

Oracle® Load Testing

Load Testing User's Guide

Release 12.5.0.2

E15484-15

April 2015

Oracle Load Testing Load Testing User's Guide, Release 12.5.0.2

E15484-15

Copyright © 1997, 2015, 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	xii
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	1-3
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	Setting Up Servers for ServerStats.....	3-10
3.5.1	Solaris SNMP Server	3-10
3.5.1.1	Starting the SNMP Agent on Solaris 2.6/2.7	3-11
3.5.1.2	Stopping the SNMP Agent.....	3-11
3.5.1.3	Enabling SNMP Agent on Startup	3-11
3.5.2	Oracle SNMP Server.....	3-11
3.5.2.1	Configure Master Agent.....	3-12
3.5.2.2	Configure the Encapsulator	3-12
3.5.2.3	Verify the start_peer Script	3-12
3.5.2.4	Start the SNMP Components.....	3-12
3.6	Changing the Web Server Port.....	3-13
3.7	Using SSL	3-13
3.8	Changing the OLT Controller Heap Settings.....	3-14
3.8.1	Basic Guidelines for the Controller Heap Settings	3-14
3.8.2	Modifying the JVM Heap Settings On Windows Machines.....	3-14
3.8.3	Modifying the JVM Heap Settings On Linux Machines	3-15
3.8.4	Limitations.....	3-15
3.9	Oracle Application Testing Suite Tools Menu.....	3-16
3.10	Administrator	3-16
3.10.1	Menu Options.....	3-17
3.10.1.1	Tools Menu	3-17
3.10.1.2	Help Menu.....	3-17
3.10.1.3	Logout	3-17
3.10.2	Users Tab.....	3-17
3.10.3	Usage Audit Tab	3-18
3.10.4	Adding Users.....	3-18
3.10.5	Editing Users	3-19
3.10.6	Deleting Users	3-19
3.10.7	Restoring Users	3-19
3.10.8	Auditing Usage	3-19
3.11	Main Window Features.....	3-20
3.11.1	Overview of the Menu Options	3-20
3.11.1.1	Scenario Menu.....	3-20

3.11.1.2	Session Menu.....	3-20
3.11.1.3	ServerStats Menu.....	3-20
3.11.1.4	Tools Menu.....	3-21
3.11.1.5	Manage Menu.....	3-21
3.11.1.6	Help Menu.....	3-21
3.11.1.7	Logout.....	3-21
3.11.2	Toolbar.....	3-21
3.11.3	Build Scenarios Tab.....	3-22
3.11.4	Set Up Autopilot Tab.....	3-23
3.11.5	Watch VU Grid Tab.....	3-24
3.11.6	View Run Graphs Tab.....	3-24
3.11.7	Create Reports Tab.....	3-24
3.12	Systems Manager.....	3-24
3.13	Defining Systems.....	3-26
3.13.1	Adding New VU Agent Systems.....	3-26
3.13.2	Adding New System Groups.....	3-27
3.13.3	Adding Systems to Groups.....	3-27
3.13.4	Adding New ServerStats Data Collectors.....	3-28
3.13.5	Adding New Monitored Systems.....	3-29
3.13.5.1	Configuring JMX Monitors.....	3-29
3.13.5.2	Adding a Monitored System.....	3-31
3.13.6	Adding Cloud Service Configurations.....	3-38
3.13.6.1	Importing the Enterprise Manager Certificate.....	3-38
3.13.6.2	Adding the Cloud Service Configuration.....	3-39
3.13.7	Renaming Systems.....	3-41
3.13.8	Editing Systems.....	3-41
3.13.9	Deleting Systems.....	3-41
3.14	Setting Options.....	3-41
3.14.1	Custom Browser Options.....	3-42
3.14.2	Repository Options.....	3-42
3.14.3	Setting Scenario Defaults.....	3-42
3.14.4	Setting Autopilot Defaults.....	3-53
3.14.5	Setting Session Start and Stop Options.....	3-53
3.14.6	Setting Session Profile Options.....	3-53
3.14.7	Setting Reporting Options.....	3-53
3.14.8	Setting General Options.....	3-53

4 Defining Virtual User Scenarios

4.1	Defining Scenarios.....	4-1
4.1.1	Selecting Profiles.....	4-1
4.1.2	Specifying Scenario Profile Attributes.....	4-2
4.1.2.1	Load Test Script Settings.....	4-2
4.1.2.2	Functional Test Script Settings.....	4-11
4.1.3	Determining the Number of Virtual Users.....	4-17
4.1.4	Managing Sessions.....	4-18
4.1.5	Using the Data Bank Control.....	4-19
4.1.6	Using Synchronization Points.....	4-21

4.2	Using IP Spoofing	4-22
4.3	Working with Scenario Files	4-23
4.3.1	Saving Scenarios	4-23
4.3.2	Opening Existing Scenarios.....	4-24
4.3.3	Renaming Scenarios	4-24
4.3.4	Deleting Scenarios	4-25
4.3.5	Removing Profiles From a Scenario	4-25
4.3.6	Running Scenarios from the Command Line	4-25
4.3.7	Estimating Hardware	4-25
4.3.7.1	Estimate for an Oracle Load Testing Scenario	4-25
4.3.7.2	Generating Hardware Estimation Reports	4-26
4.4	Submitting Scenarios to Autopilot	4-26
4.4.1	Submit without Starting the Scenario	4-27
4.4.2	Submit and Start Scenario in Autopilot.....	4-27

5 Using the Autopilot

5.1	Specifying Start and Stop Settings.....	5-1
5.1.1	Specifying When to Start Virtual Users.....	5-1
5.1.2	Specifying When to Stop Virtual Users	5-2
5.2	Specifying Virtual User Rampup.....	5-2
5.2.1	Specifying How Many Virtual Users to Start at a Time.....	5-2
5.2.2	Examples	5-3
5.3	Manually Starting and Stopping the Autopilot.....	5-3
5.3.1	Starting the Autopilot	5-3
5.3.2	Stopping the Autopilot	5-4
5.4	Using the Advanced Scenario Ramp Options	5-4
5.5	Working with Virtual Users	5-6
5.5.1	Overview of the Virtual Users Grid	5-6
5.5.2	Controlling Individual Virtual Users.....	5-7
5.5.3	Managing Synchronization points	5-7
5.5.3.1	Releasing Individual Virtual Users.....	5-8
5.5.3.2	Releasing Multiple Virtual Users	5-8
5.5.3.3	Releasing Synchronization Points.....	5-8
5.5.4	Modifying Run Attributes	5-8
5.5.5	Viewing Server Warnings.....	5-9
5.5.6	Stopping All Virtual Users	5-9
5.5.7	Stopping the Autopilot From Starting New Virtual Users.....	5-9
5.5.8	Aborting All Virtual Users	5-9

6 Using Graphs and Reports

6.1	Setting the Reporting Options.....	6-2
6.2	View Run Graphs Tab	6-3
6.3	Using the Performance Statistics	6-5
6.4	Using Run Graphs	6-7
6.4.1	Viewing Runtime Statistics and Graphs.....	6-7
6.4.2	Performance Vs. Users Graph.....	6-9
6.4.3	Errors Vs. Users Graph	6-9

6.4.4	Performance Vs. Time Graph.....	6-9
6.4.5	Errors Vs. Time Graph	6-10
6.4.6	Users Vs. Time Graph	6-10
6.4.7	Users Vs. Time Graph	6-10
6.5	Using Data Counters	6-10
6.5.1	Using Data Counters	6-10
6.6	Create Reports Tab	6-11
6.6.1	Viewing Reports	6-13
6.6.2	Exporting Graphs and Data	6-13
6.6.3	Exporting CSV Data	6-14
6.6.4	Exporting to Excel.....	6-15
6.6.5	Exporting to PNG	6-15
6.6.6	Exporting to JPG	6-15
6.6.7	Working With Graph Queries.....	6-15
6.6.7.1	Saving Graph Queries.....	6-16
6.6.7.2	Running a Saved Query.....	6-16
6.6.7.3	Editing Queries	6-16
6.6.7.4	Deleting Queries	6-17
6.6.8	Creating Report Templates	6-17
6.6.8.1	Creating a Custom Graph and Session List.....	6-18
6.7	Generating Summary Reports.....	6-19
6.7.1	Scenario Reports	6-19
6.7.2	Session Reports	6-19
6.7.3	Hardware Estimation Reports	6-19
6.7.4	Running Automatic Database Diagnostic Monitor (ADDM) Reports.....	6-20
6.7.5	Running Automatic Workload Repository (AWR) Reports.....	6-20
6.7.6	Using Time Range Filters	6-21
6.8	Managing Session Data	6-22
6.9	Using Raw Data.....	6-23
6.9.1	Enabling Raw Data Collection	6-23
6.9.2	Working with Raw Data Files.....	6-23
6.9.3	Creating Tables in an Oracle Database	6-25
6.9.4	Importing Raw Data CSV Files	6-25
6.9.5	Example SQL Command File.....	6-25
6.9.6	Example Counter Definition Control File	6-26
6.9.7	Example Counter Data Control File.....	6-26

7 Using the Virtual User Logs

7.1	Starting the Virtual User Logs Viewer.....	7-1
7.2	Overview of the Main Window	7-1
7.2.1	Overview of the Toolbar Options.....	7-1
7.2.2	Overview of the Summary View	7-2
7.2.3	Overview of the Summary by VU View.....	7-3
7.2.4	Overview of the All Logs View	7-3
7.3	Opening VU Logs	7-3
7.4	Saving Log Files	7-4

8 Database and File Management

8.1	Database Management	8-1
8.1.1	Adding an Oracle Database	8-1
8.1.2	Selecting the Database to Use	8-2
8.1.3	Removing Databases	8-2
8.1.4	Updating Databases	8-3
8.1.5	Exporting and Importing Databases	8-3
8.1.5.1	Exporting Databases Using Export Utility	8-3
8.1.5.2	Importing Databases Using Import Utility	8-4
8.1.5.3	Importing Databases Using Data Pump Import Utility	8-4
8.2	Importing Files	8-5
8.3	Exporting Files	8-5

9 Working With Sessions

9.1	Attaching to a Session	9-1
9.2	Detaching from a Session	9-1
9.3	Exporting Session Data	9-2
9.3.1	File Format	9-2
9.4	Importing Session Data	9-2

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

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, "Defining Virtual User Scenarios"](#) Defining Virtual User Scenarios: explains how to define virtual user scenarios based upon the virtual user profiles.

[Chapter 5, "Using the Autopilot"](#) explains how to use the Autopilot to specify start and stop times for scenario profiles and the playback rampup specifications for each Virtual User.

[Chapter 6, "Using Graphs and Reports"](#) explains how to view real-time performance, statistics, and error information in graphs and reports.

[Chapter 7, "Using the Virtual User Logs"](#) explains how to use the virtual user logs to view virtual user actions during an Autopilot session.

[Chapter 8, "Database and File Management"](#) 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 9, "Working With Sessions"](#) explains how to attach and detach from running sessions.

[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. .

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 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 Load Testing Load Testing ServerStats 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.

Convention	Meaning
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.
- Integrates with Enterprise Manager 11g to analyze middleware performance diagnostics under load.

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.

Enterprise Manager 11g customers can also use Oracle Load Testing in conjunction with application middleware diagnostics in Enterprise Manager, to get deeper insight into application performance under load. Users can link from their load test sessions in Oracle Load Testing to middleware targets in Enterprise Manager 11g so they can analyze J2EE performance diagnostics during their load test.

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 Netscape;
- 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 Netscape Web server, a ColdFusion application server, a Tuxedo 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 Netscape (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 Netscape.

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 Autopilot 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 Create 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 3.10, "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'(3.6,6,10).
- 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 `wrapper.app.parameter.2` setting in the `<installdir>\agentmanager\bin\AgentManagerService.conf` 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 **Systems** from the **Manage** menu then selecting **VU Agent Systems**. 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 `AgentManagerService.conf` 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 **Systems** from the **Manage** menu.

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 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 the Autopilot.

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 **Systems Manager**.

When you define your Oracle Load Testing 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 the Autopilot.

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\10.2.0\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=oatsbtkp_x.x.dmp logfile=oatsbtkpexp_x.x.log
```

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

```
expdp system/<passwd>@xe schemas=OTM directory=bpk_dir
dumpfile=otmbtkp_x.x.dmp logfile=otmbtkpexp_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=oatsbtkp_x.x.dmp logfile=oatsbtkpimp_x.x.log
```

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

```
impdp system/<passwd>@xe schemas=OTM directory=bpk_dir
dumpfile=otmbtkp_x.x.dmp logfile=otmbtkpimp_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 **Tools** menu.
2. Select **Repositories** in 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 Setting Up Servers for ServerStats

Before ServerStats can monitor server-side statistics, the server(s) that you plan to monitor must be configured so that the ServerStats client has remote access to the server(s).

This section explains the server-side requirements for ServerStats remote access.

3.5.1 Solaris SNMP Server

The ServerStats Solaris SNMP client gathers performance statistics from a Solaris SNMP agent. ServerStats uses Sun Microsystems' proprietary SNMP extensions to report on the overall status of a machine and its individual processes.

The Solaris SNMP agent is installed and enabled by default on versions 2.6 and above of Solaris and is officially part of the *Solstice Enterprise Agents* suite:

<http://www.sun.com/software/entagents/>.

3.5.1.1 Starting the SNMP Agent on Solaris 2.6/2.7

To start the SNMP agent on Solaris:

1. Verify the Solaris SNMP agent is installed and that you have the following files and directories.

Files and Directories	Description
/usr/lib/snmp/snmpdx	Sun Solstice Enterprise Master Agent
/usr/lib/snmp/mibiisa	Sun SNMP Agent
/etc/init.d/init.snmpdx	Initialization script
/etc/snmp/conf	configuration files directory
/var/snmp/mib/snmpdx.mib	Master agent MIB file
/var/snmp/mib/sun.mib	Sun agent MIB file

The above directory and file locations apply to the default installation of Solaris and SNMP. Refer to the Solaris installation documentation for additional information about installing Solstice Enterprise Agents suite.

2. Start the Solaris SNMP agent.

If the snmpdx process is not already running, execute the following command to start the Solaris SNMP agent:

```
/etc/init.d/init.snmpdx start
```

Verify that snmpdx and mibiisa are in the process list.

3.5.1.2 Stopping the SNMP Agent

To stop the SNMP server at any time, execute the following command:

```
/etc/init.d/init.snmpdx stop
```

3.5.1.3 Enabling SNMP Agent on Startup

To enable SNMP at startup, make a new entry in the `/etc/rc*.d` directories, or run the startup command from your local initialization script.

Refer to the Solaris documentation for more information on installing, configuring and running the SNMP agent. Online information can be found at:

<http://www.sun.com/software/entagents/docs.html>

3.5.2 Oracle SNMP Server

The ServerStats Oracle SNMP client gathers performance statistics from the Oracle Enterprise Manager Intelligent Agent. The Oracle Intelligent Agent uses the Simple Network Management Protocol (SNMP). The agent exposes an extensive set of statistics through SNMP. The ServerStats Oracle SNMP client reports on statistics relating to I/O, load, and query activity.

You must configure Oracle SNMP support before starting the Intelligent Agent. Note that all configuration files for the following steps are located in the `$ORACLE_HOME/network/snmp/peer` directory.

This section provides basic instructions for configuring Oracle SNMP access. Additional documentation for the Oracle Enterprise Manager Intelligent Agent is located on the Oracle Documentation CD in the Server section (see section 4-19 of the

Oracle Installation Guide). The installation documentation for the specific platforms describe how to start the agent on Windows and Linux.

3.5.2.1 Configure Master Agent

In the CONFIG.master file, make the following change:

Search for the line beginning with MANAGER.

Change the `ipaddr` field, coded as 130.35.10.210, to the hostname or IP address of the Oracle server to monitor using ServerStats.

You can also make other changes to the CONFIG.master file, as documented within the file.

3.5.2.2 Configure the Encapsulator

Add the following line to the snmpd.conf file:

```
trap hostname_or_IP_address
```

where `hostname_or_IP_address` represents the hostname or IP address of the Oracle server to monitor using ServerStats.

In the CONFIG.encap file, you can optionally modify the port number, which is set to 161 in the default file. If you modify the port number, you must also modify the port number for NEW_SNMPD_PORT in the `start_peer` script.

NEW_SNMPD_PORT is the port on which the `snmpd` agent (the native Solaris 2.x SNMP agent) listens. Make sure this is the same port as specified in the CONFIG.encap file. NEW_TRAPD_PORT is the PEER encapsulator port to which the `snmpd` agent sends traps.

NEW_SNMPD_PORT and NEW_TRAPD_PORT in the `start_peer` script must have different port numbers. You may also modify the NEW_TRAPD_PORT port number.

3.5.2.3 Verify the start_peer Script

The `start_peer` script contains a line like the following:

```
SNMPD = snmpd_executable_path
```

If the `snmpd` executable on your system is not in the location indicated by the `start_peer` script, edit `snmpd_executable_path` to indicate the correct location of the `snmpd` executable.

3.5.2.4 Start the SNMP Components

Perform the following steps to start the SNMP components:

Verify that the SNMP components, `master_peer`, `encap_peer`, and `snmpd`, are not running:

```
$ ps -aef | grep peer
```

```
$ ps -aef | grep snmp
```

If any of the components are running, log in as the root user and use the `kill` command to terminate the processes before proceeding.

As the root user, run the `start_peer` script to start the PEER master agent, PEER encapsulator, and native Solaris 2.x SNMP agent:

```
# cd $ORACLE_HOME/network/snmp/peer
```

```
# ./start_peer -a
```

Warning: If you do not have the native Solaris 2.x SNMP agent on your system, you must not use the PEER encapsulator. To start the master agent only, run `start_peer -m`.

3.6 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 **Systems** from the **Manage** menu and then select **ServerStats Data Collectors**.
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.

3.7 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.

3.8 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.

3.8.1 Basic Guidelines for the Controller Heap Settings

The following are the basic guidelines for the settings to use for the maximum JVM heap size:

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 - Do not modify (default Heap size is `-Xmx1024m`)
- 32bit with 4GT enabled - Set the maximum JVM Heap size to `-Xmx2g` (see: Windows 4GT Tuning and Windows 32-bit Weblogic patch)
- 64bit - Set the maximum JVM Heap size to `-Xmx3g` (regardless if the JVM is 32-bit or 64-bit)

Where:

- `-Xms` = Starting heap size (default is 256m)
- `-Xmx` = Maximum Heap size (default is 1024m)

3.8.2 Modifying the JVM Heap Settings On Windows Machines

To modify the heap setting on Windows XP and Windows 7 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 `-Xmx1024m` 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.

3.8.3 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 -Xmx1024m"`.
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 `-Xmx1024m` 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`.

3.8.4 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`). To enable the 4GT feature see:

[http://msdn.microsoft.com/en-us/library/windows/desktop/bb613473\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/bb613473(v=vs.85).aspx).

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`).

3.9 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.

3.10 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**.

3.10.1 Menu Options

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

3.10.1.1 Tools Menu

Unlock Locked Records - unlocks locked records in the database. This is an emergency feature to be used to clear out locks in the database when the product fails and leaves entries locked for editing.

Purge Deleted Records - physically removes all deleted data from the database. Deleted items are marked as such and are not shown in the user interface; however, they are kept in the database until the database is purged.

Setup E-mail Config - displays the Setup E-mail Configuration dialog box for configuring the mail server to use for e-mail notifications.

Manage Default Reports - displays the Manage Default Reports dialog box for selecting which default reports will be visible to individual users.

3.10.1.2 Help Menu

Contents - displays the online help table of contents.

About Admin - displays version and copyright information.

3.10.1.3 Logout

Exits the Administrator.

3.10.2 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.

3.10.3 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.

3.10.4 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 when the **Set Oracle Load Testing Login** option is enabled (using the Tools menu).

- **None** - the user cannot access 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 Administrator access options.

4. Click **OK**.

3.10.5 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 when the **Set Oracle Load Testing Login** option is enabled (using the Tools menu).

- **None** - the user cannot access 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**.

3.10.6 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.

3.10.7 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**.

3.10.8 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 3.10.3, "Usage Audit Tab"](#) for additional information.

3.11 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 five dialog tabs.

3.11.1 Overview of the Menu Options

The Oracle Load Testing main menu has the following options:

- Scenario
- Session
- ServerStats
- Tools
- Manage
- Help
- Logout

The following sections explain each of the menu options.

3.11.1.1 Scenario Menu

These menu options let you work with scenario files. The following options are available:

New- creates a new Oracle Load Testing scenario.

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

Save- 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.

Save As - saves the currently open Oracle Load Testing scenario using a different filename.

3.11.1.2 Session Menu

The Sessions Menu options let you manage sessions. The following options are available:

Attach - opens a dialog box for selecting another running session that you want to view.

Detach - detaches from the current session.

Stop - stops the current session.

Terminate Idle Agents - stops all idle agent processes running on your agent systems.

3.11.1.3 ServerStats Menu

The ServerStats Menu options let you configure ServerStats configurations, metric profiles, and metrics, as well as start the virtual user logs.

Configurations - opens a dialog box for managing ServerStats configurations.

Metric Profiles - opens a dialog box for configuring ServerStats metric profiles.

Metrics - opens a dialog box for configuring ServerStats metrics.

ServerStats Display - opens the ServerStats Status dialog box that shows the results and status of ServerStats monitors.

3.11.1.4 Tools Menu

The following options are available:

Options - opens a dialog box for setting the Oracle Load Testing preferences.

VU Logs - starts the virtual user logs, which lets you monitor the progress of virtual users and view any errors virtual users may run into during playback.

Sync Point Status - opens a dialog box that displays the status of all sync points and lets you release individual sync points or all sync points.

View Cloud Agents - opens a dialog box that displays the agent and status information for cloud agents.

Import - opens a dialog box for importing a file to the repository.

Export - opens a dialog box for exporting a file to the local system.

Hardware Estimation - opens a dialog box for selecting a scenario for a hardware estimation session. A background session and local agent will start and run one Virtual User for the selected scenario.

3.11.1.5 Manage Menu

The following options are available:

Systems- opens a dialog box for adding, editing, and deleting systems that can be configured as VU Agents, Data Collectors, or Monitored Systems.

Databases - opens a dialog box for adding, editing, and deleting databases.

Sessions- opens a dialog box for editing and deleting sessions from the database.

Scenarios - displays a dialog box for editing and deleting scenarios.

Graphs - displays the Graph Query Manager for editing and deleting saved graph queries.

3.11.1.6 Help Menu

The following options are available.

Contents - opens the help system contents.

About - provides version, licensing, and serial number information.

3.11.1.7 Logout

Exits Oracle Load Testing.

3.11.2 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.

Start Load Test - Submits the current scenario to the Autopilot and automatically starts the scenario.

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.

Pause Autopilot - Pauses the autopilot.

3.11.3 Build Scenarios Tab

The Build Scenarios tab is where you specify which scripts to include in the scenario.

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

The **Configure parameters of the scenario** list shows the scripts selected for the current Oracle Load Testing scenario. You can configure each using the options provided here.

You can change the fields that are displayed and the default values for each field by selecting **Options** from the **Tools** menu then selecting **Scenario Defaults** and checking or unchecking the **Show** field.

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 **Tools** menu then selecting **Repositories**.

<**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"](#).

Configure parameters of the scenario - a list of scripts selected to be in the load scenario. The fields displayed here can be customized by selecting **Options** from the **Tools** menu then selecting **Scenario Defaults**. Select the fields you want to display by checking the field's corresponding checkbox in the Show column.

Scripts - lists the names of the scripts added to the scenario.

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.

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 Autopilot.

VUs 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.

Each line also includes the following buttons:

Configure all paramters - Displays the Edit Scenario Details dialog box for configuring the parameters script in the scenario.

Configure Data Bank - Displays the Data Bank Control dialog box for configuring the Data Bank options for individual scripts.

Delete - Deletes the selected profile from the scenario.

Configure Sync Point - Displays the Sync Point Status dialog box for configuring the Synchronization point parameters for the scripts in the scenario.

For information about the Selected VU Profile settings see [Chapter 4, "Defining Virtual User Scenarios"](#).

3.11.4 Set Up Autopilot Tab

The Set up Autopilot tab is where the information needed to control the running of the scenario is specified. The Autopilot controls the starting and stopping of the scenario, the frequency with which new virtual users are started and the number of virtual users that are started from among the profiles submitted to it.

You specify the start and stop times, and the virtual user rampup specifications for the Submitted Scenario Profile. The Set up Autopilot tab also shows the list of virtual user profiles submitted in the Oracle Load Testing scenario and the available ServerStats Configurations for monitoring back-end systems during a load test.

The **Timing and event controls** section is where you specify when the Scenario profiles should start and when they should end and the rate at which the virtual users within the Scenario profile list should start.

The **ServerStats Configuration** section is where you specify an Oracle Load Testing ServerStats configuration to run during the load test. Each ServerStats configuration

contains a collection of monitors for monitoring performance of back-end systems during a load test to identify bottlenecks.

The **Submitted Scenario Profiles** list shows the virtual user profiles submitted to the Autopilot as part of the Oracle Load Testing scenario. The list also shows the number of virtual users specified for each profile, the number of virtual users remaining to be started, and other details of the Scenario run.

The **Advanced Scenario Options** list shows the advanced ramp-up settings specified for each script in the scenario. The advanced ramp-up options show the start conditions, stop conditions, and ramp-up rate, if specified, for each script. The advanced ramp-up options are specified using the **Advanced options** link in the **Timing and event control** section.

For information about using the Autopilot, see [Chapter 5, "Using the Autopilot"](#).

3.11.5 Watch VU Grid Tab

The virtual user grid lists the currently running virtual users and the profile and playback details associated with each.

For information about using the virtual users grid, see [Chapter 5, "Using the Autopilot"](#).

3.11.6 View Run Graphs Tab

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

The View Run Graphs tab is refreshed according to what is set in the **Graph refresh interval** setting in the reporting options (select **Options** from the **Tools** menu).

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 Overview tab shows a thumbnail view of each graph. Click on a thumbnail to see a full view of that graphs or reports.

Click the New Graph tab to create a custom run time graph.

For information about reports and graphs, see [Chapter 6, "Using Graphs and Reports"](#).

3.11.7 Create Reports Tab

The Create 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 6, "Using Graphs and Reports"](#).

3.12 Systems Manager

The System Manager, accessed by selecting **Systems** from the **Manage** 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 **Systems** option of the Build Scenario tab.

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

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

- **New** - displays the Add System Group dialog box for configuring a new system group.
- **Edit** - displays the Edit System Group dialog box for adding or deleting systems from the group.
- **Delete** - 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** - displays the Add System dialog box for adding a new ServerStats data collector.
- **Edit** - displays the Edit System dialog box for editing the selected data collector.
- **Delete** - 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** - displays the Add System dialog box for adding a new system that you want to monitor using ServerStats.
- **Edit** - displays the Edit System dialog box for editing the selected system.
- **Delete** - deletes the selected systems. To select more than one system, hold down the CTRL key.
- **Name** - lists the available systems.

Cloud Services - these are cloud service configurations that you use to run Virtual User agents on a cloud. Configured cloud services are available on the **Build Scenarios** tab as a **System** choice.

- **New** - displays the Add Cloud Service dialog box for adding a new cloud configuration that you want to use to run Virtual User agents on a cloud.
- **Edit** - displays the Edit System dialog box for editing the selected cloud configuration.
- **Delete** - deletes the selected cloud configurations. To select more than one cloud configuration, hold down the CTRL key.
- **Name** - lists the available cloud configuration.

Note: Systems, system groups, and cloud service configurations appear in the System list on the Build Scenarios tab.

For more information about using the System Manager, see [Section 3.13, "Defining Systems"](#).

3.13 Defining Systems

Before you can select systems in Oracle Load Testing Scenarios, you must define the machines that are Oracle Load Testing agent systems. The Oracle Load Testing System Manager lets you define system names or IP addresses and create system groups that the Oracle Load Testing scenarios can use as agents.

Note: See the installation section at the beginning of this 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.

3.13.1 Adding New VU Agent Systems

To add a new VU Agent system:

1. Select **Systems** from the **Manage** menu to display the Systems Manager.
2. 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 Name or 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 **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 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 Name or IP field, and specify the operating system type.
5. Enter the port and authentication settings.
6. Click **Test** to verify the system is available.
7. Click **OK**.
8. Click **Close**.

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

3.13.2 Adding New System Groups

To add a new system group:

1. Select **Systems** from the **Manage** menu.
2. Select **VU Agent System Groups**.
3. Click **New** to display the Add System dialog box.

Name - enter the name of the system group.

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. Enter the name of the group in the **Name** field.
5. Select the systems you want to add from the **Systems** list.
6. Click **OK**.

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

3.13.3 Adding Systems to Groups

To add systems to groups:

1. Select **Systems** from the **Manage** menu.
2. 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.

4. Select the systems you want to add from the Systems list.
5. Click **OK**.
6. Click **Close**.

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

3.13.4 Adding New ServerStats Data Collectors

To add a new ServerStats Data Collector:

1. Select **Systems** from the **Manage** menu to display the Systems Manager.
2. Select **ServerStats Data Collector**.
3. Click **New** to display the Add ServerStats Data Collector dialog box.

General - enter the system information.

- **Name** - enter the name of the data collector.
- **Host Name or 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.

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.

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

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

3.13.5 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.

3.13.5.1 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 `<installdir>\DataCollector\classes` directory on all machines that will be used as data collectors for sampling from that server. The default `<installdir>` 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	<installdir>\DataCollector\classes\jboss
Redhat JBoss 5.x	all *.jar files under <JBoss5 install dir>/jboss-as/client/	<installdir>\DataCollector\classes\jboss
Redhat JBoss 6.x	all *.jar files under <JBoss6 install dir>/client/	<installdir>\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, webservice.jar	<installdir>\DataCollector\classes\weblogic
Oracle WebLogic 9.1	weblogic.jar, webservice.jar	<installdir>\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.)	<installdir>\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, wlmsrver.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

3.13.5.2 Adding a Monitored System

To add a new monitored system:

1. Select **Systems** from the **Manage** menu to display the Systems Manager.
2. Select **Monitored System**.
3. Click **New** to display the Add Monitored System 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.

New - displays the Add Component dialog box for manually adding components and component types.

Delete - deletes the selected component.

Discover Components - displays the System Discovery dialog box for specifying the components to discover and the data source to use.

Note: Discover Components is not used with the Enterprise Manager data source. The Enterprise Manager data source does not use the Oracle Load Testing data collector. Session metrics for Enterprise Manager are retrieved using the query string parameters specified in the ServerStats metric profile. For Enterprise Manager monitored systems, select the Enterprise Manager data source and specify the base URL. See the Enterprise Manager data source in this section for additional details.

General

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

Components - lists the component types and components that are configured for this 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.

Enterprise Manager - specifies the Enterprise Manager instance from which to retrieve session metrics. The Enterprise Manager data source is a special case in that it does not use the Oracle Load Testing data collector. Session metrics are accessed directly from an Enterprise Manager instance. Specify the base URL as follows:

- **Enterprise Manager URL** - enter the base URL of the Enterprise Manager instance from which to retrieve Weblogic Domain Metrics or Weblogic JVM Metrics. For example, `https://<machine name or IP>.us.oracle.com:7799/em`. The metrics specified for each ServerStats Enterprise Manager metric profile will be used as query string parameters for this URL when accessing Enterprise Manager Diagnostic metrics from the Oracle Load Testing reports. See the *Oracle Load Testing ServerStats User's Guide* for information about configuring Enterprise Manager metrics and configurations.

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

<http://java.sun.com/j2se/1.4.2/docs/guide/jdbc/bridge.html>.

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.
- **Host** - Specify the host name of the machine running the database. 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.
- **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.
- **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.

IBM WebSphere PMI

- **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 10.x
 - Oracle WebLogic 9.1
 - Oracle WebLogic 9.0
 - IBM WebSphere 7.0
 - IBM WebSphere 6.1
 - IBM WebSphere 6.0
 - IBM WebSphere 5.1
 - IBM WebSphere 5.0
 - Redhat JBoss 4.x
 - Redhat JBoss 5.x
 - Redhat JBoss 6.x
- **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.

SNMP

- **Port** - enter the port number for the connection.
- **Community String** - the access key required for remote access. The Community String is typically "public" unless otherwise configured by the System Administrator. Contact the Administrator to find out the Community String required for remote access to the system.
- **SNMP Version** - enter the SNMP version.

Url

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

Virtual Agent

- **Remote Port** - Specifies the port number. The default port for Telnet is 23, and the default port for SSH is 22.
- **Remote 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 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.

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.

4. Enter the name of the system in the Name field, and enter the name or IP address of the system in the Host Name or IP field.
5. Enter data source information for the type of system you are adding.
6. Click **Discover Components** to find the components available on this system. Components that are found will be available in ServerStats. If you do not discover components when setting up the monitored system, you can discover them later when you set up your ServerStats monitor. The System Discovery dialog box is displayed.

Select Data Sources - select the data sources you want to use for discovery. When you select a data source, the components that it can discover are selected in the **Select Component Types to Discover** list.

Select Component Types to Discover - deselect component types that you do not want to discover.

7. Select the data sources you want to use for discovery. The components types that this data source can discover are automatically selected in the **Select Component Types to Discover** list.
8. Deselect and component types that you do not want to discover and click **OK**. The Discovery Setup dialog box for the data sources you selected is displayed. For example, if you selected Perfmon, the following dialog box is displayed.

This dialog box has the following options based on the selected data sources.

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

<http://java.sun.com/j2se/1.4.2/docs/guide/jdbc/bridge.html>.

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.
- **Host** - Specify the host name of the machine running the database. 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.
- **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.
- **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.

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.

SNMP

- **Port** - enter the port number for the connection.
- **Community String** - the access key required for remote access. The Community String is typically "public" unless otherwise configured by the System Administrator. Contact the Administrator to find out the Community String required for remote access to the system.
- **SNMP Version** - enter the SNMP version.

9. Enter the discovery information and click **OK**. The Discovery Status dialog box is displayed showing the progress of the discovery process.

This dialog box displays the progress of the discovery process. It also displays any errors encountered. The message, "Discovery Done," is displayed when discovery is complete.

10. When Discovery Done is displayed, click **OK**. If previously configured components could not be found, the Confirm Remove Components dialog box is displayed. If the configuration of previously configured components has changed, the Confirm Replace Components dialog box is displayed.

Check All - checks all of the listed components. Checked components will be removed or replaced.

Uncheck All - unchecks all of the listed components.

<components> - lists the previously configured components that either were not found or whose configuration has changed.

11. Deselect any components that you do not want to remove or replace and click **OK**.
12. The discovered components are added to the Components tree. Click on a component to view its data source configuration. Click **Delete** to remove the data source. Click **Add** to add an available data source.
13. Click **New** to manually add component types and components. The Add Component dialog box is displayed.

Add Type

- **Component Type** - select this option to add a new component type. This is the only option available from the system node.
- **Component** - select this option to add components for the selected component type.

Definition

- **Component Type** - when adding a new component type, enter a meaningful name for the component type. When adding a new component, this field defaults to the component type that was selected when you clicked the **Add** button.

- **Component** - enter a meaningful name that allows you to identify the specific component you are adding.

Data Sources to Add to Component - select the data sources to add to this component.

14. Select whether you are adding a component or component type.
15. Specify the component type, if necessary, and the component. Select the data sources to apply to this component and click **OK**.
16. The data sources available for this component are displayed on the right. Enter the appropriate information. Click **Delete** to remove a data source. Click **Add** to add an available data source.
17. Click **OK**.
18. Click **Close**.

Note: Systems, system groups, and cloud service configurations appear in the Systems list on the Build Scenarios tab.

3.13.6 Adding Cloud Service Configurations

The cloud services provide an available, elastic, and practically limitless set of agent machines to use when running a large load test. Combined with some basic monitoring of agent load metrics, Oracle Load Testing is able to detect when an agent process or agent machine has reached its load limits and add additional cloud instances for generating additional load as needed.

Note however, that Virtual User rampup cannot proceed at a faster rate than new agent machines can be added over time. Planning is required to ensure that agent machines can be added in the cloud at the rate the Virtual User rampup for the test session requires. For example, if a cloud agent machine requires X minutes to initialize, the **Agents per Session** settings for the Cloud Service configuration should factor in the X minutes to initialize an agent to accommodate the Virtual User rampup-rate. The best practice is to initialize as many Agents machines in the cloud as the test run can possibly require to ensure enough agents are available to provide seamless rampup of Virtual Users.

Note: The Oracle Load Test controller machine must have a static IP address for the DNS lookup from the deployed agent to the controller to function correctly.

3.13.6.1 Importing the Enterprise Manager Certificate

By default, Enterprise Manager uses a self-signed certificate for SSL/HTTPS communication. Communication between Oracle Load Testing and Enterprise Manager is performed using HTTPS and requires configuring Oracle Load Testing (and the underlying Java SSL support) to accept the Enterprise Manager certificate by importing the Enterprise Manager certificate into the Oracle Load Testing cacerts keystore.

To import the Enterprise Manager certificate into the Oracle Load Testing keystore:

1. Save the Enterprise Manager certificate to a file:
 - Connect to Enterprise Manager using Internet Explorer.

- View certificate details and copy the certificate to a file (for example, `c:\emserver1234.cer`).
2. Import the certificate into the keystore cacerts:
 - `cd \<installDir>\jdk\jre\lib\security`
 - `c:\<installDir>\jdk\jre\keytool -importcert -keystore cacerts -file c:\emserver1234.cer -alias emserver1234`
 - (password=changeit)
 3. Restart the Oracle Application Testing Suite server:
 - Open Control Panel, Administrative Tools
 - Open Services
 - Select Oracle ATS Server
 - Click **Restart**.

3.13.6.2 Adding the Cloud Service Configuration

Before you can select cloud services in Oracle Load Testing Scenarios, you must define the cloud service configuration(s) that will be used as Oracle Load Testing Virtual User agent systems. The Oracle Load Testing System Manager lets you define the number of agents per session, agent system policies, and the cloud configuration details. You must also have access to an Oracle Cloud managed by Enterprise Manager as a source of agent machines for a load test, have the details of the assembly, and have the Enterprise Manager host information.

To add a new cloud service configuration:

1. Select **Systems** from the **Manage** menu.
2. Select **Cloud Services**.
3. Click **New** to display the Add Cloud Services dialog box.

Name - enter the name of the cloud service.

Agents per Session - specifies the initial size of the agent pool to create and the behavior of agent pool reallocation.

- **Initial** - specifies the initial size of the agent pool.
- **Minimum** - specifies the low threshold of available machines.
- **Increment** - specifies the number of instances to allocate when the low threshold is triggered.
- **Maximum** - specifies the maximum number of instances per session.

Agent System Policies - specifies the agent resource thresholds and behavior.

- **Metric Type** - specifies the metric to monitor to determine agent pool reallocation.
 - **System CPU Used** - monitors agent process CPU usage. Specify the maximum process CPU usage percentage to trigger adding new agent machines.
 - **System Memory Available** - monitors system memory usage. Specify the minimum system memory to trigger adding new agent machines.
- **<operator>** - specifies the operator to use to test the metric against the threshold value.

- **Value** - specifies the value to use as the threshold for the specified metric type.
- **Action** - specifies the action to perform if the metric reaches the specified threshold value.
- **New Policy** - adds a new Agent System policy row to the configuration. Setting an added policy **Metric Type** to *(none)* will remove the new policy when the cloud service configuration is saved.

Cloud Configuration - specifies the Oracle Enterprise Manager instance connection information and Assembly instance creation details.

- **EM Host** - specifies the Oracle Enterprise Manager instance to use for the cloud.
- **Port** - specifies the port number to use for the connection.
- **Username** - specifies the user name used to connect to the Oracle Enterprise Manager instance. The credentials for the cloud must refer to a repository user not an SSO user in Enterprise Manager. The user account must have the correct roles and privileges in Enterprise Manager to access the cloud. For example:

- Roles: EM_SSA_USER, EM_USER
- Privileges: View_Any_Infrastructure_Cloud

The user account must also have access to the required Assemblies needed from the software library.

- **Password** - specifies the password used to connect to the Oracle Enterprise Manager instance.
 - **Connect** - connects to the specified Oracle Enterprise Manager instance and retrieves Assembly details.
 - **Assembly** - specifies the Assembly on the Oracle Enterprise Manager machine, which is part of the software library.
 - **Zone** - specifies the deployment Zone in the Assembly.
 - **Network Profile** - specifies the network profile to use.
 - **Assembly Prefix** - specifies the Assembly prefix to use.
4. Enter the name of the cloud service in the **Name** field.
 5. Specify the Agent cloud services rampup by entering the **Initial**, **Minimum**, **Increment**, and **Maximum** values.
 6. Specify the Agent System Policies by selecting the **Metric Type**, operator, **Value**, and **Action** to perform if the metric values meets the specified operator criteria.
 7. If necessary, click **New Policy** to add additional Agent System Policies. Setting an added policy **Metric Type** to *(none)* will remove the new policy when the cloud service configuration is saved.
 8. Specify the **EM Host**, **Port**, **Username**, and **Password** credentials for the Oracle Enterprise Manager instance to use for the cloud.
 9. Click **Connect** to access the connection to the cloud service and retrieve assembly data.
 10. Select the **Assembly**, **Zone**, and **Network Profile**.
 11. Enter the **Assembly Prefix**.

12. Click **OK**.
13. Click **Close**.

Note: Systems, system groups, and cloud service configurations appear in the Systems list on the Build Scenarios tab.

3.13.7 Renaming Systems

To rename a system:

1. Select **Systems** from the **Manage** menu.
2. 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**.
7. Click **Close**.

3.13.8 Editing Systems

To edit a system:

1. Select **Systems** from the **Manage** menu.
2. 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**.
6. Click **Close**.

3.13.9 Deleting Systems

To delete a system:

1. Select **Systems** from the **Manage** menu.
2. 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**.

3.14 Setting Options

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 **Tools** menu. Selecting this option opens the Options dialog box.

3.14.1 Custom Browser Options

The following custom browser options are available:

New - displays a new line in the table.

Delete - deletes the selected browser.

Name - any name for the customized browser emulator. This name will appear in the Browser Emulation list in the Edit Scenario Details dialog box.

User Agent String - specifies the string to send to the server as the User Agent header string for the customized browser emulator.

3.14.2 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. Refer to "Configuring Oracle Load Testing Agents" earlier in this chapter.

New - adds a new entry to the table.

Delete - deletes the selected 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 **Tools** menu.
2. 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.

3.14.3 Setting Scenario Defaults

You can change the default settings for profiles using the Scenario Defaults 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 Build Scenarios tab, then add them back.

Each setting in the right panel has two columns:

Show - when checked, this field is displayed on the Build Scenarios tab.

Default Value - shows the value to which the option is set when a new script is added to a scenario.

Main - the main settings are as follows:

- **# 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.
- **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 Autopilot.
- **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 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.
- **Automatically dismiss Javascript 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.
- **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 4.2, "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:

- **Auto Generate Timers For All 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.
- **Auto Generate Timers For All 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.
- **Auto Generate Timers For All 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:

- **Object Download Errors Are Fatal** - 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:

- **Maximum Users Per Process** - sets the maximum number of virtual users per single agent process. When running virtual users as threads in a single process, **Maximum 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.

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

Java 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 a set of CSV files. The files are saved locally on the agent machines in directories specified as follows:

```
<oats_install>/agent/rawdata/<controller-identifier>/<session_
name>/<agent-id>/<YYYY-MM-DD HH:mm:ss>
```

See [Section 6.9, "Using Raw Data"](#) for additional information about the counter files and 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 Number of 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.
- **SSL Version** - 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.
- **Ignored Url** - 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 could be, 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 - 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.

OpenScript 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.

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.

OpenScript 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.

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 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.

3.14.4 Setting Autopilot Defaults

The Autopilot Defaults options let you specify the default settings for the Set up Autopilot tab. Refer to [Section 5.1, "Specifying Start and Stop Settings"](#) for a description of these options.

3.14.5 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 can be saved to a database for post-testing analysis in the Analyze Results tab.

You can specify default settings for how sessions start and end data collection by selecting **Options** from the **Tools** menu then selecting **Start and Stop**. See [Section 4.1.4, "Managing Sessions"](#) for a description of these options.

3.14.6 Setting Session Profile Options

These options specify the default characteristics for graphing and reporting. Unique session profiles are created for multiple instances of a script if the selected settings have different values.

For example, if you are running two profiles emulating different browsers, check the Cache Emulation attribute to view separate plot lines in the graphs for each browser.

3.14.7 Setting Reporting Options

Use these options to specify the parameters for refresh intervals and for creating profile timer names when generating timers for all resources. See [Section 6.1, "Setting the Reporting Options"](#) for a description of these fields.

3.14.8 Setting General Options

These options specify the general settings for validation, restarts, and timeouts.

Validate hostname/ip when user adds 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 monitors when user adds or modifies a monitor - checks to see if the monitor can be applied to the target system when you create a monitor.

When Oracle Load Testing Server IP address changes - specifies the action to take if the Oracle Load Testing Application Server IP address changes, as follows:

- **Restart Application Service** - restarts the Oracle Load Testing Application Service when the server IP address changes.
- **Stop Application Service** - stops the Oracle Load Testing Application Service when the server IP address changes.

Polling interval for network status check - specifies the interval, in seconds, for checking the network status.

Databank Setup Timeout - specifies the time, in seconds, to wait for databanking operations before timing out.

Defining Virtual User Scenarios

This chapter explains how to define virtual user scenarios based upon the virtual user profiles.

4.1 Defining Scenarios

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.

4.1.1 Selecting Profiles

Once you have a set of virtual user profiles, you can define Oracle Load Testing 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 **Build Scenarios** 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 **Select Script** list.

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

5. Click the **Add to Scenario** button or double-click the script name in the **Select Script** list.
6. If 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. 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. When finished adding scripts, click **Add to Autopilot** to specify the start, ramp-up, and stop setting for the scenario.

4.1.2 Specifying Scenario Profile Attributes

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

To set the parameters for a specific script:

1. If necessary, open a saved Scenario file.
2. Select the Build Scenarios tab.
3. Select the scripts to include in the Scenario.
4. Click the **Scenario Detail** button to display the Edit Scenario Details dialog box. This dialog box shows the settings for the scripts in the current scenario. Select the script name in the left pane and set the settings. Note that the settings 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 Build Scenarios tab. To specify which parameters are displayed in the Build Scenarios tab, select **Options** from the **Tools** menu then select **Scenario Defaults** and select the fields you want to display by selecting the **Show** 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 attributes.
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.
8. Click **OK**.

4.1.2.1 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**. Right click on the script or user defined profile that you want to copy the settings to and click **Paste**.

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

Main - the main settings are as follows:

- **# 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.
- **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. You specify the number of iterations using the Autopilot.
- **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 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, each Virtual User will clear its own cache after each iteration (after the script completes each iteration of its `run()` section).
- **Automatically (when page is out of date)**: when true, the web server is checked for newer versions if the page is out of date. The Web server is not checked for newer versions of unexpired cached pages. This setting behaves like the "Automatically" cache setting in Internet Explorer.
- **Every visit to the page**: when true, the Web server is always checked for newer versions of all cached pages. This setting behaves like the "Every time I visit the web page" cache setting in Internet Explorer.
- **Maximum In-Memory Cache Size** - specifies the maximum amount of in-memory storage to allocate for cached document contents. This setting applies to all virtual users in the process, even though each virtual user keeps its own cached documents. After the in-memory cache is exhausted, document contents will be cached to a temporary folder on disk in `<installDir>\agent\cache`. There is no upper bound on how much disk storage may be used to store cached documents. The disk cache is cleared every time the agent process starts. The default value is 128MB.
- **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 4.2, "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:

- **Auto Generate Timers For All 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.

- **Auto Generate Timers For All 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:

- **Maximum Users Per Process** - sets the maximum number of virtual users per single agent process. When running virtual users as threads in a single process, **Maximum 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.

- **Maximum 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.

Java 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 a set of CSV files. The files are saved locally on the agent machines in directories specified as follows:

```
<oats_install>/agent/rawdata/<controller-identifier>/<session_name>/<agent-id>/<YYYY-MM-DD HH:mm:ss>
```

See [Section 6.9, "Using Raw Data"](#) for additional information about the counter files and 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 Number of 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.
- **SSL Version** - 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.
- **Ignored Url** - 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
```

OpenScript 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, or Ignore, 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.

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.

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 - 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.

OpenScript 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.

OpenScript 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.

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.

4.1.2.2 Functional Test Script Settings

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

Main - the main settings are as follows:

- **# 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.
- **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. You specify the number of iterations using the Autopilot.
- **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.
- **Automatically dismiss Javascript 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.

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:

- **Auto Generate Timers For All 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.
- **Auto Generate Timers For All 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.

Java 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 a set of CSV files. The files are saved locally on the agent machines in directories specified as follows:

```
<oats_install>/agent/rawdata/<controller-identifier>/<session_name>/<agent-id>/<YYYY-MM-DD HH:mm:ss>
```

See [Section 6.9, "Using Raw Data"](#) for additional information about the counter files and 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.

4.1.3 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 2000/2003, to determine system resource usage. See [Section 4.3.7, "Estimating Hardware"](#) for additional information about using the Oracle Load Testing Hardware Estimation tool.

4.1.4 Managing Sessions

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, 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 **Tools** menu then selecting **Session Start/Stop**. Selecting this option opens the Session Start and Stop Options dialog box:

Define how a session starts - defines actions for a new session.

- **Auto assign session name** - when selected, the specified session name prefix, plus a four-digit increment number, is assigned when a new session starts.
- **Session name prefix** - specifies a fixed name to add before the session name. Enter a name to use or "Default". Oracle Load Testing adds the increment number to the name you define. When set to "Default," the name is "session" (for example, session0001).

Define how a session ends - 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 in the Autopilot.
- **Stop attached session after browser closes** - 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.

4.1.5 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 setting the **Use Data Bank** option to True in the Edit Scenario Details dialog box 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. Add a script or profile that uses a Data Bank to the Configure parameters of the scenario list.
2. Click the **Edit Scenario Details** button.
3. In the Main section, set the **Use Data Bank** field to True.
4. Select other Scenario Profile options as needed to configure how each Oracle Load Testing agent uses Data Bank records.
5. Click **OK** to exit the dialog box.
6. Click the **Configure Data Bank** button.

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

Databank Source - this section shows the following information about the selected databank:

- **Type** - shows the type of the selected databank file. Databanks can be CSV text files or databases.
- **Source** - shows the path and filename of CSV text files or the database source of database databanks. While there is not a maximum file size, the recommended maximum sizes is 200 MB. The only limitation is how long it takes to generate the index. The databank must be indexable within 30 seconds, by default. This setting is configurable in the "Databank Timeout" setting in the General Options. See [Section 3.14.8, "Setting General Options"](#) for additional information about setting the Databank Timeout.
- **Records** - shows the number of records in the databank. The Oracle Load Testing application sets the databank buffer size as 20 records per agent process by default.

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

- **Name** - lists the names of the Data Bank variables.
- **Value** - list the values of the Data Bank variables.

- **Type** - shows the type of Data Bank variable: Internal, External, or Not Mapped.

Databank Settings - this section specifies the settings to use for the selected databank:

- **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 a 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 **Let Each User Iterate Over Records 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 the 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 the 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()`.
- **Let Each User Iterate Over Records 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. Make sure the script or profile is selected in the tree view.
8. Use the arrow buttons to change the record data.
9. Click OK.

4.1.6 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. Select **Sync Point Status** from the **Tools** menu.

-or-

Click the **Watch VU Grid** tab and right-click on a running virtual user.

4.2 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 Edit Scenario Details dialog box, set **Use IP Spoofing** to True in the Browser Settings section.
7. Submit and Start the scenario in the Autopilot.

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.

4.3 Working with Scenario Files

Once you select virtual user profiles and set the attributes, you can save the Scenario to a file for future use. Oracle Load Testing automatically assigns the name Scenario### for the scenario.

4.3.1 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.
 - Workspace** - lists the workspaces available in the selected repository.
 - <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 Autopilot settings as part of a saved scenario.

4.3.2 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.

File Type - specifies the type of scenario file to open.

- **Scenario** - specifies a scenario that has been save using the **Save As** option on the **Scenario** menu. Scenario files have a .scn filename extension.
- **Self Contained Scenario** - specifies a scenario saved as a self contained zip file using the **Export** option on the **Tools** menu. Self Contained Scenarios can include recorded data, playback results, and databanks. Scenario files have a .scnzip filename extension.

Repository - a list of repositories.

Workspace - lists the workspaces available in the selected repository.

<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**.

4.3.3 Renaming Scenarios

To rename a scenario:

1. Select **Scenario** from the **Manage** menu to display the Scenario Manager.

Repository - a list of repositories.

Workspace - lists the workspaces available in the selected repository.

Edit - opens the Edit Scenario Dialog Box for editing the selected scenario.

Delete - deletes the selected scenario.

Name - lists the scenarios in the selected workspace.

2. Select the scenario you want to edit.
3. Click **Edit** to display the Edit Scenario dialog box.

Name - enter a name for the scenario.
4. Enter a new name for the scenario.
5. Click **OK**.
6. Click **Close** to exit the Scenario Manager.

4.3.4 Deleting Scenarios

To delete a scenario:

1. Select **Scenarios** from the **Manage** menu to display the Scenario Manager.
 - Repository** - a list of repositories.
 - Workspace** - lists the workspaces available in the selected repository.
 - 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.
 - Name** - lists the scenarios in the selected workspace.
2. Select the scenarios you want to delete. To select more than one scenario, hold down the CTRL key.
3. Click **Delete**.
4. Click **Yes** to confirm the deletion.
5. Click **Close** to exit the Scenario Manager.

4.3.5 Removing Profiles From a Scenario

To remove profiles from the Scenario Profiles list:

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

4.3.6 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.

4.3.7 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.

4.3.7.1 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 **Build Scenarios** tab and save the scenario.

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

2. Select **Hardware Estimation** from the **Tools** menu.
3. Select the Repository and workspace folder in which the scenario file is located.
4. Select the scenario to use for the hardware estimate session.
5. Click **Estimate** 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.
6. When the estimation finishes, click **View Report** to view the results.

4.3.7.2 Generating Hardware Estimation Reports

To run a hardware estimate report for an Oracle Load testing scenario:

1. Make sure you have generated a Hardware Estimation using the **Hardware Estimation** option on the **Tools** menu.
2. Click the **Create Report** tab.
3. Click the **Reports** tab.
4. Select Estimation Report from the **Report** list.
5. Select the Repository and workspace folder in which the report is located.
6. Select the Scenario report and click **Generate** to generate the Hardware Estimation report.

The Hardware Estimation Report provides the following:

- Hardware configuration for each agent, including CPU information, memory size, and network bandwidth limitations.
- Process Runtime Configuration for each agent, including Operating System information, Max Virtual Users per Process setting, and Max heap size setting.
- Estimated consumption for each script running on the corresponding agent, including CPU usage, heap size usage, and network bandwidth usage.
- Suggested Configuration, including idle resources monitored before start estimation, estimated number of required machines, and suggested configurations for better performance.

4.4 Submitting Scenarios to Autopilot

Once you have defined a scenario and set the profile attributes, you submit the scenario to the Oracle Load Testing Autopilot.

4.4.1 Submit without Starting the Scenario

Submitting the scenario to the Autopilot without starting allows you to specify the start and stop times for running the scenario profiles.

To submit a Scenario without starting the Autopilot click the **Add to Autopilot** button.

Oracle Load Testing automatically opens the Autopilot tab with the scenario loaded.

4.4.2 Submit and Start Scenario in Autopilot

If you want to use the default Autopilot settings and start the Scenario, you can submit scenarios and automatically start the Autopilot in one step.

To submit a Scenario and start the Autopilot click the **Run Test** button.

Oracle Load Testing automatically opens the Autopilot tab with the scenario loaded and starts the virtual user run.

Perform baseline for Oracle DB - specifies if an Oracle Database snapshot is taken before the load test session. This option only appears if the session has a ServerStats configuration with Oracle Database Metrics applied when the load test is launched. Oracle metrics are identified by the database driver connection properties. If you add an Oracle Database metric *after* the load test is running, there is no way to take snapshots during the load test. The following options are available:

- **No** - no database snapshot will be taken during the load test.
- **Yes** - a database snapshot and baseline will be created during the load test. An Oracle Database "snapshot" is a PL/SQL call that causes the database to record a timestamp during the gathering of performance data. A "baseline" is a PL/SQL call that uses the ID of two snapshots, which causes the data recorded between the snapshots to be persisted indefinitely rather than expiring after a user-configured persistence interval (the default is 8 days). The snapshot and baseline allow you to retrieve Enterprise Manager drilldown information for a load test session long after the session has been run. When Yes is selected, two snapshots - one at the beginning and one at the end of a load test session - and a baseline cause the data gathered during the session to be persisted indefinitely.

Using the Autopilot

This chapter explains how to use the Autopilot 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 **Tools** menu then selecting **Autopilot Defaults**.

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 autopilot defaults does not affect the current scenario. The changed autopilot defaults go into effect when you start a new scenario.

Before you can use the Autopilot, you must first define virtual user profiles and scenarios and submit the scenario profiles to the Autopilot. [Chapter 4, "Defining Virtual User Scenarios"](#) explains how to define virtual user profiles.

5.1 Specifying Start and Stop Settings

The **Start the load test** and **Stop the load test** groups of the Set Up Autopilot tab is where you specify when to start and stop running of virtual user scenarios.

5.1.1 Specifying When to Start Virtual Users

The **Start the load test** selector specifies when to start running virtual user scenarios and has the following options:

- **When the start button is pressed** - starts running the scenario as soon as you start the Autopilot.
- **After a delay of** - allows you to specify an amount of time to wait before starting the virtual user scenario. Selecting this option enters a default time in the associated editbox in the format: hh:mm:ss. Change the amount of time as necessary. The Autopilot waits the specified amount of time after you start the Autopilot to start running the virtual users.
- **At a specific time** - allows you to specify the time to start running the virtual user scenario. Selecting this option enters a default time in the associated editbox in the format: hh:mm:ss. Change the time as necessary. The Autopilot waits until the specified time after you start the Autopilot to start running the virtual users.
- **Synchronize VU start up** - fully initializes the first group of virtual users in their respective agent processes before actually beginning the iteration. This procedure

attempts to eliminate any startup overhead that may amplify the iteration performance times of the virtual users' first iterations.

5.1.2 Specifying When to Stop Virtual Users

The **Stop the load test** selector specifies when to stop virtual user scenarios that are currently running and has the following options:

- **When the stop button is pressed** - repeats running of the virtual user scenario continuously until you manually stop the virtual users.
- **After each user iterates [X] times** - allows you to specify the number of times the virtual users iterate over the scripts submitted to the Autopilot. Selecting this option enters a default number of iterations in the associated edit box. 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.
- **After a delay of** - allows you to specify an amount of time to continue running the virtual user scenario. Selecting this option enters a default time in the associated editbox in the format: hh:mm:ss. Change the amount of time as necessary. The Autopilot runs the scenario profiles for the specified amount of time and then automatically stops running the scenario.
- **At specific time** - allows you to specify the time to stop running the virtual user scenario. Selecting this option enters a default time in the associated edit box in the format: hh:mm:ss. Change the time as necessary. The Autopilot plays back the scenario profiles until the specified time and then automatically stops running the scenario.

5.2 Specifying Virtual User Rampup

The **Virtual user (VU) ramp-up** group is where you specify the rate at which the Autopilot starts running virtual users.

The **Add per step** and **After every** settings work together to specify the start rate of the virtual users.

Use the **Advanced options** to specify start, stop, and ramp-up conditions on a per-script basis. When Advanced options are set, the Virtual user (VU) ramp-up settings in the Autopilot tab are disabled with the notice "These settings are overridden by advanced options. Revert". Click "Revert" to clear the Advanced options and re-enable the Virtual user (VU) ramp-up settings in the Autopilot tab. See [Section 5.4, "Using the Advanced Scenario Ramp Options"](#) for additional information about the Advanced options.

5.2.1 Specifying How Many Virtual Users to Start at a Time

Add per step 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.

After 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.

5.2.2 Examples

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

Example: If you specify the number as 5 and the **After every** setting is set to 1 iteration, the Autopilot starts running five virtual users 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 per step** setting is set to 5 users at a time, the Autopilot first starts five virtual users. After the first five complete one iteration of the profile, the Autopilot starts another five virtual users. The Autopilot starts five additional virtual user after each iteration until all virtual users are running.

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

5.3 Manually Starting and Stopping the Autopilot

Once you have submitted scenario profiles and set the Autopilot options, you can manually start and stop the Autopilot using the following:

- Autopilot buttons
- Toolbar

You need to manually stop the Autopilot to submit more scenario profiles or stop any new virtual users from being started.

5.3.1 Starting the Autopilot

To start the Autopilot manually:

1. Click the **Run Test** button.
2. If the Deferred Start options appear, allow the delay to continue until the start time.

or:

Click **Run Test** to start the start running the scenario immediately.

or

Click **Stop** from the **Session** menu to cancel the start.

Note: The Autopilot starts running the virtual users based upon the **Start the load test** settings and the **Virtual User (VU) ramp-up** settings.

If you set the Start option to either **At Specified Time** or **After Delay Of**, Oracle Load Testing displays the remaining time until the Autopilot starts in the status bar.

5.3.2 Stopping the Autopilot

To stop running the scenario profiles submitted to the Autopilot click the **Stop** button.

This stops Autopilot from starting any new virtual users.

5.4 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. When the defaults are changed, these settings override the start, stop, and ramp-up settings specified in the Autopilot tab. The Virtual user (VU) ramp-up settings in the Autopilot tab are disabled with the notice "These settings are overridden by advanced options. Revert". Click "Revert" to clear the Advanced options and re-enable the Virtual user (VU) ramp-up settings in the Autopilot tab.

Select the script in the left pane and set the start, stop, and ramp-up settings for the selected script in the right pane.

To specify advanced ramp-up settings:

1. Create a scenario in the Build Scenarios tab by adding scripts to define virtual user profiles. [Chapter 4, "Defining Virtual User Scenarios"](#) explains how to define virtual user profiles.
2. Add the scenario to the Autopilot tab.
3. Click **Advanced options**. The advanced ramp-up settings are as follows:

Start the script - specifies the start conditions for the script selected in the left pane. 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 Autopilot tab or when the **Run Test** button is clicked in the Autopilot tab.
- **After a delay of (hh:mm:ss)** - when selected, the script starts the specified amount of delay time after the scenario starts.
- **When this script 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 script 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 the script - specifies the stop conditions for the script selected in the left pane. 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 Autopilot tab or when the Stop or Abort button is clicked in the Autopilot tab.
- **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 script 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 script 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.

Virtual user (VU) ramp-up - specifies the rate at which the virtual users for the selected script should start. The available ramp-up settings are as follows:

- **Add per step** 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.
 - **After 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.
4. Select a script name in the left pane.
 5. Specify the start, stop, and ramp-up settings for the script selected in the left pane.
 6. Repeat steps 4 and 5 for all scripts in the left pane.
 7. Click **OK**.

The advanced ramp-up settings for each script will be shown in the **Advanced Scenario Options** section of the Autopilot tab.

5.5 Working with Virtual Users

Once you start running virtual users using the Autopilot, the Virtual User Grid tab lists the running virtual users and shows the run details for each.

You can sort the data by a column by clicking on the column title.

5.5.1 Overview of the Virtual Users Grid

The virtual users grid has the following columns:

Items - displays the currently selected item. Enter the item number you want to view and click Enter to go directly to that item.

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.

OLT Server Warnings - this button is displayed when a virtual agent encounters a server error. Click it to view errors.

5.5.2 Controlling Individual Virtual Users

You can control the running of specific running virtual users using the virtual users grid.

1. Define virtual user profiles.
2. Specify the Scenario.
3. Submit the Scenario to the Autopilot.
4. Start the Autopilot.
5. Select the virtual user row in the Watch VU Grid 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.

5.5.3 Managing Synchronization points

If you create scripts that include synchronization points, you can manually release one or more virtual users from any synchronization point. You can release specific virtual users, specific synchronization points, or all virtual users and synchronization points.

5.5.3.1 Releasing Individual Virtual Users

To release an individual virtual user from a synchronization point:

1. Select the virtual user in the Virtual User grid.
2. Click the right-mouse button and select **Release Sync**.

5.5.3.2 Releasing Multiple Virtual Users

To release multiple virtual users from a synchronization point:

1. Select the virtual users in the Virtual User grid.
2. Click the right-mouse button and select **Release Sync**.

5.5.3.3 Releasing Synchronization Points

To release synchronization points:

1. Select **Sync Point Status** from the **Tools** menu to display the Sync Point Status dialog box.
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 name of the synchronization point.
Users At - displays the number of users waiting at the synchronization point.
Status - displays the user's status.
2. Select the Sync Point that you want to release and click **Release** or click **Release All** to release all Sync Points.
3. Click **Close**.

5.5.4 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 3.14.3, "Setting Scenario Defaults"](#) or [Section 4.1.2, "Specifying Scenario Profile Attributes"](#) for details about Standard and Extended log messages.

5.5.5 Viewing Server Warnings

The Oracle Load Testing Server Warnings dialog box displays errors encountered by virtual agents. When an error is encountered, the **OLT Server Warnings** button is displayed at the bottom right-hand corner of the Watch VU Grid. Click on it to display this dialog box.

5.5.6 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.

5.5.7 Stopping the Autopilot From Starting New Virtual Users

To stop the Autopilot from starting new Virtual Users click the **Pause** button

5.5.8 Aborting All Virtual Users

To abort all running Virtual Users click the **Abort All Virtual Users** button.

The virtual users stop immediately without completing the current iteration of the script.

Using Graphs and Reports

Oracle Load Testing lets you evaluate the performance of your Web application in real time using Performance Statistics and graphs on the **View Run Graphs** tab. You can create custom graphs on the **Create Reports** tab for post-run analysis for sessions where data has been saved. The following default reports and graphs are available on the **View Run Graphs** tab:

- **Performance Statistics** - shows a summary of performance statistics data generated by running 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 in the Autopilot. 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.
- **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 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 running in the Autopilot.
- **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 Autopilot runs the virtual users.
- **Errors 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 running in the Autopilot. 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.
- **Errors 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 in the Autopilot. 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 Autopilot ramps up to the total number of virtual users for all scenario profiles.

- **Performance, Users vs. Time** - this graph combines the Performance vs. Time 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.

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.

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 15 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 Autopilot session, the View Run Graphs tab may not have enough time to update the screen with the real time data. However, if you use the Session Start/Stop options to save the data for reporting, 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 Create Reports tab to generate a variety of reports and graphs from the data stored in the database.

6.1 Setting the Reporting Options

To set the Oracle Load Testing Report options:

1. Select **Options** from the **Tools** menu.
2. Select **Reporting** to set additional options.

Intervals - the Intervals options are as follows:

- **Reporting data collection interval** - specifies how often the aggregate Oracle Load Testing agent performance data is collected for reporting to the Oracle Load Testing database.
- **User Interface 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 Logs auto display interval** - specifies how often to refresh the virtual user logs when Auto mode is being used.

VU Grid - the VU Grid options are as follows:

- **Maximum number of 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 - the VU Logs options are as follows:

- **Maximum number of 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 Statistics and Session Report - specifies the default setting for Performance Statistics and Session Reports, as follows:

- **Show Server Times Only** - specifies that only server response time is included in performance statistics and session reports.
- **Show End-to-End Times (include think times)** - 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.

Report Template - the Report Template option is as follows:

- **Delete temporary images/data generated during report template** - when selected, any temporary images or data generated by report templates is automatically deleted.

3. Enter the how often you want data to be collected in the **Reporting data collection interval** field.
4. Enter the how often you want to refresh the user interface in the **User interface 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 how you want timer names to be displayed.
7. Specify whether to include think time in the profile time.
8. Specify the maximum number of data series to include per graph.
9. Click **OK**.
10. Start an Oracle Load Testing Autopilot session.

6.2 View Run Graphs Tab

The View Run Graphs tab is where you can view runtime reports. These graphs are only available while the load test is running. Use the Create Reports tab to view reports and graphs after the load test has finished running.

The View Run Graphs tab is refreshed according to what is set in the **Graph Refresh Interval** setting in the reporting options.

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.

Reports & Graphs - displays options for creating a new graph and viewing default graphs.

- **New Graph** - displays a new tab with filters for customizing the graph. The tab label is the name you enter in the Graph Name field in the Filters section.
- **Overview** - shows a thumbnail view of all graphs. Click on a thumbnail to enlarge it. The Overview tab includes the Default Graphs selector list and the Show Chart Statistics button, as follows:
 - **Default Graphs** - select a graph and click Add Data Series to view a default graph.
 - **Show Chart Statistics** - toggles the chart statistics table view on and off. The chart statistics show the Data Series and values for each series.
- **Reports** - displays the performance statistics for the session.

Custom graph tabs have the following options:

- **Remove Graph** - the (X) button on the tab deletes the selected graph.
- **Session** - displays the session that is running.
- **Available Data Series** - shows a list of session data generated by Oracle Load Testing and ServerStats. The data series listed is a subset of all of the available data series. Click **Show All** to display all available data series.

Default Graphs - select a graph and click Add Data Series 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:

- **PGA/SGA** - groups "Program Global Area (PGA)", "Shared Global Area (SGA)" and "Total Global Area" metrics.
- **CPU vs. DB Sessions** - groups "Average active sessions on CPU", "CPU recursive usage", and "CPU parse time usage" metrics.
- **Transaction Overview** - groups "User login rates per second" and "User Transaction Commit Rate" 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 Autopilot 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 Autopilot 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 Auto generate timers for all resources check box in **Options** from the **Tools** menu then selecting **Scenario**

Defaults. Select the timer(s) and click the arrow button or 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 Monitors - shows the ServerStats counter configuration.

[>] **Add Data Series** - adds the selected data series to the graph. You can move the mouse over a data series line in the graph to view details about specific data points in the graph.

Show Legend - opens a view of the names and plot line color of each data series in the graph. click a data series in the legend popup to show or hide plot lines in the graph. Double click a data series in the legend popup to show /hide the y-axis scale on the right side of the graph.

Show Chart Statistics - toggles the chart statistics table view on and off. The chart statistics show the Data Series and values for each series.

Show Graph Options - opens the options for customizing, exporting, opening saved queries, and saving queries.

Clear Graph - removes all data series from the graph.

Zoom - toggles the zoom tool on and off.

- zoom in - click and drag a bounding box over the area of the graph. The zoom indicator shows on the graph when zoomed in from full size.
- zoom out - press and hold the Ctrl key and click the mouse button to zoom out one step to the previous zoomed size. Press and hold the Ctrl key and double-click the mouse button to zoom out to the full size.

6.3 Using the Performance Statistics

You can view the Performance Statistics report by selecting the **View Run Graphs - Reports** tab or the **Create Reports** tab for sessions for which data has been saved, when you start an Oracle Load Testing session.

View - specifies which performance times to show in the report.

- **Show Server Times Only** - when selected, only server response time is included in performance statistics and session reports.
- **Show End-to-End Times (include think times)** - when selected, think time is included with server times in performance statistics and session reports.

Detail Level - lets you specify the amount of detail shown in the report. This field is only displayed for the Performance Statistics report when viewed in the **View Run Graphs** tab.

The statistics show the values for the following performance categories:

<Session Name> **Current**

- **Active Virtual Users** - the number of virtual users currently running in the Autopilot. Clicking the link opens a Users vs. Time graph in a new sub-tab.

- **Virtual Users with Errors** - the number of virtual users with errors. Clicking the link opens a Errors vs. Users graph in a new sub-tab.
- **Transactions Per Second** - the number of times the virtual user played back the script per second. Clicking the link opens a Trans/sec, Users vs. Time graph in a new sub-tab.
- **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. Clicking the link opens a Pages/sec, Users vs. Time graph in a new sub-tab.
- **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. Clicking the link opens a Hits/sec, Users vs. Time graph in a new sub-tab.
- **Kilobytes Per Second** - the number of kilobytes transferred between the server and browser client per second. Clicking the link opens a KB/sec, Users vs. Time graph in a new sub-tab.

<Session Name> Totals

- **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 and Timer - this section appears when the **Detail Level** is set to Summary + Profile Timers or Full View.

- **<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.
- **Pass** - the number that passed.
- **Fail** - the number that failed.
- **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.

6.4 Using Run Graphs

The Load graphs lets you view real-time performance, statistical, and error information about the virtual users being run by the Autopilot. 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

6.4.1 Viewing Runtime Statistics and Graphs

To view runtime statistics:

1. Click the **View Run Graphs** tab. The **Overview** tab is displayed showing a small view of the default graphs. This is useful for getting an at-a-glance view of all of the run graphs.
2. Click the graph you want to view to enlarge it and display it in the **Reports** tab. You can switch between Load graphs using the **Default Reports** selector list at the bottom of the Overview tab.

To create a custom graph:

1. Click **Graph**.
2. In the **Available Data Series** section, select the data you want to view. Use Ctrl-click and Shift-click to select multiple data series. If you do not see the data series you are looking for, click **Show All** to display all available data series.
3. Click the [**>**] **Add Data Series** button to add the data series to the graph.
4. Click the **Show Legend** button to open a view of the names and plot line color of each data series in the graph.
 - Click a data series in the legend popup to show or hide plot lines in the graph.
 - Double-click a data series in the legend popup to show/hide the y-axis scale on the right side of the graph.
 - Click the **Show Legend** button again to close the legend view.
5. Click the **Show Chart Statistics** 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.

6. Click the **Show Graph Options** button to open the options for customizing, exporting, opening saved queries, and saving queries.

Customize options - these options let you customize the current graph.

- **Graph Tab**

- **Graph Title** - enter the name of the graph. This name is displayed in the tab in the Reports & graphs section.
- **Data Series** - lists the data series contained in the graph. The (x) button marks the Data Series for removal. Click the **Apply** button to Refresh the graph and remove the selected Data Series.
- **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.
- **Data Series Label** - displays the label displayed in the legend.

- **X-Axis Tab**

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 to avoid having to reload the data from the server. The Filter icon appears on the graph when the Range filter is set.

- **Manual** - when selected, the Range Filter minimum and maximum values are enabled. Edit the minimum and maximum values to specify the data points slice to include in the new graph. Click **Apply** to generate a new graph that includes only the data points between the minimum and maximum values.
- **Format** - shows the Range format to use to filter the graph data. The format is based upon the **Scale** setting.
- **Minimum** - enter the minimum X-axis value. This field is only available when **Manual** is selected.
- **Maximum** - enter the maximum X-axis value. This field is only available when **Manual** is selected.

Session Details

- **Start Time** - displays the date and time that the session started.
- **End Time** - displays the date and time that the session ended.
- **Duration** - displays the length of the session.

- **Y-Axis Tab**

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.

- **Displays** - when selected the threshold line will be included in the graph.
- **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.

7. Enter a name for the graph.
8. Select how to plot the graph in the **X-Axis** and **Y-Axis** tabs.
9. Click **Apply**. Each custom report is displayed under a separate tab so that you can select the one you want to view. Click **Remove Graph** to delete the graph and its tab.

6.4.2 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 in the Autopilot.

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.

6.4.3 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 in the Autopilot.

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.

6.4.4 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 in the Autopilot.

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.

6.4.5 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 in the Autopilot.

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.

6.4.6 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 in the Autopilot.

6.4.7 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.

6.5 Using Data Counters

Oracle Load Testing automatically generates a set of counters for Oracle Load Testing virtual user data that can be selected in the **Available Data Series** field in the **Filters** section. The Oracle Load Testing data includes overall data for the session at the top of the list followed by a folder for each Virtual User profile included in the **Configure parameters of the scenario** list of the **Build Scenarios** tab.

Oracle Load Testing also generates counters for each ServerStats monitor in the selected ServerStats configuration. You can create a new graph to monitor any of the counters in the list during the Load Test to view both VU performance and ServerStats monitor performance.

Note: Sessions with Oracle DB Metrics and Enterprise Manager URL that were run using an Oracle Load Testing version prior to version 9.30 will not automatically show the links for ADDM & AWR reports. To show the ADDM & AWR links under Enterprise Manager Diagnostics, generate the ADDM & AWR reports using the Reports sub-tab of the Create Reports tab.

6.5.1 Using Data Counters

The Virtual User profile folders contain the data counters and page timer folders for the specific profile in the load testing Scenario.

You can create a new ServerStats chart to monitor any of the counters in the list during the Load Test.

The page timer folders contain the performance data for the individual pages of scripts used as a Virtual User profile, similar to the following:

You can create a new ServerStats chart to monitor any of the timers in the list during the Load Test.

6.6 Create Reports Tab

The Create Reports tab is where you can view historical reports and graphs for sessions for which you have saved data for reporting. This tab has the following options:

Reports & Graphs - displays default and custom graphs, and gives options for exporting graphs.

- **New Graph** - opens a blank tab and displays the filter options for creating a new graph.
- **Overview** - displays a thumbnail view of each created graph.
- **Reports** - displays the Session Performance and Hardware Estimation reports. These reports can be exported to CSV or printed.
- **Graph <#>** - displays custom graphs that you have created. These graphs can be exported in CSV, Excel, PNG, or JPG format.

The Reports tab has the following options for Session Reports:

- **Session** - lets you select the session for which you want to view the session report.
- **Report** - if you have created custom report templates, they are listed here so that you can select them and display a report using the template. Refer to Creating Report Templates later in this chapter for instructions on how to create a report template. The default is 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 template.
- **Generate** - generates the report based on the session and report template selected.
- **Show Server Times Only** - when selected, only server response time is included in performance statistics and session reports.
- **Show End-to-End Times (include think times)** - when selected, think time is included with server times in performance statistics and session reports.
- **Export to CSV** - opens the data in a separate browser window in comma separated value format.
- **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 options for (hardware) Estimation Reports:

- **Repository** - lets you select the repository containing the saved scenario file to use for the hardware estimation report.
- **Folder** - lets you select the workspace folder containing the saved scenario file to use for the hardware estimation report.
- **File** - lets you select the saved scenario file to use for the hardware estimation report.
- **Generate** - generates the report based on the scenario file selected.

- **Export to CSV** - opens the data in a separate browser window in comma separated value format.
- **Print/Save HTML** - opens the hardware estimation report in a separate browser window. From there you can either print the report or save it in HTML format.

Custom graph tabs have the following options:

- **Remove Graph** - deletes the selected graph.
- **Session** - lets you select the session for which you want to view data series.
- **Available Data Series** - shows a list of session data generated by Oracle Load Testing and ServerStats. The data series listed is a subset of all of the available data series. Click **Show All** to display all available data series.
- **Default Graphs** - select a graph and click Add Data Series 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:
 - **PGA/SGA** - groups "Program Global Area (PGA)", "Shared Global Area (SGA)" and "Total Global Area" metrics.
 - **CPU vs. DB Sessions** - groups "Average active sessions on CPU", name="Oracle: CPU recursive usage", and "CPU parse time usage" metrics.
 - **Transaction Overview** - groups "User login rates per second" and "User Transaction Commit Rate" 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 Autopilot session. Select the counter(s) and click the **Add Data Source** button or double-click to add counters(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 Autopilot 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 Auto generate timers for all resources check box in **Options** from the **Tools** menu then selecting **Scenario Defaults**. Select the timer(s) and click the arrow button or double-click with the mouse to add timer(s) to the Selected Data Series list.
 - **ServerStats Nodes** - list the 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 Monitors** - shows the ServerStats counter configuration.
- **[>] Add Data Series** - the [>] button adds the selected data series to the graph. You can move the mouse over a data series line in the graph to view details about specific data points in the graph.
- **Show Legend** - opens a view of the names and plot line color of each data series in the graph. click a data series in the legend popup to show or hide plot lines in the graph. Double click a data series in the legend popup to show/hide the y-axis scale on the right side of the graph.
- **Show Chart Statistics** - toggles the chart statistics table view on and off. The chart statistics show the Data Series and values for each series.
- **Show Graph Options** - opens the options for customizing, exporting, opening saved queries, and saving queries.
- **Clear Graph** - removes all data series from the graph.
- **Zoom** - toggles the zoom tool on and off.
 - zoom in - click and drag a bounding box over the area of the graph. The zoom indicator shows on the graph when zoomed in from full size.
 - zoom out - press and hold the Ctrl key and click the mouse button to zoom out one step to the previous zoomed size. Press and hold the Ctrl key and double-click the mouse button to zoom out to the full size.

6.6.1 Viewing Reports

You can quickly view session data by creating custom graphs.

1. Click the **Create Reports** tab. A blank tab is displayed for creating a new graph.
2. Select the session you want to graph.
3. Select the data you want to view in the **Available Data Series** field. Use Ctrl-click and Alt-Click to select multiple data series.
4. Click **Add Data Series**.
5. Repeat for all of the data series you want to graph.
6. Click the **Show Graph Options** button to display the Graph Options for naming the graph and selecting how to plot the graph.
7. Click **OK**.
8. Each custom report is displayed under a separate tab so that you can select the one you want to view. Click the **Remove Graph** button located on the graph's tab to delete the graph and its tab. Click the **Show Graph Options** button and use the export options to save a graph or the graph data to a file.

You can save this query as described later in this chapter for later use against another session.

6.6.2 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. Click the **Show Graph Options** button on the right of the graph.

3. Select **Export**. The Export options are as follows:

Format - specifies the file format to export.

- PNG - when selected, the exported graph image is saved in Portable Network Graphics file format. Specify the size of the image file.
- JPG - when selected, the exported graph image is saved in Joint Photographic Experts Group compression file format. Specify the size and quality level of the image file.
- CSV - when selected, the exported graph data is saved in Comma Separated Value file format. You can open the file in Microsoft Excel or save it to a file.
- XLS - when selected, the exported graph data is saved in Microsoft Excel file format. You can open the file in Microsoft Excel or save it to a file.

Size - specifies the pixel size of exported PNG and JPG image files.

Quality - specifies the quality level of exported JPG image files. The quality level dictates the ratio between file size and image clarity. A lower quality level produces a smaller file size but the image may lose clarity. A higher quality level produces a higher clarity image but with a larger file size. Experimentation may be the best way to determine the best acceptable quality level for your image file size and clarity requirements if the default quality level is insufficient.

6.6.3 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. Create a report.
2. Click the **Show Graph Options** button on the right of the graph.
3. Select **Export**.
4. Select **CSV** format and click **Export**. The File Download dialog box is displayed.

5. Click **Open** to open the file in Excel in CSV format. Click **Save** to save the file in CSV format.
6. If you chose to save the file, the Save As dialog box is displayed. Enter a new name for the file or accept the default, select the location to save to, and click **Save**.

6.6.4 Exporting to Excel

To export as an Excel Workbook file:

Note: The Oracle Load Testing Server must have Microsoft Excel installed to export to Excel format.

1. Create a report.
2. Click the **Show Graph Options** button on the right of the graph.
3. Select **Export**.
4. Select **XLS** format and click **Export**. The File Download dialog box is displayed.
5. Click **Open** to open the chart in Excel. Click **Save** to save the chart in Excel format.
6. If you choose to save the file, the Save As dialog box is displayed. Enter a name for the file or accept the default, select the location to save the file, and click **Save**.

6.6.5 Exporting to PNG

You can export reports to a PNG file. To export to a PNG file:

1. Create a report.
2. Click the **Show Graph Options** button on the right of the graph.
3. Select **Export**.
4. Select **PNG** format and set the size.
5. Click **Export**.
6. Enter a name for the file and click **Save**.

6.6.6 Exporting to JPG

You can export reports to a JPG file. To export to a JPG file:

1. Generate a report.
2. Click the **Show Graph Options** button on the right of the graph.
3. Select **Export**.
4. Select **JPEG** format and set the size and quality level.
5. Click **Export**.
6. Enter a name for the file and click **Save**.

6.6.7 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.

6.6.7.1 Saving Graph Queries

To save a graph query:

1. Select the sessions and data series that you want to graph.
2. Click the **Show Graph Options** button on the right of the graph.
3. Select **Save Query** and 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 **Save**.
5. To create a new query, enter a name for the query and click **Save**. To overwrite an existing query, select the query and click **Save**, then click **Yes** when asked to confirm.

6.6.7.2 Running a Saved Query

To run a saved query:

1. Click the **Show Graph Options** button on the right of the graph.
2. Click **Open Query**.
<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.
Add to Current Graph - when selected, the new session data is added to the graph. When deselected, the graph is cleared and the new session data is displayed.
3. Select the query you want to use.
4. Select the session against which you want to run the query.
5. Select whether or not to add this session data to the current graph.
6. Click **OK**.
7. 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.
8. The new graph is generated.

6.6.7.3 Editing Queries

To edit a query:

1. Click **Graphs** from the **Manage** menu.
Edit - displays the Edit Graph Query dialog box.
Delete - deletes the selected query.

Clone - displays the Edit Graph Query dialog box with the configuration of the selected query. This lets you copy the selected query and edit it to create a new query.

Query Name - lists the existing queries.

2. Select the query you want to change and click **Edit**.
3. Make and changes and click **OK**.
4. Click **Close** to exit the Graph Query Manager dialog box.

6.6.7.4 Deleting Queries

To delete a query:

1. Click **Graphs** from the **Manage** menu.
2. Select the query you want to delete and click **Delete**.
3. Click **Yes** to confirm the deletion.
4. Click **Close** to exit the Graph Query Manager dialog box.

6.6.8 Creating Report Templates

You can create custom report templates that you can import to Oracle Load Testing. These templates are shown on the Reports tab in the Create Reports tab. Once imported you can select a session, select the report template, and generate the report.

Report templates are created using the Oracle Load Testing Report Generator tool in Microsoft Word. This tool can be installed on any Windows machine that has Microsoft Word. Oracle Application Testing Suite does not have to be installed to use the tool.

Once installed, report templates are created in Microsoft Word and then imported to Oracle Load Testing. Then generated reports are displayed in Microsoft Word.

To install the Report Generator tool:

1. Close Microsoft Word.
2. Run the OLTOfficeToolbarInstaller.exe program located in the `<installdir>\tools\ms-office\` folder in the Oracle Application Testing Suite installation directory.

The **Insert OLT Data** button is added to the toolbar in Microsoft Word.

You can now create report templates. Report templates can contain text, graphics, Oracle Load Testing graphs and session data. In addition, you can include custom Java code. To create a report template:

1. Click the **Insert OLT Data** button to display the following dialog box.

This dialog box has the following options:

Graphs - lists the default Oracle Load Testing graphs or a custom list of graphs if you have created a template.properties file.

Optional Settings - leave these settings blank to use the defaults.

- **Title** - enter a title for the graph.
- **Height** - enter a height for the graph.
- **Width** - enter a width for the graph.

Session data - select the session data that you want to display, session name, overview, page data, or profile data. The generated data is comma-separated so that you can easily convert it to a table by selecting **Convert** from the **Table** menu then select **Text to Table**.

Custom code - this section can be used to create Java code using the methods that are available in the Report Generator API. Documentation for this API can be accessed from the online help table of contents.

Note: To move to the next line use CTRL-Enter not Enter. Enter exits the dialog box.

2. Select the graph or session data that you want to display or enter custom code.
3. Click **Insert**. A field code is inserted into the document and it is displayed as a comment. To edit the field code, select it, right-click on it and select **Edit Field**.
4. In the template, press **Enter** to move to the next line.
5. When you are finished creating the template, click Close to exit the Insert Oracle Load Testing Reference dialog box.
6. Click **Save** from the **File** menu. Enter a name for the file and save it as file type .rtf. The file can be saved to any location.

You can now import the file to Oracle Load Testing and start generating reports. To import the file:

1. In Oracle Load Testing, select **Import** from the **Tools** menu.
2. In the Type field, select **Report Template**.
3. Click **Browse**.
4. Select the report template and click **Open**.
5. Click **Import**.
6. Click **Close**. The template will be listed on the Reports tab in the report list.
7. To generate a report, select the session, select the template, and click **Generate**. The report is generated and displayed in Word.

6.6.8.1 Creating a Custom Graph and Session List

You can change the list of graphs and session data that is available in the Insert OLT Reference dialog box by creating a template.properties file in the installdir/data/olt/reports folder. If this file exists, the graphs and session data listed in it are used. If it does not exist, the defaults are used. This file lists the graphs and session data that you want to make available. For example:

```
[GRAPH]
Performance vs. Time@<%out.write(session.getGraph('Performance vs. Time'))%>
Errors vs. Time@<%out.write(session.getGraph('Errors vs. Time'))%>
Statistics vs. Time@<%out.write(session.getGraph('Statistics vs. Time'))%>
[SESSION]
Session Name@<%=session.getName()%>
Overview@<%=session.getOverview()%>
```

6.7 Generating Summary Reports

You can generate summary reports containing statistical information about the last data collection and reporting session and the Scenario Profile settings.

On the **Create Reports** tab, select the **Reports** tab and then select the session for which you want to view reports and click **Generate**. The Session Performance report is displayed followed by the Oracle Load Testing Scenario report.

- **Scenario Report** - displays a summary of Scenario settings and options that were used to run virtual users.
- **Session Report** - displays a summary of performance and error statistics data for the virtual user session.
- **Estimation Report** - displays a hardware estimation report for the selected saved scenario file.

The following sections explain more about the summary reports.

6.7.1 Scenario Reports

The generated report file is the same name as the Scenario in Oracle Load Testing (i.e. the Scenario Description field of the Scenario tab).

To save the report, click **Print/Save: HTML**. The report is displayed in a separate browser window. Select **Save As** from the **File** menu to save the report in HTML or text format.

You can print the report by clicking **Print/Save HTML**. The report is displayed in a new browser window. Select **Print** from the **File** menu. A print dialog box opens for setting the printer options. The available options are specific to the type of printer. Set the options and click the appropriate **OK** or **Print** button.

To save the Scenario as a different name:

1. Select **Save As** from the **Scenario** menu.
2. Enter a new Name in the Name field of the **Save Scenario** dialog box.
3. Select the workspace to save the report in and click **OK**.

6.7.2 Session Reports

To save the report, click **Print/Save: HTML**. The report is displayed in a separate browser window. Select **Save As** from the **File** menu to save the report in HTML or text format.

You can print the report by clicking **Print Friendly**. Select **Print** from the **File** menu. A print dialog box opens for setting the printer options. The available options are specific to the type of printer. Set the options and click the appropriate **OK** or **Print** button.

6.7.3 Hardware Estimation Reports

On the **Create Reports** tab, select the **Reports** tab and then select the scenario file for which you want to view reports and click **Generate**.

The Hardware Estimation Report provides the following:

- Hardware configuration for each agent, including CPU information, memory size, and network bandwidth limitations.

- Process Runtime Configuration for each agent, including Operating System information, Max Virtual Users per Process setting, and Max heap size setting.
- Estimated consumption for each script running on the corresponding agent, including CPU usage, heap size usage, and network bandwidth usage.
- Suggested Configuration, including idle resources monitored before start estimation, estimated number of required machines, and suggested configurations for better performance.

See [Section 4.3.7.2, "Generating Hardware Estimation Reports"](#) for additional information.

6.7.4 Running Automatic Database Diagnostic Monitor (ADDM) Reports

The Automatic Database Diagnostic Monitor (ADDM) analyzes data in the Automatic Workload Repository (AWR) to identify potential performance bottlenecks. For each of the identified issues, it locates the root cause and provides recommendations for correcting the problem.

To create an ADDM Report:

1. Create a load testing script in OpenScript (for example, "script1") against an application using an Oracle Database.
2. Select **Systems** from the **Manage** menu and add a new Monitored System in Oracle Load Testing (for example, "db1") corresponding to your application's database server, and set the Database data source properties to that of the database.
3. Create a ServerStats configuration (for example, "config1") and apply the Oracle Database profile to it using "db1".
4. Add "script1" to an Oracle Load Testing scenario.
5. Apply "config1" to the scenario in the ServerStats Configuration of the Set up Autopilot tab.
6. Click the Run button and select "Yes" to perform a baseline for oracle DB. After the load test finishes, the "ADDM Report" selection will appear in the selection list of the Create Reports - Reports subtab.
7. Click the **Create Reports** tab.
8. Click the **Reports** tab.
9. Select the session for which you want to view the report from the **Session** option list.
10. Select **ADDM Report** from the **Report** option list.
11. Click **Generate**.

6.7.5 Running Automatic Workload Repository (AWR) Reports

For applications under test using Oracle Database and using Enterprise Manager to monitor the Oracle database performance, Enterprise Manager provides data on various tiers in the application, including Automatic Workload Repository (AWR). Oracle Load Testing provides integration points with Enterprise Manager that can drill down to AWR reports.

You will need to know the snapshot id and the "target" domains to get to the correct location within Enterprise Manager to view reports for the duration of load test.

Statistics are collected continuously during database operation and show statistics in the form of a report called AWR Report for performance analysis. The AWR report provides a detailed report of the statistics collected between two snapshots. The snapshots do not have to be consecutive. Any period of time bordered by AWR snapshots can be analyzed.

To create an AWR Report:

1. Create a load testing script in OpenScript (for example, "script1") against an application using an Oracle Database.
2. Select **Systems** from the **Manage** menu and add a new Monitored System in Oracle Load Testing (for example, "db1") corresponding to your application's database server, and set the Database data source properties to that of the database.
3. Create a ServerStats configuration (for example, "config1") and apply the Oracle Database profile to it using "db1".
4. Add "script1" to an Oracle Load Testing scenario.
5. Apply "config1" to the scenario in the ServerStats Configuration of the Set up Autopilot tab.
6. Click the Run button and select "Yes" to perform a baseline for oracle DB. After the load test finishes, the "AWR Report" selection will appear in the selection list of the Create Reports - Reports subtab.
7. Click the **Create Reports** tab.
8. Click the **Reports** tab.
9. Select the session for which you want to view the report from the **Session** option list.
10. Select **AWR Report** from the **Report** option list.
11. Click **Generate**.

6.7.6 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 **Create Reports** tab, then click the **Reports** tab.
2. Select the session and click **Generate**.
3. After the report loads, click the **Filter** icon. The following options are available:
 - Time Range** - specifies the date and time range to use to filter the report.
 - **Custom (M/d/yyyy HH:mm:ss)** - when selected, the minimum and maximum date and time ranges can be set for the filter.
 - **Minimum** - specifies the start date and time for the data to include in the report. The specified minimum must be within the Session Start Time and the Session End time.
 - **Maximum** - specifies the end date and time for the data to include in the report. The specified maximum must be within the Session Start Time and the Session End time.

Session Details - shows the date and time range details for the session data.

- **Session Start Time** - shows the date and time when the session was started.
 - **Session End Time** - shows the date and time when the session ended.
 - **Duration** - shows the length of time the session was running.
4. Select the **Custom** option and set the minimum and maximum date and time values to the time range to include in the report.
 5. Click **OK** to generate a new report that include only the data between the minimum and maximum date and times.

6.8 Managing Session Data

To delete session data from the database or edit your session file names and descriptions:

1. Select **Sessions** from the **Manage** menu to display the Session Manager.

This dialog box lets you manage session data in the database.

Edit - displays the Edit Session dialog box for changing the name and description of the session.

Delete - deletes the entire selected session data from the database. To select more than one session, hold down the CTRL key.

Delete Virtual User Logs - deletes only the VU log data for the selected sessions. To select more than one session, hold down the CTRL key.

Session Name - the names of the virtual user sessions run by the Oracle Load Testing Autopilot. Oracle Load Testing lets you specify a name for the session data when you start the Autopilot 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.

Description - shows the status of the session. One of the following status entries may appear:

- **[Blank]** - 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.
- **end of session** - the data for the session has been committed to the database and is ready for reporting.
- **saving data** - session data is being saved.

VU Logs - indicates whether VU log data was saved for this session.

ADDM/AWR Data - indicates whether the session included Oracle Database metrics and enabled baseline in order to generate Automatic Database Diagnostic Monitor (ADDM) and Automatic Workload Repository (AWR) reports. This will indicate No if the session is still running even if the session includes Oracle Database metrics and enabled baseline because the ADDM/AWR reports cannot be generated while a session is running.

2. Select the Session Name to delete. To select more than one session, hold down the CTRL key.

3. Click **Delete** to delete the session or click **Delete Virtual User Logs** to delete the VU log only.

6.9 Using Raw Data

This section explains how to use the raw data files that can be generated by a load test session. The raw data provides the ability to upload non-aggregated data collected from all Virtual Users running in a load test across all Load Agent machines for reporting.

A database administrator can then create tables in an Oracle database (Oracle RDBMS only) and import the CSV files into the tables. Database administrators and end users can execute SQL queries against the raw data tables to generate reports.

6.9.1 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 **Tools** menu.
2. Select **Scenario Defaults**.
3. Set the **Persist Raw Data** setting to **True** in the **Java Client 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 **Configure all parameters** button next to the first script/profile.
3. Set the **Persist Raw Data** setting to **True** in the **Java Client 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).

6.9.2 Working with Raw Data Files

When raw data collection is enabled, Oracle Load Testing saves every single measured data point in a set of CSV files. However, ServerStats data is not included in the CSV files as the data is not present on the agent. The files are saved locally on the agent machines in directories specified as follows:

```
<oats_install>/agent/rawdata/<controller-identifier>/<session_
name>/<agent-id>/<YYYY-MM-DD HH:mm:ss>
```

All raw data files in the agent machine directories remain there and are not delivered to the Oracle Load Testing controller, reconciled between the agents of the same session, or manipulated in any way after the capture is complete. Management of the raw data files must be performed manually.

The files are as follows:

- `counter_definitions.csv` - for every session where the **Persist Raw Data** option is enabled, for every agent where the session is running, every counter definition will be saved in this CSV file. This file has the following fields:
 - ID - a unique identifier for each counter definition record in the file.
 - SOURCE_ID - the ID of the virtual user. For example, if a session is running 10 virtual users, there will be source IDs numbered 1 through 10.
 - SESSION_NAME - the name of the load test session.
 - PROFILE_NAME - the name of the profile run in the load test session.
 - TIMER_NAME - the name of the Oracle Load Testing timer.
 - COUNTER_NAME - the name of the Oracle Load Testing counter.
 - AGGREGATION_TYPE_NAME - the aggregation type for each counter as follows:
 - * 0, Current - the counter is a current value counter.
 - * 1, Cumulative - the counter is a cumulative value counter.
 - * 2, Average - the counter is an average value counter.
 - * 3, Min - the counter is a minimum value counter.
 - * 4, Max - the counter is a maximum value counter.
 - * 5, No data - the counter has no data.
 - * 6 Last - the counter is a last value counter.
 - CONTROLLER_TIME_DELTA_MS - this is a time stamp offset value for multiple agent machines. It is used to indicate the difference between time stamps (AGENT_TME_MS) on different agent machines and the Oracle Load Testing controller. For example if the controller has a time stamp of "1" and an agent machine has a time stamp of 2, the delta would be 1. if another agent machine has a timestamp of 3 the delta would be 2. Typically in milliseconds.
- `counter_data.csv` - for every session where the **Persist Raw Data** option is enabled, for every agent where the session is running, every counter value for every counter will be saved in this CSV file.

There are some conditions in which *all* of the data is not captured for the particular session. How quickly a session is shutdown may affect the raw data capture. This file has the following fields:

- ID - corresponds to the Counter Definition ID in the `counter_definitions.csv` file.
- AGENT_TIME_MS - a timestamp indicating the first time an agent starts reporting. It indicates the time, in EPOCH time, when a session request arrived on an agent.
- VALUE - the value for the counter. The AGGREGATION_TYPE_NAME values in the `counter_definitions.csv` file specifies the type of value.

6.9.3 Creating Tables in an Oracle Database

Once you have generated the raw data files by running an Oracle Load Testing scenario with the **Persist Raw Data** setting enabled, you can create tables in an Oracle Database for the counter definitions and counter data CSV files. This section explains the basic procedure for creating tables. [Section 6.9.5, "Example SQL Command File"](#) provides an example of the SQL commands required to create the tables in the database.

1. Create a SQL command file or use the `RawTables.sql` command file provided with the Oracle Application Testing Suite installation.
2. Copy your command file or the `RawTables.sql` file to `Oracle Home\BIN`.
3. Connect to the Oracle Load Testing database schema (e.g. `olt/password`) using SQL Plus.
4. Enter following command in SQLPlus (use the appropriate path for where your SQL command file is located):

```
@C:\OracleATS\oxe\app\oracle\product\10.2.0\server\BIN\RawTables.sql
```

6.9.4 Importing Raw Data CSV Files

Once the raw data tables have been created in the database, you can import the counter definitions and counter data CSV files into the database. This section explains the basic procedure for importing the CSV files into the database.

1. Create a control file to import the counter definitions CSV file or use the `RawDataDef.ctl` command file provided with the Oracle Application Testing Suite installation. [Section 6.9.6, "Example Counter Definition Control File"](#) provides an example of the control file required to import the counter definitions CSV file into the database.
2. Create a control file to import the counter data CSV file or use the `RawData.ctl` command file provided with the Oracle Application Testing Suite installation. [Section 6.9.7, "Example Counter Data Control File"](#) provides an example of the control file required to import the counter data CSV file into the database.
3. Copy your control files or the `RawDataDef.sql` and `RawData.ctl` files to `Oracle Home\BIN`.
4. Execute the following commands:

```
sqlldr olt/yourpassword control=RawDataDef.ctl
sqlldr olt/yourpassword control=RawData.ctl
```

6.9.5 Example SQL Command File

This section provides an example of the SQL commands used to create tables in an Oracle database.

```
CREATE TABLE counterRun_rawdata (
  counterHdrID int NULL,
  counterVal float NULL,
  counterTS timestamp(3) NULL,
  iteration int NULL,
  numberVus int NULL,
  errFlag int NULL
)
;
```

```

CREATE TABLE counterHdr (
    counterHdrID int NOT NULL,
    sessionRunID int NULL,
    source int NULL,
    machineName nvarchar2 (255),
    counterName nvarchar2 (255),
    objName nvarchar2 (255),
    instName nvarchar2 (255),
    niceName nvarchar2 (255),
    aggregationType int NULL,
    CONSTRAINT counterHdr_PK PRIMARY KEY (counterHdrId)
)
;

create or replace FUNCTION GETSESSIONID
( SessionName1 IN VARCHAR2
) RETURN NUMBER AS

ID number(38):=0;
BEGIN
--dbms_output.put_line('before 1' || length(SessionName1)|| ' ' || SessionName1);
select sessionrunid into ID from sessionrun where sessionname =trim(SessionName1);
--dbms_output.put_line('end');
RETURN ID;
EXCEPTION
    WHEN OTHERS THEN
        -- DBMS_OUTPUT.put_line(sqlerrm);
        -- dbms_output.put_line(dbms_utility.format_error_backtrace);
    RETURN 0;
END GETSESSIONID;
/

```

6.9.6 Example Counter Definition Control File

This section provides an example of a control file used to input a counter definition CSV file into an Oracle Database (use the appropriate path for where your control file is located).

```

Options (Skip = 1)
Load Data
  infile 'C:\OracleATS\oxe\app\oracle\product\10.2.0\server\BIN\counter_
definitions.csv'
replace into table COUNTERHDR
  fields terminated by ","          OPTIONALLY ENCLOSED BY '"'
(
    COUNTERHDRID,
    source, sessionrunid "getSessionID(:sessionrunid)",      objname,
    INSTNAME,
    COUNTERNAME,
    AGGREGATIONTYPE "decode(trim(:AGGREGATIONTYPE), 'Average',2, 'Cumulative',
1, 'Current',0, 'Min',3, 'Max' ,4, 'No data',5, 'Last' ,6,-1)" ,
    NICENAME "decode(:COUNTERNAME, null, :COUNTERNAME, :COUNTERNAME)"
)

```

6.9.7 Example Counter Data Control File

This section provides an example of a control file used to input a counter data CSV into an Oracle Database (use the appropriate path for where your control file is located).

```
Options (Skip = 1)
infile 'C:\OracleATS\oxe\app\oracle\product\10.2.0\server\BIN\counter_data.csv'
replace into table COUNTERRUN
fields terminated by ","          OPTIONALLY ENCLOSED BY ''
(
    COUNTERHDRID,
    COUNTERTS "new_time( to_date('01011970', 'ddmmyyyy') + 1/24/60/60 *
:COUNTERTS/1000, 'GMT', 'GMT')",
    COUNTERVAL
)
```

Using the Virtual User Logs

The virtual user logs lets you monitor the progress of virtual users as they are running or view errors as they occur.

The following is a summary of the virtual user logs features:

- View virtual user playback as pages are accessed and displayed;
- View only selected virtual user pages;
- View only pages with errors.

7.1 Starting the Virtual User Logs Viewer

To start the Virtual User Logs viewer:

1. Specify Scenarios.
2. Submit the Scenario to the Autopilot.
3. Start the Autopilot.
4. Select **VU Logs** from the **Tools** menu.

7.2 Overview of the Main Window

The main window of the virtual user logs is where you select the virtual user actions to view. The main window has the following parts:

- Toolbar
- Summary Views
- All Logs view

7.2.1 Overview of the Toolbar Options

The virtual user logs window has the following toolbar options:

Open - Opens an existing virtual user log.

Save as CSV - Exports the current VU log to a comma-separated value file.

Clear - Clears the virtual user logs.

View Text - Displays the source HTML for pages accessed by the virtual user.

Render HTML - Displays actual pages as accessed by the virtual user.

Auto Display - Displays pages as they are received by the viewer.

Manual Display - The VU log stays on the selected page and does not go to the next page; however, it continues to collect pages.

Refresh On - Allows the viewer to collect new pages that are being accessed by the virtual user.

Refresh Off - Stops the viewer from accepting new pages being accessed by the virtual user.

Navigate to First Page - Displays the first page in the log.

Navigate to Previous Page - Navigates to the previous page in the log.

Navigate to Next Page - Navigates to the next page in the log.

Navigate to Last Page - Displays the last page of the log.

[View Buttons] - Selects the view to display: Summary view, Summary by VU view, or All Logs view.

Filter - Displays and hides the filter pane on the main screen. The filter fields corresponds to the table columns and can be used to enter particular values to filter the tabular data based upon specified values.

Details/Content - Displays or hides a text area used to display the contents of messages. You can only select a button if it is enabled. Only one can be selected at a time. If neither button is selected, the text area is hidden.

The Details button shows a text report that contains all the values of fields of the currently selected row in the particular view's table.

The Content button shows any message associated with a log entry. For example a message with type ResponseContent is associated with the full response content received by the agent. A message with type Parameter substitution is associated with a string specifying all the parameter/value pairs. Any messages with contents are shown in the contents pane.

Level/Threshold - The level selector filters the tabular data to the selected levels. If Threshold is cleared, only the messages of the selected levels will be shown. If Threshold is selected, all the messages of the selected level and the levels of higher priority are shown.

7.2.2 Overview of the Summary View

The Summary view provides an overview of how many times a particular type message has occurred and by what number of Virtual Users. The Summary view shows the following information:

[Level image] - Indicates the message level. The level can be Info, Debug, Warning, or Error.

Sample Message - Shows the message of the first record of the group as a sample 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.

Occurrence - Specifies the number of time this message has occurred.

Vusers - Specifies the number of distinct Virtual Users that have thrown this message.

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.

7.2.3 Overview of the Summary by VU View

The Summary by VU view provides an overview of how many times a particular type message has been thrown by a particular VU. The Summary by VU view 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.

Sample Message - Shows the message of the first record of the group as a sample 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.

Occurrence - Specifies the number of time this message has occurred.

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.

7.2.4 Overview of the All Logs View

The All Logs view 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:

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.

7.3 Opening VU Logs

You can open an existing VU log for viewing in the VU Logs viewer. To open a VU log:

1. Start the VU Logs Viewer.
2. Click **Open**.
Select from Server - lets you open a log file that is saved on the server. Select the session from the list.
Select from local machine - lets you open a log file that has been saved on the local machine. Click **Browse** to navigate to the file.
3. Select whether you want to open a log file on the server or on the local machine.
4. Select the log that you want to execute in the virtual user logs. If you are opening a log file from the local machine either enter it's path or click **Browse** to navigate to the file.
5. Click **OK**.

7.4 Saving Log Files

You can save VU log to a CSV file. To save log files:

1. Click **Save to CSV**. The File Download dialog box is displayed.
2. Click **Save**. The Save As dialog box is displayed.
3. Navigate to the location where you want to save the file.
4. Click **Save**.
5. Click **Close** when the download is complete.

Database and File Management

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.

8.1 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.

8.1.1 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.

8.1.2 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.

8.1.3 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.

8.1.4 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**.

8.1.5 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.

8.1.5.1 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
```

8.1.5.2 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 8.1.1, "Adding an Oracle Database"](#) for additional information.

8.1.5.3 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
```

8.2 Importing Files

You can import files from the local machine to the Oracle Load Testing server. This can be useful if you are accessing Oracle Load Testing on a remote machine via the Web UI and want to import scripts from your local Oracle OpenScript.

1. Select **Import** from the **Tools** menu.

File Type - select the type of file.

- **Type** - lists the types of files that can be imported.

File to Import - the options listed here depend on the type of file being imported.

- **<extension>** - enter the file you want to import. For MIB files, the file must have a .mib file extension to be recognized in the Import File dialog box. If you have a MIB file that does not have the .mib file extension, you will need to rename the file to include the .mib file extension.
- **Browse** - opens the Choose File dialog box for browsing and selecting a file.

Destination - the options displayed here depend on the type of file you are importing and only apply to certain file types.

- **Repository** - select the repository where you want to import the file.
- **Workspace** - select the workspace where you want to import the file.

2. Select the type of file that you want to import.

Note: If you are importing a ServerStats Metric Profile, the metrics associated with the profile must be imported separately.

3. Enter the file name or click **Browse** to locate the file.
4. Select the destination location to which you want to copy the file.
5. Click **OK**.
6. Click **OK**.

8.3 Exporting Files

You can export files from the server to the local machine. This is useful if you want to access a script from the Oracle Load Testing Server and edit it on your local machine.

1. Select **Export** from the **Tools** menu.

File Type - select the type of file you want to export from the Oracle Load Testing server to the local machine.

Note: Exporting ServerStats configurations does not export the metrics required for those ServerStats configurations. You must explicitly download any custom metrics defined by user.

Repository - a list of repositories from which you can download files.

Folder - lists the folders available in the selected repository.

<file list> - lists the files of the selected type in the selected folder that are available to export.

Recorded Data - when selected, recorded data is included in the exported scenario package zip file.

Playback Result - when selected, playback result data is included in the exported scenario package zip file.

Hardware Estimation Data - when selected, hardware estimation data is included in the exported scenario package zip file. This data is used with the Testing as a Service (TaaS) features of the Oracle Cloud.

Databank - when selected, databank playback result data is included in the exported scenario package zip file.

2. Select the type of file that you want to export.
3. Select the Repository and Workspace in which the file is located if applicable to the type of file you are exporting.
4. Select the file you want to export. If you are exporting scripts, the Download Multiple Files dialog box is displayed.

This dialog box lists script parts available for exporting for the selected script. Select the part you want to export and click **Close**.

5. Select the script part that you want to export. Script parts must be exported separately.
6. The File Download dialog box is displayed.
7. Click **Save** to open the Save As dialog box.
8. Select the location where you want to save the file and click **Save**.
9. Click **Close** in the Download Complete dialog box.
10. If you are downloading scripts, select another script part to download or click **Close** to exit the Download Multiple Files dialog box.
11. Select another file to download or click **Close** to exit the Download File dialog box.

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.

9.1 Attaching to a Session

You can attach to another running session on the Oracle Load Testing Server. To attach to a session:

1. Select **Attach** from the **Sessions** menu.

<session list> - lists all sessions that are running on the Oracle Load Testing Server. Select a session and click **OK**.

Attach mode - select how you want to attach to the session:

- **View Only** - select this option to attach to the session in view only mode. This mode allows you access to graphs.
 - **Full Control** - select this option to attach to the session with full control. This option allows you to control tests.
2. Select the session to which you want to attach.
 3. Select the mode in which you want to attach.
 4. Click **OK**.

9.2 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.

9.3 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 **Export** from the **Tools** menu.
2. Select **OLT Session Data** as the **Type**. The **Source** list shows the list of sessions stored in the Oracle Load Testing database. The Source list uses the following markers to indicate if a session can be exported or not:
 - Check mark: indicates the session can be exported.
 - X-mark: indicates the session cannot be exported. Sessions that cannot be exported include running sessions, sessions currently being imported, and sessions in a Reservation status.
3. Select the session to export and click **Export**. You can also double-click a session name to start the export.
4. Click **Save** when the File Download dialog box appears.
5. Specify the location and file name and click **Save**.

9.3.1 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  
    volog.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.

9.4 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 **Import** from the **Tools** menu.

2. Select **OLT Session Data** as the **Type**. The **File To Import** list shows the list of sessions stored in the current Oracle Load Testing instance database.
3. Click **Browse** in the **File To Import** section and select the session `.osd` file to import and click **Open**.
4. Click **Import**. The Session Import Wizard shows the session details.

Session Details - shows the details for the data included in the `.osd` file being imported, as follows:

 - **Name** - shows the name of the load testing session.
 - **Start Time** - shows the date and time the session started.
 - **End Time** - shows the date and time the session ended.
 - **Machine** - shows the name of the machine on which the session ran.
 - **VUDisplay Data** - shows if Oracle Load Testing virtual user logs data is included in the session data `.osd` file being imported or not. Yes indicates the `.osd` file includes virtual user logs data. No indicates the `.osd` file does not include virtual user logs data.
 - **ADDM/AWR Data** - shows if Automatic Database Diagnostic Monitor (ADDM) Report and Automatic Workload Repository (AWR) Report data is included in the session data `.osd` file being imported or not. Yes indicates the `.osd` file includes ADDM/AWR data. No indicates the `.osd` file does not include ADDM/AWR data.
 - **Counter Definitions** - shows the number of counter definitions included in the in the session data.
 - **Counter Values** - shows the number of counter values included in the session data.
5. Click **Next**. The Session Import Wizard shows the session name details.

Original Session Name - shows the name of the session that was exported from the original Oracle Load Testing instance. This session name will be the name of the session as it was exported from the original Oracle Load Testing instance regardless of the `.osd` file name.

New Session Name - provides a text box for renaming the session being imported (optional). The default is the original session name.

<Session List> - shows the sessions already in the current Oracle Load Testing instance.
6. Optionally specify a new session name and click **Import**. If you attempt to import a session with the same name as a session that already exists in the Session List, a Confirm message appears asking if you wish to overwrite the existing session. Select **Yes** to import and overwrite or **No** to cancel.
7. Click **OK** when the Info box appears indicating the import was successful. The imported session will now appear in the Session Lists of the current Oracle Load Testing instance.
8. Click the **Create Reports** tab and select the imported session name to generate reports from the session data.

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>

MSDN Knowledge Base article Q193625:

<http://support.microsoft.com/directory/article.asp?ID=KB;EN-US;Q193625>

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/Utilities/TcpView.html>

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:

- The install was bad, leaving JMSWrapper.dll unregistered.
- 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 register JMSWrapper.dll:

1. Open a command prompt.
2. Set path=*<installdir>\jdk\jre\bin;<installdir>\jdk\jre\bin\client;%path%*
3. Type the following command: `Regsvr32 jmswrap.`

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 *<installldir>\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 *<installldir>\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 -Xmx1024m* (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 *<installldir>\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 Systems from the Manage menu then selecting VU Agent Systems in the Oracle Load Testing UI. 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 Systems from the Manage menu then selecting VU Agent Systems in the Oracle Load Testing UI. 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 Autopilot 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=mymystem.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=myspass123</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=<i>EM Password</i></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=<i>Assembly Name</i></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=<i>Zone Name</i></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=<i>Deployment Plan</i></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=<i>Network Name</i></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=<i>Initial Agent Count</i></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=<i>Minimum Free Agent Count</i></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=<i>Maximum Free Agent Count</i></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=<i>Cloud Agent Increment</i></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>-groupdatapercen=% 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>-groupdatapercen=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>

Numerics

4GT feature
enabling, 3-15

A

Abort All Virtual Users toolbar button, 3-22
About menu option, 3-21
Absolute Time setting, 6-8
Accept String setting, 3-48, 4-7
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, 3-18, 3-19
Active Virtual Users, 6-5
Add Data Series button, 6-5, 6-7, 6-13
Add per step setting, 5-2, 5-5
Add to Autopilot button, 4-27
Additional Arguments setting, 3-49, 4-8, 4-15
ADDM reports, 6-20
ADDM/AWR Data, 6-22
Administrator, 3-16
adding users, 3-18
auditing usage, 3-19
deleting users, 3-19
editing users, 3-19
Help menu options, 3-17
logging in, 3-17
Logout menu option, 3-17
restoring users, 3-19
starting, 3-16
Tools menu options, 3-17
Usage Audit tab, 3-18
Users tab, 3-17
Administrator Access
enabling, 3-18
Advance to Next Record setting, 4-20
Advanced Ramp-up options, 5-4
Advanced Scenarion Options, 3-24
Advanced settings, 3-46, 4-6, 4-14
advantages, 1-2

After a delay of setting, 5-4, 5-5
After each user iterates setting, 5-5
After every setting, 5-2, 5-6
Agent error
dropping failed agents, 4-19
stopping rampup, 4-18
Agent error handling setting, 4-18
agents
configuring, 3-5
error handling, 4-18
installing, 3-4
terminating, 4-18
verifying network access, 3-5
All Logs view, 7-3
All users in scenario setting, 4-22
analysis reports
generating, 2-11
application service messages, B-4
Application stability
defining test process, 2-6
Architecture Validation, 2-2
Attach menu option, 3-20
Auto assign session name setting, 4-18
Auto Generate Timers For All Pages setting, 3-46
Auto Generate Timers For All Resources
setting, 3-46, 4-6, 4-14
Auto Generate Timers For All Step Groups
setting, 3-46, 4-5, 4-14
Auto Mode toolbar button, 7-1
Automatic Database Diagnostic Monitor
reports, 6-20
Automatic Workload Repository reports, 6-20
Automatically (when page is out of date) setting, 4-4
Autopilot
advanced ramp-up settings, 5-4
defaults, 3-53
pausing, 3-22, 5-9
setting rampup, 5-2, 5-4
setting start and stop, 5-1
starting, 5-3
starting and stopping manually, 5-3
stopping, 5-4
using, 5-1
AWR reports, 6-20

B

- basics, 3-1
- Binary Decoding Error setting, 3-49, 4-9, 4-16
- Browser Additional Arguments setting, 3-44, 4-13
- Browser Emulation setting, 3-44, 4-4
- Browser options
 - setting, 3-42
- Browser Path Override setting, 3-44, 4-12
- Browser Settings, 3-44, 4-3, 4-12
- Browser Type setting, 3-44, 4-12
- Browsers to simulate
 - defining test criteria, 2-7
- Build Scenarios tab, 3-22

C

- Cache Download Pages setting, 3-44, 4-4
- Call Function Failed setting, 3-50, 4-10, 4-16
- Cannot Open Scenario, B-6
- Categories of users
 - defining test criteria, 2-7
- Child Script Failed setting, 3-50, 4-9, 4-16
- Clear Cache After Iteration setting, 4-4
- Clear Cache Before Playback setting, 3-52, 4-17
- Clear Cache Between Iterations setting, 3-52, 4-17
- Clear Graph button, 6-5, 6-13
- Clear Persistent Cookies Before Playback setting, 3-52, 3-53, 4-17
- Clear Session Cookies Before Playback setting, 3-52, 4-17
- Clear Session Cookies Between Iterations setting, 3-52, 4-17
- Clear toolbar button, 7-1
- Client Certificate Keystore Error setting, 3-50, 4-10
- Cloud services, 3-25
 - adding configurations, 3-38
- 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, 3-35
- Community String setting, 3-34, 3-37
- Components
 - discovering, 3-35
 - starting SNMP, 3-12
- CONFIG.encap file, 3-12
- CONFIG.master file, 3-12
- Configurations menu option, 3-20
- Connect String setting, 3-33, 3-36
- Connection Idle Timeout setting, 3-46, 4-6
- Connection Speed Emulation setting, 3-44, 4-4
- Contents menu option, 3-21
- CPU vs. DB Sessions graph, 6-4

- Create Reports tab, 3-24, 6-11
- Create Variable Fail setting, 3-49, 4-9, 4-16
- Criteria for Each Business Transactions
 - defining test criteria, 2-7
- CSV data
 - exporting, 6-14
- Custom Browsers, 3-42
- Custom Graphs setting, 6-3

D

- data collectors
 - adding, 3-28
 - startup errors, B-9
- data counters
 - using, 6-10
- Data Driven Tests, 2-8
- Data Series Label setting, 6-8
- Data Series list, 6-8
- Data Series Name setting, 6-8
- Data Sources, 3-32
- Databank Configuration settings, 3-52, 4-11, 4-17
- databanks
 - advance records settings, 4-20
 - configuring, 4-19
 - next record settings, 4-20
 - out of records settings, 4-21
 - settings, 4-20
 - using the control, 4-19
- Database Name or Database SID setting, 3-33, 3-36
- databases
 - configuring existing, 3-7
 - deleting, 8-2
 - deleting session data, 6-22
 - exporting and importing, 8-3
 - exporting files, 8-5
 - importing files, 8-5
 - managing, 8-1
 - migrating data, 8-3
 - purging records, 3-17
 - removing, 8-2
 - selecting, 8-2
 - transferring schemas, 3-7
 - unlocking records, 3-17
 - updating, 8-3
- Databases menu option, 3-21
- DB Throughput graph, 6-4
- default administrator account, 3-2, 3-16
- defining systems, 3-26
- Delay Between Business Transaction Runs, 2-7, 2-9
- Deployment environment
 - defining test process, 2-6
- Detach menu option, 3-20
- Detail Level options, 6-5
- Development Engineers and Architecture Groups, 2-5
- Discover Components, 3-32
- Download Local Files setting, 3-48, 4-8
- Driver setting, 3-33, 3-36
- Driver String setting, 3-33, 3-36

Drop failed agents from session setting, 4-19
Duration details, 6-8

E

Each Iteration of Script setting, 4-20
Each Occurrence setting, 4-20
Edit Scenario Details button, 4-19
Enable Cookies setting, 3-44, 4-4
Enable Deflate setting, 3-47, 4-7
Enable GZIP setting, 3-47, 4-7
Enable Keep Alive setting, 3-48, 4-7
Enable Logging setting, 3-44, 4-4, 4-13, 5-8
encap_peer component, 3-12
Encrypted Scripts
 requirements, 3-3
 specifying passwords, 4-2
Encrypting/Decrypting Failed setting, 3-50, 4-10, 4-16
Encryption Service not Initialized setting, 3-49, 4-9, 4-16
End Time details, 6-8
Enterprise Manager server, 3-32
Enterprise Manager URL setting, 3-32
error codes, A-1
 viewing, B-1
Error Handling settings, 3-46, 4-6, 4-14
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
Errors Vs. Time graph, 6-1, 6-10
Errors Vs. Users graph, 6-1, 6-9
Event Timeout settings, 3-53
Every visit to the page setting, 4-4
Excel files
 exporting, 6-15
Execute User Defined Tests setting, 3-44, 4-4
Export File menu option, 3-21
exporting graphs, 6-14
Extensibility settings, 3-44, 4-4

F

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, 8-1
File Not Found setting, 3-49, 4-9, 4-15
filters
 using, 6-21
Firefox browser, 3-3
firewall considerations, 3-3
Forms Action Timeout setting, 3-53
Forms Component not Found setting, 3-50, 4-10
Forms Connect Error setting, 3-50, 4-10

Forms Content Match Failed setting, 3-50, 4-10
Forms I/O Communication Error setting, 3-50, 4-10
Forms LT Playback settings, 3-51, 4-11
Forms Playback Error setting, 3-50, 4-10
Forms Response Timeout setting, 3-53
Forms Startup Timeout setting, 3-53
FTP API Error codes, A-8
functional test scripts
 default settings, 3-42
 preconditions, 3-2
 scenario settings, 4-11

G

General options settings, 3-53
Global Headers setting, 3-49, 4-8
Gopher API Error codes, A-8
Graph refresh interval setting, 3-24, 6-2
Graph Title setting, 6-8
graphs
 creating, 6-13
 Errors Vs. Time graph, 6-1, 6-10
 Errors Vs. Users graph, 6-1, 6-9
 exporting data, 6-13
 exporting images, 6-13
 exporting to CSV file, 6-11, 6-12
 exporting to Excel, 6-15
 exporting to JPG, 6-15
 exporting to PNG, 6-15
 load test, 6-7
 Oracle Database Diagnostics, 6-4
 Performance Vs. Time graph, 6-1, 6-9
 Performance Vs. Users graph, 6-1, 6-9
 setting maximum data series, 6-3
 setting options, 6-8
 setting refresh interval, 6-2
 showing chart statistics, 6-7
 Statistics Vs. Time graph, 6-1, 6-10
 Users Vs. Time graph, 6-1, 6-10
 using, 6-1
 working with queries, 6-15
Graphs menu option, 3-21
Group type setting, 4-22

H

Hardware
 estimating, 4-25
Hardware Allocation
 defining test process, 2-6
Hardware Estimation menu option, 3-21
Hardware Estimation reports, 6-19
 generating, 4-26
Help menu, 3-21
Hits Per Second statistics, 6-6
Host setting, 3-33, 3-36
HTML Parsing Error setting, 3-50, 4-10
HTML Test Failed setting, 3-51, 4-16
HTTP
 defining test criteria, 2-7

HTTP API Error codes, A-9
HTTP Version setting, 3-48, 4-7

I

IBM WebSphere server, 3-33
Ignore HTTP Proxy Settings setting, 3-47, 4-6
Ignored Url setting, 3-48, 4-8
Ignored URLs that Match Regex setting, 3-49
images
 using in tests, 2-9
Import File menu option, 3-21
Information codes, A-1
installing, 3-1
 agents, 3-4
 Linux agent, 3-6
 Oracle Load Testing, 3-1
Instance setting, 3-33, 3-36
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, 3-50, 4-10
Invalid URL setting, 3-50, 4-10
IP spoofing
 using, 4-22
ipaddr field, 3-12
Iteration Delay setting, 3-43

J

Java Client Preferences settings, 3-47, 4-6, 4-14
JMX monitors
 configuring, 3-29
JPG files
 exporting, 6-15
JVM Heap settings
 changing, 3-14
 guidelines, 3-14
 limitations, 3-15
 modifying on Linux, 3-15
 modifying on Windows, 3-14

K

Keep the Same Record setting, 4-21
Key Store File Name setting, 3-34
Key Store Password setting, 3-34
Kilobytes Per Second statistics, 6-6

L

Language setting, 3-47, 4-7
Let Each User Iterate Over Records Independently
 setting, 4-21
Linear scale setting, 6-9
Linux agent
 installing, 3-6
Load Distribution Capability, 2-4

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, 6-9
Logged Messages setting, 3-45, 4-5, 4-13, 5-8
Logging Tools, 2-4
Login, 3-6
Logout, 3-21
Loop Over Range setting, 4-21

M

Main Window
 Build Scenarios, 3-22
 features, 3-20
Manage Default Reports menu option, 3-17
Manage menu, 3-21
Manual Display toolbar button, 7-2
Manual range filter, 6-8
master_peer component, 3-12
Max Content Download Size setting, 3-48, 4-8
Max Number of Keep Alive Requests setting, 3-48,
 4-8
Maximum HTTP Connections Per User setting, 3-46,
 4-6
Maximum In-Memory Cache Size setting, 4-4
Maximum JVM Heap Size (MB) setting, 3-47, 4-7,
 4-15
Maximum number of rows per page setting, 6-3
Maximum range filter, 6-8
Maximum Users Per Process setting, 3-46, 4-6
Memory
 determining requirements, 2-4
Menu options
 Abort, 5-7
 About, 3-21
 Attach, 3-20
 Autopilot Defaults, 5-1
 Configurations, 3-20
 Contents, 3-21
 Databases, 3-21
 Detach, 3-20
 Export File, 3-21
 Graphs, 3-21
 Hardware Estimation, 3-21
 Import File, 3-21
 Manage Default Reports, 3-17
 Metric Profiles, 3-20
 Metrics, 3-20
 Modify Run Attributes, 5-7

- New, 3-20
- Open, 3-20
- Options, 3-21
- overview, 3-20
- Purge Deleted Records, 3-17
- Release Sync, 5-7
- Save, 3-20
- Save As, 3-20
- Scenarios, 3-21
- ServerStats Display, 3-21
- Sessions, 3-21
- Setup E-mail Config, 3-17
- Stop, 3-20, 5-7
- Sync Point Status, 3-21
- Systems, 3-21
- Terminate Idle Agents, 3-20
- Unlock Locked Records, 3-17
- View Cloud Agents, 3-21
- VU Logs, 3-21
- Message Delivery setting, 3-45, 4-4, 4-13, 5-8
- Metric Profiles menu option, 3-20
- Metrics menu option, 3-20
- migrate data between databases, 8-3
- Minimum range filter, 6-8
- Modify Run attributes, 5-8
- monitored systems, 3-25
 - adding, 3-29, 3-31
- Monitoring and Operations Groups, 2-5

N

- Navigate toolbar buttons, 7-2
- New menu option, 3-20
- New Scenario toolbar button, 3-21
- NEW_SNMPD_PORT, 3-12
- NEW_TRAPD_PORT, 3-12
- No Delay setting, 3-43, 4-3, 4-12
- Non Proxy Hosts setting, 3-47, 4-7
- Number of Business Transactions to Simulate
 - defining test criteria, 2-7
- Number of Users, 2-9

O

- Object Download Errors Are Fatal setting, 3-46
- Object Not Found setting, 3-51, 4-16
- Object Test Failed setting, 3-50, 4-16
- Object Test setting, 3-46, 4-14
- Object Timeout setting, 3-46, 4-14
- OLT Server Warnings button, 5-7, 5-9
- On Error Stop Virtual User setting, 3-46, 4-6, 4-14
- Open gate time setting, 4-22
- Open menu option, 3-20
- Open Scenario toolbar button, 3-21
- Open VU Log toolbar button, 7-1
- OpenScript Download Manager settings, 3-51, 4-10
- OpenScript Error Recovery settings, 4-9
 - Cache and Cookies, 3-52, 4-17
 - Forms Load Test, 3-50
 - Functional test, 3-50, 4-16

- General, 3-49, 3-50, 4-9, 4-10, 4-15
- Oracle Forms Functional test, 3-51, 4-16
- Web Functional test, 3-51, 4-16
- Operating System
 - determining requirements, 2-4
- Operating System setting, 3-35
- options
 - Autopilot defaults, 3-53
 - Browser, 3-42
 - general, 3-53
 - reporting, 3-53, 6-2
 - repository, 3-42
 - scenario defaults, 3-42
 - session profiles, 3-53
 - session start and stop, 3-53
 - setting, 3-41
- Options menu option, 3-21
- Oracle Application Testing Suite Application Service
 - stopping, 3-16
- Oracle Database
 - creating tables in, 6-25
- Oracle Database setting, 3-33, 3-36
- Oracle Forms Error setting, 3-51, 4-17
- Oracle Functional Testing, 2-3
- Oracle Load Testing, 2-3
 - advantages of, 1-2
- Oracle Load Testing Agent Authentication
 - Manager, 3-16
- Oracle Load Testing Agent Service, 3-5
- Oracle Load Testing Server Settings options, 3-54
- Oracle Load Testing ServerStats, 2-4
- Oracle SNMP server, 3-11
 - configuring encapsulator, 3-12
 - configuring master agent, 3-12
 - starting SNMP components, 3-12
 - verifying start_peer script, 3-12
- Oracle Thin JDBC Driver, 3-33, 3-36
- 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, 6-6
- Pages Per Second statistics, 6-6
- Password setting, 3-33
- Pause Autopilot toolbar button, 3-22
- PEER master agent, 3-12
- Perfmon authentication, 3-34, 3-37
- Perform baseline for Oracle DB setting, 4-27
- 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 Statistics, 6-1, 6-5
- Performance Vs. Time graph, 6-1

- Performance Vs. Users graph, 6-1, 6-9
- Performance, Users vs. Time graph, 6-2
- Persist Raw Data setting, 3-47, 4-6, 4-15
- Personnel
 - defining test process, 2-6
- PGA/SGA graph, 6-4
- Playback Failed setting, 3-51, 4-16
- PNG files
 - exporting, 6-15
- Polling interval for network status check
 - setting, 3-54
- popup blockers, 3-3
- port
 - changing, 3-13
- Port setting, 3-33, 3-36, 3-37
- preferences
 - setting, 3-41
- Preserve Connections Between Iterations
 - setting, 3-48, 4-7
- Preserve Cookies Between Iterations setting, 3-48, 4-8
- Preserve Variables Between Iterations setting, 3-48, 4-8, 4-15
- Processor
 - determining requirements, 2-4
- profiles
 - selecting, 4-1
- Proxy Host setting, 3-47, 4-7
- Proxy Port setting, 3-47, 4-7
- Purge Deleted Records menu option, 3-17

Q

- Quality Assurance Organizations, 2-5
- queries, 6-15
 - deleting, 6-17
 - editing, 6-16
 - running saved queries, 6-16
 - saving, 6-16

R

- Rampup Specification
 - advanced, 5-4
- Random setting, 3-43, 4-3, 4-12
- Randomly setting, 4-20
- Range Filter settings, 6-8
- Range Format, 6-8
- Raw Data
 - example counter definition file, 6-26
- Raw data
 - creating database tables, 6-25
 - enabling collection, 6-23
 - example command file, 6-25
 - example counter data control file, 6-26
 - importing CSV files, 6-25
 - using, 6-23
 - working with data files, 6-23
- Recorded setting, 3-43, 4-3, 4-12
- Recorded/Random setting, 3-43
- Recorded/Rnd setting, 4-3, 4-12
- Redirection codes, A-2
- Refresh Off toolbar button, 7-2
- Refresh On toolbar button, 7-2
- Relative Time setting, 6-8
- Release trigger setting, 4-22
- releasing
 - synchronization points, 5-8
 - Virtual Users, 5-8
- Remote Agent setup, 3-4
- Remote Password setting, 3-35
- Remote Port setting, 3-35
- Remote Protocol setting, 3-35
- Remote Username setting, 3-35
- Render HTML toolbar button, 7-1
- Replace URLs setting, 3-49, 4-8, 4-15
- Report Counters setting, 3-47, 4-6, 4-15
- Report Sender Interval setting, 3-47, 4-7, 4-15
- report template
 - deleting temporary images/data, 6-3
- Report Template setting, 6-3
- Reporting data collection interval setting, 6-2
- Reporting options, 3-53
- Reporting settings, 3-46
- reports
 - Create Reports tab, 6-11
 - creating custom graph and session list, 6-18
 - creating templates, 6-17
 - exporting to CSV data, 6-14
 - exporting to Excel files, 6-15
 - exporting to JPG files, 6-15
 - exporting to PNG files, 6-15
 - generating, 2-11
 - generating summary reports, 6-19
 - including think time, 6-3
 - managing default, 3-17
 - performance statistics, 6-5
 - scenario summary, 6-19
 - session summary, 6-19
 - setting options, 4-5, 4-14, 6-2
 - using, 6-1
 - viewing, 6-13
- repositories
 - adding, 3-9
- repository options, 3-42
- Request Timeout setting, 3-46, 4-6
- Required Applications
 - defining test process, 2-6
- Resolution Size setting, 3-44, 4-13
- Response Time Error setting, 3-50, 3-51, 4-10, 4-16
- Restart Application Service setting, 3-54
- RSWISERV errors, B-3
- Rules of Thumb, 2-4
- Run Test button, 4-27, 5-3
- Running users setting, 4-22
- Run-Time Data, 2-8
- runtime reports
 - viewing, 6-7

S

- Save as CSV toolbar button, 7-1
- Save As menu option, 3-20
- Save menu option, 3-20
- Save Scenario toolbar button, 3-21
- scalability tests
 - planning, 2-8
- Scenario Default options, 3-42
- Scenario menu, 3-20
- Scenario reports, 6-19
- scenarios
 - advanced ramp-up, 5-4
 - defining, 4-1
 - deleting, 4-25
 - estimating hardware, 4-25
 - opening existing, 4-24
 - removing profiles, 4-25
 - renaming, 4-24
 - running from command line, 4-25
 - saving, 4-23
 - specifying profile attributes, 4-2
 - submitting to Autopilot, 4-26
 - using synchronization points, 4-21
 - working with files, 4-23
- Scenarios menu option, 3-21
- Scheduling
 - defining test process, 2-6
- scripts
 - advanced ramp-up, 5-4
 - creating, 2-9
- Segment Parser Error setting, 3-49, 4-9, 4-16
- Select Next Record setting, 4-20
- Sequentially setting, 4-20
- Server Error codes, A-4
- Server not initialized, B-4, B-5
- Server Status codes, A-1
- Server Type setting, 3-34
- Server Warnings
 - viewing, 5-9
- servers
 - Oracle SNMP server, 3-11
 - setting up, 3-10
 - Solaris SNMP server, 3-10
- ServerStats
 - data collectors, 3-25
 - data counters, 6-10
 - Oracle SNMP server, 3-11
 - page timers, 6-10
 - setting up servers, 3-10
 - Solaris SNMP, 3-11
- ServerStats Configuration, 3-23
- ServerStats Display menu option, 3-21
- ServerStats menu, 3-20
- ServerStats Monitors options, 3-53
- Session Details, 6-8
- Session menu, 3-20
- Session Name Prefix setting, 4-18
- Session Profile options, 3-53
- Session reports, 6-19
 - using filters, 6-21
- Session Start and Stop options, 3-53, 4-18
- sessions
 - attaching to, 9-1
 - defining how to end, 4-18
 - defining how to start, 4-18
 - deleting, 6-22
 - detaching from, 9-1
 - exporting data, 9-2
 - importing data, 9-2
 - managing, 4-18
 - viewing performance statistics, 6-5
 - working with, 9-1
- Sessions menu option, 3-21
- Set Up Autopilot tab, 3-23
- Setup E-mail Config menu option, 3-17
- Show Chart Statistics button, 6-5, 6-7, 6-13
- Show End-to-End Times option, 6-5
- Show Graph Options button, 6-5, 6-8, 6-13
- Show Legend button, 6-5, 6-7, 6-13
- Show Server Times Only option, 6-5
- Shuffle setting, 4-20
- SID setting, 8-2
- SNMP Agent
 - enabling on startup, 3-11
 - starting, 3-11
 - stopping, 3-11
- SNMP Community String setting, 3-34
- SNMP server, 3-10
- SNMP Version setting, 3-37
- snmpd component, 3-12
- Socket Timeout setting, 3-46, 4-6
- Solaris SNMP Server, 3-10
- Solve Variable Fail setting, 3-50, 4-10
- Solve Variable Failed setting, 3-51, 4-16
- SQL Database setting, 3-33, 3-36
- SSL
 - defining test criteria, 2-7
 - using, 3-13
- SSL Version setting, 3-48, 4-8
- Start Load Test toolbar button, 3-22
- Start Time details, 6-8
- start_peer script, 3-12
- Statistics Vs. Time graph, 6-1, 6-10
- Status Bar Test Error setting, 3-51, 4-17
- Stop All Virtual Users toolbar button, 3-22
- Stop Application Service setting, 3-54
- Stop attached session after browser closes
 - setting, 4-18
- Stop menu option, 3-20
- Stop ramp-up on agent error setting, 4-18
- Stop Remaining Iterations on Failure setting, 3-46, 4-6, 4-14
- Stop session on last VU completion setting, 4-18
- Stop the User setting, 4-21
- Submitted Scenario Profiles, 3-24
- Success codes, A-1
- Summary by VU view, 7-3
- summary reports, 6-19
- Summary view, 7-2
- Sun JDBC

- ODBC Bridge Driver, 3-33, 3-36
- support files, B-4
 - creating, 3-16
- Synchronization points
 - naming, 4-22
- synchronization points
 - adding to profiles, 4-21
 - managing, 5-7
 - releasing, 5-8
 - using, 4-21
- System Identifier setting, 8-2
- System Monitoring Tools, 2-4
- system requirements
 - functional test scripts, 3-2
- systems
 - adding new groups, 3-27
 - adding to groups, 3-27
 - defining, 3-26
 - deleting, 3-41
 - editing, 3-41
 - renaming, 3-41
- Systems Manager, 3-24
- Systems Manager., 3-6
- Systems Manager options, 3-53
- Systems menu option, 3-21

T

- Table Test Failed setting, 3-50, 4-16
- Table Test setting, 3-45, 4-14
- TCP/IP Properties, 4-22
- templates
 - creating, 6-17
- Terminate all agents at end of session setting, 4-18
- Terminate Idle Agents menu option, 3-20
- test execution
 - verifying on multiple machines, 2-10
- test results
 - evaluating, 2-11
- Testing Environment
 - defining test process, 2-6
- tests
 - executing, 2-10
- Text Match Fail setting, 3-50, 4-10
- Text Matching Failed setting, 3-50, 4-16
- Threshold Line settings, 6-9
- Throughput Required, 2-8
- Timing and event controls, 3-23
- Title Test Failed setting, 3-51, 4-16
- TLSv1.2, 3-48, 4-8
- Toolbar, 3-21, 7-1
- Tools menu, 3-16, 3-21
- Transaction Overview graph, 6-4
- Transactions Per Second statistics, 6-6
- Transactions statistics, 6-6
- Transactions to Perform, 2-8
- Transactions with Errors statistics, 6-6
- Trans/sec, Users vs. Time graph, 6-2
- Trust Store File Name setting, 3-34
- Trust Store Password setting, 3-34

- 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, 3-50, 4-9, 4-16
- Unlock Database Records menu option, 3-17
- Use Data Bank setting, 3-44, 4-3, 4-12, 4-19
- Use IP Spoofing setting, 3-44, 4-4, 4-23
- User Agent String, 3-42
- User Interface refresh interval setting, 6-2
- Username setting, 3-33
- users
 - adding, 3-18
 - deleting, 3-19
 - editing, 3-19
 - number of, 2-9
 - restoring, 3-19
 - types of, 2-8
- Users setting, 6-8
- Users Vs. Time graph, 6-1, 6-10

V

- Validate hostname/ip when user adds a system setting, 3-53
- Validate monitors when user adds or modifies a monitor setting, 3-53
- Variable Not Found setting, 3-49, 4-9, 4-16
- View Run Graphs tab, 3-24, 6-3
- View Text toolbar button, 7-1
- Virtual Agent authentication, 3-35
- Virtual user (VU) ramp-up, 5-2
- Virtual User profiles
 - selecting, 4-1
- Virtual User scenarios
 - defining, 4-1
 - specifying profile attributes, 4-2
- Virtual Users
 - aborting all, 5-9
 - controlling, 5-7
 - determining how many, 4-17
 - modifying run attributes, 5-8
 - pacing, 3-23
 - pausing, 5-9
 - releasing, 5-8
 - selecting scenarios, 4-1
 - specifying how many, 5-2
 - specifying rampup, 5-2
 - specifying start time, 5-1
 - specifying stop time, 5-2
 - stopping all, 5-9
 - synchronizing start up, 5-1
 - using the display, 7-1
 - using the grid, 5-6

- verifying multiple, 2-9
- with Errors statistics, 6-6
- working with, 5-6
- VU Agent System Groups, 3-25
- VU Agent Systems, 3-25
 - adding, 3-26
 - changing port settings, 3-4
- VU Logs
 - auto display interval setting, 6-2
 - exit codes, B-8
 - main window, 7-1
 - menu option, 3-21
 - opening log files, 7-3
 - saving log files, 7-4
 - setting options, 3-44, 4-4, 4-13
 - starting, 7-1
 - toolbar, 7-1
 - using, 7-1
- VU Pacing (Think Time) setting, 3-43, 4-3, 4-12

W

- Wait for Page Timeout setting, 3-51, 4-16
- Watch VU Grid tab, 3-24
- web server port, 3-13
- Web-Based Interface, 1-2
- WebLogic, 3-29
- Weblogic patch, 3-15
- WebSphere, 3-29
- When counter setting, 5-4, 5-5
- When Oracle Load Testing Server IP address changes setting, 3-54
- When Out of Records setting, 4-21
- When scenario starts setting, 5-4
- When scenario stop setting, 5-5
- When Script Requests a Record setting, 4-20
- When script runs setting, 5-4, 5-5
- When script stops setting, 5-4, 5-5

X

- X-Axis Scale settings, 6-8
- XML Test Failed setting, 3-51, 4-16
- XML Test setting, 3-46, 4-14

Y

- Y-Axis Scale settings, 6-9

Z

- Zero Downloads Fatal setting, 3-50, 4-10
- Zoom tool, 6-5, 6-13

