerStats configuration is used with the session run. The graphs and metrics are as follows:
 - **Average Active Sessions** - shows Average Active Sessions metrics.
 - **Host CPU Utilization** - shows Host CPU Utilization metrics.
 - **Host PGA Utilization** - shows Host "Program Global Area (PGA)" Utilization metrics.
 - **DB Throughput** - groups "Physical I/O per second" and "Physical I/O requests total" metrics.
- **Session Counters** - the counters that are available for the overall session.
- **<profiles>** - the counters that are available for each Oracle Load Testing virtual user profile.
 - **Profile Nodes** - list the counters for virtual user profiles available to include in the report graphs. The profile is the name of the virtual user profile(s) that were run in the Session. Select the counter(s) and click the Add Data Source button or double-click to add counters to the Selected Data Series list.
 - **Timer Nodes** - list the page timers available to include in the report graphs. Timers correspond to the individual pages of the script(s) run by the virtual user profile. They allow you to view the performance of a specific page download within a test run, allowing you to identify bottlenecks. The timer is the name of the virtual user profile(s) that were run by the Session. The timers consist of the virtual user profile name/Script name and the page number of the script. The timers are added to the session data when you select the **Generate timers for all resources** check box in the **Default Properties** section of the **Profiles** group of the user **Options**. Select the timer(s) and double-click with the mouse to add timer(s) to the Selected Data Series list.
 - **ServerStats Nodes** - list the Oracle Load Testing ServerStats counters available to include in the report graph. The ServerStats counters are based upon the data source(s) and counters that were included in the ServerStats Data Source configuration when the Oracle Load Testing session was run. Select the counter(s) and click the **Add Data Series** button or double-click with the mouse to add counter(s) to the Selected Data Series list.
- **ServerStats Counters** - shows the ServerStats counter configuration.

Graphs have the following toolbar buttons:

- **Load Query** - opens a dialog box for opening a saved graph data query.
- **Save Query** - opens a dialog box for saving the current graph data query.
- **Export** - opens options for saving the current graph to a file. Select a file type.
 - **PNG** - opens options for saving the current graph to an PNG image file.
 - **SVG** - opens options for saving the current graph to an SVG image file.

- **CSV** - opens options for saving the current graph data to a comma-separated-value file.
- **Excel HTML** - opens options for saving the current graph data to an Excel HTML file.
- **Toggle Legend** - displays or hides the names and plot line color of each data series in the graph.
- **Toggle Stats** - displays or hides the chart statistics table view. The chart statistics show the Data Series and values for each series.
- **Refresh Graphh** - refreshes the data in the graph from the database.
- **Zoom** - toggles the zoom tool on and off. When in zoom mode, a magnifier appears in the graph area. Click the magnifier and select the zoom options from the popup menu.
- **Options** - opens the options for customizing graph and x/y axis options.

Graph Tab

- **Graph Title** - enter the name of the graph. This name is displayed in the tab in the Reports & graphs section.
- **Series** - lists the data series contained in the graph. The (x) button deletes the Data Series from the graph.
- **Data Series Name** - displays the name of the data series that is displayed when you place the mouse over the data series in the graph and the label displayed in the legend.
- **Set Label to Default** - resets the data series label name to the default name.
- **Delete Series** - removes the data series from the graph.

Axes Tab

- **X-Axis**

Scale - specifies the X-axis data range format. Changing the scale causes the graph to be refreshed from the server.

- **Absolute Time** - when selected, the generated report tables and graphs are based upon the actual start and end date and times of the profile/timer.
- **Relative Time** - when selected, the generated report tables and graphs are based upon relative time. Profiles/Timers are graphed from a shared zero point. This option can be used to compare two or more metrics from different sessions in the same scale.
- **Users** - when selected, the generated report tables and graphs are based upon users.

Range Filter - specifies a filter for data coming from the server for a particular session. For example, if a session has two hours of data and filter is set to get a one hour time slice, the graph will only show the one hour time slice. Alternatively, you can use the Graph Zoom feature.

- **Start** - enter start date and time.
- **End** - enter the end date and time.

- **Y-Axis**

Scale - specifies the Y-axis data range format. Changing the scale causes the graph to be refreshed from the server.

- **Linear** - displays the Y-axis linearly.
- **Logarithmic** - displays the Y-axis logarithmically.

Threshold Line - displays a threshold line on the graph at the configured location.

- **Value** - enter the value of the threshold line, that is, the point on the Y-axis at which you want the line to be displayed.
- **Label** - enter the label to display in the legend.

- **Clear Graph** - clears the graph.

10.5 Viewing Virtual User Logs

To view Virtual User Logs for the Active Session:

1. Specify Scenarios on the Composer tab.
2. Click **Run Now** or schedule the scenario to run at a later time using the Schedule.
3. Select the **Virtual User Logs** sub-tab from the **Active Session** tab while a session is running.

The Virtual User Log shows all messages as they are stored without grouping. You can use the filter to see only messages for a particular VU and its chronological path flow. Use this view to debug problems in depth by viewing message contents. The Virtual User Log shows the following information:

VU-ID - Specifies the virtual user number.

Level - Indicates the message level. The level can be Info, Debug, Warning, or Error.

Message - Shows the message.

Type - Specifies the type of message. Some examples of valid types are: Action, BeginPage, Cached Data, Customized Log, DatabankRecord, FoundResource, ParameterSubstitution, RequestHeader, ResponseContent, ResponseHeader, ScriptError, Sync Point, ThinkTime, Verification.

Script - Shows the name of the script that generated the message.

Step - Specifies the step number in the script.

Parent Chain - Specifies the chain of scripts starting with the parent script if the message occurred in a child script called from another script.

Iteration - Shows the iteration number.

Time - Shows the date and time the message occurred.

10.6 Viewing Sync Point Status

You can view the status of Sync Point during a running session. To view Sync Point status:

1. While a session is running, select the **Active Session** tab.
2. Click the **Sync Point Status** sub-tab.

Release - releases the Virtual Users from the selected synchronization point that may have been reached during the execution.

Release All - releases all Virtual Users waiting at synchronization points.

Sync Name - displays the Sync Point name defined in the script.

Users At - displays the number of Virtual User waiting at the Sync Point.

Status - displays current status of the Sync Point.

10.6.1 Releasing Synchronization Points

To release synchronization points:

1. While a session is running, select the **Active Session** tab.
2. Select the Sync Point that you want to release and click **Release** or click **Release All** to release all Sync Points.

10.6.2 Releasing Individual Virtual Users

To release synchronization points:

1. While a session is running, select the **Active Session** tab.
2. Select the virtual user in the Virtual User Status sub-tab.
3. Click the right-mouse button and select click **Release Sync**.

10.7 Attaching to a Session

You can attach to another running session on the Oracle Load Testing Server. To attach to an active session:

To attach to a session with full control:

1. With a session running, click the **Active Session** tab.
2. Select **Attach to** from the **Sessions** menu.

<session list> - lists all sessions that are running on the Oracle Load Testing Server.

3. Select the session to which you want to attach.

To attach to a session in view only mode:

1. With a session running, click the **Active Session** tab.
2. Select **Watch** from the **Sessions** menu.

<session list> - lists all sessions that are running on the Oracle Load Testing Server. Select a session and click **OK**.

3. Select the session to which you want to watch.

10.8 Detaching from a Session

If you have attached to a session, you will need to detach from it to run a separate test.

To detach from a session, select **Detach** from the **Sessions** menu on the **Active Session** tab.

Using Graphs and Reports

You can create custom graphs on the **Reports** tab for post-run analysis for sessions where data has been saved.

11.1 Reports Tab Overview

The Reports tab is where you can view session data for post-run analysis. Use the **Active Session** tab to view reports and graphs while a load test session is still running.

The following options are available on the **Reports** tab:

- **Session** - lets you select the session for which you want to view the session report.
- **Filter** - opens a dialog box for specifying custom date and time range filter options. The filter options specify a specific date and time range to include in the report. When no date and time filter is specified, the report displays the session report for the entire session duration.
- **Delete** - deletes the selected report template.
- **Generate** - generates the report based on the session selected.
- **Export to CSV** - generates the report data into a comma separated value format. You can open the data into a spreadsheet application or save the data to a file.
- **Print/Save HTML** - opens the session report in a separate browser window. From there you can either print the report or save it in HTML format.

The Reports tab has the following sub-tabs after you generate a session report:

- **Session Overview** - shows the start time, end time, and duration, and overview of the number of virtual users, transactions, performance totals and performance by profile.
- **Graphs** - this sub-tab is used to create a custom graph reports.
- **Virtual User Logs** - this sub-tab shows the virtual user information saved to the log file.

11.2 Session Overview

The **Session Overview** sub-tab shows the generated Session Performance Report. You can view the Performance Statistics using the **Session Overview** sub-tab on the **Reports** tab for sessions for which data has been saved.

Once you generate a session report, the statistics show the values for the following performance categories:

<session name> - shows the name of the session used to generate the report.

Start Time - shows the date and time the session was started.

End Time - shows the date and time the session ended.

Duration - shows how long the session ran.

Virtual Users - shows the number of Virtual User that were run during the session and the number with errors.

Transactions - show the total number of transactions and the Average number of transactions per second.

Overview - shows a summary of the performance statistics:

- **Active Virtual Users** - the number of virtual users currently running.
- **Virtual Users with Errors** - the number of virtual users with errors.
- **Transactions Per Second** - the number of times the virtual user played back the script per second.
- **Pages Per Second** - the number of pages returned by the server per second. A "page" consists of all of the resources (i.e. page HTML, all images, and all frames) that make up a Web page.
- **Hits Per Second** - the number of resource requests to the server per second. Each request for a page, individual images, and individual frames is counted as a "hit" by Oracle Load Testing. If Oracle Load Testing does not request images from the server (as specified in the Download Manager), images are not included in the hit count. The **Hits Per Second** and **Pages Per Second** counts will be the same if images are not requested and there are no frames in the page.
- **Kilobytes Per Second** - the number of kilobytes transferred between the server and browser client per second.

Totals - show the totals of the performance statistics:

- **Transactions** - the total number of times the virtual user played back the virtual user profile.
- **Transactions with Errors** - the total number of virtual user profile iterations that had errors.
- **Pages** - the total number of number of OpenScript step groups in the script.
- **Hits** - the total number of resource requests to the server.
- **Kilobytes** - the total number of kilobytes transferred between the server and browser client.

Performance by Profile - shows the performance statistics for the Virtual User profile(s) run during the load test.

- **<Profile Name>** - the latest, minimum, maximum, and average performance for the virtual user profile in seconds.
- **<Timer Name>** - the latest, minimum, maximum, and average performance for the server response timers in seconds. Server Response timers are added to scripts using Oracle OpenScript.
- **Min** - the minimum performance for the virtual user profile or server response timer in seconds.
- **Max** - the maximum performance for the virtual user profile or server response timer in seconds.

- **Avg** - the average performance for the virtual user profile or server response timer in seconds.
- **Std Dev** - the number of seconds that the performance of virtual user profiles or server response timers deviated from the mean (average value).
- **90th %** - shows the number of seconds that the performance of ninety percent of the virtual users or server response timers was at or below.
- **Pass** - the number that passed.
- **Warning** - the number with warnings.
- **Fail** - the number that failed.

11.3 Graphs

The **Graphs** sub-tab is where you can generate graph reports from the counter data saved during the session.

Initially the Graphs tab displays a blank dashboard. Click **Add Graph** to open the graph options. Expand the Session tree to view the available default graphs and counters.

<session tree> - the Session tree lists the available graphs and counters for the selected scenario.

Filter Counters - selects the filter to use for the session data counters.

- **Show Average Performance Only** - when selected, the counters tree shows only average performance counters.
- **Show All Counters** - when selected, the counters tree shows all counters.

The Session tree has the following available graphs and counters for the selected session:

Default Graphs - lists the built-in graphs that can be used for reporting:

- **Performance Vs. Time** - shows the average virtual user run time over a period of time. If you have multiple profiles in the running Scenario the Performance Vs. Time graph shows separate plot lines for each scenario profile.
- **Trans Failed/sec Vs. Time** - shows the number of failed transactions-per-second for virtual user over a period of time. If you have multiple profiles in the running Scenario the Errors Vs. Time graph shows separate plot lines for each scenario profile. The failed transactions-per-second depends upon the number of virtual users being run by individual profiles and the number of server requests (page, image, frame, etc.) that fail during the Oracle Load Testing session.
- **Statistics Vs. Time** - shows averages for virtual user hits, pages, transactions, and Kilobytes per second over time. The Average Statistic values are a graphical representation of the Average Value data from the Performance Statistics over time. The plot points are updated while the session runs the virtual users.
- **Performance Vs. Users** - shows the average run time of the script for each running virtual user in each profile. If you have multiple profiles in the running Scenario, the Performance Vs. Users graph shows separate bars for each scenario profile running. For example, if you run 100 VUs and ramp 10 at a time, you will see the average response time plotted at 10 VUs, 20 VUs, 30 VUs, etc. up to 100 VUs.
- **Trans/Failed Vs. Users** - the number of failed transactions-per-second for each virtual user profile. If you have multiple profiles in the running Scenario the

Errors Vs. Users graph shows separate plot lines for each scenario profile running. The failed transactions-per-second depends upon the number of virtual users being run by individual profiles and the number server requests (page, image, frame, etc.) that fail during the Oracle Load Testing session.

The graph adds plot points based upon the number of running virtual users as the session ramps up to the total number of virtual users for all scenario profiles.

- **Users Vs. Time** - shows the relative time when the virtual users for each profile started running. The graph represents the Autopilot ramp up times and the number of virtual users ramped up for each profile.
- **Performance, Users vs. Time** - this graph combines the Performance vs. Time and the Users vs. Time graphs.
- **Hits/sec, Users vs. Time** - this graph combines the Hits per second and the Users vs. Time graphs.
- **KBs/sec, Users vs. Time** - this graph combines the Kilobytes per second and the Users vs. Time graphs.
- **Pages/sec, Users vs. Time** - this graph combines the Kilobytes per second and the Users vs. Time graphs.
- **Trans/sec, Users vs. Time** - this graph combines the transactions per second statistic from the Statistics vs. Time graph and the Users vs. Time graph.
- **All Trans/sec, Users vs. Time** - this graph combines the transactions per second statistic from the Statistics vs. Time graph and the Users vs. Time graph.

Note: For OpenScript scripts, steps run in the **Initialize** and **Finish** sections do not contribute to the overall script execution time reported in Oracle Load Testing reports. The time taken for the **Initialize** and **Finish** sections are not included in the script times so as not to skew the average time when the script runs for multiple iterations.

Session Counters - lists the built-in session counters that can be used for reporting:

- **Hits/sec** - this counter graphs the hits per second.
- **Kb Rcvd/sec** - this counter graphs Kilobytes received per second.
- **Number of VUs** - this counter graphs Kilobytes received per second.
- **Pages Rcvd/sec** - this counter graphs Kilobytes received per second.
- **Trans Failed/sec** - this counter graphs the number of transactions failed per second.
- **Trans sec** - this counter graphs the number of transactions failed per second.
- **VUs with Errors** - this counter graphs the number Virtual Users with errors.

Profile Counters - lists the counters from the Virtual User profile that can be used for reporting:

- **Profile Nodes** - list the counters for virtual user profiles available to include in the report graphs. The profile is the name of the virtual user profile(s) that were run in the session. Double-click the counter name to add counter(s) to the Selected Data Series list.
- **Timer Nodes** - list the page timers available to include in the report graphs. Timers correspond to the individual pages of the script(s) run by the virtual user profile.

They allow you to view the performance of a specific page download within a test run, allowing you to identify bottlenecks. The timer is the name of the virtual user profile(s) that were run by the session. The timers consist of the virtual user profile name/Script name and the page/Step group number defined the script. The timers are added to the session data when you select the Generate timers for resources check box in **Options** from the **User** menu then selecting **Default Properties**. Double-click with the mouse to add timer(s) to the Selected Data Series list.

ServerStats Counters - lists the counters from the metrics included in the ServerStats counter configuration. The ServerStats counters are based upon the data source(s) and counters that were included in the ServerStats Data Source configuration when the Oracle Load Testing session was run. Double-click with the mouse to add counter(s) to the Selected Data Series list.

Oracle Database Diagnostics - groups a set of Oracle Database profile metrics on the same graph. These metrics appear only if an Oracle Database ServerStats configuration is used with the session run. The graphs and metrics are as follows:

- **Average Active Sessions** - shows Average Active Sessions metrics.
- **Host CPU Utilization** - shows Host CPU Utilization metrics.
- **Host PGA Utilization** - shows Host "Program Global Area (PGA)" Utilization metrics.
- **DB Throughput** - groups "Physical I/O per second" and "Physical I/O requests total" metrics.

Graphs have the following toolbar buttons:

- **Load Query** - opens a dialog box for opening a saved graph data query.
- **Save Query** - opens a dialog box for saving the current graph data query.
- **Export** - opens options for saving the current graph to a file.
 - **PNG** - opens options for saving the current graph to a PNG format image file.
 - **SVG** - opens options for saving the current graph to a SVG format image file.
 - **CSV** - opens options for saving the current graph data to a comma-separated-value file.
 - **Excel HTML** - opens options for saving the current graph data to an Excel spreadsheet file.
- **Toggle Legend** - displays or hides the names and plot line color of each data series in the graph.
- **Toggle Stats** - displays or hides the chart statistics table view. The chart statistics show the Data Series and values for each series.
- **Refresh Graph** - refreshes the data in the graph from the database.
- **Toggle Zoom** - toggles the zoom tool on and off. When in zoom mode, a magnifier appears in the graph area. Click the magnifier and select the zoom options from the popup menu.
- **Options** - opens the options for customizing graph and x/y axis options. The options are the same as graphs created on the Active Session Runtime Graphs sub-tab. See the options in [Section 10.4.2, "Custom Graphs"](#).
- **Clear Graph** - clears the graph.

11.4 Virtual User Logs

The **Virtual User Logs** sub-tab is where you can view log messages generated during the session.

To view Virtual User Logs:

1. Specify Scenarios on the Composer tab.
2. Click **Run Now** or schedule the scenario to run at a later time using the Schedule.
3. Select the **Virtual User Logs** sub-tab from the **Reports** tab for post session analysis.

The Virtual User Log shows all messages as they are stored without grouping. You can use the filter to see only messages for a particular VU and its chronological path flow. Use this view to debug problems in depth by viewing message contents. The All Logs view shows the following information:

Show - Specifies the message level. The level can be Info, Debug, Warning, or Error.

- **Info** - show only Info level messages.
- **Debug** - show only Debug level messages.
- **Warning** - show only Warning level messages.
- **Error** - show only Error level messages.
- **Content** - show all Info level messages.
- **Filter** - toggles the Filter bar for specifying the filter method to use.

VU-ID - Specifies the virtual user number.

Level - Indicates the message level. The level can be Info, Debug, Warning, or Error.

Message - Shows the message.

Type - Specifies the type of message. Some examples of valid types are: Action, BeginPage, Cached Data, Customized Log, DatabankRecord, FoundResource, ParameterSubstitution, RequestHeader, ResponseContent, ResponseHeader, ScriptError, Sync Point, ThinkTime, Verification.

Script - Shows the name of the script that generated the message.

Step - Specifies the step number in the script.

Iteration - Shows the iteration number.

Timestamp - Shows the date and time the message occurred.

11.5 Setting the Reporting Options

To set the Oracle Load Testing Report options:

1. Select **Options** from the **User** menu.
2. Select **Reporting** to set additional options.

Intervals - the Intervals options are as follows:

- **Data collection interval** - specifies how often the aggregate Oracle Load Testing agent performance data is collected for reporting to the Oracle Load Testing database.

- **Active Session UI refresh interval** - specifies how often to refresh the user interface including the VU grid, ServerStats Display, and Sync Point Status.
- **Graph refresh interval** - specifies how often to redisplay graphs in the View Run Graphs tab.

VU Status Grid - the VU Grid options are as follows:

- **Default rows per page** - specifies the maximum number of rows to display on a page on the Watch VU Grid tab. The value must be between 10 and 100, inclusive.

VU Logs Grid - the VU Logs options are as follows:

- **Default rows per page** - specifies the maximum number of rows per page to display in the VU logs. The value must be between 10 and 100, inclusive.

Performance Reporting - specifies the default setting for Performance Statistics and Session Reports, as follows:

- **Calculate performance end-to-end** - specifies that think time is included in performance statistics and session reports.

Custom Graphs - the Custom Graphs option is as follows:

- **Maximum Data Series Per Graph** - specifies the maximum number of data series that can be graphed. The default is 20.
- **Maximum Data Points Per Graph** - specifies the maximum number of data points that can be graphed. The default is 100.

3. Enter the how often you want data to be collected in the **Data collection interval** field.
4. Enter the how often you want to refresh the user interface in the **Active Session UI refresh interval** field. This includes the VU grid, ServerStats Status Display, and the Sync Point Status display.
5. Enter the how often you want to refresh the report display in the **Graph refresh interval** field.
6. Specify the maximum number of data series to include per graph.
7. Click **OK**.
8. Start an Oracle Load Testing session.

11.6 Creating Reports

Oracle Load Testing statistics and performance reports and graphs can be viewed for sessions for which data has been saved.

To create custom reports:

1. Click the **Reports** tab.
2. Select the session and click **Generate** to generate a Performance report.
3. To create graphs, click **Add Graph** on the Graphs sub-tab.
4. Double-click data counters you want to view in the tree view.
5. Repeat for all of the data series that you want to graph.
6. Click the **Options** button display the Graph Options for naming the graph and selecting x/y axis options.

7. Click **OK**.
8. Click **Back to Graphs** to go back to the thumbnail view. Each custom report is displayed under a separate thumbnail.

11.7 Exporting Graphs and Data

The graph export options let you specify export options for the current graph.

To export graph image or data:

1. Create a graph report.
2. Select the appropriate export type from the **Export** menu on the top of the graph.

The following sections provide specific steps and information.

11.7.1 Exporting CSV Data

When you export generated data from a report, Oracle Load Testing creates a comma separated value (.csv) file containing the raw data used to generate reports. The first line of the file contains the field names separated by commas. Each additional line contains the data values separated by commas. The following is an example of a .csv file from a Performance vs. Time report:

```
Time,Entire Job,tutor1,tutor4,
04-09-2001 15:59:48,,.6,,
04-09-2001 16:00:00,,.13,
04-11-2001 17:08:04,,1.15,,
04-11-2001 17:08:29,1.39,,.24,
04-11-2001 17:09:19,,.98,,
04-11-2001 17:09:25,,.12,
```

This example includes the following fields:

```
Time
Entire Job
tutor1
tutor4
```

The additional lines of the .csv file list the actual data (the time values followed by the number of seconds).

The .csv file for each type of report will contain the type of data specific to the report type.

You can export the raw data used to generate reports to a comma separated value (.csv) file. To export generated report data:

1. Click the **Reports** tab
2. Create a session report.
3. Select the **Graphs** sub-tab.
4. Click **Add Graph**.
5. Double-click data counters you want to view in the tree view.
6. Repeat for all of the data series that you want to graph.
7. Select **CSV** from the **Export** menu on the top of the graph.
8. Select the save or open option from the browser file popup.

11.7.2 Exporting for Excel

To export as file that can be used with Excel:

1. Click the **Reports** tab
2. Create a session report.
3. Select the **Graphs** sub-tab.
4. Click **Add Graph**.
5. Double-click data counters you want to view in the tree view.
6. Repeat for all of the data series that you want to graph.
7. Select **Excel HTML** from the **Export** menu on the top of the graph.
8. Select the save or open option from the browser file popup.

11.7.3 Exporting to PNG

You can export reports to a PNG file. To export to a PNG file:

1. Click the **Reports** tab
2. Create a session report.
3. Select the **Graphs** sub-tab.
4. Click **Add Graph**.
5. Double-click data counters you want to view in the tree view.
6. Repeat for all of the data series that you want to graph.
7. Select **PNG** from the **Export** menu on the top of the graph.
8. Select the save or open option from the browser file popup.

11.8 Working With Graph Queries

A graph query consists of the sessions, data series, and X and Y-axis settings. Once you select this information you can save it and reuse it later in the following ways:

- save a graph of a single session and apply it to a new session to create a new graph.
- save a graph of a baseline session and a run session and apply it to a new run session to create a new graph with the original baseline.
- save a graph of a single session and apply it to a new session and add the counters to the current graph definition.

11.8.1 Saving Graph Queries

To save a graph query:

1. Select the sessions and data series that you want to graph.
2. Click the **Save Query** button on the top of the graph.
3. Enter a name for the query.

<query list> - lists the existing saved queries.

Name - enter a name for the new query or select an existing query to overwrite it.

4. Click **OK**.

11.8.2 Running a Saved Query

To run a saved query:

1. Click the **Load Query** button on the top of the graph to open the Load Query dialog box.
<**Query List**> - lists the existing queries. Select the query that you want to run.
Session - select the session against which you want to run the query.
2. Select the saved query.
3. Select the session against which you want to run the query.
4. Click **OK**.
5. A warning message is displayed if some of the data series cannot be applied to the selected session. If this happens, click **OK** to continue.
6. The new graph is generated.

11.8.3 Editing Queries

To edit a query:

1. Select **Options** from the **User** menu.
2. Select **Graphs** in the **Manage** group.
Edit - displays the Edit Graph Query dialog box.
Delete - deletes the selected query.
Query Name - lists the existing queries.
3. Select the query you want to change and click **Edit**.
4. Make and changes and click **OK**.
5. Click **Close** to exit the Graph Query Manager dialog box.

11.8.4 Deleting Queries

To delete a query:

1. Select **Options** from the **User** menu.
2. Click **Graphs** from the **Manage** group.
3. Select the query you want to delete and click **Delete**.
4. Click **Yes** to confirm the deletion.
5. Click **Close** to exit the Graph Query Manager dialog box.

11.9 Using Time Range Filters

The filter options specify a specific date and time range to include in the report. When no date and time filter is specified, the report displays the session report for the entire session duration.

To specify a custom date and time range filter:

1. Click the **Reports** tab.
2. Select the session and click **Generate**.
3. After the report loads, click the Filter icon.
4. Set the Start and End date and time values to the time range to include in the report.
5. Set the **Calculated Percentile** option.
6. Click **Apply** to generate a new report that include only the data between the Start and End date and time.

11.10 Using Raw Data

This section explains how to use the raw data that can be generated by a load test session. The raw data provides the ability to view non-aggregated data collected from all Virtual Users running in a load test across all Load Agent machines for reporting.

11.10.1 Enabling Raw Data Collection

To enable raw data collection by default:

Caution: The raw data files generated during a load session can increase in size rapidly causing file size and disk space issues on the agent machine. A brief 15 minute load test session can produce files of approximately 40MB. When running load test sessions with the **Persist Raw Data** setting enabled, be sure to plan for file size and disk space issues.

1. Select **Options** from the **User** menu.
2. Select **Default Properties** in the **Profiles** group.
3. Set the **Persist Raw Data** setting to `True` in the **Agent Preferences** section of the default settings.
4. Be sure to set the **Persist Raw Data** setting to `False` after completing the raw data collection load test session.

To enable raw data collection for a specific session:

1. Add the script to a Scenario.
2. Select the script name in the Profiles panel.
3. Set the **Persist Raw Data** setting to `True` in the **Agent Preferences** section of the default settings.

Note: The options for the first script in the session govern the settings for same options for the other scripts in the same session (even though user interface shows different values for the options in different scripts).

11.10.2 Viewing Raw Data in Reports

When raw data collection is enabled, Oracle Load Testing saves every single measured data point to the database.

To view raw data in reports:

1. After running a load session with raw data collection, click the **Create Reports** tab.
2. Select the session in the **Session(s)** list.
3. Click **Filter Counters**.
4. Select **Show All Counters**. The Raw Data node is added to the counters tree.
5. Expand the Raw Data node and select the counters to add to the report.

11.10.3 Raw Data Database Tables and Counters

This section lists the raw data database tables and counters. Advanced users can use customized database queries to extract additional reports from the raw data collected during a load test.

COUNTERHDR: Contains metadata for an individual counter.

- COUNTERHDRID: unique counter ID.
- SESSIONRUNID: ID of the session for which this counter was created (index into SESSIONRUN table).
- SOURCE: Indicates the type of counter (1=ServerStats Counter, 4=OLT Aggregated Counter, 8=OLT Raw Counter).
- COUNTERNAME: Name of the counter.
- OBJNAME: The name of the script or ServerStats profile to which this counter applies, if applicable. Null indicates a session-level counter.
- INSTNAME: The OpenScript-generated UUID of the step group to which the counter applies, if applicable.
- NICENAME: The user-readable name of the step group to which this counter applies, or a collection of UUIDs indicating additional information, if applicable. This data is generated by OpenScript.

COUNTERRUN: Contains counter values.

- COUNTERHDRID: ID of the counter to which this data applies (index into COUNTERHDR table).
- COUNTERVAL: Numeric counter value.
- COUNTERTS: Timestamp that the counter value was measured.
- ITERATION: Iteration of the script for which the value was measured (aggregated counters only).
- NUMBERVUS: Number of VUs that were running when the counter was measured (aggregated counters only).

SESSIONRUN: Contains metadata about load sessions.

- SESSIONRUNID: Unique Session ID.
- SESSIONNAME: Unique Session Name.
- MACHINE: Weblogic URL of the server which started this session (ie. t3://hostname:port).
- START_TS: Timestamp of the start of session.
- FINISH_TS: Timestamp of the end of session. Null if the session is still running.

- INFOSTRING: Indicates status of the load test. Possible values are "end of session" or "running".

11.10.4 Example Database Queries for Raw Data

This section provides example SQL queries for generating custom reports from the raw data saved to the database during a load test session.

- List all session names in reverse chronological order:

```
SELECT sessionname FROM sessionrun ORDER BY sessionrunid DESC;
```

- List raw performance data for VU 5 in session named "MySession1":

```
SELECT dat.counterts as "Timestamp", dat.counterval as "Avg Response Time
(sec)" FROM counterrun dat, counterhdr hdr
WHERE hdr.counterhdrid = dat.counterhdrid AND
      hdr.countername = 'Avg Performance (sec) (Raw, VU=5)' AND
      hdr.source=8 AND
      hdr.sessionrunid=(SELECT sessionrunid FROM sessionrun s WHERE
s.sessionname='MySession1')
ORDER BY dat.counterts;
```

- List average of raw page performance counters ordered by page name (similar to OLT session report) for session named "MySession1":

```
SELECT hdr.nicename AS "Name", AVG(dat.counterval) AS "Avg Response Time (sec)"
FROM counterrun dat, counterhdr hdr
WHERE hdr.counterhdrid = dat.counterhdrid AND
      hdr.countername LIKE 'Avg Performance (sec) (Raw, VU=%)' AND
      hdr.source=8 AND
      hdr.sessionrunid=(SELECT sessionrunid FROM sessionrun s WHERE
s.sessionname='MySession1')
GROUP BY hdr.nicename
ORDER BY hdr.nicename;
```

- Calculate 75th percentile for the step group named "[1] Landing page" in session named "MySession1":

```
SELECT PERCENTILE_DISC(0.75) WITHIN GROUP (ORDER BY dat.counterval) AS "75th
Percentile"
FROM counterrun dat, counterhdr hdr
WHERE hdr.counterhdrid = dat.counterhdrid AND
      hdr.nicename='[1] Landing Page' AND
      hdr.countername LIKE 'Avg Performance (sec) (Raw, VU=%)' AND
      hdr.source=8 AND
      hdr.sessionrunid=(SELECT sessionrunid FROM sessionrun s WHERE
s.sessionname='MySession1');
```

Working With Sessions

This chapter describes the how to work with Oracle Load Testing session data. The following session actions are available:

- Attach to and detach from sessions that are running on the Oracle Load Testing server. This enables a tester to share real-time test results and to collaborate with team members during testing.
- Export specific session data from and import session data to any Oracle Load Testing instance. This enable users to share session data stored on one instance of Oracle Load Testing with other users running another instance of Oracle Load testing without having to export and import an entire Oracle Load Testing database. The user importing the session data can then generate reports for the data using their own Oracle Load Testing instance.

12.1 Managing Sessions

The Session options lets you rename sessions, delete sessions, purge VU logs, and import/export session data.

To use sessions options:

1. Select **Options** from the **User** menu.
2. Expand Options and select **Sessions** in the **Manage** group.

This dialog box lets you manage session data in the database.

Edit - displays the Edit Session dialog box for changing the name and status of the session.

Delete - deletes the entire selected session data from the database. To select more than one session hold down the CTRL key.

Purge VU Logs - deletes the Virtual User logs for the selected session. To select more than one session hold down the CTRL key.

Import - opens a dialog box for selecting the session archive (.osd) file to upload.

Export - saves the selected session to an archive (.osd) file.

Refresh - refreshes the session list from the database.

Session - the names of the virtual user sessions run by Oracle Load Testing. Oracle Load Testing lets you specify a name for the session data when you start the session to play back virtual users.

Start Time - shows the date and time of day the virtual user session was started.

End Time - shows the date and time of day the virtual user session ended.

Status - shows the status of the session. One of the following status entries may appear:

- end of session - the session data was stored to the database successfully.
- start of session - the session database is currently being accessed by an Oracle Load Testing session.
- saving data - session data is being saved.

Result - shows the result for this session: No result | Passed | Failed.

Logs - indicates whether VU log data was saved for this session.

3. Select the session you want to manage.
4. Click an option on the toolbar to manage the session.
5. When editing a session, change the Name, Status, and Result settings as needed.
6. .Click OK.

12.2 Exporting Session Data

You can export Oracle Load Testing session data to an external file which can then be imported into another instance of Oracle Load Testing installed elsewhere.

To export Oracle Load Testing session data:

1. Select **Options** from the **User** menu.
2. Expand Options and select **Sessions** in the **Manage** group.
3. Select the session to export and click **Export**.
4. Click **Save** when the File Download dialog box appears.
5. Specify the location and file name and click **Save**.

12.2.1 File Format

The default name of the file is the same as the session name. The filename can be changed to any other name. The data is not affected.

The exported file has a `.osd` (OLT Session Data) file extension. The file is in `.zip` compression format. You can change the extension to `.zip` and open the file in any ZIP utility to access the raw data. However, the file extension must be `.osd` to import the file into another Oracle Load Testing instance. The file structure is as follows:

```
Session Data\  
  db\  
    sessionRun.csv  
    scenarioReports.csv  
    counterHDR.csv  
    counterRun.csv  
    awrInfo.csv  
    vlog.csv  
    metadata.csv
```

All the Database related data is stored in `/Session Data/db/`. The `db` folder contains comma-separated-value (`.csv`) files which represent corresponding tables and their session related data in the database. Each `.csv` file contains the column names as their first row and the values on subsequent lines. All the `.csv` files are encoded in UTF-8 format.

12.3 Importing Session Data

You can import Oracle Load Testing session data that has been previously exported from another Oracle Load Testing instance as an Oracle Session Data (.osd) file.

To import Oracle Load Testing session data:

1. Select **Options** from the **User** menu.
2. Expand Options and select **Sessions** in the **Manage** group.
3. Click **Import**.
4. Click **Browse**.
5. Select the session .osd file and click **Open**.
6. Click **Upload**. The imported session will now appear in the Session Lists of the current Oracle Load Testing instance.
7. Click the **Reports** tab and select the imported session name to generate reports from the session data.

Managing Files and Databases

This chapter describes the how to import and export files to and from the server, how to manage the Oracle Load Testing database, and how to select the database to use.

13.1 Database Management

The Oracle Load Testing database is used to store the results of your load test sessions. You use the **Oracle Application Testing Suite Database Configuration** to configure databases for use in Oracle Load Testing. The **Databases** option on the **Manage** menu in Oracle Load Testing is used to set the current database to use. The Database Manager dialog box has the following options:

Name - displays the ODBC System Data Source name of the database.

Description - displays a description of the database.

Type - displays the type of database.

Connection String - displays the connection string.

13.1.1 Adding an Oracle Database

To add an Oracle database:

1. Use the Oracle Application Testing Suite Database Configuration utility to configure database connections to Oracle Load Testing databases. On Windows machines, you can access the Database Configuration utility from the **Tools** sub menu of the Oracle Application Testing Suite **Start** menu. On Linux machines, you can access the Database Configuration utility from `<oats_install>/bin/DbConfig.sh`.
2. Select **Oracle Load Testing**.
3. Click **New**.
4. Select **Use existing schema** or **Create new schema & tables**.

This dialog box has the following options:

System user credentials - specify the login authentication credentials if you select to create a new schema and tables, as follows:

- **User name** - enter the login user name for the database system.
- **Password** - enter the login password for the database system.

Connection details - enter the Oracle database connection information.

- **Name** - enter the name of the database. This name appears in the Database selection lists for Oracle Load Testing and as the ODBC Data Source name.
 - **Description** - enter any description text to identify the database.
 - **Host** - enter the host name of the machine on which the database server is running.
 - **Port** - enter the port number that you want to communicate with. If you are running multiple instances of the database on the same host, specify the port number you want to communicate with.
 - **SID** - enter the System Identifier for the Oracle database.
 - **User name** - enter the login user name for the database.
 - **Password** - enter the login password for the database.
5. Enter the database information.
 6. Click **Save**.
 7. If creating a new schema, enter the password for both the Administrator and default accounts and click **OK**.
 8. Click **Close**.
 9. Either restart the computer or restart the Oracle Application Testing Suite Application service by going to the Control Panel Administrative Tools and open **Services** and restart the Oracle Application Testing Suite Application Service.

13.1.2 Selecting the Database to Use

1. Select **Programs** from the **Start** menu then select **Oracle Application Testing Suite Database Configuration** from the **Oracle Application Testing Suite - Tools** menu.
2. Select Oracle Load Testing as the product type.
3. Select the database you want to use to store Oracle Load Testing data.
4. Click **Set current**.
5. Click **Close**.
6. Restart Oracle Load Testing.

13.1.3 Removing Databases

You can remove databases from Oracle Load Testing. Doing so does not delete the database files. It removes the database from the database selection lists so that the database is no longer available in Oracle Load Testing. The database cannot be deleted if it is open in Oracle Load Testing.

To remove a database from Oracle Load Testing:

1. Select **Programs** from the **Start** menu then select **Oracle Application Testing Suite Database Configuration** from the **Oracle Application Testing Suite - Tools** menu.
2. Select the database that you want to remove.
3. Click **Remove**.
4. Click **Yes** to confirm the deletion.

13.1.4 Updating Databases

You can change the name, location, and login information for databases that you are using in Oracle Load Testing.

1. Select **Programs** from the **Start** menu then select **Oracle Application Testing Suite Database Configuration** from the **Oracle Application Testing Suite - Tools** menu.
2. Select the database that you want to update.
3. Click **Update**.
4. Make any changes.
5. Click **OK**.

13.1.5 Exporting and Importing Databases

Oracle Application Testing Suite Release 12.1+ supports Oracle Database 10g, 11g or XE 10g as the RDBMS for its load testing and test management components. Oracle Application Testing Suite Release 12.5+ supports Oracle Database 11g, 12c or XE 11g as the RDBMS for its load testing and test management components. In some cases, it may be necessary to transfer data between databases. The basic procedure to is to export a database dump from one database and then import the dump file into the new database. The following sections provide basic details for exporting and importing an Oracle XE database. See the *Oracle Database Utilities* documentation for additional information about using the original Export (`exp`) and Import (`imp`) utilities and the Data Pump Export (`expdp`) and Data Pump Import (`impdp`) utilities.

13.1.5.1 Exporting Databases Using Export Utility

The Database Export utility provides a way to transfer data objects between Oracle databases, even if the database reside on different platforms. Running the Export utility on an Oracle database extracts objects (such as tables) with all related objects (such as indexes, comments, and grants), if any. The extracted data is written to an Export file.

To export a database using the export utility:

1. Open a command prompt.
2. Change directory to Oracle home. For example,
C:\OracleATS\oxe\app\oracle\product\10.2.0\server\BIN
3. Enter the following command:

```
exp userid=username/password owner=schema name file=Exportfile.dmp
log=Export.log
```

where:

`userid=` specifies the username/password of the user performing the export.

`owner=` specifies the schema name to export ("oats", "olt", "otm").

`file=` specifies the file name to which the schema will be exported.

`log=` specifies the log file name to save any warnings or errors that may have occurred during export.

For additional parameter information, type `exp help=y` at the command prompt.

Example:

```
exp userid=system/oracle owner=olt file=olt.dmp log=olt.log statistics=none
```

13.1.5.2 Importing Databases Using Import Utility

The Import utility reads the object definitions and table data from an Export dump file and inserts the data objects into an Oracle database.

To import a database from an exported dump file:

1. Open Sql Plus and create a user (for example, Test) and grant roles (connect, resource, create any view) to the newly created user.
 - create user Test identified by password;
 - grant connect, resource, create any view to Test;
2. Open a command prompt.
3. Change directory to Oracle home. For example,
C:\OracleATS\oxe\app\oracle\product\10.2.0\server\BIN

4. Enter the following command.

```
imp userid=user/pwd fromuser=OATS touser=Test file=Exportfile.dmp log=test.log
```

where:

userid= specifies the username/password of the user performing the import.

fromuser= specifies a list of owner usernames. This parameter enables you to import a subset of schemas from an export file containing multiple schemas.

touser= specifies a list of usernames whose schemas will be targets for import.

file= specifies the name of the exported dump file. For example, Exportfile.dmp.

log= specifies the log file name to save any warnings or errors that may have occurred during import.

For additional parameter information, type `imp help=y` at the command prompt.

Example:

```
imp userid=system/oracle fromuser=olt touser=tempuser file=olt.dmp log=t.log
```

5. Add the new database to the Oracle Application Testing Suite using the Oracle Application Testing Suite Database Configuration utility. See [Section 13.1.1, "Adding an Oracle Database"](#) for additional information.

13.1.5.3 Importing Databases Using Data Pump Import Utility

In cases where the database was exported using the Data Pump Export (`expdp`) utility, you use the Data Pump Import (`impdp`) utility to import the database.

To import a database from a Data Pump Export utility:

1. Create a folder. (for example c:\test)
2. Open Sql Plus and enter the following command:

```
SQL> create directory test as '/backup/db_name/exports/dpump' ;
```
3. Grant read and write permissions to the users who will be performing the data pump import. Note: before granting following permission please log in as sys as sysdba (different user than logged in user and should be logged in as dba)

```
SQL> grant read, write on directory test to system, user1, user2, user3;
```

4. Open a command prompt.

5. Change directory to Oracle home. For example,
C:\OracleATS\oxe\app\oracle\product\10.2.0\server\BIN
6. Enter the following command:

```
impdp system/<Password> DIRECTORY=<dir name> DUMPFILE=<dmpfile.dmp>
LOGFILE=<logfile.log> remap_schema=<source_schema>:<target_schema>
```

For additional parameter information, type `impdp help=y` at the command prompt.

Example:

```
impdp system/oracle DIRECTORY=test DUMPFILE=expdp_Exportfile.dmp
LOGFILE=impfull.log remap_schema=oats:oats
```

13.2 Importing and Exporting Data Bank Files

You can import and export Data Bank files (.csv) to and from the Oracle Load Testing server and the local machine. This can be useful if you are accessing Oracle Load Testing on a remote machine via the Web UI and want to copy data files to another server.

To export Data Bank files from the server to the local machine:

1. Select **Options** from the **User** menu.
2. Select **Data Bank** in the **Manage** group.
3. Select the Data Bank file (.csv) file that you want to export.
4. Click **Export**.
5. Click **Save As**.
6. Select the location where you want to save the file and click **Save**.

To import Data Bank files from the local machine to the server:

1. Select **Options** from the **User** menu.
2. Select **Data Bank** in the **Manage** group.
3. Click **Import Data Bank**.
4. Enter the file name or click **Browse** to locate the file.
5. Select the destination location from which you want to copy the file.
6. Click **Open**.
7. Click **Upload**.

13.3 Importing and Exporting Data Files

You can import and export Download Manager data files (.dat) to and from the Oracle Load Testing server and the local machine. This can be useful if you are accessing Oracle Load Testing on a remote machine via the Web UI and want to copy data files to another server.

To export data files from the server to the local machine:

1. Select **Options** from the **User** menu.
2. Select **Data File** in the **Manage** group.

3. Select the data file (.dat) file that you want to export.
4. Click **Export**.
5. Click **Save As**.
6. Select the location where you want to save the file and click **Save**.

To import data files from the local machine to the server:

1. Select **Options** from the **User** menu.
2. Select **Data File** in the **Manage** group.
3. Click **Import Data File**.
4. Enter the file name or click **Browse** to locate the file.
5. Select the destination location from which you want to copy the file.
6. Click **Open**.
7. Click **Upload**.

13.4 Using the Reset Password Utility

The Reset Password utility allows the installing user of the Oracle Application Testing Suite web applications to reset the password for the Administrator account in an Oracle Load Testing or Oracle Test Manager database.

The installing user of the Oracle Application Testing Suite can reset the password for:

- Oracle Load Testing Administrator user
- Oracle Test Manager Administrator user

The default Administrator user password is the password specified during the Oracle Application Testing Suite installation.

See the *Oracle Application Testing Suite Installation Guide* for additional information about using the Reset Password utility.

Setting Options

This chapter explains how to specify default options used by Oracle Load Testing.

You can set Oracle Load Testing options for custom browsers, repositories, scenario defaults, session start and stop, session profile, and reporting options using **Options** from the **User** menu. Selecting this option opens the Options dialog box.

14.1 Setting General Options

The General options let you specify the default settings for validation and timeouts.

Validate hostname automatically when adding/editing a system - checks to see if the Oracle Load Testing server can connect to the specified system. If it cannot, Oracle Load Testing displays a dialog box asking you if you want to proceed anyway.

Validate monitor automatically when creating a monitor - checks to see if the monitor can be applied to the target system when you create a monitor.

Databank Setup Timeout - specifies the time, in seconds, to wait for databanking operations before timing out.

14.2 Setting System Options

The System options let you add, edit, and delete systems, system groups and data collectors used to run Virtual users. See [Chapter 4, "Managing Systems"](#) for additional information.

14.3 Setting Manage Options

The Manage options let you perform add, edit, delete, and import/export tasks for Repositories, Sessions, Scenarios, Graphs, Scripts, Data Banks, and Data Files in the Manage group of the user options.

14.3.1 Repository Options

Repositories give you the ability to share files. Any shared network directory can be used as a repository. Since the Oracle Application Testing Suite Application Service runs as the local SYSTEM user, there may be a policy restricting access to your network share. You can fix this by doing one of the following:

- Ensure that both sharing permissions and security permission on the remote network directory allows for other SYSTEM users to gain access. The least restrictive setting is to allow the windows user "Everyone" to be given permission.

- Configure the Oracle Application Testing Suite Application Service to run under a specific user account rather than the local System user account.

New - adds a new entry to the table.

Delete - deletes the selected repository.

Repository Name - enter the name of the repository.

Path - enter the path of the repository. If the path you specify is a shared network drive, the Oracle Load Testing Server must have access to that drive. By default, the Oracle Load Testing Server runs under the "Local System" account. You may need to change this to a user account in the Services panel.

To add a repository:

1. Select **Options** from the **User** menu.
2. Expand **Manage** and click **Repositories**.
3. Click **New**. A new entry is made in the table.
4. Enter the name of the repository.

Note: If you plan to use OpenScript scripts with Oracle Load Testing, the repository names you specify should match the repository name specified in OpenScript (including case).

5. Enter the location of the repository.

14.3.2 Session Options

The Session options let you rename sessions, delete sessions, purge VU logs, and import/export session data. See [Chapter 12, "Working With Sessions"](#) for additional information.

14.3.3 Scenario Options

The Scenario options lets you edit, delete import and export scenarios. See [Chapter 9, "Running and Saving Scenarios"](#) for additional information.

14.3.4 Graph Options

The Graph options lets you edit and delete saved graph queries. See [Chapter 11, "Using Graphs and Reports"](#) for additional information.

14.3.5 Scripts Options

The Script options let you import and export scripts that have been exported from OpenScript as .zip files. See [Chapter 5, "Creating Scenario Profiles"](#) for additional information.

14.3.6 Data Bank Options

The Data Bank options let you import and export Data Bank files (.csv) to and from the Oracle Load Testing server and the local machine. See [Section 13.2, "Importing and Exporting Data Bank Files"](#) for additional information.

14.3.7 Data File Options

The Data File options let you import and export Download Manager data files (.dat) to and from the Oracle Load Testing server and the local machine. See [Section 13.3, "Importing and Exporting Data Files"](#) for additional information.

14.4 Setting Profiles Options

You can set Oracle Load Testing Profile options for custom browsers, scenario default properties, and ramp options in the Profiles group of the user options.

14.4.1 Setting Browser Emulation Options

The Browser Emulation options let you add, edit, and delete custom browser emulator options. The following custom browser options are available:

New - displays a new line in the table.

Delete - deletes the selected browser.

Browser - any name for the customized browser emulator. This name will appear in the Browser Emulation list in the Edit Scenario Details dialog box.

Agent String - specifies the string to send to the server as the User Agent header string for the customized browser emulator.

14.4.2 Setting Scenario Default Properties

You can change the default settings for profiles using the Default Properties in the Profiles section of the Options dialog box. Changes made are applied to profiles as they are added to the scenario. Note that changes are not applied to profiles that are already in the scenario. To apply changes to profiles already in the scenario, remove them from the scenario on the Composer tab, then add them back.

The Default Properties has two columns:

Default Value - shows the value to which the option is set when a new script is added to a scenario.

Table View - when checked, this field is displayed on the Composer tab.

General - the main settings are as follows:

- **Max. # VUs** - specifies the number of virtual users to run for the selected profile. For each virtual user, Oracle Load Testing runs a separate instance of the script(s) specified in the virtual user profile.
- **Agent System** - specifies the machine on which the virtual users will run. When running virtual users across systems on a LAN/WAN, enter the machine name of a system running either Oracle Load Testing or Oracle Load Testing Agent. Systems are defined using the Systems Manager. Initially, you must define the machine names or IP addresses of the system(s) in the Systems Manager. Once the name(s) or IP addresses have been specified, you can select the system name from the drop-down list for future load tests.

When determining the number of virtual users to run per process or system, you need to include the Client overhead in the resource allocation. Each VU in Thin or Java Client requires approximately 350 KB-500 KB of memory to run. When calculating the available memory to run VUs on an agent system, you must account for a 20-30% client system overhead. Therefore, you only have 70-80% of the physical memory (RAM) available to run VUs.

- **Iteration Delay** - specifies the amount of time (in seconds) to wait between iterations of virtual user runs. You specify the number of iterations using the Profile Properties.
- **VU Pacing (Think Time)** - specifies the script playback delay for each virtual user. There are four options:
 - **Recorded** - uses the delay times that were recorded in the Oracle OpenScript script. You can set minimum and maximum delay times (in seconds) that override the script delay times in the Minimum and Maximum edit boxes.
 - **Recorded/Random** - uses random delay times based upon the recorded user delay. Oracle Load Testing sets the low end of the random range as the actual user delay minus the Lower percentage setting. Oracle Load Testing sets the high end of the random range as the actual user delay plus the Upper percentage setting. For example, if the actual recorded delay time was 100 seconds and the Lower and Upper settings are 10% and 25% respectively, Oracle Load Testing uses random delay times between 90 and 125 seconds.
 - **Random** - uses random times for Virtual User pacing. You can set minimum and maximum delay times for random delay in the Minimum and Maximum edit boxes.
 - **No Delay** - plays back the scripts at the fastest possible speed.

Note: For OpenScript scripts, the VU Pacing overrides the times specified in `think()` and `beginStep()` methods.

Data Bank - the Data Bank load testing playback settings are as follows:

- **Use Data Bank** - when true, scripts that have Oracle OpenScript Data Banks will use the Data Banks as part of the virtual user playback. When false, scripts playback using the recorded data rather than the Data Bank.
- **Setup Timeout:** Specifies how much time to spend preparing a databank for use before timing out. The value is in seconds. This setting includes the total time to do all of the following activities:

If using a Database-backed databank:

- Connect to the database
- Query
- Read records, write into the file
- Create the index simultaneously
- Disconnect

If using a CSV-backed databank:

- Time required to parse the CSV file and create the index

If using Random Unique:

- Time to shuffle the index

- **Read Timeout** - specifies the amount of time to wait for a databank read or get operation for a script at run-time before timing out.

See [Section 5.1.5, "Using the Data Bank Control"](#) for additional information.

Sync Points Settings- the Sync Point settings are as follows:

- **Name** - specifies the name of the Synchronization point. This is the same name as specified for the sync point in the OpenScript script.
- **Group type** - specifies how to determine when to release virtual users based on the following options:
 - **All users in scenario** - calculates when to release virtual users based on all virtual users whether or not they are running.
 - **Running users** - calculates when to release virtual users based on the number of running users only.
- **Release trigger** - specifies the percentage of virtual users that must be waiting before they are released. For example, if ten virtual users are running and the release trigger is set to fifty percent, then the waiting virtual users are released when five of them are waiting. Setting the value greater than one hundred percent causes the virtual users to wait until they are manually released.
- **Open gate time** - specifies the amount of time to keep the release open after it has been triggered.
- **Maximum Wait** - specifies, per individual virtual user, the maximum amount of time, in seconds, that the synchronization point will wait before releasing the virtual user.

Browser Settings - the browser settings are as follows:

- **Browser Emulation** - specifies the type of browser to emulate. Default is the browser used to record the script.
- **Browser Type** - specifies the type of browser to use for functional test scripts: Internet Explorer or Firefox. The default is Internet Explorer on Windows. The default is Firefox on Linux.
- **Browser Path Override** - specifies an alternative path to use when launching the specified browser type. Explorer and Firefox browser processes physically exist in the file system. In case the path to one of these browsers is incorrect, specify an alternative path to use when launching the specified browser type. This setting is not intended to be used to specify the path to an unsupported browser.
- **Browser Additional Arguments** - specifies any additional startup arguments that should be used when launching the browser process on playback. The default is no additional arguments other than what may be required internally.
- **Connection Speed Emulation** - specifies the line speed to simulate for the virtual user's Internet connection. Set the speed to a specific number if you want the virtual user to simulate a dial-up connection using a modem, DSL, or other speed. Set the speed to True Line Speed if you want the virtual user to run using the actual connection speed.
- **Resolution Size** - specifies the screen resolution of the playback agent machine. Functional test scripts with mouse-click actions and screenshot capturing are dependent upon this setting.
- **Dismiss Alert Dialogs** - specifies if JavaScript alert dialog boxes are automatically dismissed if they appear during playback. The default is false; do not automatically dismiss alert dialogs.
- **Cache Download Pages** - when true, downloaded pages are stored in a local cache and caching options are enabled. Caching places less of a load on the server as only newer pages are requested and brought down from the Web server. When false, caching is not used. No caching places more of a load on the Web server

because pages and images are brought down from the Web server for every request.

- **Clear Cache After Iteration** - when true, the cache is cleared after each iteration of the script.
- **When Page Out of Date** - when true, the cache is cleared automatically when the page is out of date.
- **Every Visit** - when true, the cache is cleared on every visit to the page.
- **Max. Cache Size (MB)** - specifies the maximum cache size.
- **Use IP Spoofing** - when true, Oracle Load Testing uses different IP addresses for Virtual User agents. Each virtual user must get a defined IP address. You must define the IP addresses available for use by Oracle Load Testing Agents in the TCP/IP network protocols of the system. All IP addresses must be added to each Agent system. See [Section 5.1.7, "Using IP Spoofing"](#) for additional information.
- **Enable Cookies** - when true, the virtual user profiles will use cookies. Use this setting if your Web application uses cookies to manage session and other context information.

Extensibility - the extensibility settings are as follows:

- **Execute User Defined Tests** - when true, Oracle Load Testing runs Oracle OpenScript Text Matching and Server Response tests.

Virtual User Logs - the VU Logs settings are as follows:

- **Enable Logging** - turns VU logging on and off. The default is on. When On, the **Message Delivery** and **Logged Messages** settings are also enabled.
- **Message Delivery** - specifies when messages are delivered to the Virtual User log, as follows:

On Error - enables delivery of messages only when an error occurs. Using this a user can debug what happened on a particular step or transaction when an error occurred. All messages, as specified by the Logged Messages settings, for steps or transactions are cached an error occurs.

Always - all messages generated by Virtual Users will be logged.

- **Logged Messages** - specifies the type of logged messages, as follows:

Standard - The standard messages consist of basic level messages which provide an overview of the chronological flow of a Virtual User. The types of messages included in this are as follows:

- **BeginPage** - Logs the step-group (page) name, when VU starts a page.
- **FoundResource** - Logs the resources' urls when download manager is turned on and discovers resources from pages.
- **ScriptError [Without stack trace]** - Logs the script exception type and messages, when an OATS defined exception happens. It does not matter if the 'Error Recovery' settings handles it as 'warn' or 'ignore'. The name of the exception class is appended to "ScriptError" as a whole message type, for example, ScriptError<SolveException>
- **CachedData** - Logs the cached resources' urls, when a VU requests on a cached resource (304 NOT MODIFIED or Found In Cache).
- **ThinkTime** - Logs a message with the think time in seconds when a VU is in iteration delay, step delay, or manual delay.

- SyncPoint - Logs whether a VU is suspended by a Sync Point or continues from a Sync Point.
- Action - Logs the details of an action when a VU is navigating to a page (http), or executing a sql statement (util).

Extended - The extended messages consist of all the message types included in Standard plus selective inclusion of extended message types, which can have a substantial overhead. Selecting this option enables the selection of the previously excluded message types. All these message types or their groups are turned off by default. The extended message types or their groups are as follows:

- Server Communication Content - Enables logging of all contents that are communicated with the server. For example, for an HTTP script it will consist of RequestHeader, ResponseHeader and ResponseContent.
 - Parameter Substitution - Enables logging of the variables name/value being substituted when parameters are transformed (messages of type ParameterSubstitution).
 - Error Stack Trace - Enables logging of messages of type ScriptError to be reported with the stack trace in the content.
 - Verification Notifications - Enables logging of the test type, test name, and test result of all types of verifications/tests (messages of type Verification).
 - User Defined Messages - Enables logging of the messages if API 'info()', 'warn()', 'fail()', 'reportFailure()' methods are used (messages of type CustomizedLog).
- **Table Test** - turns Table Test logging on and off. The default is off.
 - **Object Test** - turns Object Test logging on and off. The default is off.
 - **Screenshot** - turns Screenshot logging on and off. The default is off.
 - **XML Test** - turns XML Test logging on and off. The default is off.

Reporting - the Reporting settings are as follows:

- **Generate Timers For Step Groups** - when true, Oracle Load Testing automatically adds timers for each OpenScript Step Group for reporting. The timers are used in Oracle Load Testing to provide performance monitoring and timing information for each Step Group the script(s) played back by a scenario.
- **Generate Timers For Pages** - when true, Oracle Load Testing automatically adds timers for each OpenScript script page for reporting. The timers are used in Oracle Load Testing to provide performance monitoring and timing information for each page of the script(s) played back by a scenario.
- **Generate Timers For Resources** - when true, Oracle Load Testing automatically adds timers for all resources for monitoring and reporting purposes. Resources include images and other objects downloaded from the server as specified by the **OpenScript Download Manager** section of the Scenario Defaults.

Error Handling - the Error Handling settings are as follows:

- **Fail on Object Download Errors** - when true, a Web page object download error is considered a fatal error that ends the current iteration.
- **Fail on Missing Frames** - when true, a Web page object download error is considered a fatal error that ends the current iteration.
- **On Error Stop Virtual User** - when true, all virtual users are stopped if an error is encountered.

- **Stop Remaining Iterations on Failure** - when true, all remaining iterations for a virtual user are stopped if an error is encountered.
- **Socket Timeout** - specifies the maximum amount of time a virtual user waits for a socket connection before timing out.
- **Request Timeout** - specifies the maximum amount of time a virtual user waits to access a page before timing out.
- **Connection Idle Timeout** - specifies the socket 'idle timeout' and uses a new connection when reusing an idle-timeout socket. This is used to specify the timeout for a socket that gets closed by the server side after a long idle period.

Advanced - the Advanced settings are as follows:

- **Max Users Per Process** - sets the maximum number of virtual users per single agent process. When running virtual users as threads in a single process, **Max Users Per Process** sets the maximum number of virtual user threads in a single process. Oracle Load Testing spawns new processes if the number of virtual users exceeds the maximum number in any single process and runs the additional virtual users as threads in the new process.

The default setting is unlimited virtual users per agent process.

- **Object Timeout** - specifies the maximum time to wait for an object to download before timing out.
- **Max HTTP Connections Per User** - specifies the maximum number of server connections per process per server. Each VU makes multiple connections to request additional resources for images and additional frames for example. Setting this option specifies a limit on the total number of connections that the VU s can make to the server. The default setting is "Default," which means use the default connection limits as configured on the agent machine. (See Microsoft KBbase article Q183110 for more information.)
- **Ignore HTTP Proxy Settings** - specifies whether to ignore the agent machine's default proxy setting as defined in Internet Explorer.
- **Debug Users** - specifies whether to enable debug features.

Agent Client Preferences - When a setting is set to the default value, this means that the value that will be used is what is set in the OracleATS\OFT\jagent\JavaAgent.properties file, unless a value is not set in the JavaAgent.properties file. In this case, the Java Agent uses the internal default value.

- **Persist Raw Data** - when true, Oracle Load Testing saves every single measured data point in to the database.
See [Section 11.10, "Using Raw Data"](#) for additional information about how to use the raw data.
- **Report Counters** - when true, Oracle Load Testing counters are reported.
- **Report Sender Interval** - when you select other, enter the time in milliseconds for how frequently the agent reports its status and accrued counters. The default in the JavaAgent.properties file is 5000.
- **Maximum JVM Heap Size (MB)** - specifies the maximum size of the JVM heap. The default is 256MB. This value cannot be more than 90% of the total memory size.
- **Proxy Host** - select other to enter the proxy host and override the system-specified proxy host.

- **Proxy Port** - select other to enter the proxy port and override the system-specified proxy port.
- **Non Proxy Hosts** - select other to enter non-proxy hosts. Delimit multiple hosts with a bar (|).
- **Enable GZIP** - when true, support for gzip compression is enabled. The browser Request includes the Accept-Encoding: gzip header indicating a gzip compressed page response will be accepted. If the server uses gzip compression, the response includes the Content-Encoding: gzip header indicating the returned page is in gzip compressed format. The browser unzips the compressed file before rendering the HTML page. Gzip compression is typically used to provide faster transfer of large HTML pages between the browser and the server.
- **Enable Deflate** - when true, support for deflate compression is enabled. The browser Request includes the Accept-Encoding: deflate header indicating a deflate compressed page response will be accepted. If the server uses deflate compression, the response includes the Content-Encoding: deflate header indicating the returned page is in deflate compressed format. The browser inflates the compressed file before rendering the HTML page. Deflate compression is typically used to provide faster transfer of large HTML pages between the browser and the server.
- **Language** - specifies which language to use for script playback. When you select **Other**, enter the language to override the Accept-Language header. The default is the locale assigned by the JVM.
- **HTTP Version** - select the HTTP protocol version to specify in the GET or POST request/response between client and server. The HTTP/1.0 protocol is an early implementation of the Hypertext Transfer Protocol. HTTP/1.1 is a standards-based enhancement to the HTTP/1.0 protocol. See the Key Differences between HTTP/1.0 and HTTP/1.1 at <http://www8.org/w8-papers/5c-protocols/key/key.html>
- **Accept String** - this setting specifies what the Accept: HTTP header value looks like. When you select other, enter the string. The default in the JavaAgent.properties file is: text/html, image/gif, image/jpeg, */*. If you modify a navigation in a script by adding a custom Accept: header, the custom header value from the script is used instead.
- **Enable Keep Alive** - when true, the Connection: Keep-Alive header is set to indicate requests should use a persistent connection. The "Keep-Alive" keyword indicates that the request should keep the connection open for multiple requests. For HTTP/1.0, the socket connection is kept open until either the client or the server drops the connection. For HTTP/1.1 all connections are kept alive unless a Connection: close header is specified.
- **Preserve Connections Between Iterations** - used to preserve connections between Virtual User agents and the browser between successive iterations of the script. Set to True if the browser should attempt to reuse any open browser connections if possible between iterations. Each virtual user maintains its own set of connections that it never shares with other virtual users. The default value is True, preserve connections between iterations.
- **Preserve Variables Between Iterations** - used to preserve or automatically clear variables added in the **Run** section of OpenScript scripts between successive iterations of the **Run** section.

- **Preserve Cookies Between Iterations** - used to preserve or automatically clear cookies added in the **Run** section of OpenScript scripts between successive iterations of the **Run** section.
- **Max Keep-Alive Requests** - select other to specify the maximum number of requests to make on a keep alive connection before closing it.
- **Download Local Files** - when true, the Java Agent retrieves the requested local file contents.
- **Max Content Download Size** - specifies the maximum size for downloads. You can specify Unlimited or Other. If you select Other, specify the maximum size in kilobytes.
- **Secure Protocol** - select the Secure Socket Layer version to use for the proxy server. When recording a secure site in the browser, the user only sees the Proxy Recorder's certificate not the secure web site's certificate. The Browser, Proxy Recorder, and Secure Server each have their own private and public keys which are used to encrypt/decrypt data.
 - **SSL**: Use Secure Socket Layer protocol with the proxy server. OpenScript supports SSLv1.0, SSLv2.0, SSLv3.0 and TLSv1.0.
 - **TLSv1.2**: Use Secure Socket Layer Transport Layer Security version 1.2.
- **Do Not Request URLs ending in** - ignores requests in the script that end with the extensions specified in this field. Specify the file extensions, separated by commas, that should not be requested. For example, `.png, .ico, .gif, .jpg, .jpeg, .css, .js`. This setting only applies to only to OpenScript load test scripts.
- **Ignored URLs that Match Regex** - specify the Regular Expression that specifies URLs that should not be requested.
- **Global Headers** - specifies any custom "Global Headers" string to use in the Request header for script playback. The format is in the form:
`name1:value1;name2:value2;name3:value3`. For example:
`x-oracle-slm-message-id: bcn=<beacon_name>; svc=<service_name>`.
- **Replace URLs** - specifies the URL replacement string in the form:
`originalURL1=replacementURL1,originalURL2=replacementURL2,[...]`. During playback, anytime the agent makes a request to a URL starting with a segment, `originalURL`, the agent replaces the original URL segment with `replacementURL`. This feature is only supported for Load Test scripts.
 - `originalURL` - Specify the starting segment of the `URL:port` that appears in the script that should be replaced. This value is case-sensitive.
 - `replacementURL` - Specify the new starting segment `URL:port` that the agent requests instead of `originalURL`.

For both parameters, if the protocol is omitted, HTTP protocol is assumed. If no port is specified after the host, port 80 is assumed for HTTP protocol, and port 443 is assumed for HTTPS protocol. URLs are replaced after all correlations are applied. One or more URL replacement pairs may be specified, separating each replacement pair with a comma. The following examples show the format of **Replace URLs** strings:

```
test_server:7789=production_server:7789
```

```
test:7789=prod:7789,https://stage.oracle.com/main=https://prod.oracle.com/home
```

- **Additional Arguments** - specifies custom OpenScript script.java code arguments. You can create your own settings in OpenScript scripts. For example, you can create custom settings in OpenScript script.java code, as follows:

```
if (getSettings().get("MyCustomSetting").equals("abc")) {
    info("We're running in ABC mode.");
}
```

You can then set the additional arguments in the **Additional Arguments** field as follows:

```
-MyCustomSetting abc
```

Error Recovery - General - the General Error Recovery settings are as follows:

- **File Not Found** - specifies the error recovery action if a file is not found.
- **Segment Parser Error** - specifies the error recovery action if the XPath Segment Parser cannot verify the correctness of an XPath.
- **Create Variable Fail** - specifies the error recovery action if a script fails to create a variable.
- **Encryption Service not Initialized** - specifies the error recovery action when the password encryption service was not initialized.
- **Binary Decoding Error** - specifies the error recovery action if a binary post data parameter error occurs.
- **Variable Not Found** - specifies the error recovery action if a variable cannot be found when parsing transformed strings.
- **Unexpected Script Error** - specifies the error recovery action if any unexpected script error occurs.
- **Child Script Failed** - specifies the error recovery action if an error occurs in a script that is a child of another script.
- **Call Function Failed** - specifies the error recovery action if an error occurs in a script that calls a function of another script.
- **Encrypting/Decrypting Failed** - specifies the error recovery action if an error occurs encrypting or decrypting a script.

Error Recovery - HTTP - the HTTP Module Error Recovery settings are as follows:

- **HTML Parsing Error** - specifies the error recovery action if an HTML parsing error occurs.
- **Text Match Fail** - specifies the error recovery action if a text matching test fails.
- **Solve Variable Fail** - specifies the error recovery action if the value of any variable cannot be solved.
- **Response Time Error** - specifies the error recovery action if a Server Response Time test fails.
- **Invalid HTTP Response** - specifies the error recovery action if the sever returns an invalid HTTP response.
- **Invalid URL** - specifies the error recovery action if the server returns an Invalid URL response code.
- **Zero Downloads Fatal** - specifies the error recovery action if a server response indicates zero bytes length.

- **Client Certificate Keystore Error** - specifies the error recovery action if the Client Certificate Keystore indicates an error.

Error Recovery - Oracle Forms Load - the Oracle Forms Load Test Module Error Recovery settings are as follows:

- **Forms Connect Error** - specifies the error recovery action if a server connection error occurs.
- **Forms I/O Communication Error** - specifies the error recovery action if a read/write or communication error occurs with an Oracle Forms message.
- **Forms Playback Error** - specifies the error recovery action if an error occurs during forms playback.
- **Forms Component not Found** - specifies the error recovery action if a component of a form is not found.
- **Forms Content Match Failed** - specifies the error recovery action if a content matching test fails.

Error Recovery - Functional Test - the Functional Test Module Error Recovery settings are as follows:

- **Text Matching Failed** - specifies the error recovery action if a text matching test error occurs.
- **Object Test Failed** - specifies the error recovery action if an Object test error occurs.
- **Table Test Failed** - specifies the error recovery action if a Table test error occurs.
- **XML Test Failed** - specifies the error recovery action if an XML test error occurs.

Error Recovery - Web Functional Test - the Web Functional Test Module Error Recovery settings are as follows:

- **Response Time Error** - specifies the error recovery action if a Response time error occurs.
- **Solve Variable Failed** - specifies the error recovery action if a Solve Variable error occurs.
- **Wait for Page Timeout** - specifies the error recovery action if a Wait for Page Timeout occurs.
- **Object Not Found** - specifies the error recovery action if an Object Not Found error occurs.
- **Playback Failed** - specifies the error recovery action if a Playback Failed error occurs.
- **Title Test Failed** - specifies the error recovery action if a Title Test Failed error occurs.
- **HTML Test Failed** - specifies the error recovery action if a Title Test Failed error occurs.

Error Recovery - Oracle Forms Functional Test - the Oracle Forms Functional Test Module Error Recovery settings are as follows:

- **Oracle Forms Error** - specifies the error recovery action if an Oracle Forms error occurs.
- **Status Bar Test Error** - specifies the error recovery action if a Status Bar Test error occurs.

Download Manager - the OpenScript Download Manager settings are as follows:

- **Use OpenScript Download Manager** - when true, the Download Manager is enabled during playback. When false, the Download Manager is not enabled during playback.
- **CSS Resource** - when true, css resources in <Link> tags are downloaded during playback. When false, css resources are not downloaded during playback.
- **Image Resource** - when true, image resources in tags, in the "background" attribute of a tag, or in <style> tags with "background:url" patterns are downloaded during playback. When false, image resources are not downloaded during playback.
- **Embedded Object Resource** - when true, object resources in <Embed> tags or in <Object> tags are downloaded during playback. When false, object resources are not downloaded during playback.
- **Script Resource** - when true, script resources in <Script> tags are downloaded during playback. When false, script resources are not downloaded during playback.
- **Applet Resource** - when true, applet resources in <Applet> tags are downloaded during playback. When false, applet resources are not downloaded during playback.

Forms LT Playback - the Oracle EBS/Forms load testing playback settings are as follows:

- **Capture Message Details** - Specifies if forms message details are captured during playback. When selected, OpenScript captures and stores Forms message requests, responses, and information about all loaded Forms components during playback. This information is useful to have when debugging the script.

OpenScript displays captured details in the "Messages" and "Object Details" tabs of the Details view. Oracle Load Testing displays this information in the virtual user logs based on the "virtual user logs" settings.

Capturing message details is a memory-intensive operation. During heavy load testing, it is recommended to clear this setting to reduce the amount of heap space required by the agent.

- **Heart Beat Interval (sec)** - Specifies how often to notify the forms server that the forms client is still alive when there is no user activity in the forms client. This value is used to override the timeout configured for the EBS Application that indicates how long the client has no activities. The default "0" value means no heart beat is sent to the server.

Cache and Cookies - the Cache and Cookies settings are as follows:

- **Clear Cache Before Playback** - specifies if the browser cache is cleared before script playback begins.
- **Clear Cache Between Iterations** - specifies if the browser cache is cleared between each playback iteration.
- **Clear Session Cookies Before Playback** - specifies if the session cookies are cleared before script playback begins.
- **Clear Session Cookies Between Iterations** - specifies if the session cookies are cleared between each playback iteration.
- **Clear Persistent Cookies Before Playback** - specifies if the persistent cookies are cleared before script playback begins.

- **Clear Persistent Cookies Between Iterations** - specifies if the persistent cookies are cleared between each playback iteration.

Event Timeout - the Oracle Forms Event timeout settings are as follows:

- **Forms Startup Timeout** - specifies the maximum number of seconds OpenScript should wait for a form to appear before considering the form not found. This is the default timeout when waiting for a form to appear before invoking an action against it. This is also the default timeout when waiting for a form to appear before continuing the script.
- **Forms Action Timeout** - specifies the maximum number of seconds OpenScript should wait for forms action playback until success.
- **Forms Response Timeout** - specifies the maximum number of seconds OpenScript should wait for forms response before timing out.

14.4.3 Setting Ramp Options

The ramp options let you specify the default ramp rate and stop conditions that will be used when creating scenarios. See [Chapter 6, "Defining VU Rampup"](#) for a description of these fields.

14.5 Setting ServerStats Options

The ServerStats options let you edit ServerStats configurations, metrics and metric profiles. See [Chapter 8, "Defining ServerStats Monitors"](#) for a description of these fields.

14.6 Setting Session Options

The Session options let you rename sessions, delete sessions, purge VU logs, and import/export session data.

14.6.1 Setting Session Start and Stop Options

Sessions specify the scope for Oracle Load Testing data collection and reporting. The data collected while the Autopilot is running virtual users is shown in the virtual user grid, runtime performance statistics and load graphs, and saved to a database for post-testing analysis in the Reports tab.

You can specify default settings for how sessions start and end data collection by selecting **Options** from the **User** menu then selecting **Session Start/Stop** in the **Session** group. See [Section 9.3.4, "Managing Sessions"](#) for a description of these options.

14.6.2 Setting Reporting Options

The Reporting options let you specify the parameters for data collections and refresh intervals, status and log grid row counts, and reporting default settings. See [Section 11.5, "Setting the Reporting Options"](#) for a description of these fields.

Internet Status Codes

This appendix lists the Internet status codes that can be returned in a response by a server.

A.1 Server Status Codes

The following are the numeric codes that may be returned as Internet Status and error codes in the Oracle Load Testing Autopilot and the Oracle OpenScript Resource Validation test.

A.1.1 Information Codes

100 Continue

The client can continue the request to the server.

101 Switching Protocols

The server has changed the application protocol being used on this connection at the request of the client via the Upgrade message header field.

A.1.2 Success Codes

200 OK

The request completed successfully.

201 Created

The request has been fulfilled and resulted in a new resource being created.

202 Accepted

The request has been accepted for processing, but the processing has not been completed.

203 Non-Authoritative Information

The returned meta information in the entity-header is not the definitive set as available from the origin server, but is gathered from a local or a third-party copy.

204 No Content

The server has fulfilled the request but does not need to return new information.

205 Reset Content

The server has fulfilled the request and the client should reset the document view which caused the request to be sent to allow the user to initiate another input action.

206 Partial Content

The server has fulfilled the partial GET request for the resource.

A.1.3 Redirection Codes

300 Multiple Choices

The requested resource corresponds to ambiguous or multiple choices.

301 Moved Permanently

The requested resource has been assigned a new permanent Uniform Resource Identifier (URI) and any future references to the resource should use one of the returned URIs.

302 Moved Temporarily

The requested resource resides temporarily under a different Uniform Resource Identifier (URI).

303 See Other

The response to the request can be found under a different Uniform Resource Identifier (URI) and should be retrieved using a GET method on that resource.

304 Not Modified

The client requested a resource that has not been modified.

305 Use Proxy

The requested resource must be accessed through the proxy given by the Location field.

306 Not Used

Used in a previous version of the HTTP specification but is no longer used and the code is reserved.

307 Temporary Redirect

The requested resource resides temporarily under a different Uniform Resource Identifier (URI).

A.1.4 Failure Codes

400 Bad Request

The request could not be understood by the server due to invalid syntax.

401 Unauthorized

The requested resource requires user authentication.

402 Payment Required

Not implemented. Reserved for future use.

403 Forbidden

The server understood the request, but is refusing to fulfill it.

404 Not Found

The server has not found anything matching the Request-URI (Uniform Resource Identifier).

405 Method Not Allowed

The method in the Request-Line is not allowed for the resource identified by the Request-URI.

406 Not Acceptable

The response from the request has content characteristics that are not acceptable to the accept headers sent by the client.

407 Proxy Authentication Required

The client must first authenticate itself with the proxy.

408 Request Time-out

The client did not produce a request within server timeout limit.

409 Conflict

The request could not be completed due to a conflict with the current state of the resource. The user should resolve the conflict and resubmit the request.

410 Gone

The requested resource is no longer available at the server and no forwarding address is known.

411 Length Required

The server refuses to accept the request without a defined Content-Length.

412 Precondition Failed

The precondition given in one or more of the request-header fields evaluated False when tested on the server.

413 Request Entity Too Large

The server is refusing to process a request because the request entity is larger than the server is willing or able to process.

414 Request-URI Too Long

The server is refusing to service the request because the Request-URI is longer than the server is willing to interpret.

415 Unsupported Media Type

The server is refusing to service the request because the entity of the request is in a format not supported by the requested resource for the requested method.

416 Requested Range Not Satisfiable

A request included a Range request-header field, and none of the range-specifier values in this field overlap the current extent of the selected resource, and the request did not include an If-Range request-header field.

417 Expectation Failed

The expectation given in an Expect request-header field could not be met by this server, or, if the server is a proxy, the server has unambiguous evidence that the request could not be met by the next-hop server.

A.1.5 Server Error Codes

500 Internal Server Error

The server encountered an unexpected condition that prevented it from fulfilling the request.

501 Not Implemented

The server does not support the functionality required to fulfill the request.

502 Bad Gateway

The server, while acting as a gateway or proxy, received an invalid response from the upstream server it accessed in attempting to fulfill the request.

503 Service Unavailable

The server is currently unable to handle the request due to a temporary overloading or maintenance of the server.

504 Gateway Time-out

The server, while acting as a gateway or proxy, did not receive a time out while waiting for a response from the upstream server specified by the URI (e.g. HTTP, FTP, LDAP) or some other auxiliary server (e.g. DNS) it needed to access in attempting to complete the request.

505 HTTP Version not supported

The server does not support, or refuses to support, the HTTP protocol version that was used in the request message.

A.2 Internet Codes

This section lists the Internet, FTP, Gopher, and HTTP API error codes.

A.2.1 Internet Error Codes

12001 Out of Handles

No more Internet handles could be generated at this time.

12002 Timeout

The request has timed out.

12003 Extended Error

An extended error was returned from the server. This is typically a string or buffer containing a verbose error message.

12004 Internal Error

An internal error has occurred.

12005 Invalid URL

The Uniform Resource Locator (URL) is invalid.

12006 Unrecognized Scheme

The Uniform Resource Locator (URL) scheme could not be recognized or is not supported.

12007 Name Not Resolved

The server name could not be resolved.

12008 Protocol Not Found

The requested protocol could not be located.

12009 Invalid Option

A request to `InternetQueryOption` or `InternetSetOption` specified an invalid option value.

12010 Bad Option Length

The length of an option supplied to `InternetQueryOption` or `InternetSetOption` is incorrect for the type of option specified.

12011 Option Not Settable

The request option cannot be set, only queried.

12012 Shutdown

The Win32 Internet function support is being shut down or unloaded.

12013 Incorrect User Name

The request to connect and log on to an FTP server could not be completed because the supplied user name is incorrect.

12014 Incorrect Password

The request to connect and log on to an FTP server could not be completed because the supplied password is incorrect.

12015 Login Failure

The request to connect to and log on to an FTP server failed.

12016 Invalid Operation

The requested operation is invalid.

12017 Operation Canceled

The operation was canceled, usually because the handle on which the request was operating was closed before the operation completed.

12018 Incorrect Handle Type

The type of handle supplied is incorrect for this operation.

12019 Incorrect Handle State

The requested operation cannot be carried out because the handle supplied is not in the correct state.

12020 Not Proxy Request

The request cannot be made via a proxy.

12021 Registry Value Not Found

A required registry value could not be located.

12022 Bad Registry Parameter

A required registry value was located but is an incorrect type or has an invalid value.

12023 No Direct Access

Direct network access cannot be made at this time.

12024 No Context

An asynchronous request could not be made because a zero context value was supplied.

12025 No Callback

An asynchronous request could not be made because a callback function has not been set.

12026 Request Pending

The required operation could not be completed because one or more requests are pending.

12027 Incorrect Format

The format of the request is invalid.

12028 Item Not Found

The requested item could not be located.

12029 Cannot Connect

The attempt to connect to the server failed.

12030 Connection Aborted

The connection with the server has been terminated.

12031 Connection Reset

The connection with the server has been reset.

12032 Force Retry

Calls for the Win32 Internet function to redo the request.

12033 Invalid Proxy Request

The request to the proxy was invalid.

12034 Need UI

A user interface or other blocking operation has been requested.

12035 Not Defined

This error code is not defined in WININET.

12036 Handle Exists

The request failed because the handle already exists.

12037 Sec Cert Date Invalid

SSL certificate date that was received from the server is bad. The certificate is expired.

12038 Sec Cert CN Invalid

SSL certificate common name (host name field) is incorrect.

12039 HTTP to HTTPS on Redir

The application is moving from a non-SSL (Secure Socket Layer) to an SSL connection because of a redirect.

12040 HTTPS to HTTP on Redir

The application is moving from an SSL (Secure Socket Layer) to a non-SSL connection because of a redirect.

12041 Mixed Security

Indicates that the content is not entirely secure. Some of the content being viewed may have come from unsecured servers.

12042 Chg Post is Non Secure

The application is posting and attempting to change multiple lines of text on a server that is not secure.

12043 Post is Non Secure

The application is posting data to a server that is not secure.

12044 Client Auth Cert Needed

An SSL protected resource on a Web server requires a valid client certificate.

12045 Invalid CA

A server SSL certificate was issued by an unknown or invalid certificate authority.

12046 Client Auth Not Setup

Client authorization is not setup up on this computer.

12047 Async Thread Failed

The application could not start an asynchronous thread.

12048 Redirect Scheme Change

The function could not handle the redirection, because the scheme changed (for example, HTTP to FTP).

12049 Dialog Pending

Another thread has a password dialog box in progress.

12050 Retry Dialog

The dialog box should be retried.

12051 Not Defined

This error code is not defined in WININET.

12052 HTTPS to HTTP Submit Redir

The data being submitted to an SSL connection is being redirected to a non-SSL connection.

12053 Insert CDROM

The request requires a CD-ROM to be inserted in the CD ROM drive to locate the resource requested.

A.2.2 FTP API Error Codes

The following are the error codes returned by the Internet API used to access FTP protocols.

12110 Transfer in Progress

The requested operation cannot be made on the FTP session handle because an operation is already in progress.

12111 FTP Dropped

The FTP operation was not completed because the session was aborted.

12112 No Passive Mode

Passive mode is not available on the server.

A.2.3 Gopher API Error Codes

12130 Protocol Error

An error was detected while parsing data returned from the Gopher server.

12131 Not File

The request must be made for a file locator.

12132 Data Error

An error was detected while receiving data from the Gopher server.

12133 End of Data

The end of the data has been reached.

12134 Invalid Locator

The supplied locator is not valid.

12135 Incorrect Locator Type

The type of the locator is not correct for this operation.

12136 Not Gopher Plus

The requested operation can only be made against a Gopher+ server or with a locator that specifies a Gopher+ operation.

12137 Attribute Not Found

The requested attribute could not be located.

12138 Unknown Locator

The locator type is unknown.

A.2.4 HTTP API Error Codes

12150 Header Not Found

The requested header could not be located.

12151 Downlevel Server

The server did not return any headers.

12152 Invalid Server Response

The server response could not be parsed.

12153 Invalid Header

The supplied header is invalid.

12154 Invalid Query Request

The request made for HTTP Query Info is invalid.

12155 Header Already Exists

The header could not be added because it already exists.

12156 Redirect Failed

The redirection failed because either the scheme changed (for example, HTTP to FTP) or all attempts made to redirect failed (default is five attempts).

12157 Security Channel Error

The application experienced an internal error loading the SSL libraries.

12158 Unable to Cache File

The function was unable to cache the file.

12159 TC/PIP not Installed

The required protocol stack is not loaded and the application cannot start WinSock.

12160 Not Redirected

The HTTP request was not redirected.

12161 Cookie Needs Confirmation

The HTTP cookie requires confirmation.

12162 Cookie Declined

The HTTP cookie was declined by the server.

12163 Disconnected

The Internet connection has been lost.

12164 Server Unreachable

The Web site or server indicated is unreachable.

12165 Proxy Server Unreachable

The designated proxy server cannot be reached.

12166 Bad Auto Proxy Script

There was an error in the automatic proxy configuration script.

12167 Unable to Download Script

The automatic proxy configuration script could not be downloaded.

12168 Redirect Needs Confirmation

The redirection requires user confirmation.

12169 Sec Invalid Cert

SSL certificate is invalid.

12170 Sec Cert Revoked

SSL certificate was revoked.

12171 Failed Due to Security Check

The function failed due to a security check.

A.2.5 Additional Information

For additional information about HTTP Status Codes and the Win32 Internet API Constants, see the following Web resources:

Section 10 of the Hypertext Transfer Protocol - HTTP/1.1 Request for Comments, 2616:

<http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>

Error Messages

This appendix lists the error messages that can be generated by Oracle Load Testing and possible remedies.

B.1 Viewing Error Codes

The error codes and messages appear either as message boxes or in the **Current Error** or **Previous Error** columns of the Oracle Load Testing Autopilot Virtual User grid.

Current Error and **Previous Error** show errors encountered by running VUs during a test. **Current Error** is for the current iteration. **Previous Error** is for the last iteration.

The following sections list the errors and meanings.

B.2 Error Messages

Failed to create server

Oracle Load Testing was unable to create an Agent process to run Virtual Users. Try the following:

1. Make sure that you have the Oracle Load Testing Agent software loaded on the Agent system(s) and that it is the same version as the Oracle Application Testing Suite software that is loaded on the Oracle Load Testing system. The systems you plan to use as agents must have either the Oracle Load Testing Agent software or the full Oracle Application Testing Suite installed to work as agents. Do not install both the Oracle Load Testing Agent software and the Oracle Application Testing Suite software on the same system. Otherwise, resource conflicts between the two programs may occur.
2. Make sure you can successfully Ping all of the Agent systems from the Oracle Load Testing system. The names you use to Ping the systems are the same names that you will specify for the Agent systems in the Oracle Load Testing system. If you cannot successfully Ping the Agent systems, contact your network administrator to resolve the issue. If you cannot Ping the Agent systems from the Oracle Load Testing system, you will not be able to run the agents from the Oracle Load Testing system.
3. Make sure that the same user is logged on both the Oracle Load Testing system and all of the Agent systems. All of the Agent systems must be logged in to be controlled by the Oracle Load Testing system. You may be able to log in as a different user on the Agent systems as long as the user login has the same administrative privileges as the user logged in on the Oracle Load Testing system.

4. If the Oracle Load Testing and Agent systems are not participating in the Domain security model, both the Oracle Load Testing and the Agent systems must log in as Administrator and have the exact same password.
5. From the Oracle Load Testing system, try mapping a drive on each of the Agent systems using Windows Explorer. Depending on how your network is setup, the Oracle Load Testing systems may not be allowed to start up processes on the Agent systems. The easiest way around this is to map a drive to the Agent system in order to authenticate with Windows.
6. In the Oracle Load Testing system, add a script to the Configure Parameters of the Scenario list. Add the Agent system on which you want to run the script using the Systems Manager, then select the system from the System field on the Build Scenarios tab.

This error may also indicate a problem with Registry entries for Oracle Application Testing Suite components. Try uninstalling and reinstalling the Oracle Application Testing Suite. If reinstalling does not fix the problem, contact Customer Support.

Failed to find

Oracle Load Testing is unable to find the navigation between two pages recorded to the script. These errors can occur in Oracle Load Testing for a variety of reasons, even if the script played back correctly in Oracle OpenScript. Oracle OpenScript uses a thick client browser mode exclusively when recording and playing back scripts.

Typically, "failed to find" errors are related to dynamically generated HREFs recorded to a script that is played back in Oracle Load Testing. For example HREFs containing:

- session information
- inline JavaScript
- other dynamically changed attributes

may cause "failed to find" errors in Oracle Load Testing depending upon the specific techniques used for links in the Web page source and Oracle Load Testing current settings.

If you experience "failed to find" errors in Oracle Load Testing, try running the Scenario/Script with one Virtual User. If the Scenario/Script runs in Oracle Load Testing, it will be possible to run small load tests with that Scenario/Script.

To resolve "failed to find" errors, contact Customer Support for assistance with adjusting Oracle Load Testing settings for your specific situation. Note that the solution to one "failed to find" situation may not resolve other "failed to find" situations. You should contact Customer Support every time you encounter a "failed to find" error that you cannot resolve.

Failed to match...

Oracle Load Testing found the navigation path (an internal representation of a clicked link that Oracle OpenScript records to the script) between two pages recorded to the script, but was unable to match one or more partial components of the navigation path.

"failed to match" errors are also related to dynamically generated HREFs recorded to a script that is played back in Oracle Load Testing.

To resolve "failed to match" errors, contact Customer Support for assistance with adjusting Oracle Load Testing settings for your specific situation. Note that the solution to one "failed to match" situation may not resolve other "failed to match"

situations. You should contact Customer Support every time you encounter a "failed to match" error that you cannot resolve.

Failed to open scenario 'scenario name'

Oracle Load Testing failed to open the specified scenario file.

Failed to OpenKey 'registry key'

Oracle Load Testing failed to open the specified registry key.

Failed to Query 'registry key'

Oracle Load Testing failed to get the specified registry value string.

Failed to start callout server 'server prog id'

Oracle Load Testing is unable to start the External Callout server specified in the script.

License Error

You are using more VUs than you have licensed. Select About from the Help menu to view the number of licenses that you have.

B.3 RSWISERV Errors

Oracle Load Testing reports its own internal RSWISERV errors to the error columns of the Virtual User grid. The following is an example of an RSWISERV error:

```
DRIVER ERROR 2010, RSWISERV Error, http:// .....
```

Code	Error Constant
100	RSW_NO_RESULTS_TO_REPORT
101	RSW_CONNECT_BUCKET_TIMEOUT
200	RSW_DLL_UNINIT_COMPLETED
1000	RSW_ERROR_INVALID_PARAMETERS
2000	RSW_INET_LAYER_NOT_INITD
2001	RSW_OPEN_REQUEST_FAILED
2002	RSW_ADD_REQUEST_HEADERS_FAILED
2003	RSW_SEND_REQUEST_FAILED
2004	RSW_QUERY_INFO_FAILED
2005	RSW_FAILED_TO_CLOSE_HANDLE
2006	RSW_ERROR_NO_SESSION_HANDLE
2007	RSW_INTERNET_SET_OPTION_FAILED
2008	RSW_INTERNET_QUERY_OPTION_FAILED
2010	RSW_INTERNET_LOGIN_FAILURE
2011	RSW_AUTH_LOGIN_PARMS_NOTAVAILABLE
2012	RSW_CANNOT_CREATE_AUTH_OBJECT
2013	RSW_READ_REQUEST_FAILED
2014	RSW_INTERNET_REQUEST_TIMEOUT

Code	Error Constant
3000	RSW_UNABLE_TO_CREATE_EVENT
3001	RSW_UNABLE_TO_CREATE_THREAD
3002	RSW_UNABLE_TO_CREATE_NEW_REQUEST
3003	RSW_UNABLE_TO_CREATE_NEW_THREAD_ENTRY
3004	RSW_UNABLE_TO_CREATE_MUTEX
3005	RSW_UNABLE_TO_CREATE_NEW_OBJECT
3006	RSW_UNABLE_TO_ALLOCATE_NEW_MEMORY

B.4 Obtaining Support Files

You can obtain support and log files to help troubleshooting by selecting **Create Support File** from the **Tools** submenu under the **Oracle Application Testing Suite** start menu. The OATSSupport.zip file will appear on your desktop.

B.5 Oracle Application Testing Suite Application Service Messages

This section lists the Oracle Application Testing Suite Application Service Messages with possible causes and resolutions.

B.5.1 Agents are Orphaned

Symptoms: Agents are orphaned, messages take forever to be dismissed, or session prematurely stops.

When it Occurs: Mostly *during* the test.

Cause(s): Communication line is interrupted between Oracle Load Testing, and Oracle Application Testing Suite Application Service, Agents.

Resolution: Check the following:

1. Make sure you can ping the machines where Oracle Load Testing, Oracle Application Testing Suite Application Service, and Agents are installed. Verify that the network is up and functional.
2. Go to <installdir>\logs directory and check the log files to see if there are any exceptions indicating the nature of the problem, for example, multiple sockets error with the subsequent forced session termination.
3. Go to System Log on the Event Viewer and check for Network Warnings.

B.5.2 Server is Not Initialized

Symptoms: "Server is not initialized". Messages take forever to be dismissed in the System Manager.

When it Occurs: Always *before* Test Run:

Cause(s): Oracle Application Testing Suite Application Service is not running or is having problems. The location for Oracle Load Testing is incorrect.

Resolution: Check the following: Check the following:

1. Make sure that right Host\Port number is being used.

2. Make sure that the Oracle Application Testing Suite Application Service is running: go to Services on that machine and check for "Oracle Application Testing Suite Application Service is started".
3. Stop and Start the server.
4. Go to <installdir>\logs directory and look at AdminServer log file to see if there are any exceptions indicating the nature of the problem, for example, the machine is disconnected from the network, etc.

B.5.3 Unable to Start Session

Symptoms: There is no page when you start Oracle Load Testing.

When it Occurs: When attempting to start Oracle Load Testing.

Cause(s): This problem can occur when:

- The Oracle Application Testing Suite Application Service has not been started.
- There is a port conflict.
- The installation was not good.
- CPU usage is at 100% because old messages from the last run are still there.

Resolution: Check the following:

To verify that the Oracle Application Testing Suite Application Service and the Oracle Load Testing Agent Service are running:

1. Close all Oracle Application Testing Suite products.
2. Open **Services** from the **Administrative Tools** in the Windows Control Panel.
3. Verify that the Oracle Load Testing Agent and Oracle Application Testing Suite services are running.
4. If necessary, restart the Oracle Load Testing Agent and Oracle Application Testing Suite services.
5. Restart the Oracle Application Testing Suite application and start the session.

To check for error messages in the server log, look in the following file located in the installation directory: C:\OracleATS\logs\wls_oats.log

Oracle Load Testing uses the following ports by default at startup: 8088, 7073, and 9001 on the target agent machine. Port conflicts are indicated by a message similar to the following:

```
java.net.BindException: Address already in use: JVM_Bind
```

in the following files located in the installation directory:

```
C:\OracleATS\logs\wls_oats.log
```

```
C:\OracleATS\logs\servers\AdminServer\logs\*.log
```

```
C:\OracleATS\logs\agentmanager.log
```

To see whether a port is available:

- Type `netstat -a` at the command prompt or
- Download and run the `tcpview` utility from <http://www.sysinternals.com>

To delete old messages when CPU usage is 100%:

1. Open **Services** from the **Administrative Tools** in the Windows Control Panel, and stop the Oracle Application Testing Suite Application Service.
2. Delete everything in the `<installdir>\oats\servers\AdminServer\data\store\default` directory.
3. Restart the Oracle Application Testing Suite Application Service.

Additional items to check:

- Make sure the service is bounded into the correct IP address if there are multiple NICs. Type `ipconfig/all` in a command prompt to see how many interfaces you have.

If the problem continues, contact Customer Support.

B.6 Problems During Load Testing

This section lists the possible problems during load testing with possible causes and resolutions.

B.6.1 Cannot Open Scenario

Symptoms: The scenario cannot be opened or scripts are missing.

Cause(s): This problem can occur when:

- The network drive cannot be accessed because the Oracle Application Testing Suite Application Service is running under a different user account than the agent.
- The saved scenario does not contain the required script or databank files. This can happen when the scenario has been copied from another machine.
- The repository and workspace structure is not the same as the recorded scenario. This can happen when the scenario has been copied from another machine.

Resolution: Check the following:

1. Check for error messages in the server log, in the following file located in the installation directory:
`<installdir>\oats\servers\AdminServer\logs\AdminServer.log`
If there are errors refer to the instructions in the "Unable to Start Oracle Load Testing Session" section.
2. Check the account that the agent is running under using `Services.msc`.
3. Verify that the network path is available to this account. Select **Log On** in the **Services** under in the **Administrative Tools** of the Control Panel.
4. Log on as this user account. See [Section 3.1.5.2, "Configuring Oracle Load Testing Agents"](#) for instructions on changing the user account.
5. Ensure that the workspace and repository structure are the same on the machine running the test as on the machine from which the scenario was copied.

Ensure that all script and databank files are present. Copy any missing script and databank files from the original machine to the new machine.

B.6.2 Unable to Start Agent Session

Symptoms: The session cannot be started.

Cause(s): This problem can occur when:

- There is a database error.
- There are agent manager configuration problems.
- There are connection problems between machines, such as dual NICs and firewalls.
- The session timeout setting is too low.
- The Oracle Load Testing version is incompatible between the server and the agent.

Resolution: Check the following:

1. Check for error messages in the server log, in the following file located in the installation directory: *<installdir>\logs\agentmanager.log*

If there are errors, refer to the instructions under ["Unable to Start Session"](#) on page B-5.

To check for a database error:

1. Open the database management user interface.
2. Check for database errors.

To change the default agent time out value:

1. Stop the Oracle Application Testing Suite Application Service.
2. Change the setting `oracle.load.agent.timeout.default`. The default value for this setting is located in the `DEFAULT_GENERAL_SETTINGS` table in the OATS schema in the OATS database. Any customer-set value is located in the `GENERAL_SETTINGS` table of the schema.
3. Change the `oracle.load.agent.timeout.default` entry to a new value in seconds.
4. Restart the Oracle Application Testing Suite Application Service.

If the machines are running different Oracle Load Testing versions, install the same version on all machines.

B.6.3 Performance Problems During the Load Test

Symptoms: The user interface froze during the load test and no longer responds.

Cause(s): This can occur when:

- CPU usage is at 100%
- Memory is out of range

Resolution: Check the following:

1. Check for error messages in the server log, in the following file located in the installation directory:

<installdir>\logs\wls_oats.log

If there are errors refer to the instructions under ["Unable to Start Session."](#)

2. Go to *<installdir>\oats\servers\AdminServer\logs* directory and look at `AdminServer.log` file to see if there are any exceptions indicating the nature of the problem, for example, multiple sockets error with the subsequent forced session termination.
3. Go to System Log on the Event Viewer and check for Network Warnings.

To delete old messages when CPU usage is 100%:

1. Open Services in the Administrative Tools of the Windows Control Panel and stop the Oracle Application Testing Suite Application Service.
2. Delete everything in the
`<installdir>\oats\servers\AdminServer\data\store\default` directory.

To increase memory if there is an out of memory error, follow these steps:

1. Stop Oracle Application Testing Suite Application Service.
2. Open Regedit and locate key HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Oracle Application Testing Suite Application Service\Parameters.
3. In the string value `CmdLine` locate arguments `-Xms256m -Xmx512m` (NOTE: Actual values may differ).
Argument `-Xms` specifies the amount of memory available to Weblogic Server, while `-Xmx` specifies the maximum.
4. After setting the desired values for both minimum and maximum memory limits, close regedit and restart the machine.

B.6.4 Unexpected Agent Exit in the VU Grid

The Oracle Load Testing Agent service is responsible for starting agent processes, monitoring when they shut down, and terminating them when asked to do so by the Oracle Load Testing controller. Whenever Agent notices that a process terminates, it asks Windows for the exit code of that process. This is the exit code value reported to Oracle Load Testing and logged in the `<installdir>\logs\agentmanager.log` file.

One of the following values is returned:

- The exit value specified in the `ExitProcess` or `TerminateProcess` function.
- The return value from the `main` or `WinMain` function of the process.
- The exception value for an unhandled exception that caused the process to terminate.

The following are known causes for commonly seen exit codes:

Unexpected Agent Exit. Code = 143

This code is returned when terminating a JavaAgent process by closing its console window manually, or by terminating the application using the Task Manager's Applications tab.

Unexpected Agent Exit. Code = 1

This code is returned when `WgPro.exe` or `JavaAgent.exe` is terminated directly using the Windows Task Manager Processes tab.

Unexpected Agent Exit. Code = 0

This code is returned when the agent form is closed manually or using the Task Manager Applications tab.

Exit code 0 is always reported if a JavaAgent process terminates itself normally. This can happen, for example, if a user callback code were to call `System.exit(1)`.

B.7 Data Collector Startup Errors

This section uses the following terminology:

Server - the server that the user pointed his/her browser at. For example, Oracle Application Testing Suite Application Service on the server machine.

Remote Data Collection Server (or RDCS) - the Weblogic server running on the machine where the user is trying to start a data collector. This may or may not be the same Weblogic instance as the server. This is also known as the Oracle Application Testing Suite Application Service on the data collection machine.

Agent Manager - the agent launching service running on the machine where the user is trying to start a data collector. This is also known as Oracle Load Testing Agent Service.

Data Collector (or DC) - the data collection process started by Agent Manager.

Ds.dcStartupFailed

This error occurs if Agent Manager was unable to launch the DC process. This may happen when the remote data collection machine does not have data collectors installed. Verify that the software is installed correctly by checking for the presence of the DataCollector directory in the install tree and that the executables are there.

Ds.dcStartError

This error occurs if the remote data collection server has trouble communicating with Agent Manager. This may be caused by the following:

- Agent Manager is not running - verify that the Oracle Load Testing Agent Service is running and start it, if necessary.
- Remote Data Collection machine does not have data collectors installed - Verify that the software is installed correctly by checking for the presence of the DataCollector directory in the install tree and that the executables are there.

Ds.dcNotAvailable

This occurs most often if the DC was started, but is unable to communicate with the RDCS, or if the DC took more than the timeout interval to contact the RDCS. This may be caused by the following:

- DC machine is slow - extend the timeout interval. The data collector startup timeout is controlled on the RDCS by the `oracle.datacollector.startup.timeout=90000` in the `oracle.properties` file.
- DC and RDCS disagree on the port number for communication - the DC determines the port number of the RDCS from `DataCollector\bin\hwmagent.txt:SERVER localhost:7073`. The RDSC port number is assigned in `oracle.properties:oracle.port.dataCollector=7073`.

errorCreatingDSClient

This occurs if the server has problems communicating with the RDCS. If the server and RDCS are the same, this error should not occur. It can be caused by the following:

- RDCS is not running - verify that the Oracle Application Testing Suite Application Service is running on the DC machine.
- Basic network connectivity between the server and RDCS - check that the basic network connectivity between the server and RDCS. If there is a firewall between

them, ports may need to be opened or, if the firewall uses NAT, it may not work at all.

C

Command Line Reference

This appendix lists the parameters for running Oracle Load Testing scenarios from the command line.

To run Oracle Load Testing scenarios from the command line, type:

```
[OATS Install Dir]/jdk/bin/java -jar OLTCommandLine.jar -command  
-option=value -option=value [...]
```

-option may consist of any number of Oracle Load Testing command line settings. Required arguments are specified based upon the command and are case-insensitive unless otherwise specified.

C.1 Specifying Command Line Settings

This section describes how to use the command line settings.

- Command line settings may be specified using the following format:

```
OLTCommandLine.jar -command -option=value [...]
```

Example 1: Run a scenario and log the results to a file.

```
cd C:[OATS Install Dir]/lib  
[OATS Install Dir]/jdk/bin/java -jar OLTCommandLine.jar -run  
-scenarioFile="C:/OracleATS/OFT/Simple Scenario.scn" -OLTServer=localhost:8088  
-user=oats -password=mypass123 -log=C:/OracleATS/OFT/mylog.log
```

Example 2: Run a scenario with rampup of 2 Users every 30 seconds.

```
cd C:[OATS Install Dir]/lib  
[OATS Install Dir]/jdk/bin/java -jar OLTCommandLine.jar -run  
-scenarioFile="c:/oracleats/OFT/Simple Scenario.scn" -session=TestSession  
-OLTServer=localhost:8088 -user=oats -password=mypass123 -iteration=1  
-groupdatanumber=2 -ramptriggerafterseconds=30
```

Example 3: Run a scenario with rampup of 2 Users every 2 iterations.

```
cd C:[OATS Install Dir]/lib  
[OATS Install Dir]/jdk/bin/java -jar OLTCommandLine.jar -run  
-scenarioFile="c:/oracleats/OFT/Simple Scenario.scn" -session=TestSession  
-OLTServer=localhost:8088 -user=oats -password=mypass123 -groupdatanumber=2  
-ramptriggerafteriterations=2
```

Example 4: Get the Oracle Load Testing version.

```
cd C:[OATS Install Dir]/lib  
[OATS Install Dir]/jdk/bin/java -jar OLTCommandLine.jar -version  
-OLTServer=localhost:8088 -user=oats -password=mypass123
```

Example 5: Stop Session0001.

```
cd C:[OATS Install Dir]/lib
[OATS Install Dir]/jdk/bin/java -jar OLTCommandLine.jar -stop
-session="Session0001" -OLTServer=localhost:8088 -user=oats -password=myspass123
```

Example 6: Generate Session Report.

```
cd C:[OATS Install Dir]/lib
[OATS Install Dir]/jdk/bin/java -jar OLTCommandLine.jar -report
-OLTServer=localhost:8088 -user=oats -password=myspass123 -session=mysession001
-thinkTime=false -filterDateMin="3/19/2013 08:00:00" -filterDateMax="3/19/2013
15:30:00" -log= C:/OracleATS/OFT/Default/mylog.log -localFile=c:/myreport.htm
```

- If many settings are required, use the `-property` setting to specify a file containing a larger number of properties. See the `-property` setting in the [Run Settings](#) for additional information.

C.1.1 Error Handling

If the OLT server is not running, an error will occur.

If the scenario file does not exist, an error will occur.

If the scenario file is not a correct xml, an error will occur.

C.2 Supported Command Line Settings

This section lists the command line settings that can be used with Oracle Load Testing.

C.2.1 Commands

The following table lists commands used for Oracle Load Testing command line execution. The options include the [General Settings](#) and any command-specific additional options.

Command	Description
<code>-addagent -[options]</code>	<p>Adds an Oracle Load Testing agent using the specified options. You can verify that the agent is been added to the systems manager by selecting Options from the User menu then selecting VU Agent Systems in the Systems group. See the AddAgent Settings for additional options.</p> <p>Example usage:</p> <pre>cd /OracleATS/lib c:/OracleATS/jdk/bin/java -jar OLTCommandLine.jar -addagent -agentname=Cloud_ Agent1 -agenthostnameorip=mysystem.example.com -agentpassword=myspass123 -OLTServer=localhost:8088 -user=oats -password=myspass123 c:/OracleATS/jdk/bin/java -jar OLTCommandLine.jar -importandrun -scenarioFile="Path_of_scenariozip" -OLTServer=localhost:8088 -user=oats -password=myspass123 -repository=Default -target=Cloud_Agent1 -session=mysessionname</pre>

Command	Description
<code>-addagentgroup -[options]</code>	<p>Adds an Oracle Load Testing agent group using the specified options. You can verify that the agent group has been added to the systems manager by selecting Options from the User menu then selecting VU Agent Systems in the Systems group. See the AddAgentGroup Settings for additional options.</p> <p>Example usage:</p> <pre>cd /OracleATS/lib c:/OracleATS/jdk/bin/java -jar OLTCommandLine.jar -addagentgroup -agentgroupname=myagentgroup -systems="agent1;agent2" -OLTServer= localhost:8088 -user=oats -password=myspass123</pre>
<code>-getdbconfig -[options]</code>	<p>Gets the database configuration information. username is the db username ('sysman' by default) & password is the database password. OLTServer may be any machine that has the Oracle Load Testing controller installed. This command returns the currently configured database-related information in XML format, for example:</p> <pre><DBConfig name="Default OLT Database"><description>description text</description><DBConnect type="Oracle" string="jdbc:oracle:thin:@//localhost:1521/XE" acct="OLT"></DBConnect></DBConfig></pre> <p>Example usage:</p> <pre>cd /OracleATS/lib c:/OracleATS/jdk/bin/java -jar OLTCommandLine.jar -getdbconfig -OLTServer= localhost:8088 -user=oats -password=myspass123</pre>
<code>-export -[options]</code>	<p>Exports the specified scenario to a scenario .scnzip file. See the Export and Import Settings for additional options.</p> <p>Example usage:</p> <pre>cd /OracleATS/lib c:/OracleATS/jdk/bin/java -jar OLTCommandLine.jar export -scenarioFile=c:/OracleATS/OFT/myTest.scn -OLTServer=localhost:8088 -user=oats -password=myspass123 -repository=Default -workspace=myfolder</pre>
<code>-import -[options]</code>	<p>Imports the specified scenario .scnzip file. See the Export and Import Settings for additional options.</p> <p>Example usage:</p> <pre>cd /OracleATS/lib c:/OracleATS/jdk/bin/java -jar OLTCommandLine.jar -import -OLTServer=localhost:8088 -user=oats -password=myspass123 -repository=Default -scenarioFile=c:/OracleATS/OFT/myTest.scnzip</pre>

Command	Description
<code>-importandrun -[options]</code>	<p>Imports the specified scenario .scnzip file and runs the scenario.</p> <p>Example usage:</p> <pre>cd /OracleATS/lib c:/OracleATS/jdk/bin/java -jar OLTCommandLine.jar -importandrun -scenarioFile=c:/OracleATS/myTest.scnzip -OLTServer=localhost:8088 -user=oats -password=myspass123 -repository=Default</pre>
<code>-report -[options]</code>	<p>Generates a session report for the specified session. See the Report Settings for additional options.</p> <p>Example usage:</p> <pre>cd /OracleATS/lib c:/OracleATS/jdk/bin/java -jar OLTCommandLine.jar -report -OLTServer=localhost:8088 -user=oats -password=myspass123 -session=mysession001 -thinkTime=false -filterDateMin="3/19/2013 08:00:00" -filterDateMax="3/19/2013 15:30:00" -log= C:/OracleATS/OFT/Default/mylog.log -localFile=c:/myreport.htm</pre>
<code>-run -[options]</code>	<p>Runs the selected scenario until the command line stop command is issued, the specified command line stopat condition is reached, or the stop conditions specified in the scenario are reached. See the Run Settings for additional options.</p> <p>Example usage:</p> <pre>cd /OracleATS/lib c:/OracleATS/jdk/bin/java -jar OLTCommandLine.jar -run -scenarioFile="C:/OracleATS/OFT/Default/myTest .scn" -OLTServer=localhost:8088 -user=oats -password=myspass123 -log= C:/OracleATS/OFT/Default/mylog.log</pre>
<code>-status -[options]</code>	<p>Returns the current Oracle Load Testing session status in the following format:</p> <pre>-1: session not found 0: session complete 1: session running</pre> <p>Example usage:</p> <pre>cd /OracleATS/lib c:/OracleATS/jdk/bin/java -jar OLTCommandLine.jar -status -OLTServer=localhost:8088 -user=oats -password=myspass123 -session=mySession</pre>

Command	Description
<code>-stop -[options]</code>	<p>Shuts down the specified session. The Stop command is not required if the specified Scenario being run has a Stop condition of some defined time or number of iterations saved in the Ramp settings. You can not use both "-Run" and "-Stop" in the same command. See the Stop Settings for additional options.</p> <p>Example usage:</p> <pre>cd /OracleATS/lib c:/OracleATS/jdk/bin/java -jar OLTCommandLine.jar -stop -session="SESSION0001" -OLTServer=localhost:8088 -user=oats -password=mypass123</pre>
<code>-version -[options]</code>	<p>Gets the current Oracle Load Testing version in XML format, for example:</p> <pre>OLT Version is <OLTVersion version="xx.x.x.x.xxx"></OLTVersion></pre> <p>Example usage:</p> <pre>cd /OracleATS/lib c:/OracleATS/jdk/bin/java -jar OLTCommandLine.jar -version -OLTServer=localhost:8088 -user=oats -password=mypass123</pre>

C.2.2 General Settings

The following table lists the command line settings used for all commands.

Setting	Description
<code>-OLTServer=servername:port</code>	<p>Specify the name and port number of the Oracle Load Testing server. The default port is 8088.</p> <p>Example usage:</p> <pre>-OLTServer=localhost:8088</pre>
<code>-user=oats</code>	<p>Specify the user name for the Oracle Load Testing server. The required username is "oats".</p> <p>Example usage:</p> <pre>-user=oats</pre>
<code>-password=password</code>	<p>Specify the password. The password is the password defined during the Oracle Application Testing Suite installation process.</p> <p>Example usage:</p> <pre>-password=mypass123</pre>

C.2.3 AddAgent Settings

The following table lists the AddAgent command line settings. These settings are used in addition to the General Settings.

Setting	Description
<code>-agentname=agentname</code>	Specify the name of the Oracle Load Testing agent. Example usage: <code>-agentname=Cloud_agent1</code>
<code>-agenthostnameorip=nameorIp</code>	Specify the host name or IP address of the Oracle Load Testing agent. Example usage: <code>-agenthostnameorip=mysystem.example.com</code>
<code>-agentpassword=password</code>	Specify the password to use for the Oracle Load Testing agent. Example usage: <code>-agentpassword=myspass123</code>

C.2.4 AddAgentGroup Settings

The following table lists the AddAgentGroup command line settings. These settings are used in addition to the General Settings.

Setting	Description
<code>-agentgroupname=name</code>	Specify a name for the Oracle Load Testing agent group. Example usage: <code>-agentgroupname=myagentgroup</code>
<code>-systems=name1;name2;name_n</code>	Specify the names of the Oracle Load Testing agents. Agents names are separated by semi-colon (;) in the <code>-systems</code> option. Example usage: <code>-systems="agent1;agent2"</code>

C.2.5 Export and Import Settings

The following table lists the import and export command line settings. These settings are used in addition to the General Settings.

Setting	Description
<code>-localFile=localFile</code>	(Export only.) Specify a name to use for the exported Oracle Load Testing scenario file. Example usage: <code>-localFile=myFile.scnzip</code>
<code>-log=logfile</code>	Optional. Specify the path and file name of a log file. Example usage: <code>-log=c:/OracleATS/OFT/mylog.txt</code>
<code>-repository=repositoryName</code>	Specify a name for the Oracle Load Testing repository. Example usage: <code>-repository=Default</code>

Setting	Description
<code>-scenarioFile=scenarioFile</code>	Specify the path and name of the Oracle Load Testing scenario (.scn) file to import or export. Example usage: <code>-scenarioFile="c:/OracleATS/OFT/Default/test1.scn"</code>
<code>-workspace=workspaceName</code>	Optional. Specify a name for an Oracle Load Testing workspace folder. Example usage: <code>-workspace=myFolder</code>

C.2.6 Report Settings

The following table lists the Report command line settings. These settings are used in addition to the General Settings.

Setting	Description
<code>-localFile=localFile</code>	Specify a name to use for the generated Oracle Load Testing session report. Example usage: <code>-localFile=mySessionReport.html</code>
<code>-log=logfile</code>	Optional. Specify the path and file name of a log file. Example usage: <code>-log=c:/OracleATS/OFT/mylog.txt</code>
<code>-session=sessionName</code>	Specify the name of the session to use to generate the report. Example usage: <code>-session=mysession001</code>
<code>-thinkTime=true false</code>	Optional. Specify if think times are included in the session report. When <code>true</code> , only Server Times are included in the report. When <code>false</code> , End-to-End Times are included in the report (includes "Think Times"). The default is <code>true</code> . Example usage: <code>-thinkTime=false</code>
<code>-filterDateMin=M/d/yyyy HH:mm:ss</code>	Optional. Specify the minimum date to use to filter the report data. Example usage: <code>-filterDateMin="3/19/2013 08:00:00"</code>
<code>-filterDateMax=M/d/yyyy HH:mm:ss</code>	Optional. Specify the maximum date to use to filter the report data. Example usage: <code>-filterDateMax="3/19/2013 015:30:00"</code>

C.2.7 Run Settings

The following table lists the Report command line settings. These settings are used in addition to the General Settings.

Setting	Description
<code>-property=propertyFile</code>	<p>Specify the file path to a properties file listing only the command line <code>-user</code> and <code>-password</code> arguments in <code>name=value</code> pair format.</p> <p>The format in the property file is:</p> <pre>name=value</pre> <p>For example:</p> <pre>user=oats</pre> <pre>password=mypass123</pre> <p>Example usage:</p> <pre>-property=c:/OLTRun.properties</pre>
<code>-scenarioFile=scenarioFile</code>	<p>Specify the path and name of the Oracle Load Testing scenario (.scn) file to run.</p> <p>Example usage:</p> <pre>-scenarioFile="c:/OracleATS/OFT/Default/test1.scn"</pre>
<code>-session=sessionName</code>	<p>Optional. Specify a name to use for the Oracle Load Testing session. If not specified, a session name is automatically assigned. Session names that are automatically assigned can be located on the console or in the log file specified using the <code>-log</code> option.</p> <p>Example usage:</p> <pre>-session=MySession001</pre>
<code>-log=logfile</code>	<p>Optional. Specify the path and file name of a log file.</p> <p>Example usage:</p> <pre>-log=c:/OracleATS/OFT/mylog.txt</pre>
<code>-useCloud</code>	<p>Required if using a Cloud agent. Indicates the Oracle Load Testing scenario will be run on an Oracle Enterprise Manager Cloud. Specify the additional required and optional cloud run settings.</p> <p>Example usage:</p> <pre>-useCloud</pre>
<code>-cloudEmHost=EM Host</code>	<p>Required if using a Cloud agent. Specify the name of the Oracle Enterprise Manager Cloud host machine.</p> <p>Example usage:</p> <pre>-cloudEmHost=mycloud.example.com</pre>
<code>-cloudEmPort=EM Port</code>	<p>Required if using a Cloud agent. Specify the port used for the Oracle Enterprise Manager Cloud.</p> <p>Example usage:</p> <pre>-cloudEmPort=17061</pre>
<code>-cloudEmUsername=EM Username</code>	<p>Required if using a Cloud agent. Specify the user name used for the Oracle Enterprise Manager Cloud.</p> <p>Example usage:</p> <pre>-cloudEmUsername=myEMusername</pre>

Setting	Description
<code>-cloudEmPassword=EM Password</code>	<p>Required if using a Cloud agent. Specify the password used for the Oracle Enterprise Manager Cloud.</p> <p>Example usage:</p> <pre>-cloudEmPassword=myEMpassword</pre>
<code>-cloudAssembly=Assembly Name</code>	<p>Required if using a Cloud agent. Specify the assembly name used for the Oracle Enterprise Manager Cloud.</p> <p>Example usage:</p> <pre>-cloudAssembly=assemblyName</pre>
<code>-cloudZone=Zone Name</code>	<p>Required if using a Cloud agent. Specify the zone name used for the Oracle Enterprise Manager Cloud.</p> <p>Example usage:</p> <pre>-cloudZone=zoneName</pre>
<code>-cloudPlan=DeploymentPlan</code>	<p>Required if using a Cloud agent. Specify the deployment plan used for the Oracle Enterprise Manager Cloud.</p> <p>Example usage:</p> <pre>-cloudPlan=myDeploymentPlanName</pre>
<code>-cloudNetwork=Network Name</code>	<p>Required if using a Cloud agent. Specify the network name used for the Oracle Enterprise Manager Cloud.</p> <p>Example usage:</p> <pre>-cloudNetwork=myNetworkName</pre>
<code>-cloudInitial=Initial Agent Count</code>	<p>Optional if using a Cloud agent. Specify the initial agent count to use when running scenarios in the Oracle Enterprise Manager Cloud. The default is 3.</p> <p>Example usage:</p> <pre>-cloudInitial=2</pre>
<code>-cloudMinimum=Minimum Free Agent Count</code>	<p>Optional if using a Cloud agent. Specify the minimum free agent count to use when running scenarios in the Oracle Enterprise Manager Cloud. The default is 0.</p> <p>Example usage:</p> <pre>-cloudMinimum=1</pre>
<code>-cloudMaximum=Maximum Free Agent Count</code>	<p>Optional if using a Cloud agent. Specify the maximum free agent count to use when running scenarios in the Oracle Enterprise Manager Cloud. The default is 10.</p> <p>Example usage:</p> <pre>-cloudMaximum=5</pre>
<code>-cloudIncrement=Cloud Agent Increment</code>	<p>Optional if using a Cloud agent. Specify the agent increment to use when running scenarios in the Oracle Enterprise Manager Cloud. The default is 2.</p> <p>Example usage:</p> <pre>-cloudIncrement=1</pre>

Setting	Description
<code>-cloudPrefix=Agent Prefix</code>	Optional if using a Cloud agent. Specify the agent prefix string to use when running scenarios in the Oracle Enterprise Manager Cloud. The default is "OLT". Example usage: <code>-cloudPrefix=myOLT</code>
<code>-cloudSystemCpuMax=System CPU threshold</code>	Optional if using a Cloud agent. Specify the percentage of CPU to use as the agent threshold when running scenarios in the Oracle Enterprise Manager Cloud. The default is 70%. Example usage: <code>-cloudSystemCpuMax=65</code>
<code>-cloudSystemMemFreeMin=System Free Memory threshold</code>	Optional if using a Cloud agent. Specify the amount of free memory to use as the agent threshold when running scenarios in the Oracle Enterprise Manager Cloud. The default is 200MB. Example usage: <code>-cloudSystemMemFreeMin=195</code>
<code>-groupdatapercents=% users</code>	Optional. Specify the percentage of Virtual Users to rampup on the rampup trigger when running scenarios. Used with a <code>ramptriggerafter</code> setting. Example usage: <code>-groupdatapercents=5</code>
<code>-groupdatanumber=# users</code>	Optional. Specify the number of Virtual Users to rampup on the rampup trigger when running scenarios. Used with a <code>ramptriggerafter</code> setting. Example usage: <code>-groupdatanumber=2</code>
<code>-iteration=# iterations</code>	Optional. Specify the number of iterations to run when running scenarios. Example usage: <code>-iteration=2</code>
<code>-ramptriggerafteriterations=# iterations</code>	Optional. Specify the number of iterations to use as the VU rampup trigger when running scenarios. Used with the <code>iteration</code> setting. Example usage: <code>-ramptriggerafteriterations=2</code>
<code>-ramptriggerafterseconds=# seconds</code>	Optional. Specify the number of seconds to use as the VU rampup trigger when running scenarios. Used with a <code>groupdata</code> setting. Example usage: <code>-ramptriggerafterseconds=2</code>
<code>-ReplaceURLs=urlA=urlB</code>	Optional. Specify the URLs to replace within scripts when running scenarios. Example usage: <code>-ReplaceURLs=abc:7011=def:7011</code>

Setting	Description
<code>-stopatdelay=# seconds</code>	Optional. Specify the number of seconds to run when running scenarios. This setting overrides any stop controls defined in the scenario file. Example usage: <code>-stopatdelay=500</code>
<code>-stopattime=hh:mm:ss</code>	Optional. Specify the hour, minute, and second to use as the stop time when running scenarios. This setting overrides any stop controls defined in the scenario file. Example usage: <code>-stopattime=10:30:00</code>
<code>-scriptname</code> <code>-scriptname[.target=target]</code> <code>-scriptname[.VUCount=count]</code>	Optional. Specify a script within a scenario to run. Optionally specify the target agent to use for the specified script(s) and/or the Virtual User count for the specified script(s) when running scenarios. Use double quotation marks if spaces are used the script name. If <code>target</code> is not specified, Oracle Load Testing will use the target agents(s) saved in the scenario file. <code>VUCount</code> is case sensitive. If <code>VUCount</code> is not specified, Oracle Load Testing will use the VU count(s) saved in the scenario file. Example usages: <code>-SimpleTest</code> <code>-SimpleTest.VUCount=5</code> <code>-"Simple Test".target=Cloud_Agent</code> <code>-"Simple Test".VUCount=2</code>
<code>-target=target</code>	Optional. Specify the target agent to use when running scenarios. The target agent will be applied to all scripts in the scenario. Example usage: <code>-target=OLTServer</code>
<code>-VUCount=count</code>	Optional. Specify the Virtual User count to use when running scenarios. The Virtual User count will be applied to all scripts in the scenario. <code>VUCount</code> is case sensitive. Example usage: <code>-VUCount=10</code>

C.2.8 Status Settings

The following table lists the Status command line settings. These settings are used in addition to the General Settings.

Setting	Description
<code>-session=<i>sessionName</i></code>	Specify the name of the session. If not specified, the status of the currently running session is returned in the following format: -1: session not found 0: session complete 1: session running Example usage: <code>-session=Session0001</code>

C.2.9 Stop Settings

The following table lists the Stop command line settings. These settings are used in addition to the General Settings.

Setting	Description
<code>-log=<i>logfile</i></code>	Optional. Specify the path and file name of a log file. Example usage: <code>-log=c:/OracleATS/OFT/mylog.txt</code>
<code>-session=<i>sessionName</i></code>	Optional. Specify the name of the session to stop. If not specified, the currently running session is stopped. Example usage: <code>-session=Session0001</code>

ServerStats Metric Descriptions

This appendix describes the default Oracle Load Testing ServerStats metrics. Use the Perfmon data source to add metrics that are not listed.

D.1 AIX 4.3 Virtual Agent

File System Free Space - Total free space on the mounted directory (from df -k).

File System Percent Used - The directory space in use (from df -k).

File System Size - The total size of the mounted directory (from df -k).

vmstat cpu(id) - CPU Idle Time - CPU Idle Time (from vmstat cpu(id) value).

vmstat kthr(b) - Number of kernel threads placed in wait queue (awaiting resource, awaiting input/output).

vmstat kthr(r) - Number of kernel threads placed in run queue.

vmstat memory(avm) - Active virtual pages.

vmstat memory(free) - Size of the free list.

D.2 Apache Web Server

Busy Servers - The busy servers.

Bytes/Request - The number of bytes per request.

Bytes/sec - The number of bytes per second.

Idle Servers - The idle servers.

Requests/Second - The number of page requests per second.

Server Uptime - The server up time in days.

D.3 Cisco Network Devices

1 Minute CPU Utilization - The overall CPU utilization for the last 1 minute period (cpmCPUTotal1min).

Chassis Temperature Alarm (1=No Alarm) - Chassis temperature alarm status (chassisTempAlarm). 1=off, 2=on, 3=critical.

Current Bandwidth Usage - The current bandwidth in megabits per second on 3500 XL series switches.

Layer 3 Packet Throughput - Number of packets per minute switched at Layer 3 by this engine (cseL3SwitchedTotalPkts).

Peak Bandwidth Usage - The most recent entry in the peak bandwidth table.

Power Supply 1 Status (2=OK) - Status of power supply number 1 (chassisPs1Status). 2=ok, 3=minor fault, 4=major fault.

Power Supply 2 Status (2=OK) - Status of power supply number 2 (chassisPs2Status). 2=ok, 3=minor fault, 4=major fault.

Traffic Utilization, Peak - Peak traffic meter value since the last time port counters were cleared or the system started (sysTrafficPeak).

Traffic Utilization - Traffic meter value for the previous polling interval (sysTraffic).

D.4 Foundry Load Balancer

CPU Util - The SNMP Foundry CPU util.

Fan 1 - The fan 1 status for SI/FIWG.

Fan 2 - The fan 2 status for SI/FIWG.

Power Supply 2 - The power supply 2 status for SI/FIWG.

D.5 General Web Server

IP Throughput - IP traffic through a host in bytes per second.

Ping - Measures response time for an ICMP echo request to determine system availability over the network.

Home Page Response Time - The number of seconds required to download the server home page.

D.6 Host

Ping (ms) - The response time for an ICMP echo request to determine system availability over the network.

Ping (percent) - Measures percentage of success for several ICMP echo requests to determine system availability over the network.

Ping (sec) - Measures response time for an ICMP echo request to determine system availability over the network.

D.7 Host Resources MIB

% Disk Space Used - Percentage of disk space used (hrStorageUsed/hrStorageSize).

% Memory Used - Percentage of memory used (hrStorageUsed/hrStorageSize).

% Swap Space Used - Percentage of swap space used (hrStorageUsed/hrStorageSize).

Process CPU Time - Number of seconds of the total system's CPU resources consumed by this process per minute (hrSWRunPerfCPU).

Process Memory Used - Kilobytes of real system memory allocated to this process (hrSWRunPerfMem).

System Processes - The number of process contexts currently loaded or running on this system (hrSystemProcesses).

D.8 IIS Web Server

ASP Queue Length - Number of ASP requests waiting for service from the queue.

ASP Request Rate - Number of ASP requests processed per minute. If high, may indicate slow response.

Blocked Request Rate - Number of requests temporarily blocked due to bandwidth throttling per minute.

CGI Request Rate - Number of CGI requests processed per minute.

HTTP 404 Not Found Error Rate - Number of HTTP 404 Not Found errors reported per minute.

KBytes Transferred Per Minute - Number of kilobytes sent and received per minute.

Rejected Request Rate - Number of requests rejected due to bandwidth throttling per minute.

D.9 Linux Virtual Agent

% Disk Space Used - How much of the file system's total capacity has been used (from df -kP).

Disk Size - The Disk Size (from df -kP).

Disk Space Available - Total amount of available disk space (from df -kP).

Disk Space in Use - Total amount of disk space in use (from df -k).

vmstat cpu(id) - CPU Idle Time - CPU Idle Time.

vmstat cpu(sy) - CPU System Time - CPU System Time.

vmstat cpu(us) - CPU User Time - CPU User Time.

vmstat memory(free) - Free Memory - Free Memory.

D.10 Memory

% Committed Bytes In Use - Percentage of available virtual memory in use. If high, may indicate low memory and slow system response.

% MDL Read Cache Hits - Percentage of memory descriptor list (MDL) read requests that hit the cache.

% Paging File In Use - Percentage of virtual memory paging file in use. If high, may indicate low memory or poor paging file configuration.

Committed Kbytes - Virtual memory kilobytes committed, i.e. really used, not just reserved. Committed memory should be less than physical RAM.

Free Kbytes - Virtual memory kilobytes available. If low, may indicate low memory and slow system response.

Page Faults Per Second - Soft and hard paging; virtual memory pages read from non-Working Set memory or from disk. Look also at Pages Per Second.

Pages Per Second - Hard paging; virtual memory pages read to/from disk per second. If high, may indicate memory shortage.

D.11 MIB-II Network Devices

icmpOutDestUnreachs Per Minute - SNMP: ICMP Destination Unreachable messages sent per minute. May indicate nonexistent address or hardware out of service.

icmpOutSrcQuenchs Per Minute - SNMP: ICMP Source Quench messages sent per minute. Indicates datagrams are arriving too quickly for the host to process.

icmpOutTimeExcds Per Minute - SNMP: ICMP Time Exceeded messages sent per minute. Indicates inability of datagrams to reach their destination.

ifInDiscards Per Minute - SNMP: Network input packets discarded per minute. Dropped packets may indicate a network hardware or congestion problem.

ifInErrors Per Minute - SNMP: Network input errors per minute. Errors may indicate a network hardware problem.

ifInNUcastPkts Per Minute - SNMP: Network non-unicast input packets per minute.

ifInOctets Per Minute - SNMP: The total number of octets received on the interface per minute, including framing characters.

ifInUcastPkts Per Minute - SNMP: Network unicast input packets per minute.

ifOutDiscards Per Minute - SNMP: Network output packets discarded per minute. Dropped packets may indicate a network hardware or congestion problem.

ifOutErrors Per Minute - SNMP: Network output errors per minute. Errors may indicate a network hardware problem.

ifOutNUcastPkts Per Minute - SNMP: Network output non-unicast packets per minute.

ifOutOctets Per Minute - SNMP: The total number of octets sent out the interface per minute, including framing characters.

ifOutUcastPkts Per Minute - SNMP: Network output unicast packets per minute.

ipInAddrErrors Per Minute - SNMP: IP input datagrams discarded per minute because their IP destination was not this machine.

ipInHdrErrors Per Minute - SNMP: IP input datagrams discarded per minute due to errors in their IP headers.

System Uptime - SNMP: The time in days since the network management portion of the system was last re-initialized (sysUpTime).

tcpOutRsts Per Minute - SNMP: TCP resets sent per minute. Resets indicate incoming connections to invalid ports were refused.

tcpOutSegs Per Minute - SNMP: TCP segments sent per minute. A measure of total TCP activity.

tcpRetransSegs Per Minute - SNMP: TCP segments retransmitted per minute. Retransmissions may indicate a network hardware problem.

D.12 Netscreen Firewall

Average CPU Usage - average CPU utilization.

CPU Usage in Last 5 Minutes - The CPU utilization in the last five minutes.

Incoming Packets - The incoming packet number through an interface.

Interface Availability - Determines whether an interface is active.

Memory Allocated - The memory allocated.

Memory Available - The memory still available.

Outgoing Packets - The outgoing packet number through an interface.

D.13 Network

Packets Received Per Second - The rate at which packets are received on the network interface.

Packets Sent Per Second - The rate at which packets are send out on the network interface

D.14 Oracle Database

Average active session - The average active sessions at a point in time. It is the number of sessions that are either working or waiting. It is the average session load in seconds per second.

Average active sessions on CPU - The average active sessions on at a point in time. It is the number of sessions that are using CPU.

CPU parse time usage - Used to identify how much time is spent parsing by the CPU. It is the Total CPU time used for parsing (hard and soft) in 10s of milliseconds. See the Oracle Database *Performance Tuning Guide* for additional information about 'parse time cpu'.

CPU recursive usage - Used to identify the total CPU time used by non-user calls (recursive calls). See the Statistics Descriptions in the Oracle Database *Reference* for additional information.

Physical IO per second - Rate of read/write requests on database blocks for all instance activity including application, backup and recovery, and other utilities.

Physical IO requests total - Number of write requests which wrote one or more database blocks from all instance activity including application activity, backup and recovery, and other utilities.

PGA (MB) - The Program Global Area (PGA) is a memory region containing data and control information for a single process (server or background). One PGA is allocated for each server process; the PGA is exclusive to that server process and is read and written only by Oracle code acting on behalf of that process. A PGA is allocated by Oracle when a user connects to an Oracle database and a session is created.

SGA (MB) - The System Global Area (SGA) is an area where data and program caches are shared among database users. It is made up of the database buffer cache, shared pool, and redo log buffer.

Total Global Area - The total of PGA(MB and SGA(MB).

User calls per second - The number of logins, parses, or execute calls per second.

User login rates per second - The number of user logins taking place in one second.

User Transaction Commit Rate - The number of database transactions per second. When a user commits a transaction, the redo generated that reflects the changes made to database blocks must be written to disk. Commits often represent the closest thing to a user transaction rate.

D.15 Solaris Virtual Agent

% Disk Space Used - How much of the file system's total capacity has been used (from `df -k`)

Disk Size - The Disk Size (from `df -k`).

Disk Space Available - Total amount of available disk space (from `df -k`).

Disk Space in Use - Total amount of disk space in use (from `df -k`).

vmstat cpu(id) - **CPU Idle Time** - CPU Idle Time.

vmstat cpu(sy) - **CPU System Time** - CPU System Time.

vmstat cpu(us) - **CPU User Time** - CPU User Time.

vmstat memory(free) - **Free Memory** - Free Memory.

D.16 Solaris (multi-CPU)

vmstat cpu(id) - **CPU Idle Time** - CPU Idle Time.

vmstat cpu(sy) - **CPU System Time** - CPU System Time.

vmstat cpu(us) - **CPU User Time** - CPU User Time.

D.17 SQL Server Databases

% Log Space Used - Percentage of transaction log space used. If the log becomes full, database transactions and recovery may fail.

Availability - Measures response time for a simple database query to verify SQL Server availability.

Average Latch Wait Time - Average time a SQL Server thread has to wait for a latch. If high, may indicate contention problem and slow response.

Average Lock Wait Time - Average time a SQL Server thread has to wait for a lock. If high, may indicate contention problem and slow response.

Batch Requests Rate - The number of SQL batch requests on the server.

Buffer cache hit ratio - Percentage of requests that hit the buffer cache. If low, may indicate poor database tuning and slow response.

Cache Hit Ratio (Ad Hoc Plans) - Percentage of ad hoc plans that hit the cache. If low, may indicate poor database tuning and slow response.

Cache Hit Ratio (Prepared Plans) - Percentage of prepared plans that hit the cache. If low, may indicate poor database tuning and slow response.

Checkpoint Pages Per Second - The number of pages flushed by Checkpoint or other operations that require all dirty pages to be flushed.

Data File Size - The cumulative size of all the files in the database.

Log File Size - The cumulative size of all the log files in the database.

Log Flushes per Second - The number of log flushes.

Number of Lock Blocks - The current number of lock blocks that are in use on the server.

Page Reads per Second - The number of database page reads issued.

Page Writes per Second - The number of database page writes issued.

Pages Read Ahead per Second - The number of pages read ahead in anticipation of its use.

Procedure Plans Cache Hit Ratio - The ratio between cache hits and lookups for procedure plans.

SQL Compilations per Second - The number of SQL compilations.

SQL Re-compilations per Second - The number of SQL re-compiles.

Total Server Memory - The total amount of dynamic memory the server is using for the dynamic SQL cache.

Transaction Rate - Number of database transactions per second. A measure of total database activity.

User Connections - Number of user connections.

D.18 WebLogic

Server Heap Utilization - SNMP: Percentage of heap used (serverHeapUsedPct).

Server Queue - SNMP: Current length of the execute queue (serverQueueLength).

Server State (1=Up, 2=Down) - SNMP: State of the WebLogic server (serverState).
1=up, 2=down.

Server Throughput - SNMP: Throughput of the execute queue in requests per second (serverQueueThroughput).

Numerics

4GT feature
enabling, 3-13

A

Abort All Virtual Users toolbar button, 3-17
Absolute Time setting, 10-8
Accept String setting, 5-8, 14-9
Acceptable failure rate
defining test criteria, 2-7
Acceptable response time under various loads
defining test criteria, 2-7
Acceptance and Scalability Fine Tuning, 2-2
Acceptance environment
defining test process, 2-6
access type, 3-16
Active Session tab, 3-22, 10-1
Active Session UI refresh interval setting, 11-7
Active Virtual Users, 11-2
Add setting, 6-3
Additional Arguments setting, 5-9, 5-16, 14-11
Administrator, 3-14
adding users, 3-15
auditing usage, 3-17
deleting users, 3-16
editing users, 3-16
Help menu options, 3-14
logging in, 3-14
Logout menu option, 3-14
restoring users, 3-16
starting, 3-14
Usage Audit tab, 3-15
Users tab, 3-15
administrator
changing user password, 13-6
resetting user passwords, 13-6
Administrator Access
enabling, 3-16
Advance to Next Record setting, 5-19
Advanced Ramp-up options, 6-2
Advanced settings, 5-7, 5-15, 14-8
advantages, 1-2
After a delay of setting, 6-3, 6-4
After each user iterates setting, 6-4

After every setting, 6-3
Agent error
dropping failed agents, 9-4
stopping rampup, 9-4
Agent error handling setting, 9-4
Agent Preferences settings, 5-7, 5-15, 14-8
Agent String, 14-3
agents
configuring, 3-5
error handling, 9-4
installing, 3-4
terminating, 9-4
verifying network access, 3-5
AIX, 8-2
AIX 4.3 virtual agent
metric descriptions, D-1
all of the following conditions are met setting, 7-1
all of these goals are met setting, 7-1
All users in scenario setting, 5-4, 5-22, 14-5
analysis reports
generating, 2-11
any of the following conditions are met setting, 7-1
any of these goals are met setting, 7-1
Apache, 8-2
Apache Web Server
metric descriptions, D-1
Applet Resource setting, 5-12
application servers, 8-1
application service messages, B-4
Application stability
defining test process, 2-6
Architecture Validation, 2-2
Attribute Name setting, 8-6, 8-14
authentication
URL configuration settings, 8-8
URL metric settings, 8-17
Autopilot
advanced ramp-up settings, 6-2
pausing, 9-4
setting rampup, 6-1, 6-2
using, 9-1
Average Active Sessions graph, 10-7, 11-5

B

basics, 3-1

- Binary Decoding Error setting, 5-10, 5-16, 14-11
- Browser Additional Arguments setting, 5-13, 14-5
- Browser Emulation setting, 5-5, 14-5
- Browser options
 - setting, 14-3
- Browser Path Override setting, 5-13, 14-5
- Browser Settings, 5-4, 5-5, 5-13, 14-4, 14-5
- Browser Type setting, 5-13, 14-5
- Browsers to simulate
 - defining test criteria, 2-7

C

- Cache and Cookies settings, 5-18
- Cache Download Pages setting, 5-5, 14-5
- Calculated Percentile option, 11-11
- Call Function Failed setting, 5-11, 5-16, 14-11
- Cannot Open Scenario, B-6
- Capture Message Details setting, 14-13
- Categories of users
 - defining test criteria, 2-7
- Child Script Failed setting, 5-11, 5-16, 14-11
- Cisco network devices, 8-1
 - metric descriptions, D-1
- Clear Cache After Iteration setting, 5-5, 14-6
- Clear Cache Before Playback setting, 5-18, 14-13
- Clear Cache Between Iterations setting, 5-18, 14-13
- Clear Graph button, 10-9, 11-5
- Clear Persistent Cookies Before Playback
 - setting, 5-18, 14-13, 14-14
- Clear Session Cookies Before Playback setting, 5-18, 14-13
- Clear Session Cookies Between Iterations
 - setting, 5-18, 14-13
- Client Certificate Keystore Error setting, 5-11, 14-12
- COM+
 - adding as metrics, 8-13
 - configuring as data source, 8-6
- Command Line setting, 8-8, 8-17
- command line settings, C-1
 - AddAgent, C-5, C-7
 - AddAgentGroup, C-6
 - commands, C-2
 - export/import, C-6
 - General, C-5
 - proxy settings, C-2
 - Run, C-7
 - specifying, C-1
 - status, C-11
 - stop, C-12
- Command Prompt setting, 4-10
- Composer tab, 3-17
- configurations
 - adding data source monitors, 8-4, 8-5
 - COM+, 8-6
 - creating new files, 8-3
 - database, 8-6
 - deleting, 8-4
 - editing, 8-3, 8-9
 - IBM WebSphere PMI, 8-9

- JMX, 8-6
- Perfmon (Windows Performance Monitor), 8-6
- Ping, 8-7
- URL, 8-7
 - using, 8-2
 - virtual agent, 8-8
- Connect String setting, 4-9
- Connection Idle Timeout setting, 5-7, 14-8
- Connection Speed Emulation setting, 5-5, 14-5
- Counter ID setting, 8-6, 8-13
- Counter Name setting, 8-6, 8-13, 8-15
- Counter property
 - IBM WebSphere PMI, 8-9
 - Perfmon (Windows Performance Monitor), 8-6
- Counter setting
 - IBM WebSphere PMI, 8-18
- Counters property
 - IBM WebSphere PMI, 8-9
- Create Variable Fail setting, 5-10, 5-16, 14-11
- Criteria for Each Business Transactions
 - defining test criteria, 2-7
- CSS Resource setting, 5-11
- CSV data
 - exporting, 11-8
- Custom Browsers, 14-3
- Custom Cookie setting, 8-8, 8-16
- Custom Graphs setting, 11-7

D

- Data Bank settings, 5-4, 14-4
- Data collection interval setting, 11-6
- data collectors
 - adding, 4-4
 - startup errors, B-9
- Data Driven Tests, 2-8
- Data Series list, 10-8
- Data Series Name setting, 10-8
- Data Sources, 4-8
- data sources
 - adding to configurations, 8-4
 - COM+, 8-4, 8-6
 - Database, 8-4, 8-6
 - IBM WebSphere PMI, 8-5, 8-9
 - JMX, 8-4, 8-6
 - Perfmon (Windows Performance Monitor), 8-4, 8-6
 - Ping, 8-4, 8-7
 - URL, 8-4, 8-7
 - Virtual Agent, 8-5, 8-8
- Databank Configuration settings, 5-17
- Databank Setup Timeout setting, 5-4, 5-17, 14-1
- databanks
 - advance records settings, 5-19
 - configuring, 5-19
 - next record settings, 5-20
 - out of records settings, 5-20
 - using the control, 5-19
- Database
 - adding as metrics, 8-13

- configuring as data source, 8-6
- supported metric profiles, 8-1
- Database Name or Database SID setting, 4-9
- Database Name setting, 4-9
- databases
 - configuring existing, 3-7
 - deleting, 13-2
 - exporting and importing, 13-3
 - managing, 13-1
 - migrating data, 13-3
 - removing, 13-2
 - selecting, 13-2
 - transferring schemas, 3-7
 - updating, 13-3
- DB Throughput graph, 10-7, 11-5
- Default rows per page setting, 11-7
- Debug Users setting, 5-7, 14-8
- default administrator account, 3-2, 3-14
- default metrics, 8-1
- Default rows per page setting, 11-7
- defining systems, 4-2
- Delay Between Business Transaction Runs, 2-7, 2-9
- Deployment environment
 - defining test process, 2-6
- Development Engineers and Architecture Groups, 2-5
- Discovery MBeans setting, 8-14
- disk performance
 - configuring for monitoring, 8-5
- Diskperf
 - configuring, 8-5
 - using, 8-19
- Dismiss Alerts Dialogs setting, 5-13, 14-5
- Do Not Request URLs ending in setting, 5-9, 14-10
- Download Local Files setting, 5-9, 14-10
- Download Manager settings, 5-11, 14-13
- Download Size in Bytes setting, 8-8, 8-16
- Download Time setting, 8-8, 8-16
- Driver setting, 4-9
- Driver String setting, 4-9
- Drop failed agents from session setting, 9-4

E

- Each Iteration of Script setting, 5-19
- Each Occurrence setting, 5-20
- Embedded Object Resource setting, 5-12
- Enable Cookies setting, 5-5, 14-6
- Enable Cookies settings, 8-7, 8-16
- Enable Deflate setting, 5-8, 14-9
- Enable GZIP setting, 5-8, 14-9
- Enable Keep Alive setting, 5-8, 14-9
- Enable Logging setting, 5-5, 5-13, 10-3, 14-6
- Enable Proxy setting, 8-17
- Encrypted Scripts
 - requirements, 3-3
 - specifying passwords, 5-2
- Encrypting/Decrypting Failed setting, 5-11, 5-16, 14-11
- Encryption Service not Initialized setting, 5-10, 5-16,

- 14-11
- End range filter, 10-8
- error codes, A-1
 - viewing, B-1
- Error Handling settings, 5-7, 5-15, 14-7
- error messages, B-1
 - application service, B-4
 - cannot open scenario, B-6
 - server not initialized, B-4
 - unable to start agent session, B-6
 - unable to start session, B-5
- Error Recovery settings
 - Forms Functional Test, 14-12
 - Functional test, 14-12
 - General, 5-10, 14-11
 - HTTP, 5-11, 14-11
 - Oracle Forms Load, 5-11
 - Web Functional test, 14-12
- Errors Vs. Time graph, 10-6, 11-3
- Errors Vs. Users graph, 10-5, 10-6, 11-3
- Event Timeout settings, 5-18, 14-14
- every setting, 6-1
- Every Visit setting, 5-5, 14-6
- Excel files
 - exporting, 11-9
- Execute User Defined Tests setting, 5-5, 14-6
- Export CSV graph, 10-7, 10-8
- Export CSV graph data, 11-5
- Export Excel HTML graph, 10-8
- Export graph option, 10-7
- Export Image graph button, 11-5
- Export PNG graph, 10-7, 11-5
- Export SVG graph, 11-5
- Export XLS graph button, 11-5
- exporting graphs, 11-8
- Extensibility settings, 5-5, 14-6

F

- Fail on Missing Frames setting, 14-7
- Fail on Object Download Errors setting, 14-7
- failed to find error messages, B-2
- failed to match error messages, B-2
- failed to open error messages, B-3
- Failure codes, A-2
- features, 1-1
- file management, 13-1
- File Not Found setting, 5-10, 5-16, 14-11
- Filter Counters, 10-6
- Filter MBeans By setting, 8-14
- Filter option, 11-1
- Firefox browser, 3-3
- firewall considerations, 3-3
- Forms Action Timeout setting, 5-18, 14-14
- Forms Component not Found setting, 5-11, 14-12
- Forms Connect Error setting, 5-11, 14-12
- Forms Content Match Failed setting, 5-11, 14-12
- Forms I/O Communication Error setting, 5-11, 14-12
- Forms LT Playback settings, 5-12, 14-13
- Forms Playback Error setting, 5-11, 14-12

- Forms Response Timeout setting, 5-18, 14-14
- Forms Startup Timeout setting, 5-18, 14-14
- Foundry Load Balancer, 8-1
 - metric descriptions, D-2
- FTP API Error codes, A-8
- functional test scripts
 - default settings, 14-3
 - preconditions, 3-2
 - scenario settings, 5-12

G

- General options settings, 14-1
- General Web Server, 8-2
 - metric descriptions, D-2
- Generate option, 11-1
- Generate Timers For Pages setting, 5-6, 14-7
- Generate Timers For Resources setting, 5-7, 5-15, 14-7
- Generate Timers For Step Groups setting, 5-6, 5-15, 14-7
- GET setting, 8-7, 8-16
- Global Headers setting, 5-9, 14-10
- Goal # settings, 7-1
- Goal is met when setting, 7-1
- Gopher API Error codes, A-8
- Graph Options button, 10-5
- Graph refresh interval setting, 3-22, 11-7
- Graph Title setting, 10-8
- graphs
 - Errors Vs. Time graph, 10-6
 - Errors Vs. Users graph, 10-5
 - exporting data, 11-8
 - exporting images, 11-8
 - exporting to CSV file, 11-1
 - exporting to Excel, 11-9
 - exporting to PNG, 11-9
 - load test, 10-4
 - Oracle Database Diagnostics, 10-7
 - Performance Vs. Time graph, 10-6, 11-3
 - Performance Vs. Users graph, 10-5, 11-3
 - setting maximum data series, 11-7
 - setting options, 10-5
 - setting refresh interval, 10-4
 - showing chart statistics, 10-5
 - Statistics Vs. Time graph, 10-6, 11-3
 - Trans Failed Vs. Users graph, 11-3
 - Trans Failed/sec Vs. Time graph, 11-3
 - Users Vs. Time graph, 10-6, 11-4
 - using, 11-1
 - working with queries, 11-9
- Group type setting, 5-4, 5-21, 14-5

H

- Hardware
 - estimating, 4-11
- Hardware Allocation
 - defining test process, 2-6
- Heart Beat Interval setting, 5-12, 14-13

- Hits Per Second statistics, 11-2
- Hits/sec counter, 11-4
- Hits/sec, Users vs. Time graph, 11-4
- Host CPU Utilization graph, 10-7, 11-5
- Host PGA Utilization graph, 10-7, 11-5
- Host Resources MIB network devices
 - metric descriptions, D-2
- Host-Resources-MIB network devices, 8-1
- HP-UX, 8-2
- HTML Parsing Error setting, 5-11, 14-11
- HTML Test Failed setting, 5-17, 14-12
- HTTP
 - defining test criteria, 2-7
- HTTP API Error codes, A-9
- HTTP Version setting, 5-8, 14-9

I

- IBM WebSphere PMI
 - adding as metrics, 8-18
 - configuring as data source, 8-9
- Ignore HTTP Proxy Settings setting, 5-7, 14-8
- Ignored URLs that Match Regexp setting, 14-10
- IIS Web Server, 8-2
 - metric descriptions, D-3
- Image Resource setting, 5-11
- images
 - using in tests, 2-9
- Include image download times setting, 8-8, 8-17
- Include object download times setting, 8-8, 8-17
- Information codes, A-1
- installing, 3-1
 - agents, 3-4
 - Linux agent, 3-6
 - Oracle Load Testing, 3-1
- Instance Name setting, 8-9, 8-15, 8-17
- Instance property
 - Perfmon (Windows Performance Monitor), 8-6
- Instance setting, 4-9
- Integration and Acceptance Organizations, 2-5
- Internet codes, A-4
- Internet error codes, A-4
- Internet Explorer browser, 3-3
- Internet Status codes, A-1
- introduction, 1-1, 2-1
- Invalid HTTP Response setting, 5-11, 14-11
- Invalid URL setting, 5-11, 14-11
- IP spoofing
 - using, 5-22
- Iterate Over Range setting, 5-19
- Iteration Delay setting, 14-4

J

- JMX
 - adding as metrics, 8-14
 - configuring as data source, 8-6
- JMX monitors
 - configuring, 4-5
- JVM Heap settings

- changing, 3-11
- guidelines, 3-12
- limitations, 3-13
- modifying on Linux, 3-12
- modifying on Windows, 3-12

K

- KB rcvd/sec counter, 11-4
- KBs/sec, Users vs. Time graph, 11-4
- Keep the Same Record setting, 5-20
- Key of value in Regexp setting, 8-8, 8-17
- Key Store File Name setting, 4-9
- Key Store Password setting, 4-10
- Kilobytes Per Second statistics, 11-2

L

- Language setting, 5-8, 14-9
- Length setting, 8-7, 8-15
- Linear scale setting, 10-9
- Linux agent
 - installing, 3-6
- Linux metric, 8-2
- Linux RH 7.1 virtual agent
 - metric descriptions, D-3
- Load Distribution Capability, 2-4
- Load Query graph button, 10-7, 11-5
- load test scenarios
 - creating, 2-9
 - planning, 2-8
 - verifying, 2-9
- load testing
 - criteria for, 2-7
 - goals of, 2-1
 - phases of, 2-2
 - planning for, 2-1
 - problems, B-6
 - process, 2-6
- Load to be Simulated
 - defining test criteria, 2-7
- Logarithmic scale setting, 10-9
- Logged Messages setting, 5-6, 5-14, 10-4, 14-6
- Logging Tools, 2-4
- Login, 3-6
- Loop Over Range setting, 5-20

M

- Main Window
 - Composer, 3-17
 - features, 3-17
- Managing Systems, 4-1
- Matching Regexp setting, 8-8, 8-17
- Max. Cache Size (MB) setting, 5-5, 14-6
- Max Content Download Size setting, 5-9, 14-10
- Max HTTP Connections Per User setting, 5-7, 14-8
- Max Keep-Alive Requests setting, 5-9
- Max Number of Keep Alive Requests setting, 14-10
- Max Users Per Process setting, 5-7, 14-8
- Maximum JVM Heap Size (MB) setting, 5-8, 5-15,

- 14-8
- MBean Name setting, 8-6, 8-14
- Measure Query Time setting, 8-6, 8-14
- Measurement setting, 8-7, 8-15
- Memory
 - determining requirements, 2-4
 - metric descriptions, D-3
- Menu options
 - Abort, 10-3
 - Modify Run Attributes, 10-3
 - overview, 3-17
 - Release Sync, 10-3
 - Stop, 10-3
- Message Delivery setting, 5-5, 5-14, 10-3, 14-6
- Metric Description setting, 8-14
- metric monitors
 - creating, 8-9
- Metric Profile List, 8-10
- metric profiles
 - adding, 8-10
 - cloning, 8-11
 - deleting, 8-12
 - editing, 8-11
 - overview of, 8-1
 - testing, 8-12
 - using metrics to create, 8-9
 - working with, 8-10
- Metric Type settings, 8-8, 8-16
- metrics
 - adding, 8-12
 - AIX 4.3 virtual agent, D-1
 - Apache Web Server, D-1
 - Cisco network devices, D-1
 - cloning, 8-18
 - COM+, 8-13
 - database, 8-13
 - deleting, 8-19
 - editing, 8-18
 - Foundry Load Balancer, D-2
 - Host, D-2
 - Host Resources MIB network devices, D-2
 - IBM WebSphere, 8-18
 - IIS Web Server, D-3
 - JMX, 8-14
 - Linux virtual agent, D-3
 - Memory, D-3
 - MIB-II network devices, D-4
 - Netscreen firewall, D-4
 - network devices, D-5
 - Oracle database, D-5
 - overview of, 8-1
 - Perfmon, 8-14
 - Ping, 8-15
 - Solaris (multi-CPU), D-6
 - Solaris virtual agent, D-6
 - SQL Server database, D-6
 - testing, 8-19
 - URL, 8-15
 - viewing, 8-19
 - Virtual Agent, 8-17

- WebLogic, D-7
 - working with, 8-12
- MIB-II network devices, 8-1
 - metric descriptions, D-4
- migrate data between databases, 13-3
- Modify Run attributes, 10-3
- monitored systems, 4-2
 - adding, 4-5, 4-7
- Monitoring and Operations Groups, 2-5
- monitors
 - adding, 8-5
 - editing, 8-9
 - overview of creating, 8-2
 - using metric profiles, 8-9

N

- Netscreen Firewall, 8-1
 - metric descriptions, D-4
- network devices, 8-1
 - metric descriptions, D-5
- New Scenario toolbar button, 3-17
- No Delay setting, 5-4, 5-13, 14-4
- No Translation setting
 - Database configuration, 8-5, 8-13
- Node setting, 8-9, 8-18
- Non Proxy Hosts setting, 5-8, 14-9
- Number of Business Transactions to Simulate
 - defining test criteria, 2-7
- Number of Users, 2-9
- Number of VUs counter, 11-4

O

- Object ID setting, 8-6, 8-13
- Object Name setting, 8-6, 8-13, 8-15
- Object Not Found setting, 5-17, 14-12
- Object property
 - Perfmon (Windows Performance Monitor), 8-6
- Object Test Failed setting, 5-17, 14-12
- Object Test setting, 5-15, 14-7
- Object Timeout setting, 5-15, 14-8
- On Error Stop Virtual User setting, 5-7, 5-15, 14-7
- Open gate time setting, 5-5, 5-22, 14-5
- Open Scenario toolbar button, 3-17
- OpenScript Error Recovery settings, 5-10
 - Cache and Cookies, 14-13
 - Forms Load Test, 14-12
 - Functional test, 5-17
 - General, 5-16
 - Oracle Forms Functional test, 5-17
 - Web Functional test, 5-17
- Operating System
 - determining requirements, 2-4
- Operating System setting, 4-10
- Options
 - Ramp Defaults, 9-1
- options
 - Browser, 14-3
 - general, 14-1

- reporting, 11-6, 14-14
- repository, 14-1
- scenario defaults, 14-3
- session start and stop, 14-14
- setting, 14-3
- Options graph button, 10-8, 11-5
- Oracle Application Testing Suite Application Service
 - stopping, 3-14
- Oracle database, 8-1
 - metric descriptions, D-5
- Oracle Database setting, 4-8
- Oracle Forms Error setting, 5-17, 14-12
- Oracle Functional Testing, 2-3
- Oracle Load Testing, 2-3
 - advantages of, 1-2
- Oracle Load Testing Agent Authentication
 - Manager, 3-13
- Oracle Load Testing Agent Service, 3-5
- Oracle Load Testing ServerStats, 2-4
- Oracle SELECT statement, 8-6
- Oracle Thin JDBC Driver, 4-8
- Oracle WebLogic Server application server, 8-1
- Other System Monitoring Tools, 2-4
- Overall Transactions-Per-Second Throughput
 - Required, 2-8

P

- Pacing mode, 2-9
 - defining test criteria, 2-7
- Pages
 - returned by server, 11-2
- Pages Per Second statistics, 11-2
- Pages/sec, Users vs. Time graph, 11-4
- Password setting, 4-9
 - URL Configuration, 8-8
- passwords
 - resetting administrator user, 13-6
- Perfmon (Windows Performance Monitor)
 - adding as metric, 8-14
 - configuring as data source, 8-6
- Perfmon authentication, 4-10
- Performance Benchmarking, 2-2
- Performance Data Logging, 2-8
- Performance Monitoring, 2-2
- Performance Problems During the Load Test, B-7
- Performance Regression, 2-2
- Performance Vs. Time graph, 11-3
- Performance Vs. Users graph, 10-5, 11-3
- Performance, Users vs. Time graph, 11-4
- Perl Expression setting, 8-17
- Persist Raw Data setting, 5-7, 5-15, 14-8
- Personnel
 - defining test process, 2-6
- Ping
 - adding as metric, 8-15
 - configuring as data source, 8-7
- Playback Failed setting, 5-17, 14-12
- PNG files
 - exporting, 11-9

- popup blockers, 3-3
- port
 - changing, 3-10
- Port setting, 4-9
- Post Data setting, 8-7, 8-16
- POST setting, 8-7, 8-16
- preferences
 - setting, 14-1, 14-3
- Preserve Connections Between Iterations
 - setting, 5-8, 14-9
- Preserve Cookies Between Iterations setting, 5-9, 14-10
- Preserve Variables Between Iterations setting, 5-9, 5-15, 14-9
- Processor
 - determining requirements, 2-4
- profiles
 - adding, 5-1
 - removing, 5-2
- Prompt for session name before running setting, 9-4
- Proxy authentication settings, 8-8, 8-17
- Proxy Host setting, 5-8, 8-8, 8-17, 14-8
- Proxy Port setting, 5-8, 8-8, 8-17, 14-9

Q

- Quality Assurance Organizations, 2-5
- queries, 11-9
 - deleting, 11-10
 - editing, 11-10
 - running saved queries, 11-10
 - saving, 11-9
- Query setting, 8-6, 8-13

R

- Ramp Rate setting, 6-1
- Rampup Specification
 - advanced, 6-2
- Random setting, 5-4, 5-13, 14-4
- Randomly setting, 5-20
- Range Filter settings, 10-8
- Rate (minutes) setting
 - Database configuration, 8-5, 8-13
- Raw data
 - enabling collection, 11-11
 - using, 11-11
 - working with data files, 11-11
- Read Result Set setting, 8-6, 8-14
- Read Timeout setting, 5-4, 5-18, 14-4
- Recorded setting, 5-3, 5-13, 14-4
- Recorded/Random setting, 14-4
- Recorded/Rnd setting, 5-3, 5-13
- Redirection codes, A-2
- Refresh button, 10-8, 11-5
- Relative Time setting, 10-8
- Release trigger setting, 5-4, 5-22, 14-5
- Remote Agent setup, 3-4
- Remote Password setting, 4-10
- Remote Port setting, 4-10

- Remote Protocol setting, 4-10
- Remote Username setting, 4-10
- Replace URLs setting, 5-9, 5-15, 14-10
- Report Counters setting, 5-7, 5-15, 14-8
- Report Sender Interval setting, 5-8, 5-15, 14-8
- Report Type settings, 8-14
- Reporting options, 14-14
- Reporting settings, 14-7
- reports
 - exporting graph files for Excel, 11-9
 - exporting to CSV data, 11-8
 - exporting to PNG files, 11-9
 - generating, 2-11
 - including think time, 11-7
 - performance statistics, 11-1
 - setting options, 5-6, 5-15, 11-6
 - using, 11-1
- Reports tab, 3-22
- repositories
 - adding, 3-9
- repository options, 14-1
- Request Timeout setting, 5-7, 14-8
- Required Applications
 - defining test process, 2-6
- Reset Password utility
 - using, 13-6
- Resolution Size setting, 5-13, 14-5
- Response Time Error setting, 5-11, 5-17, 14-11, 14-12
- Result of Perl Expression setting, 8-8, 8-16
- Result Pattern setting, 8-6, 8-14
- RSWISERV errors, B-3
- Rules of Thumb, 2-4
- Run Now toolbar button, 3-17
- Run Test button, 9-3
- Running users setting, 5-4, 5-22, 14-5
- Run-Time Data, 2-8
- runtime reports
 - viewing, 10-4

S

- Sample Multiplier setting
 - Virtual Agent configuration, 8-5, 8-13
- Save Query graph button, 10-7, 11-5
- Save Scenario toolbar button, 3-17
- scalability tests
 - planning, 2-8
- Scenario Default Properties options, 14-3
- scenarios
 - adding profiles, 5-1
 - advanced ramp-up, 6-2
 - deleting, 9-2
 - estimating hardware, 4-11
 - opening existing, 9-2
 - renaming, 9-2
 - running, 9-3
 - running from command line, 9-5
 - saving, 9-1
 - specifying profile properties, 5-2
 - using synchronization points, 5-21

- working with files, 9-1
- Schedule tab, 3-22, 9-3
- Scheduling
 - defining test process, 2-6
- Script Resource setting, 5-12
- scripts
 - advanced ramp-up, 6-2
 - creating, 2-9
- Secure Protocol setting, 5-9, 14-10
- Segment Parser Error setting, 5-10, 5-16, 14-11
- Select Next Record setting, 5-20
- SELECT statement, 8-6
- Sequentially setting, 5-20
- Server Error codes, A-4
- Server not initialized, B-4, B-5
- Server setting, 8-9
- Server Status codes, A-1
- Server Type setting, 4-9
- ServerStats
 - adding data sources, 8-4
 - adding metric profiles, 8-10
 - adding metrics, 8-12
 - configuring, 8-2
 - data collectors, 4-1
 - metric descriptions, D-1
 - overview of, 8-1
 - viewing status, 10-9
 - working with metrics, 8-12
- ServerStats Monitors options, 14-1
- Session Name Prefix setting, 9-4
- Session Start and Stop options, 9-4, 14-14
- Sessions
 - starting, 9-3
 - starting and stopping manually, 9-3
- sessions
 - attaching to, 10-10
 - defining how to end, 9-4
 - detaching from, 10-10
 - exporting data, 12-2
 - importing data, 12-3
 - managing, 9-4, 12-1
 - naming, 9-4
 - viewing performance statistics, 11-2
 - working with, 12-1
- Setting Options, 14-1
- Setup Timeout setting, 14-4
- Show All Counters option, 10-6
- Show Average Performance Only option, 10-6
- Show Legend button, 10-8, 11-5
- Shuffle setting, 5-20
- SID setting, 13-2
- Socket Timeout setting, 5-7, 14-8
- Solaris (multi-CPU) virtual agent, D-6
- Solaris metric, 8-2
- Solaris virtual agent
 - metric descriptions, D-6
- Solve Variable Fail setting, 5-11, 14-11
- Solve Variable Failed setting, 5-17, 14-12
- Specify (URL) setting, 8-7, 8-15
- SQL Database setting, 4-8
- SQL Server, 8-1
- SQL Server database
 - metric descriptions, D-6
- SSL
 - defining test criteria, 2-7
 - using, 3-11
- Start range filter, 10-8
- Statistics Vs. Time graph, 10-6, 11-3
- Status Bar Test Error setting, 5-17, 14-12
- Stop All Virtual Users toolbar button, 3-17
- Stop ramp-up on agent error setting, 9-4
- Stop Remaining Iterations on Failure setting, 5-7, 5-15, 14-8
- Stop session on last VU completion setting, 9-4
- Stop session when browser is closed setting, 9-4
- Stop the User setting, 5-20
- Sub Attribute Name setting, 8-6, 8-14
- Success codes, A-1
- Sun JDBC
 - ODBC Bridge Driver, 4-8
- support files, B-4
 - creating, 3-13
- supported metric profiles, 8-1
- Synchronization points
 - naming, 5-4, 5-21, 14-5
- synchronization points
 - adding to profiles, 5-21
 - using, 5-21
- System Identifier setting, 13-2
- System Monitoring Tools, 2-4
- system requirements
 - functional test scripts, 3-2
- systems
 - adding new groups, 4-3
 - adding to groups, 4-4
 - defining, 4-2
 - deleting, 4-11
 - editing, 4-11
 - renaming, 4-10
- Systems Manager, 4-1
- Systems Manager options, 14-1

T

- Table Test Failed setting, 5-17, 14-12
- Table Test setting, 5-15, 14-7
- TCP/IP Properties, 5-22
- Terminate all agents at end of session setting, 9-4
- test execution
 - verifying on multiple machines, 2-10
- Test Goals
 - setting, 7-1
- test results
 - evaluating, 2-11
- Testing Environment
 - defining test process, 2-6
- tests
 - executing, 2-10
- Text Match Fail setting, 5-11, 14-11
- Text Matching Failed setting, 5-17, 14-12

- Threshold Line settings, 10-9
- Throughput Required, 2-8
- Time Range Filters, 11-10
- Time to Live setting, 8-7
- Title Test Failed setting, 5-17, 14-12
- TLSv1.2, 5-9, 14-10
- Toggle Legend button, 10-5
- Toggle Stats button, 10-5
- Toggle Stats graph button, 10-8, 11-5
- Toolbar, 3-17
- Tools menu, 3-13
- Trans Failed/sec counter, 11-4
- Transactions Per Second statistics, 11-2
- Transactions statistics, 11-2
- Transactions to Perform, 2-8
- Transactions with Errors statistics, 11-2
- Translation Type settings
 - Database configuration, 8-5, 8-13
- Trans/sec counter, 11-4
- Trans/sec, Users vs. Time graph, 11-4
- Trust Store File Name setting, 4-9
- Trust Store Password setting, 4-9
- Type of Error Handling, 2-8
- Type of Transaction, 2-8
- Type of User, 2-8
- Types of Business Transactions to be Simulated
 - defining test criteria, 2-7

U

- Unable to Start Agent Session, B-6
- Unexpected Agent Exit in the VU Grid, B-8
- Unexpected Script Error setting, 5-11, 5-16, 14-11
- URL
 - adding as metrics, 8-15
 - configuring as data source, 8-7
 - specifying authentication, 8-8
 - specifying for configuration, 8-7
- URL Method settings, 8-7, 8-16
- URL Usage settings, 8-7, 8-15
- Use Data Bank setting, 5-4, 5-13, 14-4
- Use HTTP Proxy setting, 8-8
- Use IP Spoofing setting, 5-5, 5-22, 14-6
- Use OpenScript Download Manager setting, 5-11
- Use System Homepage setting, 8-7, 8-15
- User Agent setting, 8-7, 8-16
- Username setting, 4-9
 - URL configurations, 8-8
- users
 - adding, 3-15
 - deleting, 3-16, 3-17
 - editing, 3-16
 - number of, 2-9
 - restoring, 3-16
 - types of, 2-8
- Users setting, 10-8
- Users Vs. Time graph, 10-6, 11-4

V

- Validate hostname automatically when
 - adding/editing a system setting, 14-1
- Validate monitor automatically when creating a
 - monitor setting, 14-1
- Variable Not Found setting, 5-10, 5-16, 14-11
- Virtual Agent
 - adding as metric, 8-17
 - configuring as data source, 8-8
 - specifying command line execution, 8-8
- Virtual Agent authentication, 4-10
- Virtual user (VU) ramp-up, 6-1
- Virtual User scenarios
 - defining, 5-1, 6-1, 7-1
 - specifying profile properties, 5-2
- Virtual Users
 - controlling, 10-3
 - determining how many, 5-18
 - modifying run attributes, 10-3
 - pacing, 3-19
 - pausing, 9-4
 - selecting scenarios, 5-1
 - specifying how many, 6-1
 - specifying rampup, 6-1
 - stopping all, 9-4
 - using the display, 10-1
 - using the grid, 10-2
 - verifying multiple, 2-9
 - with Errors statistics, 11-2
- VU Agent System Groups, 4-1
- VU Agent Systems, 4-1
 - adding, 4-2
 - changing port settings, 3-4
- VU Logs
 - exit codes, B-8
 - setting options, 5-5, 5-13, 14-6
 - viewing, 10-9, 11-6
- VU Pacing (Think Time) setting, 5-3, 5-13, 14-4
- VUs with Errors counter, 11-4

W

- Wait for Page Timeout setting, 5-17, 14-12
- web server port, 3-10
- Web-Based Interface, 1-2
- WebLogic, 4-5
- WebLogic metrics, D-7
- Weblogic patch, 3-13
- WebSphere, 4-5
- WebSphere server, 4-9
- When counter setting, 6-3, 6-4
- When goals are met, mark load session setting, 7-2
- When goals are not met, mark load session
 - setting, 7-2
- When Out of Records setting, 5-20
- When Page Out of Date setting, 5-5, 14-6
- When scenario starts setting, 6-3
- When scenario stop setting, 6-4
- When Script Requests Next Record setting, 5-19
- When script runs setting, 6-3, 6-4

When script stops setting, 6-3, 6-4
Windows metric, 8-2

X

X-Axis Scale settings, 10-8
XML Test Failed setting, 5-17, 14-12
XML Test setting, 5-15, 14-7

Y

Y-Axis Scale settings, 10-9

Z

Zero Downloads Fatal setting, 5-11, 14-11
Zoom tool, 10-8, 11-5