Oracle® Load Testing for Web Applications

Load Testing User's Guide

Version 9.00 for Microsoft Windows (32-Bit)

E15484-01

September 2009

ORACLE®

Oracle Load Testing for Web Applications Load Testing User's Guide, Version 9.00 for Microsoft Windows (32-Bit)

E15484-01

Contents

3 Basics

4 Defining Virtual User Profiles

5 Defining Virtual User Scenarios

6   Using the Autopilot

7   Using Graphs and Reports

A Internet Status Codes

B Error Messages

Preface

Welcome to the Oracle Load Testing for Web Applications User's Guide. This guide explains how to get started using the features and options of Oracle Load Testing for Web Applications 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 for Web Applications should be familiar with Oracle Functional Testing for Web Applications Visual Scripts and the concepts of Web application testing, load testing, scalability testing, and reporting.

Prerequisites

Oracle Load Testing for Web Applications 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 provides an overview of the major features of Oracle Load Testing for Web Applications.

Chapter 2 provides a basic methodology for testing scalability and performance of Web applications throughout the life cycle.

Chapter 3 provides descriptions of the Oracle Load Testing for Web Applications main window features and menu options.

Chapter 4 explains how to define profiles that specify the actions of virtual users for load/performance testing.

Chapter 5 Defining Virtual User Scenarios: explains how to define virtual user scenarios based upon the virtual user profiles.

Chapter 6 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 7 explains how to view real-time performance, statistics, and error information in graphs and reports.

Chapter 8 explains how to use the virtual user display to view virtual user actions during an Autopilot session.

Chapter 9 explains how to manage databases for use in Oracle Load Testing for Web Applications and how to upload and download files to and from the Oracle Load Testing for Web Applications Server.

Chapter 10 explains how to attach and detach from running sessions.

Appendix A lists the status codes and Internet error codes that can be returned from a server.

Appendix B provides a reference error messages and codes generated by Oracle Load Testing for Web Applications.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at http://www.oracle.com/accessibility/.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, 7 days a week. For TTY support, call 800.446.2398. Outside the United States, call +1.407.458.2479.

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 Getting Started Guide*

- *Oracle Functional Testing for Web Applications Functional Testing User's Guide*

- *Oracle Functional Testing for Web Applications Job Scheduler User's Guide*

- *Oracle Functional Testing for Web Applications Navigation Editor User's Guide*

- *Oracle Functional Testing for Web Applications Application Programming Interface Reference*
- *Oracle Functional Testing for Web Applications Result Objects Reference*
- *Oracle Load Testing for Web Applications Load Testing User's Guide*
- *Oracle Test Manager for Web Applications 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.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Introduction

The power of the Internet is that you can make your new Web application available to thousands of users instantaneously. That's also the scary part. Fortune 1000 companies are rapidly exploiting Web technology for business-critical commerce and enterprise applications. Scalability is one of the top concerns most often cited by IT managers as they prepare to deploy new Internet and Intranet applications for their companies. With the myriad of technologies and platforms to choose from, it's no wonder that the biggest concern many developers have is "will the application scale?"

Since much of the technology is new, there is significant risk with any Web project that the application will not perform adequately under real-life conditions. The only way to mitigate this risk is by emulating the load of hundreds or thousands of simultaneous users prior to the application going live.

1.1 About Oracle Load Testing for Web Applications

Oracle Load Testing for Web Applications provides the easiest and most accurate way to test the scalability of your e-commerce and e-business applications. Oracle Load Testing for Web Applications emulates thousands of "virtual users" accessing your site simultaneously, and measures the effect of the load on application performance.

TrueLoad technology ensures that your tests closely correlate with real user-load so you can confidently use Oracle Load Testing for Web Applications results to help make key decisions about your system's architecture, tuning, and hosting alternatives.

Oracle Load Testing for Web Applications is the load and scalability testing component of the Oracle Application Testing Suite. Oracle Load Testing for Web Applications requires no programming and is operated using an intuitive graphical user interface and Visual Scripts. Oracle Load Testing for Web Applications uses the Visual Scripts you develop using Oracle Functional Testing for Web Applications, which have verification capabilities built in, to emulate the Virtual Users for load testing.

1.2 Feature Highlights

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

- Accurately emulates real user interactions and tests for correct responses.

- Requires no programming and re-uses your functional and regression tests scripts with no modifications.

- Enables you to vary load conditions on-the-fly to try "what-if" scenarios, and to create comprehensive reports and graphs with the touch of a button.

- Allows you to access Oracle Load Testing for Web Applications from any networked machine using a web browser.

- Allows multiple users to collaborate during testing by connecting to the same in progress test session to view and analyze results.

1.2.1 The Problem with Traditional Load Testing Tools

Most Web load testing tools today are simply HTTP recorders that record the low-level traffic between client browser and web server and then play back many copies of the recording simultaneously. The problem is that most web applications are dynamic, that is, the Web pages change slightly with every request to reflect such things as session IDs and dynamic content. This means that the recording you make now will be useless in five minutes.

In addition, it's very difficult to know if the server is responding correctly. For example, your server could be sending the message "Server Busy" to many users and most load testing tools will never notice the difference.

1.3 Advantages of Oracle Load Testing for Web Applications

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

- TrueLoad Technology

- Reuse of existing Visual Scripts

- User-defined Virtual User Profiles

- Single Point of Control with Distributed Agents

- Data-Driven Load Testing

- Real Time Reports and Graphs

- Real Time Server-Side Statistics Monitoring

- Optimized for Popular IDEs

- Web-based Access

- Collaborative Testing

1.3.1 TrueLoad Technology

Oracle Load Testing for Web Applications uses innovative TrueLoad technology that closely emulates the traffic from real Web users using real Web browsers. TrueLoad includes the following unique technologies:

Object-Oriented Recording - Unlike HTTP-based tools, Oracle Load Testing for Web Applications records the actual objects used by the Web browser to render a page. These objects are represented graphically in Oracle Load Testing for Web Applications Visual Scripts. The Visual Scripts that are used for load testing are richer in detail than what is captured with HTTP recorders, and are much easier to understand, maintain, and debug. As the application-under-test evolves during the development cycle, it is very easy to make incremental updates to the Visual Script.

Automatic Response Verification - Oracle Load Testing for Web Applications automatically checks for the accuracy of each of the pages returned by the server. If the pages match what is defined in the Visual Script, the test will proceed without errors.

If pages are returned that do not match the recorded baseline script, then an error will be flagged.

Concurrent Object Requests - Like IE and Netscape, each Oracle Load Testing for Web Applications Virtual User makes concurrent requests to the server using the multi-threading capabilities of the operating system; i.e. one thread gets the page, another gets images, etc. The result is that the intensity of hits generated on the back-end servers is identical to that of a real browser.

Automatic Session Management - Oracle Load Testing for Web Applications automatically recognizes and handles the session variables for popular development environments such as Microsoft ASP/ASP.NET, Weblogic, WebSphere, NetDynamics, ColdFusion, and WebObjects.

TrueLoad technology used in Oracle Load Testing for Web Applications offers the most accurate load simulation available, and produces results that closely correlate with real-life usage.

1.3.2 Reuse of Existing Visual Scripts

Oracle Load Testing for Web Applications uses the same Visual Scripts created for functional testing with Oracle Functional Testing for Web Applications to emulate hundreds or thousands of "virtual users." The Oracle Load Testing for Web Applications Build Scenarios tab lets you define any number of custom load scenarios by simply pointing and clicking on the names of the pre recorded Visual Scripts and then specifying how many virtual users of each type you wish to run.

You can change the number and type of users on-the-fly to try "what-if" scenarios as you vary the loading conditions or application settings. While running, any virtual user can be "expanded" to be able to view the actual pages seen by virtual users to aid in debugging.

1.3.3 User-Defined Virtual User Profiles

Oracle Load Testing for Web Applications also lets you combine multiple Visual Scripts as part of a single virtual user "profile." Within these profiles, you can specify Visual Scripts to use as prolog, run, epilog, and error scripts.

User-defined profiles can also include synchronization points to create realistic multi-user situations.

1.3.4 Single Point of Control with Distributed Agents

Virtual users can be simulated by a single server or distributed among multiple servers located anywhere on a LAN or WAN. The Oracle Load Testing for Web Applications controller can run virtual user "agents" on distributed client stations from a single location.

Using the Oracle Load Testing for Web Applications Build Scenarios and Set Up Autopilot tabs, you can define any number of custom load scenarios by simply pointing and clicking on the names of the pre-recorded Visual Scripts. Oracle Load Testing for Web Applications scenarios define which virtual user profiles each agent will run on which workstation, specifies how many virtual users of each type to run, and how virtual users should ramp up.

1.3.5 Data-Driven Load Testing

Oracle Load Testing for Web Applications easily supports data-driven testing to emulate realistic scenarios in which many different users are accessing different database records simultaneously. The Oracle Functional Testing for Web Applications Data Bank Wizard lets you re-use a single recorded script many times with a different data set for each virtual user. For example, you can use different usernames, passwords, account balances, etc. for different virtual users. Data is automatically read in from an external Data Bank and overlaid onto each virtual user.

1.3.6 Comprehensive Reports and Graphs

The graphing capability in Oracle Load Testing for Web Applications lets you see the results of your Oracle Load Testing for Web Applications tests in real time. Performance data can be accumulated at varying levels of granularity including profiles, scripts, groups of pages, individual pages, and objects on pages. Graphs and reports include response time, error rates, number of users, and statistics such as hits-per-second, and pages-per-second, etc.

In addition, Oracle Load Testing for Web Applications provides for post-run analysis using a comprehensive set of graphs and reports. You can also export data for use in external programs such as Microsoft Excel for further analysis.

1.3.7 Real Time Server-Side Statistics Monitoring

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

You can monitor specific counters in real time while running a load test. Server statistics can also be saved to the Oracle Load Testing for Web Applications database for later analysis.

1.3.8 Optimized for Popular IDEs

Oracle Load Testing for Web Applications is optimized for a number of popular Web development environments including NetDynamics, WebObjects, Cold Fusion and Microsoft ASP/ASP.NET, Weblogic, and WebSphere. These tools use variable session ID's and SSL, which pose problems for other load tools. Oracle Load Testing for Web Applications handles these environment features automatically.

1.3.9 Development-to-Deployment Testing

Load and performance testing should not just be a one-time event prior to deployment. Instead it should become an integral part of the process of designing, building, and maintaining Web applications. Load testing can and should be used to validate the system architecture in the early stages of development, to identify bottlenecks and to benchmark the application during its development, and to fine-tune the application for optimal performance prior to deployment. As the application and system infrastructure are modified over time, the standard set of performance benchmark tests developed should be re-run to ensure that performance has not been adversely affected.

Even after the application is deployed, application performance should be continuously monitored to identify slowdowns and performance degradation before users experience problems.

Oracle Load Testing for Web Applications is one component of the Oracle Application Testing Suite, an integrated solution for testing e-Business applications. Designed to deliver an immediate improvement in the quality of your software by offering repeatable, measurable results, the Oracle Application Testing Suite helps you keep pace with the accelerated Web application development cycle.

Oracle Application Testing Suite is fully integrated and is the first Web testing solution to leverage a single set of test assets from development through deployment, ensuring consistency in testing throughout the application life cycle and maximizing the return on investment for automated testing of your Web applications.

2

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

Oracle Load Testing for Web Applications - used to define virtual user profiles and scenarios and to perform the load test. Oracle Load Testing for Web Applications runs the Visual Scripts as multiple virtual users that test the scalability of the application. Oracle Load Testing for Web Applications defines the number and types of virtual users and which Visual 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 for Web Applications 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 for Web applications.

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 Visual 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 Visual Script should have a detailed sequence of steps that define the exact actions a user would perform. Multiple Visual Scripts can be used. For example, you can define a specific Visual Script that performs user login, several Visual Scripts that perform specific business transactions, and another Visual Script that logs users off. For each Visual Script, you should define the expected results. Oracle Functional Testing for Web Applications lets you quickly and easily record Visual 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 Functional Testing for Web Applications Data Bank Wizard lets you specify and connect external data sources to the Visual Scripts.

2.8.4 Planning the Load Test Scenarios

In addition to the business transaction details for each Visual 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 for Web Applications 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 Visual Script.

Number of Users - How many virtual users with this user profile will run over the same time interval? Oracle Load Testing for Web Applications 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 for Web Applications can run virtual users on a single system or on multiple, distributed systems running Oracle Load Testing for Web Applications agents. Oracle Load Testing for Web Applications can specify which virtual user scenarios run on which workstations.

Which Browser - Which browser will this user group emulate? Oracle Load Testing for Web Applications 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 for Web Applications 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 for Web Applications 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 for Web Applications provides this capability.

2.8.5 Create and Verify the Test Scripts

After planning the Visual Scripts, you will use Oracle Functional Testing for Web Applications to create and verify each Visual Script.

Create the Scripts - This process is defined by Oracle Functional Testing for Web Applications (recording user actions) and the individual test plans for each script. When creating the Visual 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 Visual Script is created, you should verify that the script performs as expected and produces the desired result. Each Visual Script should be verified independent of any other scripts and in a controlled manner to simplify Visual Script debugging.

2.8.6 Create and Verify the Load Test Scenarios

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

Oracle Load Testing for Web Applications 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 Visual 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 for Web Applications 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 for Web Applications, 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.

3

Basics

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

3.1 Installing Oracle Load Testing for Web Applications

The installation for Oracle Load Testing for Web Applications depends upon how you plan the virtual user test configuration. At a minimum, you need to install Oracle Load Testing for Web Applications 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 for Web Applications to use as a "Master" station and install the Oracle Load Testing for Web Applications Agent (or another seat of Oracle Load Testing for Web Applications) on the client machines to use as virtual user Agent stations.

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

3.1.1 Installing Oracle Load Testing for Web Applications

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

1. Go to:
    http://www.oracle.com/technology/software/products/app-testing/index.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.

3. Unzip the the download file and then run oats###.exe.

4. Follow the setup instructions to install the Oracle Application Testing Suite. The install program will install the WebLogic web server and the Oracle 10g Express Edition Database by default. If you chose to have the install program use an existing Oracle database, follow the instructions for using an existing database below.

5. Select **Oracle Load Testing for Web Applications** from the **Oracle Application Testing Suite** start menu to start the Oracle Load Testing for Web Applications user interface.

Oracle Load Testing for Web Applications starts in an existing browser if one is available. To set Oracle Load Testing for Web Applications 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.

3.1.1.1 Using an Existing Database

If you choose to use an Oracle Database 11*g* Enterprise Edition existing database during the installation procedure, use the following instructions to setup the database for use with the Oracle Application Testing Suite.

1. Install the Oracle Database 11*g* Enterprise Edition and create a user with a minimum of the following privileges:

   - Create/Alter/Drop Tables

   - Create/Drop Views

   - Create/Alter/Drop Sequences

   - Create/Alter/Drop Triggers

   - Create/Alter/Drop Indexes

If you are performing the database administration yourself, use the following steps:

Log Into Database Control

1. Install the Oracle Database 11g Enterprise Edition and launch Database Control.

2. Login to Database Control as a user with "system" privileges.

Create a Tablespace

1. Click the **Server** tab.

2. Click **Tablespaces**.

3. Click **Create**.

4. On the **General** tab, give the tablespace a name.

5. Under Datafiles, click **Add**. Specify a file for the new tablespace. Set the file size to 1000 MB.

6. Click **Continue**.

7. Click **OK** to create the new tablespace

Create a User

1. Return to the database instance, Server tab and click **Users**

2. Create a new user ("ATSuser"). During user creation, designate the user's default tablespace to the one created above. On the Roles tab, assign the user the RESOURCE role. On the System Privileges tab, add the required system privileges defined above to that user.

3. Click **OK** to create the new user.

3.1.1.2 Oracle Application Testing Suite System Setup

The following instructions explain how to setup an exsiting Oracle database for use with the Oracle Application Testing Suite.

Oracle Test Manager for Web Applications

1. Install the Oracle Database 11g Enterprise Edition Client OR

2. The Oracle Database Instant Client:

   a. Basic Package, AND

   b. ODBC Supplement

   > **Note:** Read the ODBC Release Notes for specific configuration instructions, including the need to add the Instant Client directory to the PATH environment variable and the creation of the TNS_ADMIN environment variable.

3. Create a text file called `tnsnames.ora` inside the folder defined by the TNS_ADMIN environment variable. The text file should contain the following text (to connect to an Oracle DB on a system with the name `oracledbsys`):

```
oracledbsys =

  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = oracledbsys.example.com)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SID = ORCL)
```

4. Create a DSN pointing to the Oracle DB Server, as follows:

   a. Open the Control Panel from the **Start** menu.

   b. Open the Administrative Tools and select **Data Sources (ODBC)**

   c. Double-click **Data Sources (ODBC)** and select **System DSN**

   d. Click **Add** and select the Oracle DB driver

   e. Click **Finish** to select this DB driver

   f. In the "Oracle ODBC Driver Configuration" dialog box enter the following information:

      – Data Source Name: <ANY name of your choice>

      – Description: optional

      – TNS Server Name: oracledbsys (correlating to the contents of `tnsnames.ora`. Use the appropriate TNS name you have given.)

      – User Id: the username assigned to you on the DB server

      – Click **Test**, enter the password for your username and click **OK**

      – If the test is successful, you can proceed.

Oracle Load Testing for Web Applications

1. Log into OLT `http://`*systemname*`:8088/olt`

2. Select **Databases** from the **Manage** menu then select **New**

3. Select Oracle from the drop-down list.

4. If you wish to create new tables in a schema, select the radio button for **Create tables in the Database**. If you wish to connect to an existing schema that has OLT tables in it, select the **User existing database** (default).

5. Enter the relevant database connection information in the fields.

6. Click **OK**

> **Note:** An SQL file is available on the Oracle Application Testing Suite system that can be used, if necessary, to delete the database components created by the product. **(This will result in loss of data!)** The file is delete tables.sql and is located in the <installdir>\OTM\Files\Oracle directory. This SQL can be executed in SQL Developer or similar utility for Oracle DB management.

3.1.2 Popup Blockers

Popup blockers must be turned off for Oracle Load Testing for Web Applications 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**.

3.1.3 Port Configuration

If you are using a firewall between the Oracle Load Testing for Web Applications 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 the agent machine, change the `wrapper.app.parameter.2` setting in the <installdir>\agentmanager\bin\AgentManagerService.conf file. In addition, change the port in the system configuration in Oracle Load Testing for Web Applications 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 for Web Applications, it is recommended that you have the following general communication ports open before starting the Oracle Load Testing for Web Applications Application Service.

```
http=8088
agent=9001
dataCollector=7073
webservice=8083
https=8089
```

To change the ports:

1. Go to `http://localhost:8088/console` to start the WebLogic Console.

2. Log in as administrator 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 config.

For more information about these settings, refer to Appendix C. See the Oracle WebLogic Server documentation for additional information about using the Console application.

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

- 9001

- 7073 - Data Collector

- 8088 for running OpenScript scripts only

3.1.4 Installing Oracle Load Testing for Web Applications Agent

The Oracle Load Testing for Web Applications Agent is a subset of the full Oracle Load Testing for Web Applications installation. You should not install the Oracle Load Testing for Web Applications Agent if you have Oracle Load Testing for Web Applications installed on the system. You install Oracle Load Testing for Web Applications Agent on the agent systems (which should be Windows 2000/2003 machines) 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 for Web Applications Agent:

1. Run the downloaded oats###.exe installaation program from the download zip file..

2. Follow the setup procedure to the Select Components and Installation Directory screen.

3. Clear the **Java Support for Oracle Application Testing Suite** and the **Oracle Application Testing Suite** check boxes.

4. Select the **Remote Agent** check box.

5. Click **Next** as necessary to complete the installation.

6. Verify network access from Oracle Load Testing for Web Applications system to the Agent systems and configure the Agent Systems as explained in the next sections.

7. Configure the agent system.

8. Define the system in Oracle Load Testing for Web Applications by selecting **Systems** from the **Manage** menu.

Once the Oracle Load Testing for Web Applications 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 for Web Applications 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 for Web Applications automatically accesses and starts the Agent when you start the Autopilot.

3.1.4.1 Verify Network Access to Agent Systems

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

1. Make sure that you have the Oracle Load Testing for Web Applications 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 for Web Applications system. The systems you plan to use as agents must have either the Oracle Load Testing for Web Applications Agent software or the

full Oracle Application Testing Suite installed to work as agents. Do not install both the Oracle Load Testing for Web Applications 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 for Web Applications 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 for Web Applications 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 for Web Applications system, you will not be able to run the agents from the Oracle Load Testing for Web Applications system.

3. In the Oracle Load Testing for Web Applications system, add a Visual 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 Visual Script into the System field on the Build Scenarios tab of Oracle Load Testing for Web Applications.

3.1.4.2 Configuring Oracle Load Testing for Web Applications 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 for Web Applications agents:

1. Open **Administrative Tools** in the **Control Panel** then open **Services**. The Services dialog box displays.

2. Select **Oracle Load Testing for Web Applications 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

The Linux agent is not part of the standard Oracle Application Testing Suite product download Zip file. To install the Linux agent:.

1. Go to:
   http://www.oracle.com/technology/software/products/app-testing/index.html

2. Download the oats-linux-agent-x.xx.xxxx.zip from the Web site.

3. Copy oats-linux-agent-x.xx.xxxx.zip to your Linux machine.

4. Enter the following commands. *<installdir>* should be the location where you want to install the agent.

```
unzip oats-linux-agent-x.xx.xxxx.zip -d <installdir>
cd <installdir>/agentmanager/bin
chmod +x startAgentManager.sh
```

5. Start the AgentManager with the following command:

```
./startAgentManager.sh
```

Once the Linux agent is installed, define the remote agent system in the Oracle Load Testing for Web Applications **Systems Manager**.

When you define your Oracle Load Testing for Web Applications testing scenarios, you specify which Agent machine to use to run the virtual users in the System field of the Build Scenarios tab and you specify Java Client as the User Mode on the same tab. Oracle Load Testing for Web Applications automatically accesses and starts the Agent when you start the Autopilot.

3.3 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.

3.3.1 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 Agent*s suite:

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

3.3.1.1 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.

3.3.1.2 Stopping the SNMP Agent

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

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

3.3.1.3 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

3.3.2 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 Oracle8 Installation Guide). The installation documentation for the specific platforms describe how to start the agent on Windows and Linux.

3.3.2.1 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.

3.3.2.2 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.

3.3.2.3 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.

3.3.2.4 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`.

3.4 Installing Remote Data Collectors

You can use remote data collectors for collecting ServerStats data. To install a remote data collector:

1. Run oats###.exe installaation program from the download zip file.

2. Follow the setup procedure to the Select Components and Installation Directory screen.

3. On the Select Components and Installation Directory dialog box, deselect all options except for **Oracle Load Testing for Web Applications DataCollector**.

4. Follow the setup instructions to install the remote data collector.

5. In Oracle Load Testing for Web Applications, select **Systems** from the **Manage** menu to display the System Manager.

6. Select **ServerStats Data Collectors** and click **New**. The Add ServerStats Data Collector Dialog Box is displayed.

7. Enter the name, host name or IP address, and port number to use for the remote data collector.

8. Click **OK**.

9. Click **Close**.

10. The remote data collector will be available to use when adding or editing monitors in a ServerStats configuration. Select the remote data collector in the **Data Collector** section of the Edit Monitor and Add Monitor: Step 2 of 3 dialog boxes.

3.5 Changing the Web Server Port

The default Oracle Load Testing for Web Applications web server port is 8088. You can change this to another port.

To change the port:

1. Go to `http://localhost:8088/console` to start the WebLogic Console.

2. Log in as administrator 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.

3.6 Using SSL

You can set up Oracle Load Testing for Web Applications 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 administrator using the password you defined during the Oracle Application Testing suite installation procedure.

3. In the Domain Structure, select Environment under "oats" and then select Servers.

4. Select AdminServer(Admin).

5. Select the SSL tab.

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

3.7 Oracle Application Testing Suite Tools Menu

The Oracle Application Testing Suite tools menu has options for viewing version and license 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, version, and license information. It also has links to information about obtaining license keys and information about your system.

Create Support File - 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.

Oracle Load Testing Agent Authentication Manager - opens the Agent Authentication Manager for defining authentication profiles for multiple load testing agent machines.

Restart Oracle Application Testing Suite Application Service - stops and restarts the Oracle Application Testing Suite Application service.

Stop Oracle Application Testing Suite Application Service - stops the Oracle Application Testing Suite Application Service.

Data Migration Utility - opens the Data Migration Utility for migrating data from Oracle Test Manager for Web Applications or Oracle Load Testing for Web Applications in MS Access or SQL Server databases to Oracle Database 11*g*.

3.8 Administrator

Using the Administrator you create user accounts, assign them user names and passwords, and assign the type of access that they have in Oracle Load Testing for Web Applications, none, full control, or view only. When Oracle Load Testing for Web Applications login is enabled, users must login to access Oracle Load Testing for Web Applications.

A default administrator account is created at installation. The username and password are specified during the installation process. You can change the password after logging in to the Administrator.

To start the Administrator:

1. Enter `http://<machine>:8088/admin` in your browser where machine is where the Oracle Application Testing Suite is installed or select the Administrator from the Oracle Application Testing Suite start menu.

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**.

3.8.1 Menu Options

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

3.8.1.1 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 - pdisplays 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.

3.8.1.2 Help Menu

Contents - displays the online help table of contents.

About Admin - displays version and copyright information.

3.8.1.3 Logout

Exits the Administrator.

3.8.2 Users Tab

The Users tab is where you add, edit, and delete users and specify what Oracle Load Testing for Web Applications 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.

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

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 for Web Applications.

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

3.8.3 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.

   Enable E-mail notification - select this option to enable email notification when new issues are created and when the owner or assigned to fields are changed for issues.

> **Enable Administrator Access** - gives this user the ability to log on to the Oracle Test Manager for Web Applications Administrator for managing the database.

2. Enter the user's information.

3. Select or clear the E-Mail notification and Administrator access options..

4. Click **OK**.

3.8.4 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.
>
> **Enable E-mail notification** - select this option to enable email notification when new issues are created and when the owner or assigned to fields are changed for issues.
>
> **Enable Administrator Access** - gives this user the ability to log on to the Oracle Test Manager for Web Applications Administrator for managing the database.

3. Make any changes.

4. Click **OK**.

3.8.5 Deleting Users

To delete a user:

1. Select the user you want to delete.

2. Click **Delete**.

3. Click **Yes** when asked to confirm the deletion.

3.9 Main Window Features

The Oracle Load Testing for Web Applications 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.

3.9.1 Overview of the Menu Options

The Oracle Load Testing for Web Applications main menu has the following options:

- Scenario
- Session
- ServerStats

- Tools

- Manage

- Help

- Logout

The following sections explain each of the menu options.

3.9.1.1 Scenario Menu

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

New- creates a new Oracle Load Testing for Web Applications scenario.

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

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

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

3.9.1.2 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.

3.9.1.3 ServerStats Menu

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

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.

3.9.1.4 Tools Menu

The following options are available:

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

VU Display - starts the virtual user display, 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.

Upload File - opens a dialog box for uploading a file to the repository.

Download File - opens a dialog box for downloading a file to the local system.

3.9.1.5 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.

User Defined Profiles - displays a dialog box for creating, editing, and deleting user defined profiles.

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

3.9.1.6 Help Menu

The following options are available.

Contents - opens the help system contents.

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

3.9.1.7 Logout

Exits Oracle Load Testing for Web Applications.

3.9.2 Toolbar

The toolbar has the following buttons:

New Scenario - Creates a new Oracle Load Testing for Web Applications scenario.

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

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

3.9.3 Build Scenarios Tab

The Build Scenarios tab is where you specify which scripts and user-defined profiles to include in the scenario.

The **Select scripts & user-defined profiles** for your scenario list shows the Oracle Functional Testing for Web Applications Visual Scripts and user-defined profiles in the current workspace.

The **Configure parameters of the scenario** list shows the scripts and user-defined profiles selected for the current Oracle Load Testing for Web Applications 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.

Repository - a list of repositories. 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**.

Workspace - lists the workspaces available in the selected repository. Workspaces store project-related Visual Script files, scenario files, and log files. You can use workspaces to organize these files.

> **Note:** Use Oracle Functional Testing for Web Applications to create and administer Workspaces.

<Script and user-defined profile list> - a list of Oracle Functional Testing for Web Applications Visual Scripts and user-defined profiles that are available to include in virtual user scenarios.

Configure parameters of the scenario - a list of scripts and user defined profiles 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.

 # VUs - specifies the number of virtual users to run for the selected profile. For each virtual user, Oracle Load Testing for Web Applications runs a separate instance of the Visual 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, select the system name of a system running either Oracle Load Testing for Web Applications Server or Oracle Load Testing for Web Applications Agent from the option dropdown. Systems are defined using the VU Agents 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.

User Mode - the mode in which to run Oracle Load Testing for Web Applications virtual users. The following options are available:

- **Thick Client** - the virtual users run using full browser capabilities. This mode requires more resources and is less scalable than Thin Client, and should only be used in special cases. Reasons for using Thick client may include Web pages that require large amounts of client side processing.

- **Thin Client** - a less resource intensive method that provides a high level of Web compatibility and Visual Script verification. Thin Client mode consumes fewer system resources per process than Thick Client, and is able to run more virtual users on a given agent system.

- **Java Client** - a highly scalable version of the Thin Client that executes a compiled Oracle Functional Testing for Web Applications script in native Java code. This agent provides a flexible code interface for performing customized scripting operations.

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.

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 Visual 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 Functional Testing for Web Applications Visual Script. You can set minimum and maximum delay times (in seconds) that override the Visual 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 for Web Applications sets the low end of the random range as the actual recorded user delay minus the Lower percentage setting. Oracle Load Testing for Web Applications 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 for Web Applications 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 Visual Scripts at the fastest possible speed with no time between page requests.

Each line also includes the following buttons:

Edit Scenario Details - Displays the Edit Scenario Details dialog box for configuring the parameters of each profile in the scenario.

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

Delete - Deletes the selected profile from the scenario.

For information about User-Defined Profiles, see Defining Virtual User Profiles in Chapter 4. For information about the Selected VU Profile settings see Defining Virtual User Scenarios in Chapter 5.

3.9.4 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 for Web Applications 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 for Web Applications 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 for Web Applications 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.

For information about using the Autopilot, see Using Autopilot in Chapter 6.

3.9.5 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 Using Autopilot in Chapter 6.

3.9.6 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 7.

3.9.7 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 8.

3.10 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.

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

For more information about using the System Manager, see Defining Systems later in this chapter.

3.11 Defining Systems

Before you can select systems in Oracle Load Testing for Web Applications Scenarios, you must define the machines that are Oracle Load Testing for Web Applications agent systems. The Oracle Load Testing for Web Applications System Manager lets you define system names or IP addresses and create system groups that the Oracle Load Testing for Web Applications 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 for Web Applications Agent software on each system and verifying network access between the Oracle Load Testing for Web Applications system and each agent system.

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

3.11.1 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 username to use for authentication.

   - **Password** - enter the password to use for authentication.

   Test - checks to see whether the Oracle Load Testing for Web Applications 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 the port and authentication settings.

6. Click **OK**.

7. Click **Close**.

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

3.11.2 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.

3.11.3 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.

3.11.4 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.

   Test - checks to see whether the Oracle Load Testing for Web Applications 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.

3.11.5 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 for Web Applications 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.

3.11.5.1 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 and NEC WebOTX 7.1

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

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 a specific configuration jar file to be copied from the JMX server installation to the OracleATS\DataCollector\classes directory on all machines that will be used as data collectors for sampling from that server. The file can be obtained from your application server installation and copied to the appropriate directory as listed in the following table.

Application Name	Files to Copy	Default Directory
BEA WebLogic 6.1	weblogic.jar	C:\OracleATS\DataCollector\classes\weblogic6.1
BEA WebLogic 7.0	weblogic.jar	C:\OracleATS\DataCollector\classes\weblogic7.0
BEA WebLogic 8.1	weblogic.jar, webservices.jar	C:\OracleATS\DataCollector\classes\weblogic8.1

Application Name	Files to Copy	Default Directory
BEA WebLogic 9.0	weblogic.jar, webservices.jar	C:\OracleATS\DataCollector\classes\weblogic9.0
IBM WebSphere 5.0	admin.jar, j2ee.jar, jflt.jar, jmxc.jar, mjxx.jar, log.jar, mail.jar, ras.jar, sas.jar, soap.jar, utils.jar, wasjmx.jar, wsexception.jar, wssec.jar	C:\OracleATS\DataCollector\classes\websphere5.0
IBM WebSphere 5.1n	admin.jar, bootstrap.jar, ffdc.jar, ibmjsse.jar, j2ee.jar, jflt.jar, jmxc.jar, jmxx.jar, log.jar, mail.jar, ras.jar, sas.jar, soap.jar, utils.jar, wasjmx.jar, wsexception.jar, wssec.jar	C:\OracleATS\DataCollector\classes\weblsphere5.1
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, ffdc.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.jar, mail-impl.jar, management.jar, pluginconfig.jar, ras.jar, runtime.jar, runtimefw.jar, sas.jar, security.jar, soap.jar, tcljava.jar, uddi4j.jar, utils.jar, validationmgr.jar, wasjmx.jar, wasproduct.jar, wccm_base.jar, webcontainer.jar, webservices.jar, wjmxapp.jar, wlmserver.jar, workspace.jar, wsdl4j.jar, wsexception.jar, wsprofile.jar, wssec.jar	C:\OracleATS\DataCollector\classes\weblsphere6.0
NEC WebOTX 7.1	j2ee.jar, wosv-rt.jar	C:\OracleATS\DataCollector\classes\webotx and C:\OracleATS\DataCollector\lib
NEC WebOTX 7.1	cacerts.jks (trust file)	C:\OracleATS\DataCollector\classes\webotx

Generating Jar Files for Weblogic 7.0, 8.1

Generate the weblogic-javax-management.jar file and copy it to the C:\OracleATS\DataCollector\classes\weblogic8.1 or 7.0 directory. This file contains all of the class files of javax.management from weblogic.jar.

Updating the Properties Files for WebLogic 8.1, WebSphere 5.0, 5.1, and 6.0

JMX must be set up for testing WebLogic and WebSphere. Use the following procedure for each application server you are testing. Note that only one type of application server can be tested at a time using a particular data collector.

1. Stop the Oracle Load Testing for Web Applications data collector.

2. Open the OracleATS\DataCollector\properties\jagent_service.properties file located in your installation directory.

3. Uncomment the line beginning with "`#JHP.commandline =`" corresponding to the application server you are testing by deleting the pound sign. Make sure that this is the only line that is uncommented.

4. Select **Save** from the **File** menu.

5. Copy the required jar files to the OracleATS\DataCollector\classes directory in the installation directory. These jar files are located on your WebSphere or WebLogic server machine in the Server LIB directory.

6. Restart the Oracle Load Testing for Web Applications data collector.

3.11.5.2 Adding a Monitored System

To add a new monitored system:

1. Select **System**s 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.

   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.

   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 for Web Applications 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 for Web Applications 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 for Web Applications.

  - **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 for Web Applications server and you are using a remotely installed SQL database for Oracle Load Testing for Web Applications, you need to install the SQL Server ODBC driver on the Oracle Load Testing for Web Applications 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 for Web Applications 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 for Web Applications 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 for Web Applications 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 for Web Applications uses the default instance as set up on your server. Refer to your database administrator for details.

- **Port** - Oracle Load Testing for Web Applications 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 Inet Sprinta driver for SQL Server, provide the database name. 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 4.0+

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

JMX

- **JServer Type** - enter 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.

  - BEA WebLogic 9.0

  - BEA WebLogic 8.1

  - BEA WebLogic 7.0

  - BEA WebLogic 6.1

  - IBM WebSphere 6.0

  - IBM WebSphere 5.1

  - IBM WebSphere 5.0

  - Fujitsu Interstage 7.0

  - Fujitsu Interstage 6.0

  - NEC WebOTX 7.1

- **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

Note: If you are using SSH for the connection method, you must download plink.exe and copy it to the OracleATS\DataCollector\bin directory on the system on which the data collector is installed. The file name must be: username_host.key. The extension must be .key not .ppk. Leave the password file blank. plink is part of the PUTTY package by Simon Tatham. It is downloadable from:

http://www.chiark.greenend.org.uk/~sgtatham/putty/

- **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 command line prompt for the host machine. If you do not specify a command line prompt, Oracle Load Testing for Web Applications 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 for Web Applications 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 for Web Applications 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, Inet Sprinta JDBC driver) are 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 for Web Applications Data collectors.

- **Inet Sprinta JDBC Driver for Microsoft SQL Server** - This driver option applies to SQL databases. This driver is installed automatically as part of Oracle Load Testing for Web Applications. For monitoring enterprise level SQL databases, the Inet Sprinta JDBC driver is recommended over the JDBC:ODBC Bridge option.

- **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 for Web Applications.

  – **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 for Web Applications server and you are using a remotely installed SQL database for Oracle Load Testing for Web Applications, you need to install the SQL Server ODBC driver on the Oracle Load Testing for Web Applications 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 for Web Applications machine in order to use this driver.

- **Driver** - Select a driver type from the list: Oracle Thin JDBC driver, Inet Sprinta JDBC driver for Microsoft SQL Server, Sun JDBC:ODBC Bridge, or Custom. You must have the appropriate driver installed on the Oracle Load Testing for Web Applications 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 for Web Applications 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 for Web Applications uses the default instance as set up on your server. Refer to your database administrator for details.

- **Port** - Oracle Load Testing for Web Applications 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 Inet Sprinta driver for SQL Server, provide the database name. 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 it's 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 and system groups appear in the Systems list on the Build Scenarios tab.

3.11.6 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**.

3.11.7 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**.

3.11.8 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**.

3.12 Setting Options

You can set Oracle Load Testing for Web Applications options for custom browsers, download rules, 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.

3.12.1 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.

3.12.2 Setting Download Rules

These options let you specify which types of objects to download in Oracle Load Testing for Web Applications Thin Client and Java Client user modes. You can specify background images, ActiveX, CAB, JAR, CSS, Flash, Applets, and JavaScript library files. To specify objects, you define the HTML tags and attributes to use to filter references from other HTML in the page source.

Filtering is enabled when the **Use Download Manager** option is selected for the Edit Scenario Details dialog box.

To use the Download Manager:

1. Determine which types of files you want Oracle Load Testing for Web Applications Thin Client or Java Client to download.

2. Select **Options** from the **Tools** menu.

3. Select **Download Rules**.

   The grid on the right side of the dialog box lists existing filters. The following options are available:

   New - adds the new filter to the existing filters list.

   Delete - removes the currently selected filter from the existing filters list.

   Existing Filters grid fields - lists the following information for the defined download filters.

   - **Tag Name** - specifies the selected HTML tag.

   - **Enabled** - shows if the filter is enabled or disabled. Filters marked as true are enabled and will be used to download objects.

   - **Attribute Name** - specifies the selected attribute used to identify which specific Tag among all tags of the same Tag Name.

   - **Attribute Value** - specifies the value for the specified attribute name used to identify which specific Tag.

   - **Download Attr** - specifies which attribute identifies the file to download.

   Filter Configuration - defines specific filter options to use for each download filter. The filter options identify specific tags and attributes in the HTML source that reference the object(s) to download. Set the following options and click **New** to add the filter to the existing filters list.

   Tag Name - specifies the HTML tag to use to locate an object within the HTML source.

   Enabled - select true to enable the filter. You can add filters to the Existing Filters list even if they are not enabled. You can select filters in the existing filters list at any time to enable or disable them for specific tests.

   Attr Name - specifies the name part of the name/value pair that identifies the specific Tag among all tags of the same Tag Name. Select the attribute name to use to identify the tag(s) in the HTML source.

   Attr Value - specifies the value part of the name/value pair that identifies the specific Tag among all tags of the same Tag Name. Specify the attribute value to use to identify the tag(s) in the HTML source.

   Download Attr - specifies the tag attribute that identifies the object to download. For example, IMG tags reference the image filename using the SRC attribute.

To modify a filter, select it, then make changes using these fields. The changes automatically appear in the existing filters list.

4. Click **New**.

5. Set the Enabled field to **true** if you want to define and enable the download filter for use in Thin Client or Java Client. Set it to **false** if you just want to define a filter for future use but not enable it right away.

6. Select the HTML tag name that has an attribute that references the object to download.

7. Select the tag attribute name to identify specific HTML tags.

8. Enter the tag attribute value that identifies specific HTML tags. The **Download** setting shows the tag attribute that references the object to download.

9. Click **New** to add another filter or select an existing filter to change it.

10. Click **OK**.

11. Click the **Edit Scenario Details** button.

12. In the **Browser Settings** section, make sure that the **Use Download Manager** field is True.

3.12.3 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 for Web Applications 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 for Web Applications Server must have access to that drive. By default, the Oracle Load Testing for Web Applications 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.

5. Enter the location of the repository.

3.12.4 Setting Scenario Defaults

Setting Scenario DefaultsYou 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 or user-defined profile 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 for Web Applications runs a separate instance of the Visual 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 for Web Applications or Oracle Load Testing for Web Applications 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.

- **User Mode** - the mode in which to run Oracle Load Testing for Web Applications virtual users. The following options are available:

  - **Thick Client** - the virtual users run using full browser capabilities. This mode requires more resources and is less scalable than Thin Client, and should only be used in special cases. Reasons for using Thick client may include Web pages that require large amounts of client side processing.

  - **Thin Client** - a less resource intensive method that provides a high level of Web compatibility and Visual Script verification. Thin Client mode consumes fewer system resources per process than Thick Client, and is able to run more virtual users on a given agent system.

  - **Java Client** - a highly scalable version of the Thin Client that executes a compiled Oracle Functional Testing for Web Applications script. This agent provides a flexible code interface for performing customized scripting operations.

    > **Note:** User-defined scriptlets will not be executed when using the Java Client regardless of how the **Execute User Defined Scriptlets** options is set.

    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 Visual Script playback delay for each virtual user. There are four options:

  - **Recorded** - uses the delay times that were recorded in the Oracle Functional Testing for Web Applications Visual Script. You can set minimum and maximum delay times (in seconds) that override the Visual 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 for Web Applications sets the low end of the random range as the actual user delay minus the Lower percentage setting. Oracle Load Testing for Web Applications 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 for Web Applications 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 Visual Scripts at the fastest possible speed.

- **Use Data Bank** - when true, Visual Scripts that have Oracle Functional Testing for Web Applications Data Banks will use the Data Banks as part of the virtual user playback. When false, Visual 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 Emulation** - specifies the type of user to simulate. This is useful for simulating different profiles of virtual users. A first time user places more of a load on the Web server because pages and image are not yet cached. A repeat user places less of a load on the server as only newer pages are requested and brought down from the Web server. There are three options:

- **Do Not Cache** - cache is not used.

> **Note:** When using WinInet, selecting **Do Not Cache** has the same effect as selecting **1st Time** as the Cache Emulation method. In this case, document cache is not created and not cleared. When using Thin Client for JHTTP, selecting **Do Not Cache** does not create cache or clear it. When the Cache Emulation method is set to **1st Time** for Thin Client for JHTTP, cache is created and cleared after each iteration which can consume considerable amounts of memory.

- **1st Time** - the virtual users are considered to be using the Web site or application for the first time for each iteration so no cache is used. The cache is used during the iteration.

- **Repeat** - the virtual users are considered to have visited or used the Web site or application previously. Pages and images are retrieved from the cache.

- **Use IP Spoofing** - when true, Oracle Load Testing for Web Applications 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 for Web Applications Agents in the TCP/IP network protocols of the system. All IP addresses must be added to each Agent system. When using IP spoofing, it is recommended that VUs run with Thin Client without WinInet or with Java Client. See Using IP Spoofing with Oracle Load Testing for Web Applications in Chapter 5 for additional information.

- **Use Download Manager** - when true with Thin Client or Java Client User Mode, filters specified and enabled in **Download Rules** (select **Options** from the **Tools** menu) will be used to download objects as part of virtual user playback. When true with Thick Client User Mode, image objects will be downloaded as part of virtual user playback. Downloaded objects will be included as timers if you have **Auto generate timers for all resources** selected in the **Scenario Defaults** (select **Options** from the **Tools** menu) and save data for reporting.

- **Use Winninet** - when true, Oracle Load Testing for Web Applications uses its Microsoft Win32 Internet (WinInet) application programming interface (API) version of the Thin Client agent. When false, Oracle Load Testing for Web Applications uses a custom (JHTTP) version of the Thin Client agent (referred to as "Thin Client for JHTTP").

- **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 for Web Applications runs Oracle Functional Testing for Web Applications Text Matching and Server Response tests.

- **Execute User Defined Scriptlets** - when true for Thick or Thin client, Oracle Load Testing for Web Applications runs Test Scriptlets included in Oracle Functional Testing for Web Applications Visual Scripts. Scriptlets are not executed when using the Java Client regardless of how this option is set. See the Oracle Functional Testing for Web Applications User Guide for more information about using Test Scriptlets with Oracle Functional Testing for Web Applications.

VU Display - the VU Display settings are as follows:

- **View All Responses** - select when you want to display all responses, on error, always, never.

- **Show Request Headers** - select when you want to display request headers, on error, always, or never. When displayed, request header information for Web page resources requested by the selected Virtual User appear in the Virtual User Display history list.

- **Show Response Headers** - select when you want to display response headers, on error, always, or never. When displayed, the Response header information for Web page resources requested by the selected Virtual User appear in the Virtual User Display history list.

Reporting - the Reporting settings are as follows:

■ **Auto Generate Timers For All Pages** - when true, Oracle Load Testing for Web Applications automatically adds timers for each Visual Script page for reporting. The timers are used in Oracle Load Testing for Web Applications to provide performance monitoring and timing information for each page of the Visual Script(s) played back by a virtual user profile.

■ **Auto Generate Timers For All Resources** - when true, Oracle Load Testing for Web Applications 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 **Download Rules** setting (select **Options** from the **Tools** menu).

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.

■ **Zero Length Download Errors Are Fatal** - when true, a server response that indicates zero bytes length is considered a fatal error. Set this setting to false if your Visual Scripts are recorded as Siebel scripts.

■ **On Error Stop Virtual User** - when true, all virtual users are stopped if an error is encountered.

■ **Request Timeout** - specifies the maximum amount of time a virtual user waits to access a page before timing out.

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 for Web Applications 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.

■ **Maximum HTTP Threads Per Process** - specifies the number of threads per agent process for optimizing the pooling of virtual user requests into a finite number of I/O worker threads.

■ **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 KBase article Q183110 for more information.) This setting is used to override the agent machine configuration for Thin Client agents using WinInet.

■ **Ignore HTTP Proxy Settings** - specifies whether to ignore the agent machine's default proxy setting as defined in Internet Explorer.

Java Client Preferences - these options only appear when User Mode is set to **Java Client**. 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.

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

- **Maximum Bytes of VU Display** - when you select other, enter the number of bytes to send to the VU display per item, that is, a request is one item, a response is an item, and content is an item. The default in the JavaAgent.properties file is-1, which returns everything. The value entered roughly equates to the number of characters.

- **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, gzip support is enabled.

- **Enable Deflate** - when true, deflate compression support is enabled.

- **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 version.

- **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, keep alive is enabled.

- **Preserve Connections Between Iterations** - used to preserve connections between OpenScript and the browser between successive iterations of the script.

- **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 SSL version.

- **Agent Inactivity Timeout** - specify the length of time, in milliseconds, that Oracle Load Testing for Web Applications will wait to hear from the agent before it assumes that the agent is not available.

- **Ignored Url** - specify the Urls, separated by commas, that should not be requested. This setting only applies to certain OpenScript scripts.

3.12.5 Setting Autopilot Defaults

The Autopilot Defaults options let you specify the default settings for the Set up Autopilot tab. Refer to "Chapter 6" for a description of these options.

3.12.6 Setting Session Start and Stop Options

Sessions specify the scope for Oracle Load Testing for Web Applications 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 bu selecting **Options** from the **Tools** menu then selecting **Start and Stop**. See Chapter 5 for a description of these options.

3.12.7 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.

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

3.12.8 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 Chapter 8 for a description of these fields.

3.12.9 Setting Data Bank Defaults

These options set the Data Bank control defaults that will be displayed when you click the Configure Data Bank button for a script or profile on the Build Scenarios tab.

Record buffer size - specifies the number of Data Bank records allocated to each Oracle Load Testing for Web Applications agent.

Data Usage - lets you select which records to use.

- **Loop Forever** - specifies that the virtual users cycle through the Data Bank records until the load test finishes.

- **Use All Records** - specifies that the virtual users use all of the records in the Data Bank and then stop. For example, if the virtual users start at record 50 of 100, they run from 50 to 100, then loop back and run from record 1 to 49 and then stop.

- **Stop After Last Record** - specifies that the virtual users stop when the end of the Data Bank has been reached regardless of the starting point in the Data Bank. For

example, if the virtual users start at record 50 of 100 records, they run from record 50 to record 100 and then stop.

- **First Record Only** - specifies that each virtual user takes the next record in the Data Bank and keeps on using that same record for each iteration. Use this setting if you want each virtual user to keep the same data for each iteration.

3.12.10 Setting General Options

These options determine whether Oracle Load Testing for Web Applications will validate the hostname/IP address when you add a system.

Validate hostname/ip when user adds a system - checks to see if the Oracle Load Testing for Web Applications server can connect to the specified system. If it cannot, Oracle Load Testing for Web Applications 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 for Web Applications Server IP address changes - restarts or stops the Oracle Load Testing for Web Applications Application Service when the state of the network changes.

- **Restart Application Service** - restarts the application service when the IP address changes.

- **Stop Application Service** - stops the application service when the IP address changes.

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

4

Defining Virtual User Profiles

This chapter explains how to define virtual user profiles for use with Oracle Load Testing for Web Applications scenarios. Virtual user profiles specify the series of actions to perform on a Web site or application, which are recorded in Oracle Functional Testing for Web Applications Visual Scripts.

4.1 Default Profiles

Visual Scripts that you record using Oracle Functional Testing for Web Applications are default virtual user profiles. When you open a Workspace in Oracle Load Testing for Web Applications, any Oracle Functional Testing for Web Applications Visual Scripts in that Workspace and repository appear in the **Select scripts and user defined profiles** list of the Build Scenarios tab.

You can include any Oracle Functional Testing for Web Applications Visual Script in the **Select scripts and user defined profiles** list to play back the Visual Script as virtual users.

4.2 Creating User-Defined Profiles

User-defined virtual user profiles allow you to combine multiple Visual Scripts as a single virtual user profile. User-defined profiles must be specifically defined in Oracle Load Testing for Web Applications and appear in the **Select scripts and user defined profiles** list of the Build Scenarios tab.

In a user-defined profile, you can specify certain Visual Scripts to use as prolog, epilog, and error scripts. You can also insert synchronization points into a user-defined profile.

4.2.1 Creating a New Virtual User Profile

To create a new User Defined Profile:

1. Select **User Defined Profiles** from the **Manage** menu to display the User Defined Profile Manager.

   Repository - lists the configured repositories.

   Workspace - lists the workspaces in the selected repository.

   New - displays the Add User Defined Profile dialog box for adding a new profile.

   Edit - displays the Edit User Defined Profile dialog box for editing the selected profile.

   Name - lists the user defined profiles in the selected workspace.

2. Click **New** to display the Add User Defined Profile dialog box.

Repository - select the repository containing the workspace with the scripts you want to add to the user defined profile.

Workspace - select the workspace containing the scripts that you want to add to the profile. The scripts are listed in the box below these fields.

<Scripts> - Oracle Functional Testing for Web Applications Visual Scripts available to include in user defined profiles.

Name - the name of the user defined profile. Enter a name for the profile.

Add Sync - opens a dialog box for defining the synchronization point's name and parameters.

Edit Sync - opens a dialog box for changing the synchronization point's name and parameters.

Delete - deletes the selected script or synchronization point from the section tree.

<Sections> - the sections of the profile, as follows:

Prolog - the Visual Scripts in this section play back only once at the beginning of the scenario run. An example of what you may add to this section is a login script.

Run - the Visual Scripts in this section iterate as many times as is specified in the Autopilot. An example of what you may add in this section is the business transaction that you wish to load test.

Epilog - the Visual Scripts in this section play back only once at the end of the Scenario run. An example of what you add to this section is a logoff script.

Errors - the Visual Scripts in this section play back only if an error occurs during the Scenario run. An example of what you may add to this section is a visual script that resets your application on an error.

Add - adds the selected script to the sections tree.

<Up arrow> - moves the selected item up within the section of the Sections tree.

<Down arrow> - moves the selected item down within the section of the Sections tree.

3. Select the repository where the workspace containing you Visual scripts are located.

4. Select the Workspace where your Visual Scripts are located. The available Visual Scripts appear in the box below these fields.

5. Enter a name for the job in the Name field.

6. Select the section in the tree where you want to add a Visual Script.

7. Double-click the Visual Script in the left-hand pane's list to add to the section or select the script and click the **Add** button.

8. Use the up and down arrows to move scripts in the section pane.

9. Repeat these steps to add more Visual Scripts to the sections tree.

10. Add synchronization points as necessary.

11. Click **OK**.

12. Click **OK** to exit the User Defined Profile Manager.

4.2.1.1 Adding Visual Scripts to the Sections Tree

To add Visual Scripts to the Sections tree of a user-defined profile:

1. Open a user-defined profile for editing.

2. Select the section on the Sections tree where you want to add a Visual Script.

3. Double-click the Visual Script (in the Profiles list) to add to the section or select the script and click the **Add** button.

4. Repeat steps 2 and 3 to add additional Visual Scripts to the Sections tree.

5. Click the **OK** button when you have finished editing the profile.

6. Click **Close** to exit the User Defined Profile Manager.

4.2.1.2 Adding Synchronization Points to the Sections Tree

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.

To add Sync point to the Sections tree of a user-defined profile:

1. Select **User Defined Profiles** from the **Manage** menu.

2. Select the Repository and Workspace that contains the profile that you want to edit.

3. Select the profile and click the **Edit** button.

4. Select the section on the Sections tree where you want to add a Sync point and click **Add Sync**.

5. Select the Sync Point and click **Edit**.

   Name - specifies the name of the Synchronization point.

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

   - **All users** - 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.

6. Specify the sync point parameters and click **OK**.

7. Repeat these steps to add additional Sync points to the Sections tree.

8. Click the **OK** button when you have finished editing the profile.

4.2.1.3 Moving Items in the Sections Tree

When you have multiple items under any section of the tree, you can move the items up and down within that section. To move an item in the Sections tree of a user defined profile:

1. Select User **Defined Profiles** from the **Manage** menu to display the User Defined Profile Manager.

2. Select the profile you want to edit and click Edit to display the Edit User Defined Profile dialog box.

3. Select the item to move in a section.

4. Click the up or down button as appropriate.

5. Click the OK button when you have finished defining the profile.

4.2.2 Editing User-Defined Profiles

After you have created a user defined profile, you can make changes to it at any time. To edit a User Defined Profile:

1. Select **User Defined Profiles** from the **Manage** menu to display the User Defined Profile Manager.

2. Select the profile you want to edit and click Edit to display the Edit User Defined Profile dialog box.

3. Click the icons to expand the nodes of the tree.

4. Add a script by opening the repository and workspace where the script is located. Select the script and click the **Add** button.

5. Use the up and down arrows as necessary to move scripts in the sections tree.

6. Remove an item from sections tree by selecting it and clicking the **Delete** button.

7. Edit synchronization points, if necessary.

8. Click the **OK** button when you have finished editing the profile.

4.2.3 Editing Synchronization Points

To edit a synchronization point:

1. Select **User Defined Profiles** from the **Manage** menu to display the User Defined Profile Manager.

2. Select the profile you want to edit and click **Edit**. Oracle Load Testing for Web Applications opens a dialog for editing the sections tree of the profile. The dialog box shows the current sections tree, the available Visual Scripts, and the default synchronization points.

3. Select the Synchronization Point you want to edit in the sections tree.

4. Click **Edit Sync** to display the Edit Sync Point dialog box.

5. Make changes and click **OK**.

6. Click **Apply**.

7. Click **OK**.

5

Defining Virtual User Scenarios

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

5.1 Defining Scenarios

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 visual scripts
- type of user to simulate
- object downloading
- use of Data Banks
- error handling for virtual users
- setting up virtual users for viewing in the Virtual User Display component

The following sections explain how to specify virtual user scenarios.

5.1.1 Selecting Profiles

Once you have a set of virtual user profiles, you can define Oracle Load Testing for Web Applications scenarios to run performance and load testing on your Web pages or application. Virtual user profiles can be either default profiles (basic Oracle Functional Testing for Web Applications Visual Scripts) or user-defined profiles created in Oracle Load Testing for Web Applications.

To add profiles to the Scenario Profiles 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 profile you want to add.

> **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 profile name in the profile list.

5.1.2 Specifying Scenario Profile Attributes

Once you select the scripts and user-defined profiles to include in the scenario, you can specify the Scenario parameters for each one. To set the parameters for a specific script or user defined profile.

1. If necessary, open a saved Scenario file.

2. Select the Build Scenarios tab.

3. Select the scripts and user-defined profiles to include in the Scenario.

4. Click the **Scenario Detail** button to display the Edit Scenario Details dialog box.

> **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.

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**.

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 for Web Applications runs a separate instance of the Visual 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 for Web Applications or Oracle Load Testing for Web Applications 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.

- **User Mode** - the mode in which to run Oracle Load Testing for Web Applications virtual users. The following options are available:

  - **Thick Client** - the virtual users run using full browser capabilities. This mode requires more resources and is less scalable than Thin Client, and should only be used in special cases. Reasons for using Thick client may include Web pages that require large amounts of client side processing.

  - **Thin Client** - a less resource intensive method that provides a high level of Web compatibility and Visual Script verification. Thin Client mode

—

consumes fewer system resources per process than Thick Client, and is able to run more virtual users on a given agent system.

– **Java Client** - a highly scalable version of the Thin Client that executes a compiled Oracle Functional Testing for Web Applications script. This agent provides a flexible code interface for performing customized scripting operations.

> **Note:** User-defined scriptlets will not be executed when using the Java Client regardless of how the **Execute User Defined Scriptlets** options is set.

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 Visual Script playback delay for each virtual user. There are four options:

  – **Recorded** - uses the delay times that were recorded in the Oracle Functional Testing for Web Applications Visual Script. You can set minimum and maximum delay times (in seconds) that override the Visual 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 for Web Applications sets the low end of the random range as the actual user delay minus the Lower percentage setting. Oracle Load Testing for Web Applications 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 for Web Applications 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 Visual Scripts at the fastest possible speed.

- **Use Data Bank** - when true, Visual Scripts that have Oracle Functional Testing for Web Applications Data Banks will use the Data Banks as part of the virtual user playback. When false, Visual 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 Emulation** - specifies the type of user to simulate. This is useful for simulating different profiles of virtual users. A first time user places more of a load on the Web server because pages and image are not yet cached. A repeat user places less of a load on the server as only newer pages are requested and brought down from the Web server. There are three options:

- **Do Not Cache** - cache is not used.

Note: When using WinInet, selecting **Do Not Cache** has the same effect as selecting **1st Time** as the Cache Emulation method. In this case, document cache is not created and not cleared. When using Thin Client for JHTTP, selecting **Do Not Cache** does not create cache or clear it. When the Cache Emulation method is set to **1st Time** for Thin Client for JHTTP, cache is created and cleared after each iteration which can consume considerable amounts of memory.

- **1st Time**- the virtual users are considered to be using the Web site or application for the first time for each iteration so no cache is used. The cache is used during the iteration.

- **Repeat** - the virtual users are considered to have visited or used the Web site or application previously. Pages and images are retrieved from the cache.

- **Use IP Spoofing** - when true, Oracle Load Testing for Web Applications 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 for Web Applications Agents in the TCP/IP network protocols of the system. All IP addresses must be added to each Agent system. When using IP spoofing, it is recommended that VUs run with Thin Client without WinInet or with Java Client. See Using IP Spoofing with Oracle Load Testing for Web Applications in Chapter 5 for additional information.

- **Use Download Manager** - when true with Thin Client or Java Client User Mode, filters specified and enabled in **Download Rules** (select **Options** from the **Tools** menu) will be used to download objects as part of virtual user playback. When true with Thick Client User Mode, image objects will be downloaded as part of virtual user playback. Downloaded objects will be included as timers if you have **Auto generate timers for all resources** selected in the **Scenario Defaults** (select **Options** from the **Tools** menu) and save data for reporting.

- **Use Winninet** - when true, Oracle Load Testing for Web Applications uses its Microsoft Win32 Internet (WinInet) application programming interface (API) version of the Thin Client agent. When false, Oracle Load Testing for Web Applications uses a custom (JHTTP) version of the Thin Client agent (referred to as "Thin Client for JHTTP").

- **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 for Web Applications runs Oracle Functional Testing for Web Applications Text Matching and Server Response tests.

- **Execute User Defined Scriptlets** - when true for Thick or Thin client, Oracle Load Testing for Web Applications runs Test Scriptlets included in Oracle Functional Testing for Web Applications Visual Scripts. Scriptlets are not executed when using the Java Client regardless of how this option is set. See the Oracle Functional Testing for Web Applications User Guide for more information about using Test Scriptlets with Oracle Functional Testing for Web Applications.

VU Display - the VU Display settings are as follows:

- **View All Responses** - select when you want to display all responses, on error, always, never.

- **Show Request Headers** - select when you want to display request headers, on error, always, or never. When displayed, request header information for Web page resources requested by the selected Virtual User appear in the Virtual User Display history list.

- **Show Response Headers** - select when you want to display response headers, on error, always, or never. When displayed, the Response header information for Web page resources requested by the selected Virtual User appear in the Virtual User Display history list.

Reporting- the Reporting settings are as follows:

- **Auto Generate Timers For All Pages** - when true, Oracle Load Testing for Web Applications automatically adds timers for each Visual Script page for reporting. The timers are used in Oracle Load Testing for Web Applications to provide performance monitoring and timing information for each page of the Visual Script(s) played back by a virtual user profile.

- **Auto Generate Timers For All Resources** - when true, Oracle Load Testing for Web Applications 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 **Download Rules** setting (select **Options** from the **Tools** menu).

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.

- **Zero Length Download Errors Are Fatal** - when true, a server response that indicates zero bytes length is considered a fatal error. Set this setting to false if your Visual Scripts are recorded as Siebel scripts.

- **On Error Stop Virtual User** - when true, all virtual users are stopped if an error is encountered.

- **Request Timeout** - specifies the maximum amount of time a virtual user waits to access a page before timing out.

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 for Web Applications 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.

- **Maximum HTTP Threads Per Process** - specifies the number of threads per agent process for optimizing the pooling of virtual user requests into a finite number of I/O worker threads.

- **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 KBase article Q183110 for more information.) This setting is used to override the agent machine configuration for Thin Client agents using WinInet.

- **Ignore HTTP Proxy Settings** - specifies whether to ignore the agent machine's default proxy setting as defined in Internet Explorer.

Java Client Preferences - these options only appear when User Mode is set to **Java Client**. 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.

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

- **Maximum Bytes of VU Display** - when you select other, enter the number of bytes to send to the VU display per item, that is, a request is one item, a response is an item, and content is an item. The default in the JavaAgent.properties file is-1, which returns everything. The value entered roughly equates to the number of characters.

- **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, gzip support is enabled.

- **Enable Deflate** - when true, deflate compression support is enabled.

- **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 version.

- **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, keep alive is enabled.

- **Preserve Connections Between Iterations** - used to preserve connections between OpenScript and the browser between successive iterations of the script.

- **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 SSL version.

- **Agent Inactivity Timeout** - specify the length of time, in milliseconds, that Oracle Load Testing for Web Applications will wait to hear from the agent before it assumes that the agent is not available.

- **Ignored Url** - specify the Urls, separated by commas, that should not be requested. This setting only applies to certain OpenScript 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
```

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 executio..

- 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 upload your script to Oracle Load Testing for Web Applications 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 for Web Applications 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.

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:

■ **Connect Error** - specifies the error recovery action if a server connection error occurs.

■ **Read/Write Message Error** - specifies the error recovery action if a read or write error occurs with an Oracle Forms message.

■ **Control Initialized Fail** - specifies the error recovery action if server control fails to initialize.

5. Select the script or user defined profile that you want to edit from the list.

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 or user defined profile to another.

8. Click **OK**.

5.1.3 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:

■ The User Mode (type of client browser) setting for the virtual user;

■ 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 for Web Applications 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.

■ The **User Mode** setting on the **Build Scenarios** tab of Oracle Load Testing for Web Applications affects the amount of memory overhead required for each Virtual User process (single or separate). Thick Client requires more system resources per Virtual User than Thin Client. Thin Client requires 350 KB to 500 KB per VU. Thick client can require anywhere from 1-2 MB to 10 MB depending on the application and the script size. The application being tested will also affect the memory requirements for virtual users depending upon the size of the Visual Script and design of the Web applications. Web applications using scripts or applets may require additional memory.

■ 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.

5.1.4 Managing Sessions

Sessions specify the scope for Oracle Load Testing for Web Applications data collection and reporting. The data collected while the Autopilot is running virtual users is shown in the virtual user grid, Oracle Load Testing for Web Applications 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.

- **Save data for reporting** - specifies if virtual user data is saved for real-time and post testing analysis reports. There are three options:

    - **No** - the virtual user data is not saved. There will be no real-time or post-run graphs.

    - **Yes** - the virtual user data is saved to the database. You can use Create Reports tab at a later time to generate reports and graphs for analysis.

    - **Ask** - Oracle Load Testing for Web Applications prompts you to save the data each time you start a new Autopilot 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 for Web Applications 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 for Web Applications stops updating runtime data in the performance statistics, load graphs, and the database (if **Save data for reporting** is used). 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 for Web Applications 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 e-Test 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.

5.1.5 Using the Data Bank Control

You can use the Data Bank Control to view Data Bank records and set Oracle Load Testing for Web Applications preferences.

> **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 Visual 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 for Web Applications agent uses Data Bank records.

5. Click **OK** to exit the dialog box.

6. Click the **Configure Data Bank** button.

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

   Data Bank Variables - 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.

   Record Usage - lets you configure how the Data Bank is used.

   - **Set/View Record #** - shows the current Data Bank record number. You can change the current record using the arrow buttons or by entering a record number and pressing Enter.

   - **Record buffer size** - specifies the number of Data Bank records allocated to each Oracle Load Testing for Web Applications agent.

   - **Data Usage** - lets you select which records to use.

     – **Sequential - Use Starting Record Only** - specifies that each virtual user takes the next record in the Data Bank and keeps on using that same record for each iteration. Use this setting if you want each virtual user to keep the same data for each iteration.

     – **Sequential - Loop Records Continuously** - specifies that the virtual users cycle through the Data Bank records until the load test finishes.

     – **Unique - Stop After All records Used** - specifies that the virtual users use all of the records in the Data Bank and then stop. For example, if the virtual users start at record 50 of 100, they run from 50 to 100, then loop back and run from record 1 to 49 and then stop.

     – **Unique - Stop After Last Record Used** - specifies that the virtual users stop when the end of the Data Bank has been reached regardless of the

starting point in the Data Bank. For example, if the virtual users start at record 50 of 100 records, they run from record 50 to record 100 and then stop.

7. Make sure the script or profile is selected in the tree view.

8. Use the arrow buttons or enter a record number in the **Set/View Record #** field.

9. Click **OK**.

5.2 Using IP Spoofing

Oracle Load Testing for Web Applications 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 for Web Applications 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 for Web Applications virtual users.

5. Repeat the above steps on each Oracle Load Testing for Web Applications Agent system.

6. In the Edit Scenario Details dialog box, set **Use IP Spoofing** to True in the Browser Settings section.

7. If you are using WinInet, set the **Maximum Users Per Process** to one in the Advanced section.

8. 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 for Web Applications 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 for Web Applications uses layering on the TCP/IP stack to perform IP spoofing with virtual users. While virtual users are running, the Oracle Load Testing for Web Applications 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 for Web Applications virtual users with IP spoofing, other applications that use the TCP/IP stack may be affected. If this occurs, use regsvr32.exe to unregister the file sporder.dll.

5.3 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 for Web Applications automatically assigns the name LoadTest# for the scenario.

5.3.1 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 for Web Applications saves the current Autopilot settings as part of a saved scenario.

5.3.2 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 for Web Applications.

To open an existing Scenario:

1. Select **Open** from the **Scenario** menu. Oracle Load Testing for Web Applications opens a dialog for selecting the scenario file. Scenario files have a .scn 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**.

5.3.3 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.

5.3.4 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.

5.3.5 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.

5.4 Submitting Scenarios to Autopilot

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

5.4.1 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 for Web Applications automatically opens the Autopilot tab with the scenario loaded.

5.4.2 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 for Web Applications automatically opens the Autopilot tab with the scenario loaded and starts the virtual user run.

6

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 select **Autopilot Defaults**.

> **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 explains how to define virtual user profiles. Chapter 5 explains how to define scenarios and submit them to the Autopilot.

6.1 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.

6.1.1 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 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.

6.1.2 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 plays** - 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.

6.2 Specifying Virtual User Rampup

The **Rampup Specification** 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.

6.2.1 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.

6.2.2 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.

6.3 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.

6.3.1 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 for Web Applications displays the remaining time until the Autopilot starts in the status bar.

6.3.2 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.

6.4 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.

6.4.1 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 played back the user profile.

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 Page - shows the title of the web page or application 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 Functional Testing for Web Applications Data Bank, if any, used by the Visual 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 Visual 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.

6.4.2 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.

- **Start Viewing** - opens the Virtual User Display window for viewing the playback actions of the virtual user.

- **Stop Viewing** - closes the Virtual User Display window.

- **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 Visual Script and terminates execution altogether.

- **Release Sync** - releases the virtual user from any synchronization point that may have been reached during the execution.

6.4.3 Managing Synchronization points

If you create user-defined profiles 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.

6.4.3.1 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**.

6.4.3.2 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**.

6.4.3.3 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**.

6.4.4 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 for Web Applications 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.

   - **View All Responses** - select when you want to display all virtual user responses, on error, always, or never.

   - **Show Request Headers** - select when you want the Request header information for Web page resources requested by the selected Virtual User to appear in the Virtual User Display history list, on error, always, or never.

   - **Show Response Headers** - select when you want the Response header information for Web page resources requested by the selected Virtual User to appear in the Virtual User Display history list, on error, always, or never.

6.4.5 Viewing Server Warnings

The Oracle Load Testing for Web Applications 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.

6.4.6 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.

6.4.7 Stopping the Autopilot From Starting New Virtual Users

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

6.4.8 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 Visual Script.

7

Using Graphs and Reports

Oracle Load Testing for Web Applications 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 for Web Applications 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 for Web Applications 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.

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 for Web Applications 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.

7.1 Setting the Reporting Options

To set the Oracle Load Testing for Web Applications Report options:

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

2. Select **Start and Stop**.

3. Set the **Save data for reporting** option to either Yes or Ask.

4. 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 for Web Applications agent performance data is collected for reporting to the Oracle Load Testing for Web Applications 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 Display auto display interval** - specifies how often to refresh the virtual user display 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 Display - the VU Display options are as follows:

- **Maximum total storage size** - specifies the maximum amount of data, in bytes, to send to the VU Display. Indicates the maximum number of bytes the Virtual User Display log file (.XML) will store for all items in the history list. The default value is 10 MB for each session. The size of the log file is affected by the size of the Web pages. The HTML source for each page is stored in the log file.

- **Maximum number of rows per page** - specifies the maximum number of rows per page to display in the VU display. The value must be between 10 and 100, inclusive.

Timer name format - specifies the format of page timer names for reporting purposes when Auto Generate Timers For All Pages is selected. The following options are available:

- **Default** - the default naming convention, e.g. MyScript.Page[2]. This setting disables the other four options.

- **URL resource name** - the base file name, e.g. home.html is added.

- **Frame name** - the HTML frame name, e.g. top, is added.

- **Page number** - the page number, e.g. Page[2] is added.

- **Profile name** - the profile name, e.g. MyScript is added.

Performance Statistics and Session Report - the Performance Statistics and Session Report options are as follows:

- **Include think time in overall profile time** - specifies whether to include think time in performance statistics and session reports. This setting only applies for tests run using Thin or Java Agent.

Custom Graphs - the Custom Graphs options are as follows:

- **Maximum Data Series Per Graph** - specifies the maximum number of data series that can be graphed. The default is 20.

5. Enter the how often you want data to be collected in the **Reporting data collection interval** field.

6. 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.

7. Enter the how often you want to refresh the report display in the **Graph refresh interval** field.

8. Specify how you want timer names to be displayed.

9. Specify whether to include think time in the profile time.

10. Specify the maximum number of data series to include per graph.

11. Click **OK**.

12. Start an Oracle Load Testing for Web Applications Autopilot session.

13. If you set the **Save data for reporting** option to Ask, Oracle Load Testing for Web Applications opens a dialog box for selecting the Oracle Load Testing for Web Applications session name and indicating whether you want to save the data.

7.2 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 and display it in the Default Graphs tab. A toolbar is also displayed for saving, printing, and emailing the selected graph when you mouse over a graph.

- **Default Graphs** - displays the default graphs in a larger format than the Overview tab. you can switch between the default graphs using the **View** pull down menu. You can also select the time range to display.

Custom graph tabs have the following options:

- **Remove Graph** - 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 for Web Applications 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.

Session Counters - the counters that are available for the overall session.

<profiles> - the counters that are available for each Oracle Load Testing for Web Applications 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 Visual 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/Visual Script name and the page number of the Visual 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 for Web Applications 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 for Web Applications 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.

Open - displays the Open Graph Query dialog box for selecting a query to use.

Edit - displays the Edit Graph Options dialog box for naming the graph, removing data series from the graph, and specifying X and Y-axis parameters.

Save - displays the Save Graph Query dialog box for saving the current query.

Clear - removes all data series from the graph.

X-Scale - the X-Scale shows the following information:

- **Absolute Time** - when selected, the generated graph is based upon the actual start and end date and times of the profile/timer.

- **Relative Time** - when selected, the generated graph is based upon relative time. Profiles/timers are graphed from a shared zero point.

- **Users** - when selected, the generated graph is based upon users.

7.3 Using the Performance Statistics

You can view the Performance Statistics report by selecting Performance Statistics from the dropdown list on the **Reports** tab on the **Create Reports** tab for sessions for which data has been saved, or by clicking on the **Performance Statistics** icon in the Overview tab on the **View Run Graphs** tab when you start an Oracle Load Testing for Web Applications session.

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.

- **Virtual Users with Errors** - the number of virtual users with errors.

- **Transactions Per Second** - the number of times the virtual user played back the Visual Script per second.

- **Pages Per Second** - the number of pages returned by the server per second. A "page" consists of all of the resources (i.e. page HTML, all images, and all frames) that make up a Web page.

- **Hits Per Second** - the number of resource requests to the server per second. Each request for a page, individual images, and individual frames is counted as a "hit" by Oracle Load Testing for Web Applications. If Oracle Load Testing for Web Applications does not request images from the server (as specified in the Download Manager), images are not included in the hit count. The **Hits Per Second** and **Pages Per Second** counts will be the same if images are not requested and there are no frames in the page.

- **Kilobytes Per Second** - the number of kilobytes transferred between the server and browser client per second.

<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 pages returned by the server.

- **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

- **<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 Visual Scripts using Oracle Functional Testing for Web Applications.

- **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.

7.4 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 for Web Applications 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

7.4.1 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. When you place your mouse over a graph in the Overview tab the following toolbar is displayed:

   Save this image - displays the Save dialog box for saving the image.

   Print this image - displays the Print dialog box for printing the image.

   Send this image in email - displays the Send Pictures via Email dialog box for emailing the image.

   Open my pictures folder - opens the My Pictures folder.

3. Click the graph you want to view to enlarge it and display it in the **Default Graphs** tab. You can switch between Load Graphs using the selector list at the top of the screen.

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 **Add Data Series**.

4. Click **Edit** to display the Edit Graph Options dialog box for changing the graph name and specifying how to plot the graph.

   Graph Node

   - **Name** - enter the name of the graph. This name is displayed in the tab in the Reports & graphs section.

   - **Delete** - deletes the selected data series from the graph.

   - **Axis** - displays the axis on which the data series will be graphed.

   - **Data Series** - displays the data series in the graph.

   - **Name** - displays the name of the data series that is displayed when you place the mouse over the data series in the graph.

   - **Label** - displays the label displayed in the legend.

   - **Secondary Y-Axis** - when checked, displays the selected data series on the secondary y-axis.

   X-Axis Scale

   - **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.

   - **Users** - when selected, the generated report tables and graphs are based upon users.

X-Axis Range

- **Automatic** - automatically creates the scale for the X axis.

- **Minimum** - enter the minimum X-axis value. This field is only displayed when **Automatic** is not selected.

- **Maximum** - enter the maximum X-axis value. This field is only displayed when **Automatic** is not selected.

Session Details

- **Start Times** - 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 and Secondary Y-Axis Scale

- **Linear** - displays the Y-axis linearly.

- **Logarithmic** -displays the Y-axis logarithmically.

Y-Axis and Secondary Y-Axis Range

- **Automatic** - automatically creates the scale for the Y axis.

- **Minimum** - enter the minimum X-axis value. This field is only displayed when **Automatic** is not selected.

- **Maximum** - enter the maximum X-axis value. This field is only displayed when **Automatic** is not selected.

Y-Axis and Secondary Y-Axis Threshold Line - displays a threshold line on the graph at the configured location.

- **Value** - enter the value of the threshold line, that is, the point on the Y-axis at which you want the line to be displayed.

- **Label** - enter the label to display in the legend.

5. Enter a name for the graph.

6. Select how to plot the graph in the **X-Axis** and **Y-Axis** nodes.

7. Click **OK**. 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.

7.4.2 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.

7.4.3 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 for Web Applications session.

7.4.4 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.

7.4.5 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 for Web Applications session.

7.4.6 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.

7.4.7 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.

7.5 Using Data Counters

Oracle Load Testing for Web Applications automatically generates a set of counters for Oracle Load Testing for Web Applications virtual user data that can be selected in the **Available Data Series** field in the **Filters** section. The Oracle Load Testing for Web Applications 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, similar to the following:

Oracle Load Testing for Web Applications 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.

7.5.1 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 Visual 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.

7.6 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 Report. 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, PDF, or JPG format.

The Reports tab has the following options:

- **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.

- **Delete** - deletes the selected template.

- **Generate** - generates the report based on the session and report template selected.

- **Export to CSV** - opens the data in a separate browser window in comma separated value format.

- **Print Friendly** - opens the session report in a separate browser window. From there you can either print the report or save it in HTML format.

- **Include Think Times in Profile timers** - specifies whether to include think time in session reports. This setting only applies for tests run using Thin or Java Agent.

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 for Web Applications 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.

- **Session Counters** - the counters that are available for the overall session.

- **<profiles>** - the counters that are available for each Oracle Load Testing for Web Applications 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 Visual 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/Visual Script name and the page number of the Visual 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 for Web Applications 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.

- **Open** - displays the Open Graph Query dialog box for selecting a query to use.

- **Edit** - displays the Edit Graph Options dialog box for naming the graph, removing data series from the graph, and specifying X and Y-axis parameters.

- **Save** - displays the Save Graph Query dialog box for saving the current query.

- **Clear** - removes all data series from the graph.

- **X-Scale** - provides the following information:

  - **Absolute Time** - when selected, the generated graph is based upon the actual start and end date and times of the profile/timer.

  - **Relative Time** - when selected, the generated graph is based upon relative time. Profile/timers are graphed from a shared zero point.

  - **Users** - when selected, the generated graph is based upon users.

- **Export to CSV** - opens the data in a separate browser window in comma separated value format.

- **Export to Excel** - opens the data in a separate browser window in Excel format.

- **Export to PDF** - opens the graph in a separate browser window in portable document format.

- **Export to JPG** - opens the graph in a separate browser window in JPG format.

7.6.1 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 **Edit** to display the Edit 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 **Export to CSV** to save the graph as a CSV file. Click **Export to Excel** to save the graph as a CSV file. Click **Export to PDF** to save the graph as a PDF file. Click **Export to JPG** to save the graph as a JPG file.

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

7.6.2 Exporting CSV Data

When you export generated data from a report, Oracle Load Testing for Web Applications 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 **Export to CSV**. The File Download dialog box is displayed.

3. Click **Open** to open the file in Excel in CSV format. Click **Save** to save the file in CSV format.

4. 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**.

7.6.3 Exporting to Excel

To export as an Excel Workbook file:

1. Create a report.

2. Click **Export to Excel**. The File Download dialog box is displayed.

3. Click **Open** to open the chart in Excel. Click **Save** to save the chart in Excel format.

4. 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**.

7.6.4 Exporting to PDF

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

1. Generate a report.

2. Click **Export to PDF**.

3. The report is displayed in Adobe Acrobat Reader in a separate browser window.

4. Click the **Save** button on the Acrobat toolbar to open the Save As dialog box.

5. Enter a name for the file and click **Save**.

7.6.5 Exporting to JPG

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

1. Generate a report.

2. Click **Export to JPG**.

3. The report is displayed in a separate browser window.

4. Move the mouse to the upper left corner of the screen to display the toolbar.

5. Click the **Save** button to open the Save As dialog box.

6. Enter a name for the file and click **Save**.

7.6.6 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.

7.6.6.1 Saving Graph Queries

To save a graph query:

1. Select the sessions and data series that you want to graph.

2. Click **Save** in the graph toolbar.

   <query list> - lists the existing saved queries.

   Name - enter a name for the new query or select an existing query to overwrite it.

3. To create a new query, enter a name for the query and click **OK**. To overwrite an existing query, select the query and click **OK**, then click **Yes** when asked to confirm.

7.6.6.2 Running a Saved Query

To run a saved query:

1. Click **Open** in the graph toolbar.

   <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.

2. Select the query you want to use.

3. Select the session against which you want to run the query.

4. Select whether or not to add this session data to the current graph.

5. Click **OK**.

6. 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.

7. The new graph is generated.

7.6.6.3 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.

7.6.6.4 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.

7.6.7 Creating Report Templates

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

Report templates are created using the Oracle Load Testing for Web Applications 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 uploaded to Oracle Load Testing for Web Applications. 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 OracleATS/OFT 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 for Web Applications 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 for Web Applications 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 for Web Applications 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 upload the file to Oracle Load Testing for Web Applications and start generating reports. To upload the file:

1. In Oracle Load Testing for Web Applications, select **Upload File** 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 **Upload**.

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.

7.6.7.1 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 OracleATS/OFT 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()%>
```

7.7 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. The Session Performance report is displayed followed by the Oracle Load Testing for Web Applications 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.

The following sections explain more about the summary reports.

7.7.1 Scenario Reports

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

To save the report, click **Print Friendly**. 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**. 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**.

7.7.2 Session Reports

To save the report, click **Print Friendly**. 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.

7.8 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 VU Display Data - deletes only the VU display 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 for Web Applications Autopilot. Oracle Load Testing for Web Applications 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 for Web Applications 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 Data - indicates whether VU data was saved for this session.

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 VU Display Data** to delete the VU display log only.

8

Using the Virtual User Display

The virtual user display 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 display features:

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

8.1 Starting the VU Display

To start the Virtual User display:

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

8.2 Overview of the Main Window

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

- Toolbar
- Browser pane
- History list

8.2.1 Overview of the Toolbar Options

The virtual user display window has the following toolbar options:

Open VU Display Log - Opens an existing virtual user error log.

Download log file from server - Downloads an existing VU display log from the server to the local drive.

Clear Display - Clears the Virtual User Display.

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

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

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

Manual Mode - The VU Display 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 history list.

Navigate to Previous Page - Navigates to the previous page in the history list.

Navigate to Next Page - Navigates to the next page in the history list.

Navigate to Last Page - Displays the last page of the history list.

8.2.2 Overview of the Browser Pane

The browser pane lets you view the pages being accessed by the virtual user. You can toggle between viewing the page or the HTML source for the page.

The Download Manager settings and the **Use Download Manager** option in the Edit Scenario Details dialog box determines if images are included separately in the history list.

8.2.3 Overview of the History List

The history list of the Virtual User Display window shows the pages accessed by the virtual users.

When viewing pages in manual mode, you can select any page in the history list to view the page in the browser.

The runtime history list has the following columns:

Id - shows a sequential identification number for the history entry. You can use the identification number to locate specific entries in the list.

Current Page - shows the profile name, and the page of the Visual Script being accessed by the virtual user.

URL - shows the Uniform Resource Locator for the page being accessed by the virtual user.

Content Type - shows the content type specification of the page being accessed by the virtual user.

VU ID - shows which virtual user accessed the page.

Iterations - shows the number of the iteration run.

Data Bank - shows data accessed from the page.

Error- shows any error encountered on the page by the virtual user.

Time- shows the date and time of day the page was accessed by the virtual user.

Size- shows the file size of the page being accessed by the virtual user.

8.3 Setting Virtual User Display Limits

You can use the **Maximum total storage size** setting in the Reporting Options dialog box to limit the total number of bytes that the VU display keeps in memory and on disk at any point during the load test.

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

2. Select **Reporting**.

3. Enter a new value in the **Maximum total storage size** field.

4. Click **Apply**.

5. Click **OK** to exit the Options dialog box.

8.4 Opening VU Display Logs

You can open and existing VU display log for viewing in the VU display. To open a VU display log:

1. Click **Open VU Display Log**.

    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.

2. Select whether you want to open a log file on the server or on the local machine.

3. Select the log that you want to execute in the virtual user display. If you are opening a log file from the local machine either enter it's path or click **Browse** to navigate to the file.

4. Click **OK**.

8.5 Downloading Log Files From the Server

You can download VU display log files from the Oracle Load Testing for Web Applications server to your local hard drive. To download log files:

1. Click **Save to Local**. 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.

9

Database and File Management

This chapter describes the how to upload and download files to and from the server, how to manage the Oracle Load Testing for Web Applications database, and how to select the database to use.

9.1 Database Management

The Oracle Load Testing for Web Applications database is used to store the results of your load test sessions. You use the **Oracle Test Manager for Web Applications Database Configuration** to configure databases for use in Oracle Load Testing for Web Applications. The **Databases** option on the **Manage** menu in Oracle Load Testing for web Applications is used to set the current database to use. The Database Manager dialog box

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.

9.1.1 Adding an Oracle Database

To add an Oracle database:

1. Select **Programs** from the **Start** menu then select **Oracle Test Manager for Web Applications Database Configuration** from the **Oracle Application Testing Suite** menu.

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 Databas selection lists for Oracle Test Manager for Web Applications 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 SQL Server 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.

9.1.2 Selecting the Database to Use

1. Select **Programs** from the **Start** menu then select **Oracle Test Manager for Web Applications Database Configuration** from the **Oracle Application Testing Suite** menu.

2. Select Oracle Load Testing as the product type.

3. Select the database you want to use to store Oracle Load Testing for Web Applications data.

4. Click **Set as current**.

5. Click **Close**.

6. Restart Oracle Load Tesitng for web Applications.

9.1.3 Removing Databases

You can remove databases from Oracle Test Manager for Web Applications. 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 Test Manager for Web Applications. The database cannot be deleted if it is open in Oracle Test Manager for Web Applications.

To remove a database from Oracle Test Manager for Web Applications:

1. Select **Programs** from the **Start** menu then select **Oracle Test Manager for Web Applications Database Configuration** from the **Oracle Application Testing Suite** menu.

2. Select the database that you want to remove.

3. Click **Remove**.

4. Click **Yes** to confirm the deletion.

9.1.4 Updating Databases

You can change the name, location, and login information for databases that you are using in Oracle Test Manager for Web Applications.

1. Select **Programs** from the **Start** menu then select **Oracle Test Manager for Web Applications Database Configuration** from the **Oracle Application Testing Suite** menu.

2. Select the database that you want to update.

3. Click **Update**.

4. Make any changes.

5. Click **OK**.

9.2 Uploading Files

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

1. Select **Upload File** from the **Tools** menu.

   File Type - select the type of file.

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

   File to Upload - the options listed here depend on the type of file being uploaded.

   ■ **<extension>** - enter the file you want to upload.

   ■ **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 uploading and only apply to certain file types.

   ■ **Repository** - select the repository where you want to upload the file.

   ■ **Workspace** - select the workspace where you want to upload the file.

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

   > **Note:** If you are uploading a ServerStats Metric Profile, the metrics associated with the profile must be uploaded 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**.

9.3 Downloading Files

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

1. Select **Download File** from the **Tools** menu.

   File Type - select the type of file you want to download from the Oracle Load Testing for Web Applications server to the local machine.

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

   Workspace - lists the workspaces available in the selected repository.

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

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

3. Select the Repository and Workspace in which the file is located if applicable to the type of file you are downloading.

4. Select the file you want to download. If you are downloading scripts, the Download Multiple Files dialog box is displayed.

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

5. Select the script part that you want to download. Script parts must be downloaded 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.

10

Working With Sessions

This chapter describes the how to attach to and detach from sessions that are running on the Oracle Load Testing for Web Applications server. This enables a tester to share real-time test results and to collaborate with team members during testing.

10.1 Attaching to a Session

You can attach to another running session on the Oracle Load Testing for Web Applications 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 for Web Applications 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**.

10.2 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.

A

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 for Web Applications Autopilot and the Oracle Functional Testing for Web Applications 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 codeds.

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.

12045Invalid 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

B

Error Messages

This appendix lists the error messages that can be generated by Oracle Load Testing for Web Applications 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 for Web Applications Autopilot Virtual User grid.

Current Error and **Previous Error** show errors encountered by running CUs 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 for Web Applications was unable to create an Agent process to run Virtual Users. Try the following:

1. Make sure that you have the Oracle Load Testing for Web Applications 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 for Web Applications system. The systems you plan to use as agents must have either the Oracle Load Testing for Web Applications Agent software or the full Oracle Application Testing Suite installed to work as agents. Do not install both the Oracle Load Testing for Web Applications 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 for Web Applications 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 for Web Applications 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 for Web Applications system, you will not be able to run the agents from the Oracle Load Testing for Web Applications system.

3. Make sure that the same user is logged on both the Oracle Load Testing for Web Applications system and all of the Agent systems. All of the Agent systems must be logged in to be controlled by the Oracle Load Testing for Web Applications

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 for Web Applications system.

4. If the Oracle Load Testing for Web Applications and Agent systems are not participating in the Domain security model, both the Oracle Load Testing for Web Applications and the Agent systems must log in as Administrator and have the exact same password.

5. From the Oracle Load Testing for Web Applications 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 for Web Applications 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 for Web Applications system, add a Visual Script to the Configure Parameters of the Scenario list. Add the Agent system on which you want to run the Visual 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 for Web Applications is unable to find the navigation between two pages recorded to the Visual Script. These errors can occur in Oracle Load Testing for Web Applications for a variety of reasons, even if the Visual Script played back correctly in Oracle Functional Testing for Web Applications. Oracle Functional Testing for Web Applications uses Thick Client mode exclusively when recording and playing back Visual Scripts.

Typically, "failed to find" errors are related to dynamically generated HREFs recorded to a Visual Script that is played back in Oracle Load Testing for Web Applications Thin Client. For example HREFs containing:

■ session information

■ inline JavaScript

■ other dynamically changed attributes

may cause "failed to find" errors in Oracle Load Testing for Web Applications Thin Client depending upon the specific techniques used for links in the Web page source and Oracle Load Testing for Web Applications current settings.

If you experience "failed to find" errors in Oracle Load Testing for Web Applications Thin Client, try running the Scenario/Visual Script with one Virtual User in Thick Client. If the Scenario/Visual Script runs in Oracle Load Testing for Web Applications Thick Client, it will be possible to run small load tests with that Scenario/Visual Script. However, Thick Client is a full implementation of the Browser, which has much higher system resource requirements than Thin Client. Thick Client is not as scaleable as Thin Client and will not run as many Virtual Users per workstation as Thin Client.

To resolve "failed to find" errors in Thin Client, contact Customer Support for assistance with adjusting Oracle Load Testing for Web Applications 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 for Web Applications found the navigation path (an internal representation of a clicked link that Oracle Functional Testing for Web Applications records to the Visual Script) between two pages recorded to the Visual 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 Visual Script that is played back in Oracle Load Testing for Web Applications Thin Client.

To resolve "failed to match" errors in Thin Client, contact Customer Support for assistance with adjusting Oracle Load Testing for Web Applications 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 for Web Applications failed to open the specified scenario file.

Failed to OpenKey 'registry key'

Oracle Load Testing for Web Applications failed to open the specified registry key.

Failed to Query 'registry key'

Oracle Load Testing for Web Applications failed to get the specified registry value string.

Failed to start callout server 'server prog id'

Oracle Load Testing for Web Applications is unable to start the External Callout server specified in the Visual 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 for Web Applications 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

Code	Error Constant
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
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_THEAD_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 for Web Applications, and Oracle Application Testing Suite Application Service, Agents.

Resolution: Check the following:

1. Make sure you can ping the machines where Oracle Load Testing for Web Applications, 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 for Web Applications 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 for Web Applications.

When it Occurs: When attempting to start Oracle Load Testing for Web Applications.

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 for Web Applications 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 for Web Applications Agent and Oracle Application Testing Suite services are running.

4. If necessary, restart the Oracle Load Testing for Web Applications 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 for Web Applications 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

Refer to the "Settings Manager" manual for instructions on changing the default ports.

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 for Web Applications 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. Refer to "Configuring Oracle Load Testing for Web Applications Agents" in Chapter 3 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 for Web Applications 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 for Web Applications versions, install the same version on all machines.

B.6.3 Performance Problems During the Load Test

Symptoms: The user interface froze during the load test and no longer responds.

Cause(s): This can occur when:

- CPU usage is at 100%
- Memory is out of range

Resolution: Check the following:

1. Check for error messages in the server log, in the following file located in the installation directory:

   <installdir>\logs\wls_oats.log

   If there are errors refer to the instructions under "Unable to Start Session."

2. Go to *<installdir>*\oats\servers\AdminServer\logs directory and look at AdminServer.log file to see if there are any exceptions indicating the nature of the problem, for example, multiple sockets error with the subsequent forced session termination.

3. Go to System Log on the Event Viewer and check for Network Warnings.

To delete old messages when CPU usage is 100%:

1. Open Services in the Administrative Tools of the Windows Control Panel and stop the Oracle Application Testing Suite Application Service.

2. Delete everything in the

   <installdir>\oats\servers\AdminServer\data\store\default directory.

To increase memory if there is an out of memory error, follow these steps:

1. Stop Oracle Application Testing Suite Application Service.

2. Open Regedit and locate key HKEY_LOCAL_ MACHINE\SYSTEM\CurrentControlSet\Services\Oracle Application Testing Suite Application Service\Parameters.

3. In the string value `CmdLine` locate command line 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 for Web Applications 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 for Web Applications 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 for Web Applications and logged in the <installdir>\logs\agentmanager.log file.

One of the following values is returned:

- The exit value specified in the ExitProcess or TerminateProcess function.

- The return value from the main or WinMain function of the process.

- The exception value for an unhandled exception that caused the process to terminate.

The following are known causes for commonly seen exit codes:

Unexpected Agent Exit. Code = 143
This code is returned when terminating a JavaAgent process by closing its console window manually, or by terminating the application using the Task Manager's Applications tab.

Unexpected Agent Exit. Code = 1
This code is returned when WgPro.exe or JavaAgent.exe is terminated directly using the Windows Task Manager Processes tab.

Unexpected Agent Exit. Code = 0
This code is returned when the Thin Client 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 JBoss 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 for Web Applications 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 for Web Applications 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.

Index