

Web Performance Suite 3.0 Manual

Table Of Contents

Quick Start Guide.....	1
Quick Start	1
Installation	1
Installation methods.....	1
Getting Help	4
Updating the software	5
Navigating the User Interface.....	7
Create a Recording	10
Inspecting a Testcase	11
Replay a Testcase.....	12
Analyze the Performance Changes.....	12
Run a Load Test.....	15
Tutorials.....	17
Tutorials Index.....	17
Introduction to Load Testing.....	18
Introduction To Load Testing	18
Virtual Users	18
Recommended Reading	20
LAN vs. WAN.....	21
Phase One - Baseline Analysis	23
Phase One Testing Procedure.....	23
Record A Testcase	25
Analyze A Recording	30
Phase Two - Test Configuration	34
Phase Two Testing Procedure.....	34
Configure and Replay a Testcase.....	35
Phase Three - Large Scale Tests.....	50
Phase Three Testing Procedure	50
Load Testing	52
FAQs.....	59
General FAQs	59
Recording FAQs.....	59
Playback FAQs.....	60
Reference Manual.....	63
Views.....	63
Toolbar buttons.....	63
Navigator	64
Testcase Editor.....	67
Dataset Editor	76
Load Test Configuration Editor	79
Load Test Results View	82
Status View.....	83
Headers View	86
Content View	90

Errors View	93
Replay View	96
Validators View	98
Fields View	100
Event Log.....	102
Servers View.....	102
Server Configuration.....	104
Engines View	105
Settings	107
General Settings.....	107
Browser Settings.....	109
Proxy Settings.....	111
Global Performance Goals.....	113
Testcase Performance Goals	115
License Management	117
Usage Logging.....	118
Configuring Memory Usage	119
Client Certificates.....	120
Load Test Settings.....	122
Charts & Reports.....	123
Performance Trend Chart	123
Baseline Performance Report.....	127
Load Test Report	128
Manual Browser Configuration	129
Step 1 - Configure recording ports.....	129
Step 2 - Configure the browser	130
Firefox.....	133
Step 3 - Select proxy server.....	134
Step 4 - Test the configuration.....	134
Workspace	135
Load Engines	136
Installing a Load Engine	136
Starting a Load Engine	136
Index.....	139

Quick Start Guide

Quick Start

This guide will help you learn basic navigation of the Web Performance Suite interface as well as the steps to record, replay and analyze the performance of a your website.

1. [Installation](#)
2. [Getting Help](#)
3. [Updating the Software](#)
4. [Navigating the User Interface](#)
5. [Record a testcase](#)
6. [Inspecting a testcase](#)
7. [Replay a testcase](#)
8. [Analyze the performance changes](#)

Installation

Supported Platforms

Web Performance Suite is supported and tested on Windows 2000 and XP. It should work on most other modern Windows versions, as well.

Installation methods

There are two methods for installing Web Performance Suite:

- Standalone - to install Suite to run without dependencies on Eclipse
- Eclipse plug-in - to install as a plug-in to the Eclipse environment

Standalone installation

1. Download the software from <http://webperformanceinc.com/download>
2. Run the installer and follow the directions provided.

Eclipse plug-in installation

Java 5 is required for the Web Performance products.

Eclipse 3.0

NOT SUPPORTED

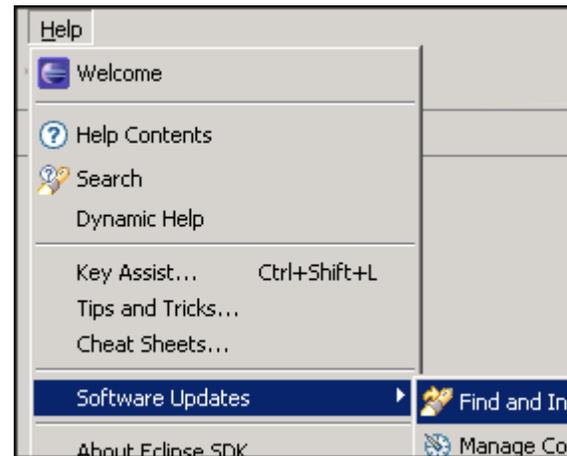
Eclipse 3.1

If you are already experienced with plug-in installation procedures, you probably just need to know this:

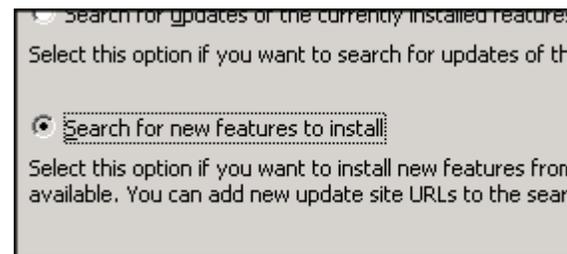
1. update site URL: <http://update.webperformance.com/download/release/3x>
2. feature to install: "Web Performance Suite Plug-in"

These instructions show each step for a plug-in installation:

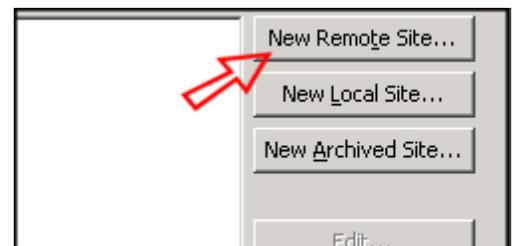
1. From the Eclipse *Help* menu, select *Software Updates* and *Find and install...*



2. Select *Search for new features to install* and click *Next*



3. Select *New Remote Site...*



Quick Start Guide

4. Enter "Web Performance Suite" for the *Name* and <http://update.webperformance.com/download/release/3x> for the *URL*

Select *OK*

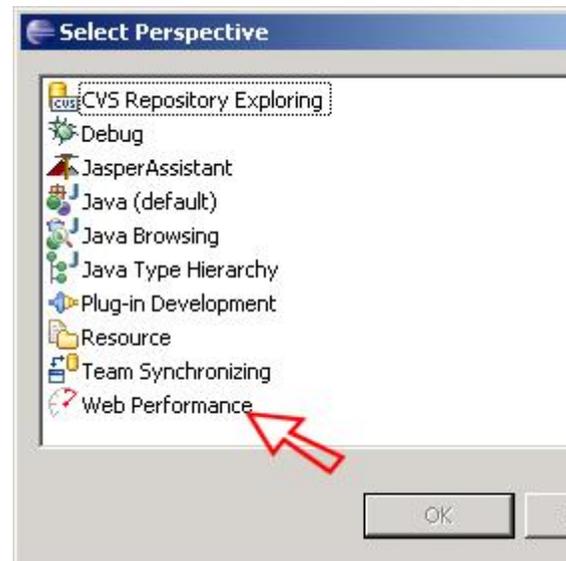
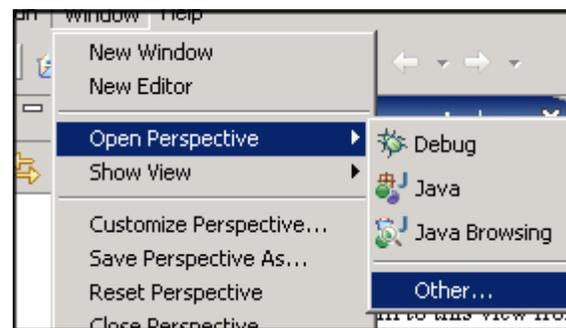
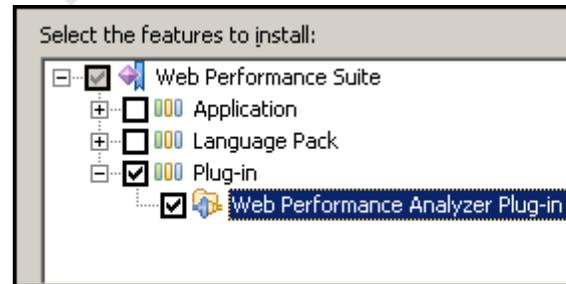
Select *Finish*

5. Select the "Web Performance Suite Plug-in" feature.

DO NOT SELECT THE "APPLICATION" FEATURE! (it is only for updates of the standalone version and could corrupt Eclipse installations)

Press the *Next* button and follow the wizard instructions to complete the installation.

6. After installing the plug-in and restarting, switch to the Web Performance perspective (*Window->Open Perspective->Other->Web Performance*).



Getting Help

Integrated help system

This page is part of the Quick Start Guide within the integrated help system. There are four books in the help system:

- Quick Start Guide - use this for a demonstration of the most commonly used features
- Tutorials - task-specific guides for accomplishing a specific goal
- FAQs - contains questions frequently asked by users and other topics not addressed by the other books
- Reference Manual - contains detailed information about various program elements and technical topics

The integrated help system is accessed from the *Help Contents* item in the *Help* menu.

Help shortcuts

Help system shortcuts are accessible from most areas of the program: Click on or near the component of interest and press the *F1* key.

1. A help window will open that describes the basic purpose of the component and links to the user manual for more detailed information on related topics.
2. After clicking one of the manual links, the pages can be difficult to read in a small window. Pressing the *Maximize* button will enlarge the help window. It can be restored to the original size using the *Restore* button. The manual page can be opened in the external help window using the *Show in External Window* button (see below).
3. Alternatively, the links may be opened in the external help window (which includes the

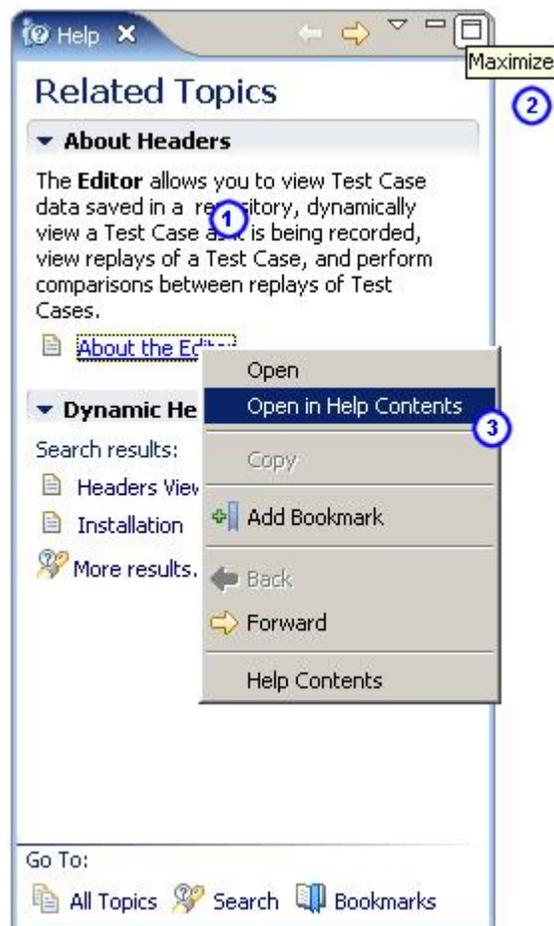


table of contents and search functions) by using the *Open in Help Contents* item from the pop-up menu.



Bookmarks

Help topics can be book-marked for quick reference in the future using the *Bookmark* button at the top of the help window. Links or tabs at the bottom of the help window displays a list of saved bookmarks.

Getting more help

If you cannot find answers in the online help resources, more help is available through our [online forum](#) and our [issue tracking system](#).

Reporting issues

You may also report bugs and request enhancements using the integrated support request form. The Support Request Form is available from the *Support Request* item on the *Help* menu.

Filling in the form

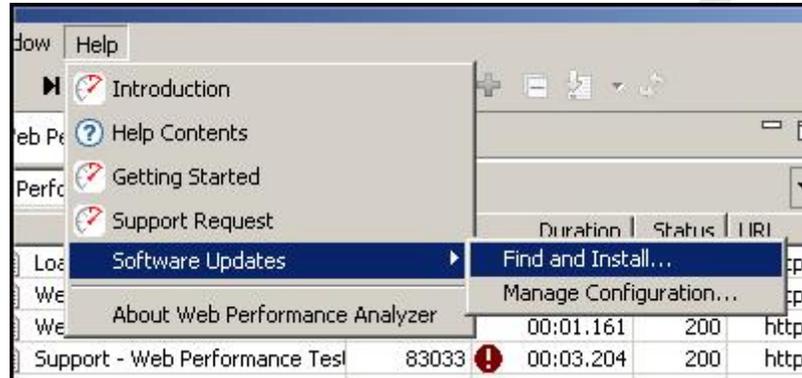
Enter your e-mail address, and choose either the *Open new request* or *Attach files to existing request* options. For new requests, please fill in a short summary, description and select a category. Additional information related to the request can be sent by selecting the appropriate items (e.g. testcases or loadtest results).

Updating the software

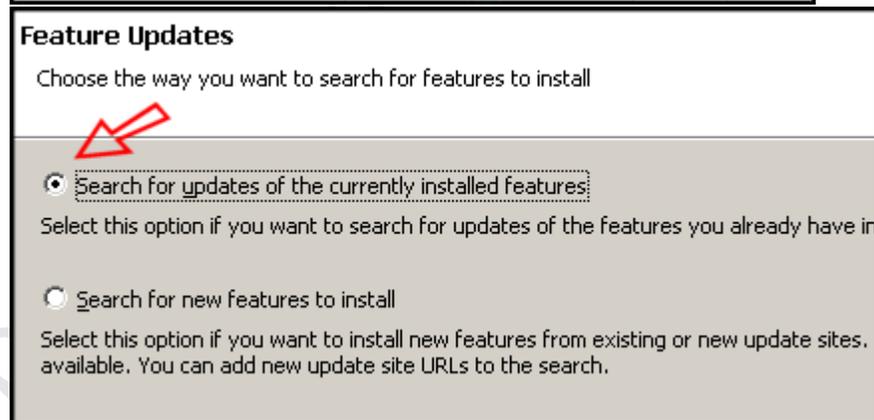
Updating to the latest patch for your version

Updates to the Web Performance software can be easily obtained via the integrated update system.

1. Open the *Install/Update* wizard from the menu: *Help*->*Software Updates*->*Find and Install*.



2. Select the "Search for updates..." option and the *Finish* button.



3. Follow the wizard to complete the installation.

You may be prompted to restart the application after updating. This is not always necessary, but is the recommended action.

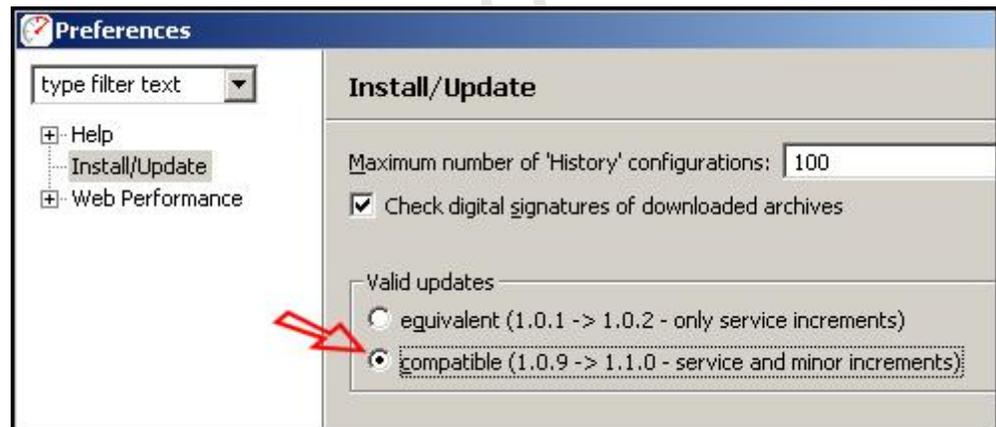
Configuring Updates to upgrade to new minor versions

By default, the software will only download patches. This prevents accidental upgrades to a version of the software for which the installed license are not valid. Once this happens, a re-install is required to get back to the previous version.

It is recommended that this option only be turned on when a minor upgrade is desired and the turned back off.

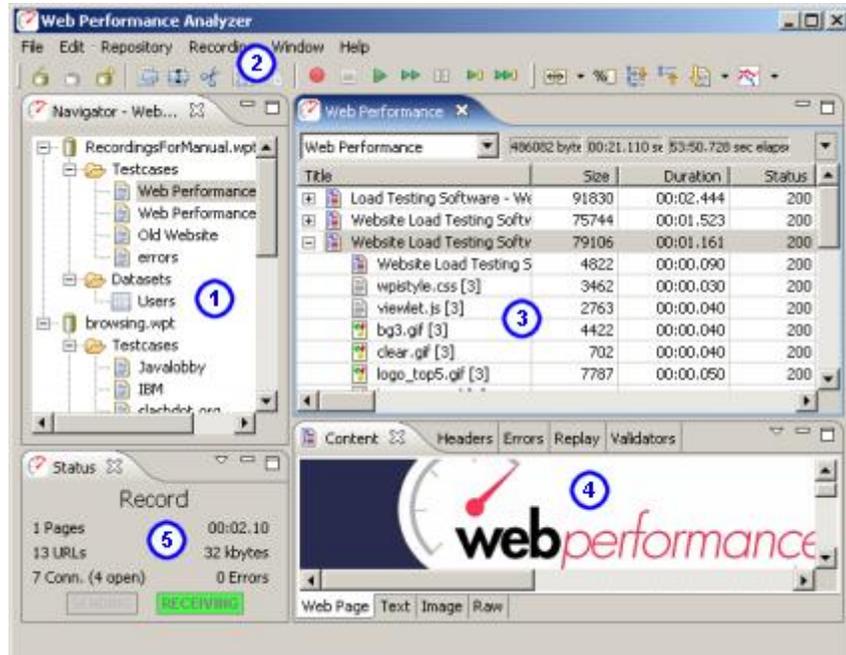
To enable the update manager to also find/install minor upgrades (e.g. 3.0 -> 3.1), follow these steps:

1. Open the preference manager from the menu: *Window->Preferences*
2. Select the *Install/Update* category
3. In the *Valid Updates* section, select *compatible*.



Navigating the User Interface

The main window consists of 3 primary components. Menu/toolbar, editors and views. The default arrangement of the interface places the Navigator view on the left, editors in the middle and other views at the bottom, as show in the screenshot.



1. Navigator view
2. Toolbars
3. Editors and Charts
4. Detailed inspection views
5. Status view

Navigator

The [Navigator](#) shows the contents of each open repository. Double-click repository items (or select a testcase and choose *Edit* from the pop-up menu) to open it in an editor.

Note: Every open repository loads the contents of each test cases into memory. Closing unused repositories will reduce the memory consumed by the test cases.

Toolbars



The [toolbars](#) provide shortcuts to the most common operations in the Navigator and Editor.

Editors and Charts

The [Testcase Editor](#) displays the recorded testcase as a tree of pages and URLs. The testcase can be [sorted](#) by the size and duration columns to quickly find the slowest pages and URLs in the testcase. An icon at the beginning of each row differentiates web pages from text files, images, forwards and errors. Secure transactions (SSL) are indicated by a lock icon at the beginning of the URL column. The editor can also [display](#) the performance differences between two [replays](#) of a testcase.

The [Dataset Editor](#) displays and edits the data and settings for a Dataset.

The [Performance Trend](#) chart shows the changes in page size and duration over multiple replays of a testcase. It is opened from either the Navigator pop-up menu or the drop-down menu in the editor.

Detailed inspection views

- [Content view](#) renders the content of web pages, images and other resources when they are selected in the editor.
- [Headers view](#) displays the HTTP request and response headers for the page/URL selected in the editor.
- [Errors view](#) lists any errors encountered during the recording or replay of a testcase
- [Replay view](#) indicates the status of the replay in progress.
- [Fields view](#) displays the HTTP fields that can be associated with modifiers.
- [Validators view](#) displays size and content validators applied to a Web Page or Transaction in a Testcase.
- [Servers view](#) displays CPU and memory usage statistics for the specified machines.

Status View

The [Status View](#) displays the status of long-running operations, such as recording, replaying testcases and opening repositories.

General GUI features

Editors vs. Views

Editors (including charts) are always located in the center of the Analyzer window. Views can be arranged around the edges of the Analyzer window. Once the size of a view has been set, it will try to maintain that size as the Analyzer window is resized. The editors will receive whatever space is left over.

Resizing

Any of the panes in the Analyzer window can be resized and the new sizes will be remembered when you restart Analyzer.

Rearranging

Any view can be reordered within the tab group or moved to other place in the window by dragging the tab for that view. The entire tab group can be moved using the *Move->Tab Group* menu item from the tab's pop-up menu.

Detaching floating windows

Any view can be detached to a separate floating window using the *Detached* menu item from the tab's pop-up menu. Editors cannot be detached.

Minimizing and Maximizing

Each editor and view can be minimized or maximized within the Analyzer window using the *minimize/maximize* buttons at the top right corner of each view. The editors and views can also be minimized and maximized by double-clicking the tab for the editor/view.

Viewing multiple editors

By default all editors appear in the same area of the Analyzer window. To view multiple editors side-by-side, drag the tab for the editor towards the desired edge. Restore the arrangement by dragging one editor tab onto another editor.

Create a Recording

To create your first website recording, follow these steps:

1. Press the *Record*  button.
2. Follow the wizard steps to auto-detect the network settings and select a browser to use for recording.
3. Enter a name for the testcase, select a repository and a network speed (optional).
4. When the browser is launched, visit your website. As you browse the website, Analyzer will record the pages and URLs retrieved from the server.
5. Press the *Stop*  button to end the recording and close the browser.

You may now review and inspect the contents of the recording in the [Editor](#). The other views ([Headers](#), [Content](#), [Errors](#), etc.) will updated to show additional information about the pages and URLs you select in the recording.

Notes:

The configuration wizard can be reopened at any time using the *Recording Configuration Wizard* menu item on the *Recording* menu.

Inspecting a Testcase

A recorded testcase is represented in the [Testcase Editor](#) by a tree of web pages and URLs. Expanding the web pages reveals the URLs that make up the web page.

The screenshot displays the 'Web Performance with SSL' interface. The top section shows a table of recorded items with columns for Title, Size, Duration, Status, and URL. The bottom section shows the 'Content' view of a selected item, displaying a web page with the 'webperformance testing tools' logo and a 'how many U' button.

Title	Size	Duration	Status	URL
Web Performance Trainer Price Lis	70593	00:01.792	200	http://webperformanceinc.com/sale
Support - Web Performance Testir	83043	00:01.372	200	http://webperformanceinc.com/supp
Support - Web Performance Testir	65015	00:01.172	200	http://webperformanceinc.com/supp
Support - Web Performance Testir	91354	00:05.928	200	http://support.webperformanceinc.
> <forward>	1033	00:00.050	301	http://support.webperformanceinc.
> <forward>	754	00:00.070	302	http://support.webperformanceinc.
Support - Web Performance T	31871	00:01.051	200	http://support.webperformanceinc.
viewlet.js [7]	2764	00:00.031	200	http://webperformanceinc.com/view
wpistyle.css [7]	3463	00:01.062	200	http://webperformanceinc.com/wpis
bg3.gif [7]	4423	00:00.030	200	http://webperformanceinc.com/imaç
clear.gif [7]	703	00:00.040	200	http://webperformanceinc.com/imaç
howmany.gif [7]	4872	00:00.050	200	http://webperformanceinc.com/imaç
logo_top5.gif [7]	7788	00:00.060	200	http://webnerformanceinc.com/imaç

The content view shows a web page with the following elements:

- Logo: webperformance testing tools
- Text: see it! demo
- Text: how many U

The interface also includes tabs for 'Content', 'Headers', and 'Errors', and a bottom navigation bar with 'Web Page', 'Text', 'Image', and 'Raw' options.

When a web page or URL is selected in the Testcase Editor, some views display details about the selected item. Web pages are displayed in an embedded

browser in the [Content View](#). Images and other text resources are displayed in text and image viewers, while other resources are displayed in the raw viewer. The [Headers View](#) displays the HTTP headers while the [Errors View](#) lists any errors encountered during the recording.

Replay a Testcase

Replaying a testcase creates a virtual user which simulates a real person using a browser to interact with a website following the same steps as the original recording.

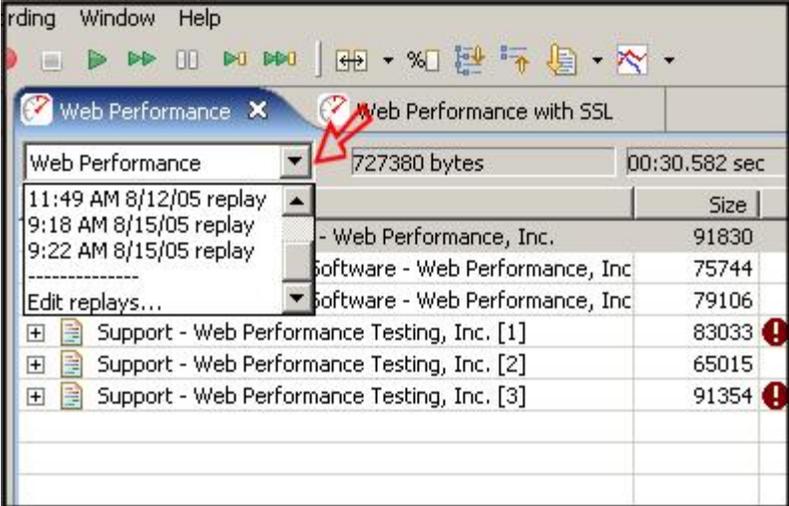
To replay the testcase, open the testcase in the Editor and press the *Play* button. A wizard appears, requesting information required to configure the testcase for replay. For simple testcases that do not require a user to log in to view content, using the same user as recording and allowing the application to automatically configure Session Tracking and Application State Management should allow the replay to run successfully. Once "Finish" is selected on the wizard, the editor clears the pages and URLs replayed appear in the editor (similar to the recording process) as the replay proceeds. The replay is added to the *Replay selection list*, at the top of the editor.

The [Replay View](#) appears, displaying the current status of the replay.

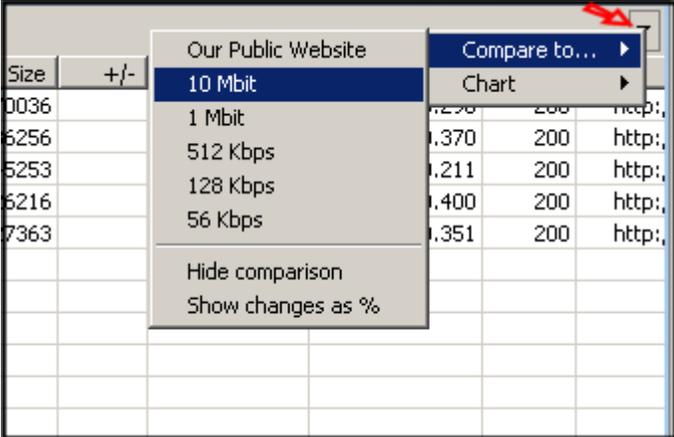
#	host	state	txns	Last URL
1	webperformanceinc.com:80	waiting	26	http://webperformanceinc.com/images/download_anl.gif
0	webperformanceinc.com:80	receiving	33	http://webperformanceinc.com/images4/analyzer_sm.jpg
2	counter2.hitslink.com:80	idle	3	http://counter2.hitslink.com/statistics.asp?v=1&s=207&

Analyze the Performance Changes

As soon a replay has been initiated, it appears in the *replay selection list*, as shown below. The testcase editor reflects the performance measurements of the selected replay for the testcase.

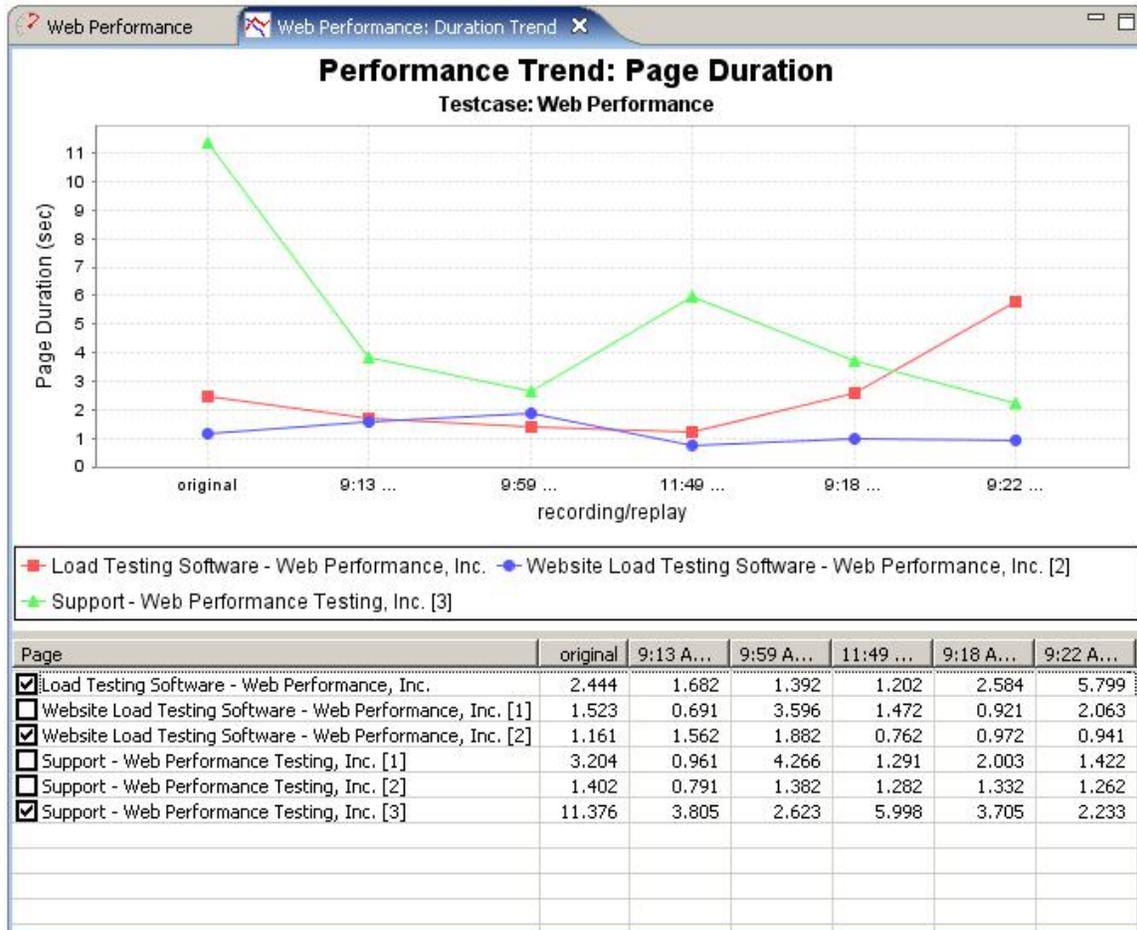


To easily see the [performance differences](#) between two replays (or a replay and the original recording), select the *Compare to...* menu item in the *Testcase Editor* menu. The testcase editor adds new columns that show the changes in the size and duration of each page and URL in the testcase.



Title	Size	+/-	Duration	+/-	Status	URL
Load Testing Software - Web Perform	91803	↓ -27	00:02.453	↑ 00:00.009	200	http://webper
Website Load Testing Software - We	75720	↓ -24	00:01.182	↓ -00:00.341	200	http://webper
Website Load Testing Software - We	79082	↓ -24	00:01.282	↑ 00:00.121	200	http://webper
Support - Web Performance Testing,	83010	↓ -23	00:00.841	↓ -00:02.363	200	http://webper
Support - Web Performance Test	4731	↓ -3	00:00.230	↑ 00:00.140	200	http://webper
wpistyle.css [4]	3467		00:00.041	↑ 00:00.001	200	http://webper
viewlet.js [4]	2768		00:00.051	↓ -00:00.009	200	http://webper
bg3.gif [4]	4427		00:00.050	↑ 00:00.020	200	http://webper
clear.gif [4]	707		00:00.040	↓ -00:00.001	200	http://webper
statistics.asp [4]	1444	↓ -20	00:00.101	↓ -00:00.049	200	http://counte
logo_top5.gif [4]	7792		00:00.060	↓ -00:00.020	200	http://webper
howmany.gif [4]	4876		00:00.070	↑ 00:00.010	200	http://webper
home_off.gif [3]	1056		00:00.040		200	http://webper
company_off.gif [4]	1242		00:00.040	↓ -00:00.010	200	http://webper
products_off.gif [2]	1255		00:00.040	↓ -00:00.040	200	http://webper
sales_off.gif [4]	1102		00:00.030	↓ -00:00.010	200	http://webper
support_on.gif [1]	1196		00:00.030	↓ -00:00.010	200	http://webper
download_off.gif [4]	1650		00:00.040	↑ 00:00.010	200	http://webper
library_off.gif [4]	1127		00:00.030	↓ -00:00.030	200	http://webper

To display the changes in performance of web pages over more than two replays, open the [Performance Trend](#) chart for the testcase. For example, selecting the *Performance Trend (duration)* item from the *Chart* submenu of the *Testcase Editor* menu results in a chart like this one:



Run a Load Test

Write this section before the 3.0 release.

Tutorials

Tutorials Index

Record and Analyze a Testcase

This is the first step in any performance analysis. This tutorial will describe how to configure Analyzer to use your browser to record your website. Then you will learn how to analyze the baseline performance of the testcase to answers questions such as:

1. How big are my web pages?
2. How long do my pages take to load?
3. Does my website meet our performance goals?
4. How does my website perform for users with limited bandwidth (e.g. dial-up or cable modems)?
5. How much bandwidth do my testcases consume?

Configure and Replay a Testcase

There are two reasons to replay a testcase:

1. Determine if the baseline performance has changed since the testcase was recorded
2. Prepare the testcase for use in a load test

This tutorial will describe the automatic and advanced options available for configuring a testcase to handle common needs such as:

1. Using different usernames and passwords each time a testcase is replayed
2. Handling dynamic URL parameters
3. Making each replay enter different information in the web page forms
4. Validating the content of the pages returned by the server

It will then describe how to replay the testcase and inspect the results.

Load Testing

Load Testing usually consists of these steps:

1. Analyze baseline performance
2. Create and validate a load test configuration
3. Run load tests
4. Analyze load test results

Step 1 is covered by the *Record and Analyze a Testcase* tutorial.

Step 2 is covered by the *Configure and Replay a Testcase* tutorial and the *Configure a Load Test* section of this tutorial.

Steps 3 and 4 are the primary focus of the *Load Testing* tutorial.

Introduction to Load Testing

Introduction To Load Testing

Web site load testing refers to subjecting a web server to a ramping number of simulated users. The resulting analysis can measure your server's existing capacity, and is an integral part of improving the performance of any web-based application.

The purpose of the tutorials is to describe the Web Performance, Inc. load testing methodology so that our customers will understand how to systematically test their websites using our software, but it is also applicable to other performance testing situations. The questions that are typically answered with load testing include:

- How Many Users Can Your Web Site Handle?
- Which Web Pages Are Slow?
- Does My Site Crash Under Load?
- How Many Hits/Sec Can My Web Site Serve?
- What's My Site's Bandwidth Requirements?
- What's My Site's Baseline Performance?

There are three phases to the testing, which roughly correspond to:

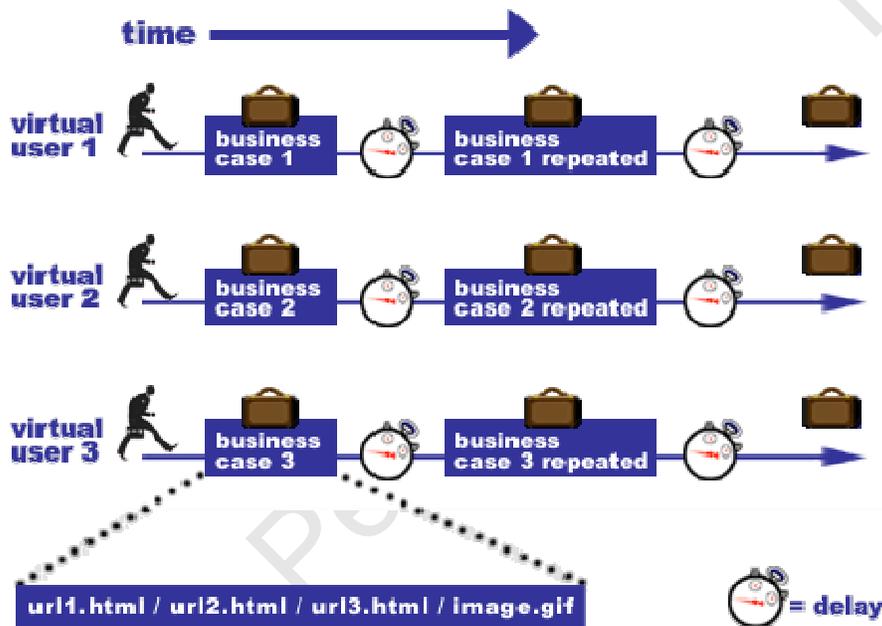
1. Baseline Analysis
2. Test Development/Verification
3. Full Scale Performance Testing

Each phase of the testing process has its own prerequisites and deliverables. The Web Performance Suite™ is designed to facilitate this testing process by separating the different parts of performance testing into different tasks, and by automatically generating the reports needed to both analyze the performance and communicate that information to others.

Virtual Users

The virtual users emulate a browser in every way; the web server can not tell the difference between a real user hitting the web site and a software-generated virtual user. Your web pages are grouped into transactions called test cases so you can get measurements that have meaning for your business. The illustration below shows how each virtual user can execute a totally different business case, creating a realistic load on your web site.

how virtual users execute business cases



Another term to know is "user identity" which describes a particular user with their own username, password, etc. A test case could have millions of user identities, but only have one hundred of these active at any given time. The software license describes the number of **simultaneous** virtual users, which are different from how many user identities exist. From a technical point of view, when you have, for example 100 active virtual users, it is really describing the level of concurrency; 100 users are active at one time. The user identities, though, will be swapped out as needed to maintain 100 concurrent test cases.

From the point of view of a virtual user, when a test case finishes, it repeats that test case using a new user identity.

How many simultaneous virtual users you need to simulate depends on a number of factors, starting with how you express your web site's capabilities.

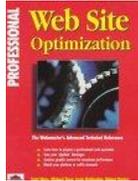
Please refer to the [hardware requirements](#) for more information.

One thing to keep in mind that performance testing starts with testing your individual back-end machines. Most large web sites scale by adding web servers and application servers. Setting up a large multiple server performance test takes significantly more time and resources than setting up a single server test. For that reason, you may want to start testing with a small number of virtual users on an individual test server.

For detailed background information about doing performance tests on a web server check out our [mini book reviews](#), or [call](#) for more information.

Recommended Reading

While [Web Performance Trainer](#)[™] makes it as easy as possible to do web performance testing, testing and tuning involve a lot more than simply using a tool. For detailed background information about planning, documenting, and tuning as well as performing tests we recommend the following books:



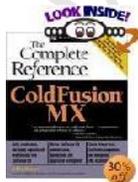
[Professional Web Site Optimization by Scott Ware, Michael Tracy, Louis Slothouber, Robert Barker](#)

Although this book was published in 1998 it still holds up as the best introduction to web performance tuning. It covers a wide range of topics, and covers everything you need to know to get started [load testing](#). The age of the book means it doesn't cover some new topics, but surprisingly enough most of the book is still relevant. If you are new to [load testing](#) and don't know where to start you should purchase this book first.



[Web Performance Tuning by Patrick Killelea](#)

Published in 1998, this book is one of the best for web performance testing, covering the technical basics for everything you need to know in order to really understand performance tuning. It includes such required information as definitions of various performance metrics, and what those should be in the real world, and moves along through networks, hardware, and operating systems. It goes to great pains to cover a variety of systems, including Windows, Linux, Macintosh, and a variety of web servers.



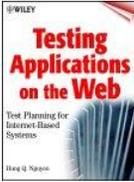
[ColdFusion MX: The Complete Reference by Jeffry Houser](#)

Although this book is specifically about ColdFusion, it does have a chapter on performance, and gives details how to both monitor and test the performance of a ColdFusion server. The basics of performance testing with Web Performance Trainer are presented in context, showing how and why it should be used in a professional setting. We've gotten good customer feedback on this book.



[The Web Testing Handbook by Steven Splaine and Stefan P. Jaskiel](#)

This book is about web testing in general, not just performance testing, and is a must have for the professional testing engineer. Chapters 7 and 8, on performance and scalability give a very good introduction to the subject, and include a great sample performance testing plan.



[Testing Applications on the Web
Test Planning for Internet-Based Systems by Hung Q. Nguyen](#)

As its title would suggest, this book is all about test planning for all types of internet based systems. Its chapters on performance testing give a lot of details about planning a performance test and analyzing the results, including examples. If you really are interested in doing a complete and thorough performance test, this book is required reading.

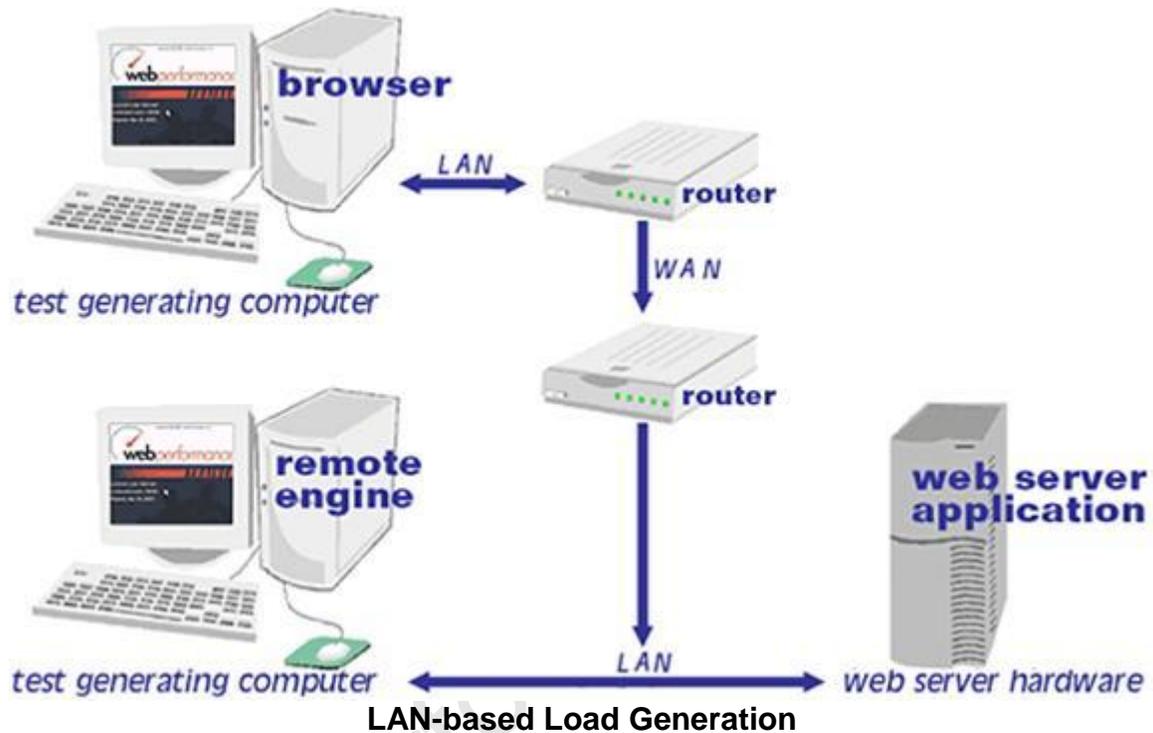


[Capacity Planning for Web Performance : Metrics, Models, and Methods by Daniel A Menasce, Virgilio A. F. Almeida](#)

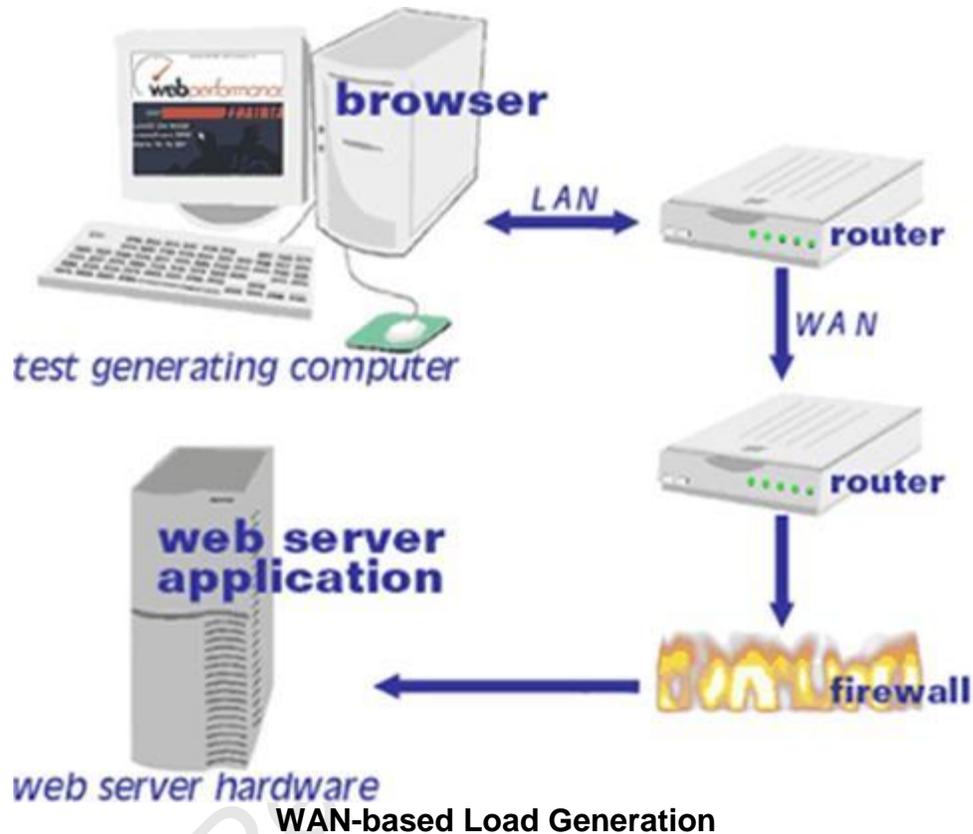
This book is a collection of technical articles on the theory of performance testing, and a good addition to the library of someone interested in the scientific and engineering aspects of web performance.

LAN vs. WAN

Load tests can either generate load from the same LAN as the web server, or from outside the LAN. The inside-the-LAN approach removes the service provider's network performance from the equation, but requires resources at the same location as the web server. This is the most cost effective way to load test since all of the network traffic is contained, and thus does not incur bandwidth charges from the web hosting provider.



When users are simulated from outside LAN, the network latency and network performance of the service provider are measured, but that also can make it more difficult to distinguish between network-related performance problems and server performance problems. This is also the most expensive way to test since the excessive bandwidth used during a load test has the potential to incur large extra fees from the hosting provider.



We strongly recommend a tiered approach to load testing regarding bandwidth:

1. First an initial bandwidth usage analysis in Phase 1 gives a rough idea of the bandwidth and network requirements.
2. Next, LAN testing is used to isolate and verify the correct operation of the web server. Once it is verified that the server, database, etc., can handle the load, then the measured bandwidth statistics can be taken to the web hosting company and used for capacity planning.
3. Finally, if there's any question of bandwidth capacity being a question an external WAN-based test can be accomplished, with the understanding that this will potentially incur bandwidth charges from the hosting company, and will increase the overall cost of the test.

Phase One - Baseline Analysis

Phase One Testing Procedure

Baseline Analysis

The preliminary analysis consists of recording one or more representative test cases to better understand the application and scope of the testing. The end goal is to get a picture of the “baseline” performance, which is the fastest the performance can be under any circumstances. Because recording is easy to do, the costs of performing this analysis are comparatively low.

Prerequisites:

The following items must be provided in order to complete Phase 1:

- A definition of acceptable performance
- A description of at least one representative test case
- Access to the web-application with non-critical data
- Access to someone who understands how the application works.

Performance Criteria

There’s no point in doing a performance test if you don’t know what to do with the results. Its best to decide on [performance goals](#) for your website right at the beginning before going through the time and expense of a test. There are no easy answers to this question, since the goals vary by application. Studies show that the percentage of people who give up on a website and go somewhere else go up as the performance decreases, but there’s no industry standard for where to draw the line. The performance goals can be expressed as:

**X percentage of web pages should load in N seconds,
with no more than Y percent errors.**

If you’re still stumped you can start with a goal of having web pages load in between two and four seconds for normal operation, although you might try staring out your watch for seconds to see how truly long a period that can be when you’re waiting for a web page. Of course, it’s possible for most pages on a site to load quickly, while a couple of pages require longer periods of time, which is why there’s a percentage qualifier. A good place to start for error percentages would be 1%. Our own published research shows that web servers will start rejecting connections while performance is still in the acceptable range, and that’s part of normal operation.

Execution:

During the execution of Phase One, one or more test cases will be recorded in the browser and then analyzed. These steps are explained in the subsequent chapters [Record A Testcase](#) and [Analyze A Recording](#).

Deliverables:

The Web Performance Analyzer™ module of the Web Performance Suite™ will generate the following information in a report which you can both edit and print.

Base Performance Analysis

The base performance is the fastest the system will perform under any circumstances. Before starting a performance test it makes sense to first investigate whether the system meets performance requirements while not under load. This report highlights which web pages do not meet your performance goals at various bandwidths.

Bandwidth Requirements Analysis

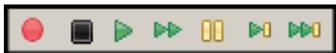
Using the base performance it is possible to estimate the bandwidth required to simulate any number of users, and is used as a ball-park figure to see if the available bandwidth is adequate before starting a load test.

Record A Testcase

Recording

Recording is the process of interacting with a website while Analyzer listens, records and analyzes the HTTP transactions between the browser and server. Analyzer constructs a tree representing the pages and URLs visited during the session. The recording may be [inspected](#) and [replayed](#) at a later time.

For a walk-through of the basic process, see the [Create a Recording](#) section of the [Quick Start Guide](#). A recording can be initiated from the *Record* (●) button and stopped with the *Stop* (■) button from the toolbar:



Once a recording is started, a new [Editor](#) tab is created to display the recording.

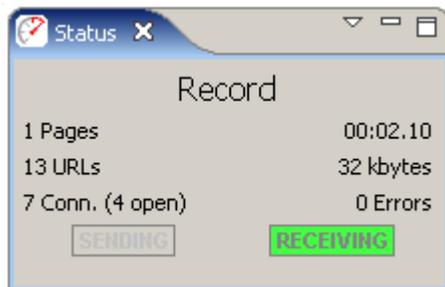
Browser and proxy configuration

In order to record the HTTP transactions, Analyzer acts as a HTTP proxy for your browser. This requires a change the the browser's proxy settings for the duration of the recording. The first time a recording is performed, the *Recording Configuration Wizard* will determine the correct proxy settings and prompt for the preferred browser for recording. To repeat this process, wizard can be restarted using the *Recording->Recording Configuration Wizard* menu item.

The preferred Browser and Proxy settings may be configured automatically in the Preferences editor. For details on configuration settings, see the [Browser Settings](#) and [Proxy Settings](#) pages.

Status display

While recording, the [Status View](#) will display the vital details about the recording status:



Recording SSL

How it works

When browsing SSL sites your browser encrypts the information sent to the server where it is decrypted. Normally, if a proxy is used by the browser, the proxy does not encrypt/decrypt the transactions - it simply passes the encrypted information through. In order for Analyzer to record the transactions, the internal recording proxy works differently - it decrypts/encrypts the transactions.

To make this work, Analyzer generates a "fake" certificate and presents it to the browser as the certificate for the server. In normal situations, this is considered a security hazard -- so when the browser detects this situation, it will display a warning message stating that it cannot verify the identity of the server. This is a good thing! If it didn't, then other programs might do what Analyzer does in order to steal your personal information.

To proceed with recording, you can simply accept the certificate and continue with the recording. This will not adversely affect Analyzer's ability to record your session, but it might produce recordings with response times that are significantly longer than a normal user would see (because of the time it takes you to dismiss the warning dialog). If a site uses multiple servers (such as most large banking and e-commerce sites), the security warning may be displayed multiple times.

How to suppress the warning messages

Analyzer generates an internal root certificate that is used to sign all of the "fake" server certificates. This root certificate may be imported into your browser as a

"trusted root certificate authority". This will allow your browser to automatically accept the certificates that are presented by Analyzer without displaying a warning message. Note that the internally generated root certificate is unique to your computer - this ensures that the certificate could not be used in a server-spoofing security breach (unless the attacker had already gained access to your computer and stolen the certificate).

To suppress the warning messages, two steps are required:

1. Export the root certificate
2. Import the root certificate into your browser

Exporting the root certificate

The root certificate may be exported in two different formats: CER or PEM. Most browsers will accept the CER format, so try it first.

1. Start a [recording](#)
2. When the Welcome Page appears, click the *test your SSL configuration* link
3. Click the appropriate link to download the certificate in either CER or PEM format
4. Save the certificate somewhere you can remember (e.g. your desktop)
5. Follow the instructions for your browser on importing the certificate. We have included instructions for a few of the most popular browsers below. If your browser is not listed here, check the documentation for your browser.

Internet Explorer 6.0

1. Select *Tools->Internet Options* from the IE menu
2. Select the *Content* tab
3. Push the *Certificates* button
4. Select the *Trusted Root Certificate Authorities* tab
5. Push the *Import...* button to start the Certificate Import wizard
6. Push the *Next* button
7. Push the *Browse...* button and locate the certificate file where you saved it
8. Then follow the Wizard to completion

After installing the certificate, you will see it listed under the name *Web Performance*. The certificate will expire in 10 years.

Firefox 1.5

1. Select *Tools->Options* from the Firefox menu
2. Select the *Advanced* icon
3. Select the *Security* tab
4. Push the *View Certificates* button
5. Select the *Authorities* tab

6. Push the *Import* button and locate the certificate file where you saved it
7. Select the "*Trust this CA to identify web sites*" option
8. Push the *OK* button

After installing the certificate, you will see it listed under the name *Web Performance*. The certificate will expire in 10 years.

Bandwidth Simulation

During recording or replays, Analyzer can simulate bandwidth-limited networks to demonstrate the effect of slower connections.

note: the simulation only limits the incoming responses from the server (i.e. requested pages, images, downloads, etc.).

Recording

Before beginning a recording, the simulated bandwidth can be selected from the list. To turn off bandwidth simulation, choose the *unlimited* option, which will deactivate the internal limiters.



Replay

To replay a testcase using bandwidth simulation, open the *Replay view* and select the bandwidth from the list. The simulated bandwidth may be changed at any time during the replay.

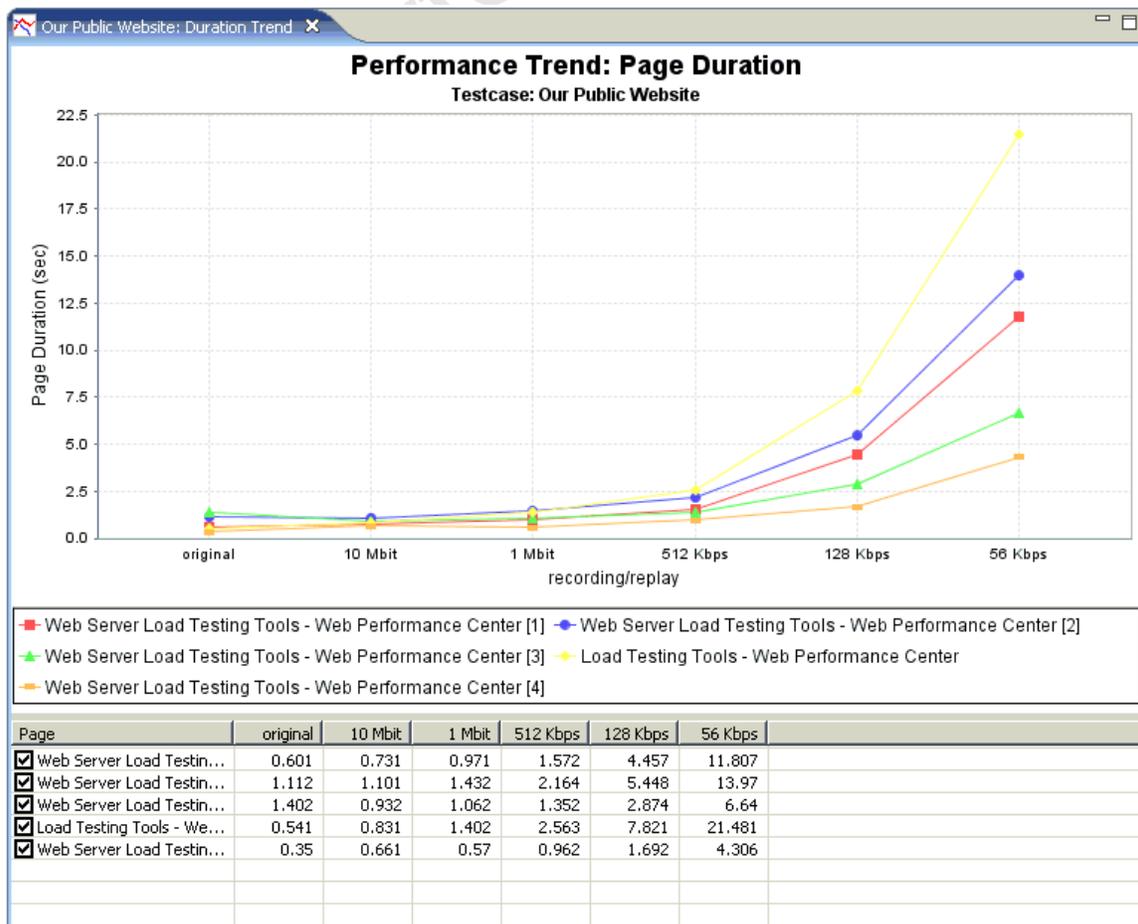
Content Headers Errors **Replay** X

Testcase: 11:49 AM 12/6/05 replay Duration: 00:08.56 Pages: 3 of 6
 Status: playing Errors: 0 URLs: 17 of 20
 Current Page: Website Load Testing Software - Web Performance, Inc. [2] - http://webper Unlimited

#	host	state	txns	Last URL
1	webperformanceinc.com:80	waiting	26	http://webperformanceinc.com/images/download_anl.gif
0	webperformanceinc.com:80	receiving	33	http://webperformanceinc.com/images4/analyzer_sm.jpg
2	counter2.hitslink.com:80	idle	3	http://counter2.hitslink.com/statistics.asp?v=1&s=207&

Connections Datasets

After replaying with bandwidth limiting activated, the timing shown in the testcase editor will reflect the effects of the simulated bandwidth limitations. The effects can also be viewed on the Performance Trend chart, which might look something like this, depending on which network limits have been tested.



Analyze A Recording

Performance Goals

Performance can be judged by setting Performance Goals that are automatically evaluated by Analyzer to give a quick visual assessment of the performance of a website.

Evaluating Performance Goals

When a testcase is displayed in the [Editor](#), the applicable performance goals will be evaluated for each page and URL. Items that do not meet the performance goals are indicated with an  icon. Hovering the cursor over the icon will display a tooltip summarizing the failed goals.

Setting Performance Goals

Performance goals can be configured at two levels: [global](#) and [testcase](#). Global performance goals are applied to all testcases automatically. Performance goals can be marked as global in the [Global Performance Goals](#) settings. Performance goals can be applied to a specific testcase in the [Testcase Performance Goals](#) settings. In either case, performance goals have the same configuration options available.

1. Name - each performance goal must have a unique name
2. Size & Duration - each performance goal can evaluate each page and URL in a testcase based on the size and/or duration of the response from the server.
3. Application Rules - the goals are only evaluated for resources that match the selections in this section. For the Server and URL rules, the additional field is used for pattern matching against the Server or URL in the transaction. If any part of the Server or URL matches the text supplied in the application rule configuration, the goal will be evaluated for that resource.

Goals summary

Goal name: default performance goal 1

Size (bytes):

Duration (sec): 6 2

Application Rules

Image 3

Web Page

Server

URL

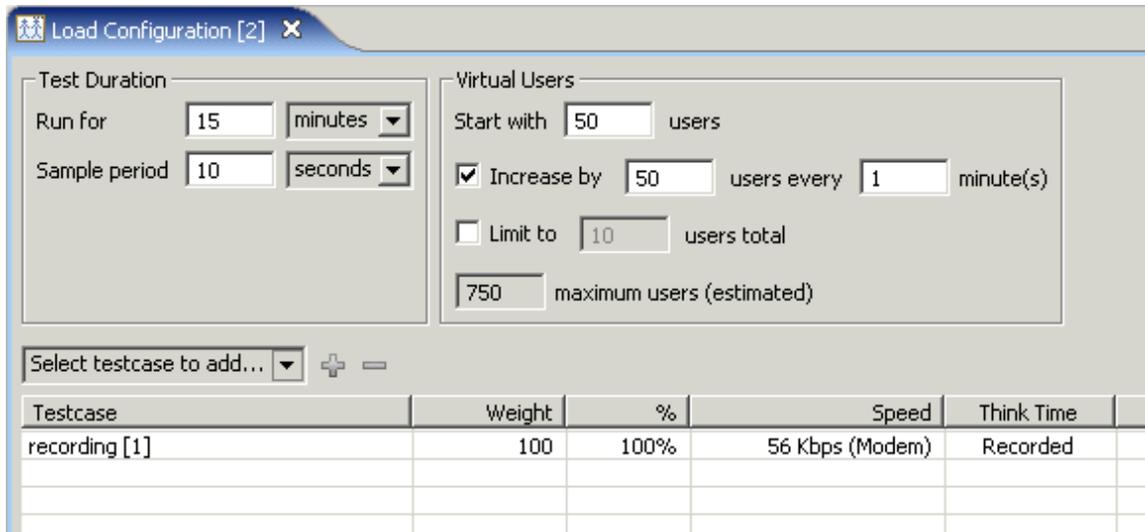
Apply

Changes made to performance goals will be applied to all testcases in open repositories.

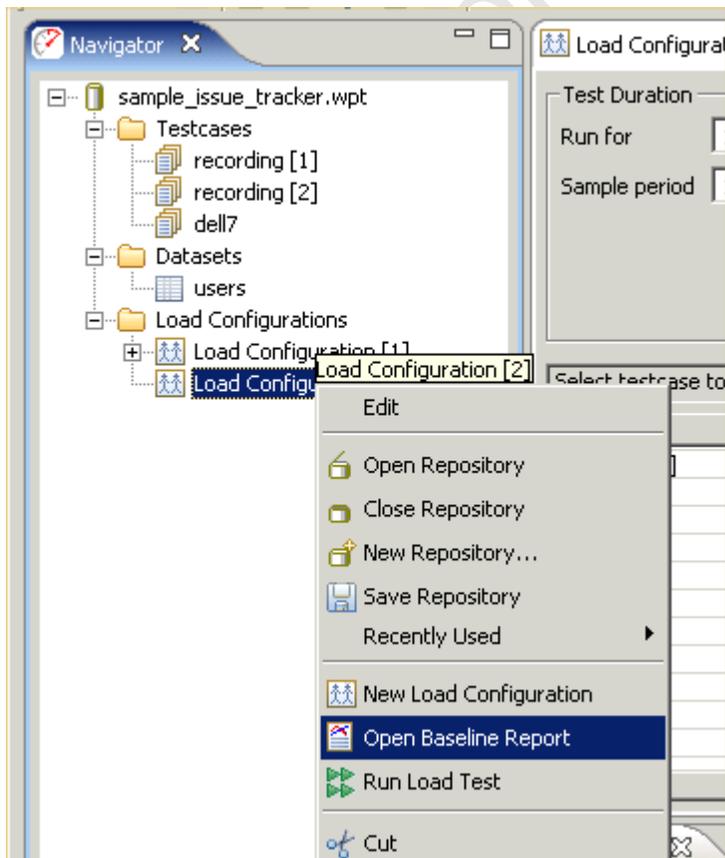
Baseline Analysis

Once a testcase is recorded then baseline analysis can be performed by creating a [Load Configuration](#). The purpose of the configuration is to specify the basic parameters of the load which will be used to generate the analysis, i.e. number of users, bandwidth of the users, etc.

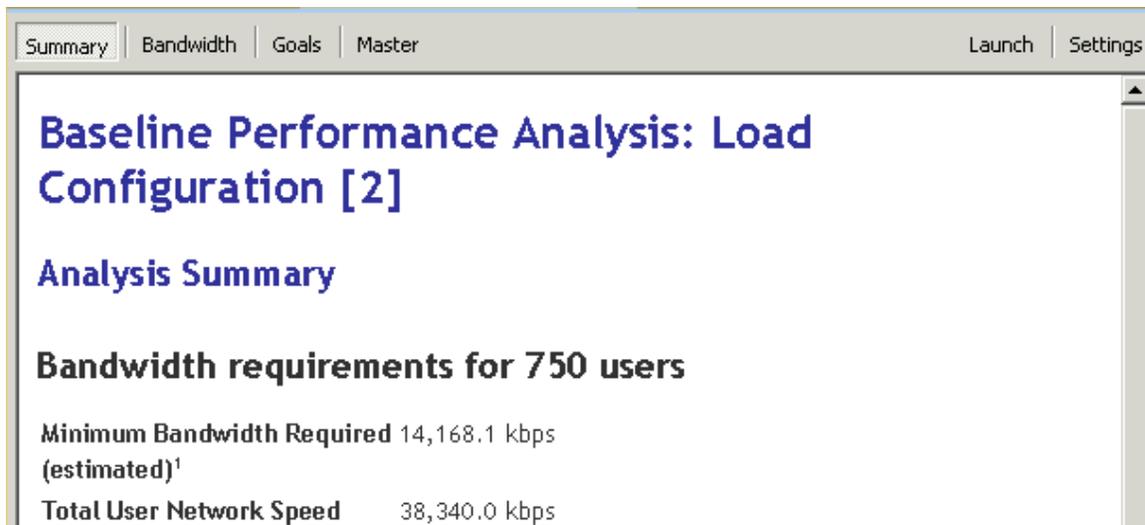
To start a Baseline Analysis right-click on the test case and select New Load Configuration.



Once this has been configured to describe a load test, then the baseline analysis can be viewed by right-clicking on the Load Test Configuration in the Navigator:



The following [Baseline Performance Analysis](#) report will appear:



The Summary gives an overall summary of the report's findings, the Bandwidth report gives estimated values for the minimum and maximum bandwidth connection needed by the hosting company to support the specified number of users, and the goals shows how many of the web pages will be estimated to meet or fail the performance goals. Of course these are just estimates and an actual load test will need to be run to get definitive answers.

Phase Two - Test Configuration

Phase Two Testing Procedure

Customize & Verify Test Cases

The goal of Phase Two is to make sure that simulation will be accurate, and that all aspects of the test meet the requirements, including network availability. This phase takes the most amount of time because it is here that the details of how the back-end works must be worked out. While Phase One relied strictly on recordings for analysis, in Phase Two the Web Performance Analyzer™ module will simulate a user with accurate data substitution, which puts extra requirements on the testing process. To make sure the virtual users are accurate the tester can actually watch the pages as they are sent to the web server, and visually confirm that everything is working correctly.

Prerequisites:

- A small number of accounts, usually between 10 and 20.
- A representative of the application/operations team to monitor the correct operation of the tests

- Test cases must be configured for multiple logins and unique data.

Execution:

- [Record](#) remaining test cases
- Configure test cases for unique data such as [separate logins](#)
- Check [application state](#) if used
- Configure [validation](#) (if needed) so you know the test cases are working
- [Replay](#) each test case with a single user to verify they are working correctly
- Repeat each test with more than one user to make sure multiple, simultaneous authentications are working

Deliverables:

- A suite of tests ready for a larger-scale multi-user simulation
- The final list of requirements for Phase 3.

Configure and Replay a Testcase

Configuration

Authentication

Authentication is the process by which the user tells the server who he/she is. The user may provide this information in a number of different ways and the server may or may not accept the credentials. After the server has accepted the user, the rest of the web application is made available.

Types

There are 4 common mechanisms of authentication in web applications, in order of frequency:

1. Web-form login - the username and password are entered on a web page and submitted by the browser. This may be done over either secure or insecure connections.
2. HTTP authentication - Uses the HTTP *WWW-Authenticate* and *Authorization* headers as described in the HTTP specification. It may use a variety of encryption algorithms depending on the client and server capabilities.
3. NTLM - An obsolete Microsoft protocol similar to #2.
4. Client Certificates - Uses identity certificates installed in the browser over a secure connection. Analyzer must be configured to use client certificates before the testcase can be configured.

Changing the Authentication in a testcase

Depending on your goals, there are different ways to change the authentication:

1. Change the username in the testcase to replay with a different user identity.
2. Configure the testcase to replay using a variety of different user identities.

Change the user in the testcase

Web-form login: you can change the values of the username and password fields in the recording. See the [Customizing a Testcase](#) section.

HTTP or NTLM: It is recommended that you re-record the testcase with the desired user identity. It is possible to change the credentials specified in the HTTP headers by manual editing, but should only be attempted by experts.

Client Certificate: Change the *Recording Certificate* in the *Client Certificates* preference page and set the playback mode to *Same as Recording Certificate*.

Replay with multiple user identities

Run the [User Identity Wizard](#) to configure the users. This can be invoked in any of three ways:

- Pop-up menu in the *Navigator*
- *Configure->User Identity* option in the *Edit* menu
- *User Identity* option from the *Configuration* button on the toolbar

Application State

Application State Management

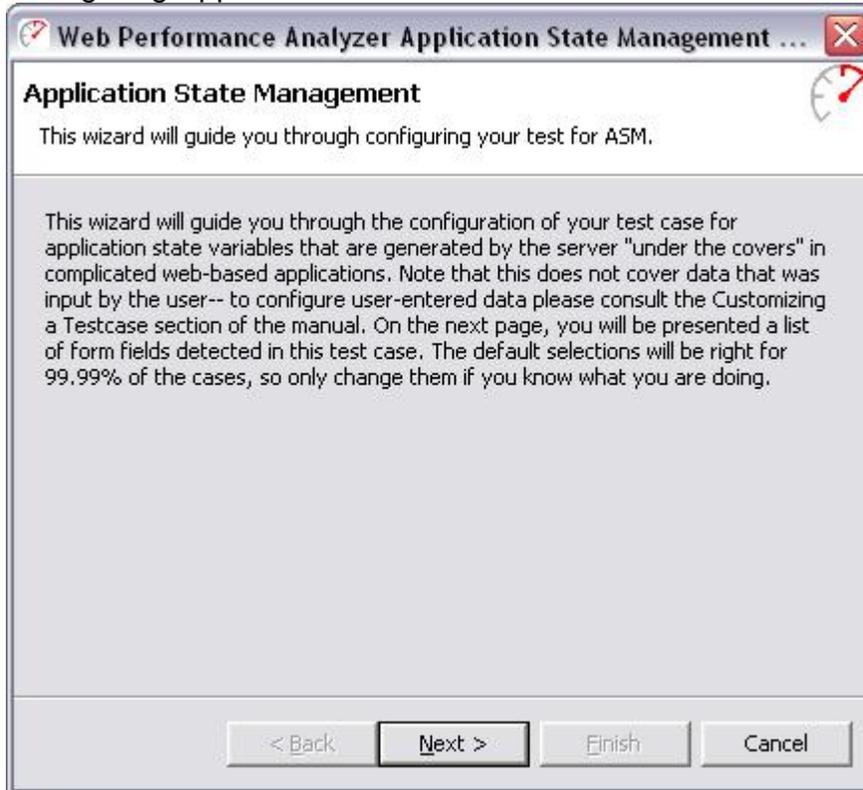
Modern web-based applications are getting more and more complex, which makes testing those applications very difficult. The web server or client-side javascript can generate unique hidden variables or separate URL paths for each user. Sessions can be specified not just through cookies, but hidden in the web page itself. Values from previous forms can be compressed, encoded, and stored inside variables with names like "`__VIEWSTATE`". Sometimes even the names of form variables changes from one user to another. **Note that Application State Management does not deal with data that the user would enter in form fields** or any other type of user-entered data. Application State Management is about all of the other complex variables changing behind the scenes. To change the input entered by the user in a form, see the section on [Customizing a Testcase](#).

With a scripting-based load tester you'd have to find each dynamic variable and where it is used, and configure it by hand, if it is supported at all. A typical application could have hundreds of dynamic variables, which means developing the test case can take days even if you understand the scripting language. With WPT the Application State Management wizard automatically finds and configures each dynamic variable for you. It locates where the variable is first used, and configures a parser to extract that value at runtime, and then detects where the value is used, and configures data replacement so that each virtual user gets its own unique runtime value.

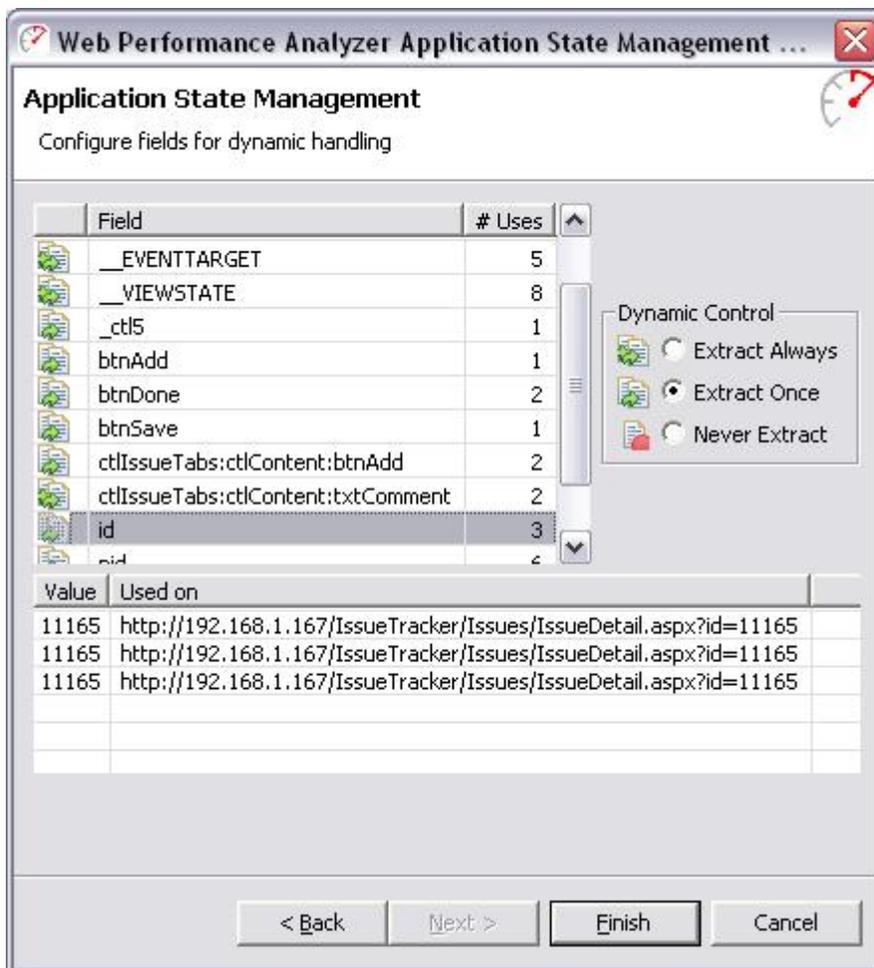
Starting the Application State Manager

The Application State will normally be automatically configured by the Testcase Configuration wizard before a replay can be attempted. To run the wizard again at a later time, select the testcase in the Navigator (or open the testcase in the Testcase Editor) and select the *Configure->Application State* option from the pop-up menu or toolbar.

Configuring Application State



Be sure to read the warning again! In order to make any changes you'll need to know how your application manages state. The following example shows some of the state management variables from making a recording on a sample ASP.NET application:



The variable "id" has been selected, and the values and where those values were used shows in the table below it. There are three options for dealing with how the dynamically changing names and values are handled. The next options deal with how often the value changes. If the value is set once, and then simply reused, the option will be set to "Extract Once". If, though, the value changes frequently, say on every page, you'll want to parse a new value every time it appears, or the "Extract Always" option. Lastly, if the field should not be managed by WPT, then you may select the "Never Extract" option, and whatever happened in the recording will happen at playback.

Why would you want to change the defaults? While the detection algorithms in WPT are smart, there is always the possibility that a variable had the same value on every page in a recording, but that might have simply been a coincidence. With another simulated user the values might have to change on every page.

Troubleshooting

The best way to determine if your application's getting the right values is to check your application state or database. For example, if you run a test case that supposed to purchase a product, check your database to see if there's a record of the purchase. If this shows there's a problem, the next step is to check your own application's error logs for trouble.

Once a problem has been verified, the next step is to walk through the the pages in the replay, looking for error messages from the server. It may be assistive later to configure validators (from the [Validators View](#)) to flag the same errors again during further replays. If the error on the first error page in the replay suggests that the cause of the error was not user entered data, but a hidden variable normally handled internally by the user's web browser, then you may use the [Fields View](#) to track down any variables on that page that do not have modifiers to update them automatically (if applicable). Once you have located a variable that is not being handled automatically, and confirmed how the application automatically updates that variable, you may consult the [Advanced Application State](#) section of the manual to give the Application State Management Wizard enough knowledge to correctly update your scheme.

Fixed value fields

The ASM Wizard will attempt to automatically configure those variables shared by the end user's Web Browser and the Application Server, but are not immediately exposed to the end user. Generally, no further configuration is required in order for your testcase to play back successfully. However, an easy optimization can be made to increase the number of virtual users supported by each load generating engine by removing those fields that never change. However, for large test cases, removing those fields from the ASM Wizard may be an inconvenient approach.

The ASM Wizard offers two configuration files for specifying fields that should be treated as fixed values. The files "ignored_field_names.txt" and "ignored_field_values.txt" (when created) will be read by the ASM Wizard each time it is run. These files are located in the product's installation directory, under "plugins\com.webperformanceinc.wpicore_<version>\config\dfc" (where <version> is the version number currently installed).

Both files follow the same format: each line contains a name (ignored_field_names) or a value (ignored_field_values). For example, suppose your case contained a HTML fragment: `<input name="btnSubmit" type="Submit" value="submit" />`

This may result in a fixed value post being sent to your server:

```
btnSubmit=submit
```

You may always remove this value from the ASM Wizard manually, or you could specify that this field always be automatically removed with either configuration file

```
ignored_field_names.txt OR ignored_field_values.txt
btnSubmit                submit
```

Be very careful not to include a blank line (especially at the end of the file) unless you intend for the Wizard to treat blank values as fixed values as well. The next time you run the ASM Wizard, any usage with their name or value specified in one of the ignore lists will be automatically ignored or hidden by the wizard.

While the Application State Management Wizard is capable of handling many multitudes of complex internal variable assignments within a web page, occasionally it is not able to definitively determine what section of a page caused the browser to post a field. One common case of this is with advanced Javascript events that are triggered by mouse movements or clicks. Often, these events only contain information indicating how the user is browsing the page, and it is not necessary for Analyzer to be able to emulate them, as by default it will simply play back the user's actions exactly as they were recorded in the Testcase. However, some web applications will use these events to insert dynamic application state data that must be more closely emulated in order for the application to be successfully automated.

Configuration files are located in the product's installation directory, under "plugins\com.webperformanceinc.wpicore_<version>\config\dfc" and end with the name `.properties` (where <version> is the version number currently installed). In order for detection to work, you should save your custom file and then run the Application State Management Wizard (or simply select the "Configure for Replay" option from the Configure menu) on any existing test cases.

Single Field String Delimited Detectors

Single Field Detectors are designed to locate segments of code within a page for a specific field.

```
detector=sdd
detector.name=Example Javascript Detector
string.prefix=setField('ID', '
string.suffix=');
field.name=ID
assignment=DEFINITE
```

Fields

Required

detector

This should always be set to either `StringDelimitedDetector` or just `sdd` for short in order to indicate that this configuration should be treated as a single field detector

For example: `detector=sdd`

field.name

The name of the field that is being assigned.

string.prefix

The prefix of the text just up to the value of the assignment.

string.suffix

The suffix of the text immediately following the value of the assignment.

Optional

assignment

the confidence that this detector has that the found assignment is accurate to the transaction. If *DEFINITE*, this detector *will* replace the value by the discovered assignment. If *PARTIAL*, this detector will only replace the value if a *DEFINITE* assignment for the same field was not found. If omitted, this field defaults to *PARTIAL*.

detector.name

The name given to this detector. If omitted this will default to the file name (less the .properties extension).

encoding.required

determines whether or not the content detected by this detector *must* be HTTP encoded before being transmitted. If false, HTTP encoding will be enabled only if characters are present in the detected value that can not be safely transmitted through HTTP. If true, then content from this detector will always be encoded for HTTP before being transmitted. If this value is omitted, it will default to false.

Variable Field String Delimited Detectors

Like the StringDelimitedDetector, this detector requires both a prefix and a suffix. However, the variable name may be substituted anywhere into the prefix or suffix by including the string "{0}" (without the quotes) wherever the name should be substituted. Single quotes (') must also be entered twice where used.

For example: Suppose the fields TX_ID and TS_ID were assigned in a page using a snippet of javascript code written as:

```
setField('TX_ID','1234'); setField('TS_ID','56789');
```

Then the VariableDelimitedDetector could be configured to detect both of these assignments (1234 and 56789, respectively) with a file in the form of:

```
detector=vdd
detector.name=Example wildcard Javascript function assignment detector
string.prefix=setField('{0}','','')
string.suffix='');
```

Fields

Required

detector

This should always be set to either VariableDelimitedDetector or just vdd for short in order to indicate that this configuration should be treated as a variable field detector

For example: `detector=vdd`

string.prefix

The prefix of the text just up to the value of the assignment.

string.suffix

The suffix of the text immediately following the value of the assignment.

Optional

accept.fieldname

A regular expression constraining which fields are subject to detection based on their names. If present, fields that do not match this pattern are omitted from this detector. If not present, all fields are examined by default.

assignment

the confidence that this detector has that the found assignment is accurate to the transaction. If *DEFINITE*, this detector *will* replace the value by the discovered assignment. If *PARTIAL*, this detector will only replace the value if a *DEFINITE* assignment for the same field was not found. If omitted, this field defaults to *PARTIAL*.

detector.name

The name given to this detector. If omitted this will default to the file name (less the .properties extension).

encoding.required

determines whether or not the content detected by this detector *must* be HTTP encoded before being transmitted. If false, HTTP encoding will be enabled only if characters are present in the detected value that can not be safely transmitted through HTTP. If true, then content from this detector will always be encoded for HTTP before being transmitted. If this value is omitted, it will default to false.

Further Configuration

If your application makes use of dynamic path segments, you may find further information under the section [Dynamic Path Segments](#).

Dynamic Path Segments

Some applications may utilize dynamic components not just in the form of traditional query parameters and field values, but also the path segments of the individual URLs. For example, a request for the URL

`http://mysite.com/widgets/14697302/index.html` may need to be dynamically replaced for the path segment 14697302 for each virtual user.

Detectors are presently limited to searching for a path segment within the location header of a previous redirect response. For further configuration options, please contact support.

A sample configuration file for this form of URL would look like

```
detector=dpsd
assignment=DEFINITE
segment.pattern=(\\d{6,})
```

This file should be located in the product's installation directory, under "plugins\com.webperformanceinc.wpicore_<version>\config\dfc" (where <version> is the version number currently installed), and saved with a ".properties" extension in the file name. The next time the the ASM Wizard is run, it will be able to apply this form of analysis to selected Testcases.

Fields

Required

detector

the type of detector to use. This style of detector may be specified as DynamicPathSegmentDetector (dpsd for short).

segment.pattern

a regular expression defining the criteria for what path segments are eligible for dynamic replacement. This detector will first ignore all path segments that do not entirely match this expression. Each dynamic component within the expression must be within a capturing group to then be eligible for replacement. In the above example, the pattern `(\\d{6,})` reads:

Look for a segment containing at least 6 decimal digits, and only decimal digits, and then replace the entire segment.

To replace just the numeric component within a path segment such as 64315_A, you could use the expression: `(\\d{5})(?>_\\w)?`

Note that \s within the expression must be double-escaped.

Optional

detector.name

the name given to this detector. If omitted this will default to the file name (less the .properties extension).

assignment

the confidence that this detector has that the found assignment is accurate to the transaction. If *DEFINITE*, this detector will replace this path segment from the first matching redirect it finds, if the redirect appears to redirect to this URL or a similarly based URL. If omitted, this field will default to *PARTIAL*.

Dependant Field Names

Occasionally your testcase will include variables that not only have changing values during playback, but also change in name as well.

Consider the case where two variables are posted to your application server:

```
serialno=1234
color1234=blue
```

In this case, you may specify that the variable *color1234* should be renamed, using a name derived from the variable *serialno* each time the test is played back. In order to configure your testcase, you must edit the file "DynamicFieldNames.txt". This file should be located in the product's installation directory, under "plugins\com.webperformanceinc.wpicore_<version>\config\dfc" (where <version> is the version number currently installed). For this case simply add a line starting with the identifying variable (*serialno*), an equals (=) sign, and the dynamically named variable (*color1234*) with the id replaced with an ampersand (&). The result is a line by itself:

```
serialno=color&
```

Once this line has been added, you may need to restart Analyzer in order for this setting to fully take effect. From there, simply re-run the ASM Wizard on your Testcase in order for the variables to be configured for playback. The next time your test is played back, Analyzer will pick up values from the variables their renamed assignments, and send them back to your server in renamed form. More elaborate testcases can also be defined using dynamically named variables. Consider if our case had been:

```
serialno=1234
color1234=blue
weight1234_in_lbs=5
1234_assembly_date=20051201
```

It is possible to specify multiple fields as having a single dependency in the "DynamicFieldNames.txt" by separating the fields by semicolons (;). In this case, we would add the single line:

```
serialno=color& ; weight&_in_lbs ; &_assembly_date
```

This configuration will allow the ASM Wizard to correctly assume name dependencies for all three dependent variables.

Customizing a Testcase

Replaying a testcase as recorded is useful in many scenarios, but it has its limitations. In many situations, it is desirable for the Virtual Users (VUs) to perform slight variations in order to more accurately judge the performance of an application. For instance, it the VU might sign onto the system using different username/password combinations. Or the VU might use different words when performing a search.

The process of configuring a testcase to submit slightly different information is referred to as *Customizing* the testcase.

Customization basics

The process usually consists of:

1. Provide some data to be substituted for data recorded in the testcase.
2. Configure *modifiers* to modify the original data with the new data.
3. Modifications to the testcase content.

Step 1 is accomplished by importing or creating data in a [Dataset](#). The data may also be [edited](#) later. These topics are addressed separately.

Step 2 will be the focus of this section.

Step 3 is accomplished using the testcase editor. See the section on [editing testcase content](#) for details.

What to customize?

In a typical web application, there are a number of ways that application data flows between the browser and server. The most common are:

1. Cookies
2. HTTP headers
3. Query parameters (and other URL elements such as path segments)
4. Form fields

Cookies

Cookies rarely need any customization because the testcases are automatically configured to handle cookies the same way that the browser would. The happens without any intervention from the user and as a result, there is little customization provided in the GUI.

Note that the automatic configuration supports cookies that are set by the server and received/returned by the browser via the HTTP headers. If the cookies are set or modified using Javascript (or any other client-side scripting mechanism), the testcase may need special configuration. Please contact support for more information.

HTTP headers

HTTP headers also rarely need customization by the user. However, support is provided for some simple customizations. See the [Headers View](#) for instruction on configuring modifiers on HTTP headers.

Query parameters

A query parameter is a part of the URL submitted in the HTTP start-line. In this example:

```
http://finance.yahoo.com/q?s=ibm
```

`s=ibm` is a query parameter. The parameter has a name (`s`) and a value (`ibm`).

A modifier may be configured for a query parameter in two ways:

1. Edit the request line in the [Headers View](#)
2. Edit the appropriate row in the [Fields View](#)

Form Fields

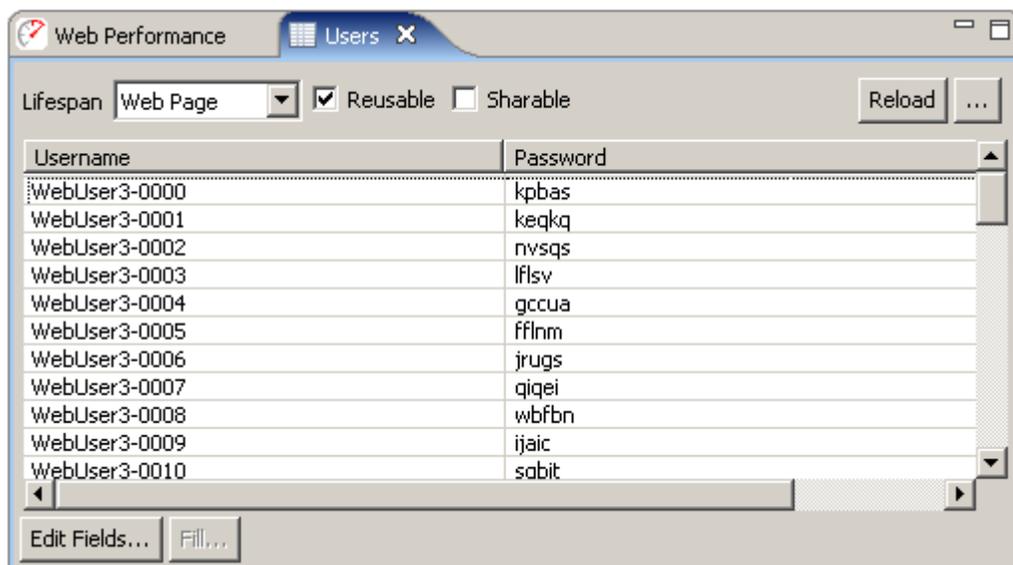
This is the most commonly-customized item in a testcase. This is how, for example, a testcase might be customized to submit different keywords in the search function of a website.

A form field is part of the HTML in a web page that allows a user to enter data into the page and submit it back to the server. There are a number of variations on this theme, including hidden fields that are not editable by the server (they are usually pre-populated by the server when the page is sent). All the fields submitted to the server may be viewed and edited in the [Fields View](#).

Datasets

In Web Performance products, a collection of data that is used to dynamically change the actions of a testcase is known as a *Dataset*. A dataset is a collection of tabular data, similar to a spreadsheet. After creating a dataset, it can be used to customize the testcase.

In this example picture of the [Dataset Editor](#), the dataset contains two fields (columns), *Username* and *Password*. It also has many rows, each of which contains values for the *Username* and *Password* fields.

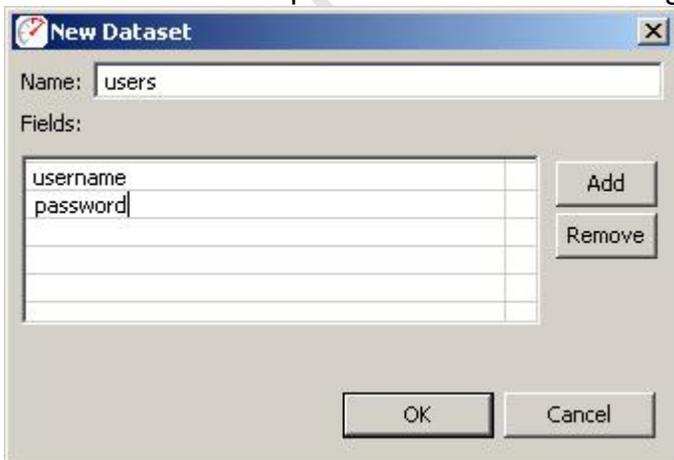


Creating a dataset

Datasets can be created with hand-entered data or imported from an external file.

Create a new dataset

In the [Navigator](#) view, the pop-up menu which appears from any existing dataset or the *Datasets* folders in each repository contains a *New Dataset* item. Selecting this menu item will open the *New Dataset* dialog:

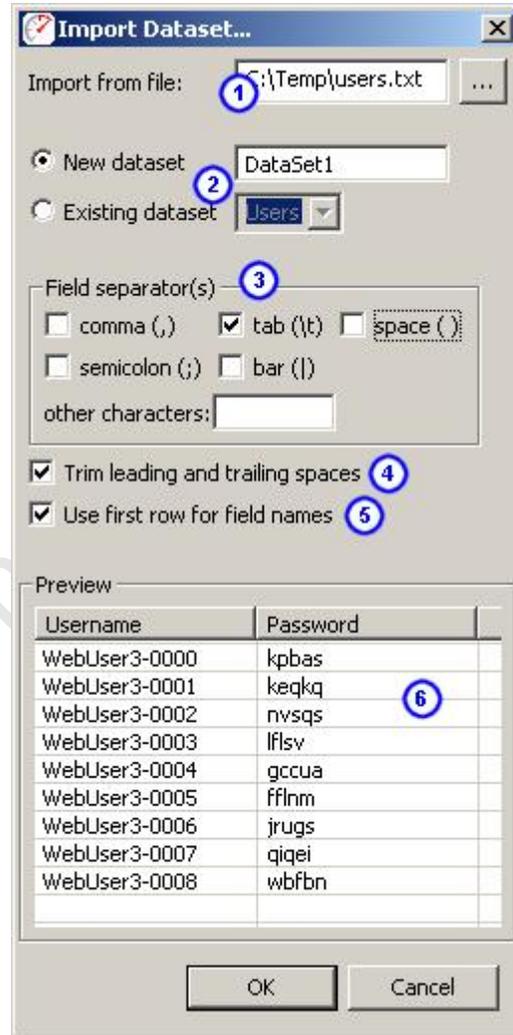


Enter a name in the name field and then press the Add button. You may then type each field name, separated by the <return> key to define the fields in the dataset. After pressing the *OK* button, the dataset will be created with one row of sample data and the [Dataset Editor](#) will be opened. Values for each field can be entered within the [Dataset Editor](#).

Import a dataset from an external file

A dataset can be created using existing data in CSV or text format. From the [Navigator](#) view, select the *Import* item from the pop-up menu on any dataset or the *Datasets* folder. Selecting this menu item will open the *Import Dataset* dialog:

1. Choose the file to import
2. The file may be imported into either a new or existing dataset
3. Choose the field separators. For CSV files, choose *comma*. This example uses tab characters between each field.
4. By default, the import process will automatically remove any leading and trailing white-space from each entry. This feature may be disabled when needed.
5. If the first row of the imported file contains the names of the fields, enable the "Use first row..." option. The import process will create a dataset with matching field names. If not, field names will be generated. They can be edited later in the [Dataset Editor](#).
6. As the import options are selected, the *Preview* section will display what the first 10 rows of the dataset would contain if the current settings were used.



Refreshing imported datasets

While the [Dataset Editor](#) provides a convenient interface for editing the values in a dataset, there are times when it may be more convenient to modify the data with external tools (such as a database query). After a dataset has been imported, it may be re-imported easily from the [Navigator](#) pop-up menu (*Reload* item) or the [Dataset Editor](#) (*Reload* button). The original settings will be remembered and re-used automatically.

Replaying

Replaying

A *Replay* is a simulation of a person using a browser to interact with a website. The pages visited are defined by the [Recording](#) being replayed. After each page is completed it will be selected in the [Testcase Editor](#) and displayed in the

[Content View](#) (unless in Fast Play mode). The *Content View* will automatically be activate when a replay is started.

Configuration

Prior to replaying a testcase for the first time, Analyzer will inspect the testcase for parts that cannot be replayed exactly as they were recorded. Then the Testcase Configuration wizard will display the recommended configuration steps. In most cases, the recommended steps should be followed.

This wizard can be re-run anytime by selecting *Configure->Testcase* option from the pop-up menu on the testcase (in the Navigator) or from the *Configure* toolbar button when a testcase editor is selected.

User Identity

If a replay should be performed using a different identity (e.g. username & password), the [User Identity](#) wizard will lead you through the steps for re-configuring the testcase to use usernames/passwords from a list. If NTLM or HTTP authentication is detected, the User Identity wizard will perform the necessary configuration steps.

If you wish to re-run the User Identity wizard later, select the testcase (in [Navigator](#) or [Testcase Editor](#)) and choose the *Configure->User Identity* option.

Application State

Many websites use dynamically-changing parameters in URLs or form fields. These testcases cannot be replayed exactly as recorded. The [Application State](#) wizard analyzes these fields and determines the most likely sources for these variables.

If you wish to re-run the Application State wizard later, select the testcase (in [Navigator](#) or [Testcase Editor](#)) and choose the *Configure->Application State* option. This wizard will lead you through the steps for re-configuring the testcase as needed. Some choices can be overridden - see the Application State section of the user manual.

Controls

For a walk-through of the basic process, see the [Replay a testcase](#) section of the [Quick Start Guide](#). A replay can be initiated from the *Play* (▶) button and stopped with the *Stop* (■) button from the toolbar:



Replays are performed using the selections under the *Recording* menu or the corresponding toolbar buttons. When any of the replay options are selected, the testcase being displayed in the active editor window will be replayed. If no editor windows are open, the testcase currently selected in the Navigator View will be replayed. The replay actions available are:

- Record - Starts recording a new testcase
- Stop - Stops the replay and cancels any outstanding transactions.

▶ Play- Replays the testcase including pauses between page selections ("think time").

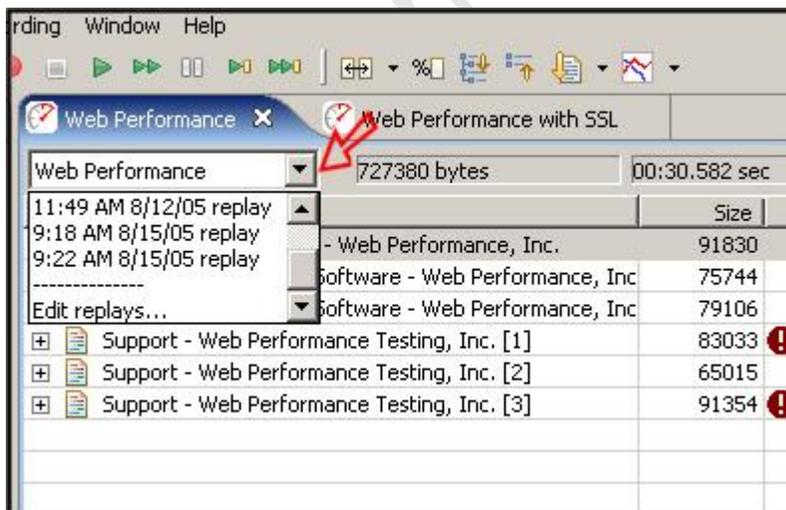
▶▶ Fast Play - Replays the testcase without think time between pages.

⏸ Pause - Pauses the replay after the completion of pending transactions. The replay may be restarted using any of the other buttons.

▶ Single Step - Replays the next transaction in the recorded testcase and pauses once the transaction is complete.

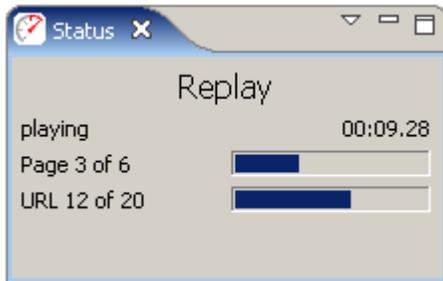
▶▶ Page Step - Replays the next page in the recorded testcase and pauses when the page is complete.

Once a replay is started, it is associated with the original testcase and is displayed in the editor window (if the editor for the testcase is open). In order to view a specific replay, select the entry from the pull-down replay menu at the top-left of the editor window, as shown below. To delete and rename replays, select the *Edit Replays...* item from the menu.

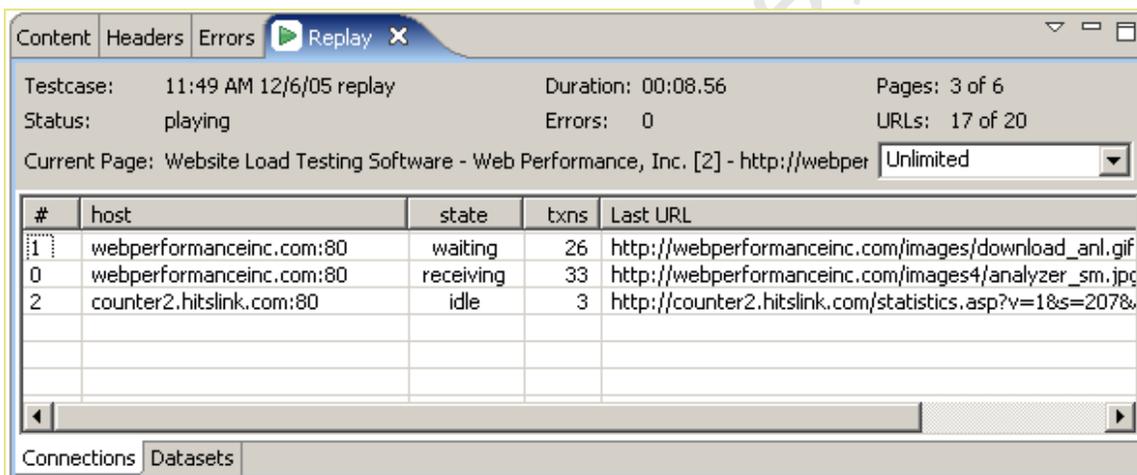


Replay status

The current status of the replay will be displayed in the [Status View](#).



More detailed information about a replay is available in the [Replay View](#).



A detailed log of the replay is available in the [Event Log](#) view.

Phase Three - Large Scale Tests

Phase Three Testing Procedure

Phase 3: Full Scale Load Testing

A full-scale load test consists of generating an increasing number of virtual users while measuring both client and server-side statistics. The result is a set of statistics that can be used to estimate the capacity of the system under test, and point the way to look for performance improvements. This stage can be repeated as necessary as changes are made to the system under test.

One area of interest is performance enhancement and code tweaking. While our performance testing consultants can suggest places to look to improvement, individual systems require the appropriate domain experts. For example, an Oracle DBA would be required to tweak the performance of stored procedures, while a .NET performance expert would be required to profile and modify .NET code.

Prerequisites

How Many Users to Simulate

A description of the load to generate must include how many users to start with, how many users to add in each time interval, and the end testing goal. Example:

“The test will start with 50 users, and add 25 users every two minutes until the goal of 500 simulated users is reached”.

Load Profile Description

A “load profile” is a description of the mix of test cases and bandwidths to be simulated. For example, if the application consists of two tasks, a load profile might be described as “40% test case 1 at DSL speeds”, and “60% test case2 and modem speed”.

Username & Passwords

If each virtual user must have a [unique identity](#), a large number of usernames and passwords must be configured in the system under test. For example, to maintain 100 concurrent users for 30 minutes when the test case lasts for 5 minutes could potentially require 600 usernames and passwords. (Each level of concurrency would repeat six times (30/5), which would be duplicated across the 100 concurrent users.)

Test Case Development

Any additional test cases needed for a complete test need to be completed and tested.

Client Access

A client representative must be available to monitor the correct operation of the tests as they run.

Execution

- Execution of Phase Three starts with [configuration](#) of a load test using the parameters specified in the prerequisites.
- Next the test is [actually performed](#)
- Finally, the test results are [analyzed](#) in a report

Deliverables

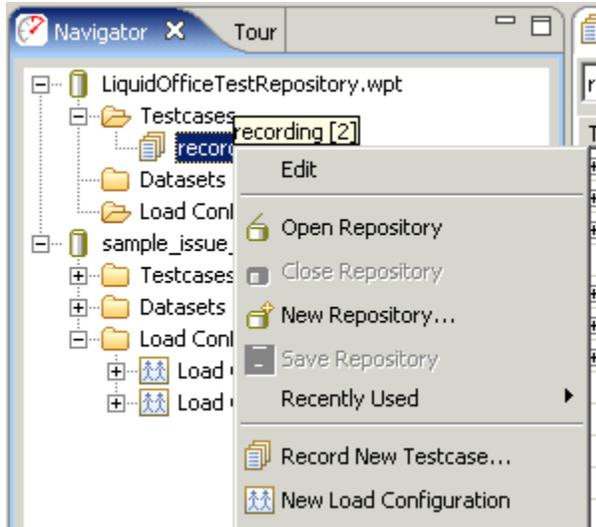
- A full [performance report](#) including a “how many users can your website handle” analysis.

- Recommendations for improving performance.

Load Testing

Configuring a Load Test

The first step in configuring a load test is to select a test case, and use the right-click menu to select New Load Configuration:

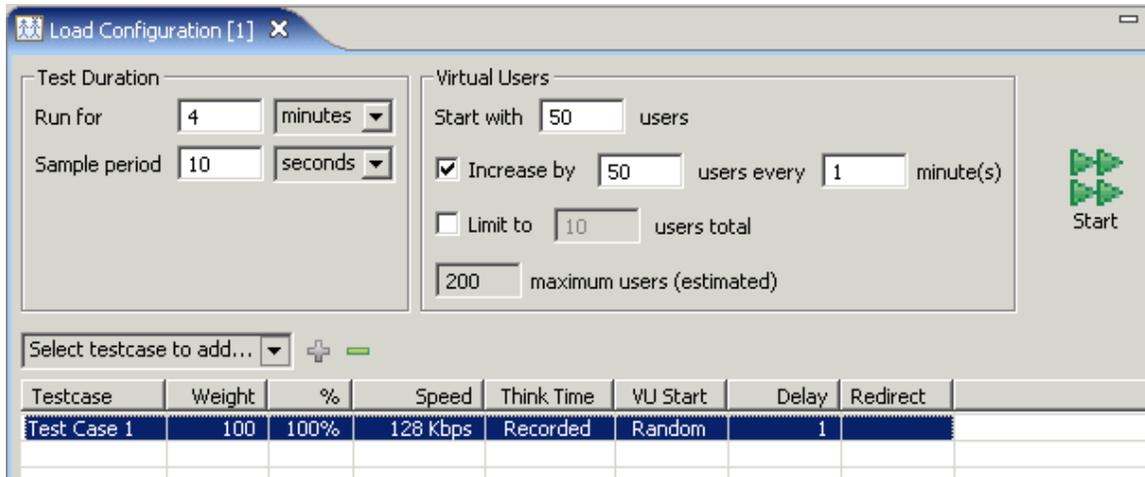


The goal of a performance test is to determine a relationship between the number of virtual users and performance. In order to do that, you'll want to describe a ramping number of virtual users and observe the changes in performance relative to the number of users.

This section of the GUI allows the user to describe the performance test in terms of the starting number of virtual users and how frequently to add new virtual users. A typical value is between 1 and 50 virtual users. The "increase by" value is how many virtual users to add in a period, usually between 1 and 5 minutes. Typically this value ranges from 1 to 50.

It is best to start with a low number of users and verify the correct operation of your server before performing tests with larger number of virtual users.

The test configuration below shows a test that will run for 4 minutes, starting with 50 users, and increasing by 50 users every minute. While the estimated maximum users that can be simulated by this configuration is shown as 200, the number of virtual users you can simulate is limited by the speed and memory of the playback machine, so that the actual number of virtual users generated is potentially lower than the value in the "potential" field.



Test Length

Duration can be specified in units of hours, minutes, or days. The duration of the test should change depending on your testing goals. If you are just trying to get an idea of the speed of certain operations on your site, useful performance information can be gained for tests that are a few minutes long. You can then tweak parameters in scripts or machine configuration and see if it has an effect on performance. If, however, you are trying to stress your web site to see if anything breaks, you'll want to run the test over a longer period of time.

Multiple Test Cases

More than one test case can be run simultaneously by adding them to the table. To add a test case to the table select the test case with the pulldown box and then click on the plus "+" sign. The distribution of test cases is determined by the "Weight" column. For example, if you were going to simulate 100 virtual users, and wanted 20% of the load to be from test case 1, and 80% of the load from test case 2, you would put a weight of "20" for test case 1, and a weight of "80" for test case 2.

Network Simulation

The "Speed" parameter describes the network bandwidth of **each** virtual user in the simulation. No matter what network configuration was used to record a test case, this setting controls the simulated network connection. For example, if the "Speed" parameter is set to 128 Kbps, that means the peak data transfer by each individual simulated user will not exceed 131,072 bits per second. (128 x 1024). This implies that if you recorded a business case over a local LAN, playing that business case back at modem speeds will take much longer. The implications of the effects of bandwidth can be studied by running a [Baseline Performance Report](#).

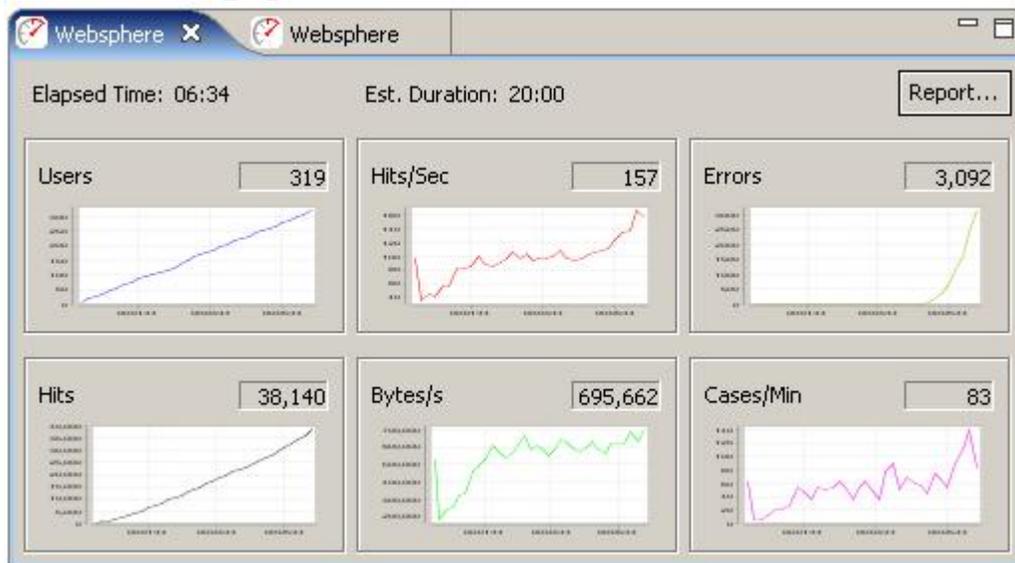
Sample Period

The sample period is the length of time over which statistics will be sampled before saving the values. For example, if the sample period is 15 seconds, the statistics views showing the results of a test will have values every 15 seconds. This value should be shorter for short tests, and longer for long tests. For example, if your test only lasts an hour, then having samples every 10 seconds makes sense. If, though, your test is intended to run overnight, then the sample period should be much longer, in the area of 5 minutes. This helps make the data easier to interpret. When running extended tests, Web Performance Load Tester™ will collect large amounts of data - which could cause the program to run out of memory and halt the test prematurely. As a rule of thumb: when running a test for multiple hours, you should have sample periods that are on the order of minutes, while short tests can handle sample periods as small as 5seconds.

For more information on the [Load Configuration view](#) please consult the reference manual.

Running a Load Test

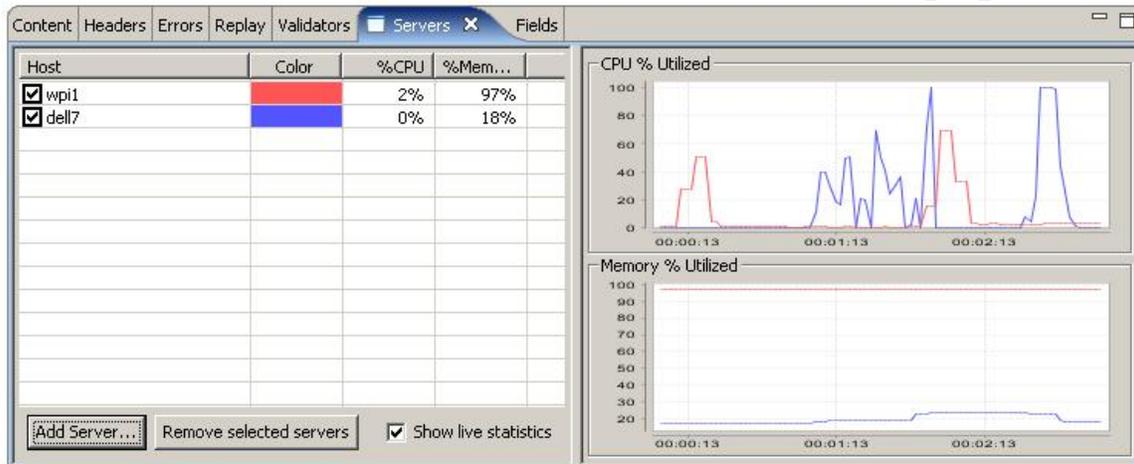
To run a load test start from the [Load Test Configuration Editor](#) and click on the Run Button. The following view will appear:



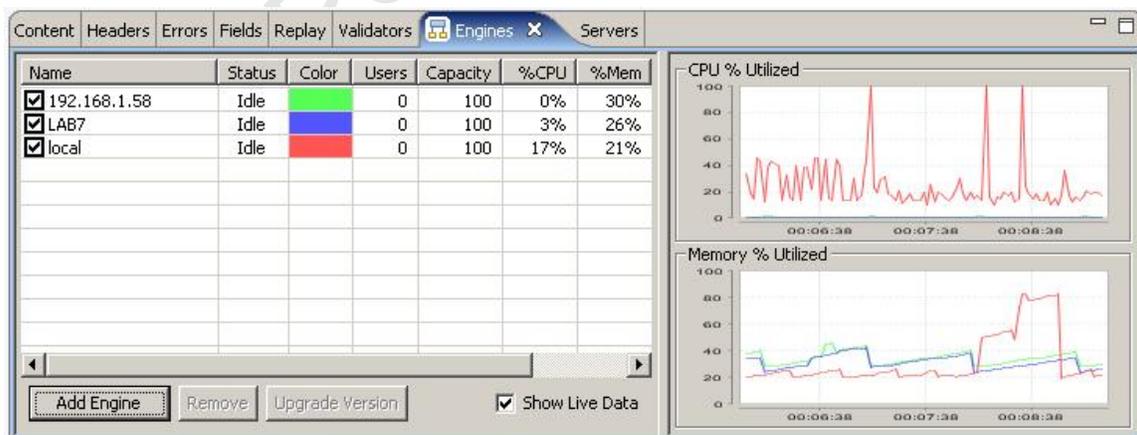
This is the [Load Test Results View](#), and will show you the test statistics being generated in real time. These statistics can be confirmed by simultaneously running a separate monitor on the web server(s) such as the Windows Perfmon

utility. Keep in mind that the statistics from multiple web servers and [Load Engines](#) are being combined for you to give an overall performance picture.

While the test is running you'll want to monitor the performance of your web servers using the [Servers View](#):



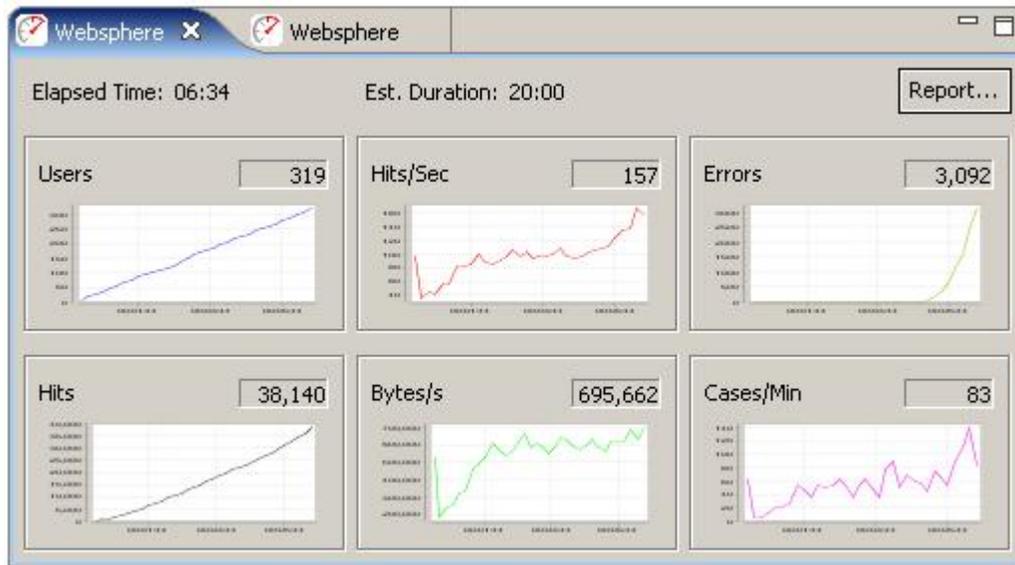
It is also important to monitor the performance of the computers generating the virtual users in the [Engines View](#):



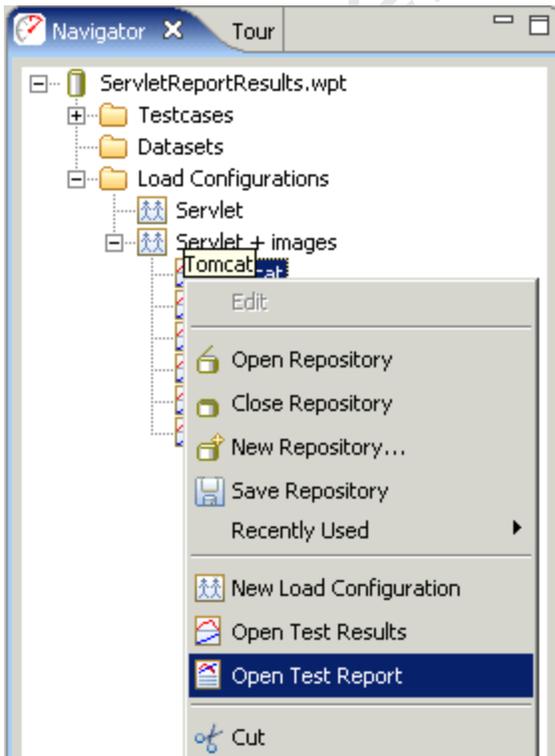
One of the major features of the software is it performs load balancing among the computers generating the virtual users in order to make sure the simulations are accurate. A computer that is overloaded with too high a CPU utilization or low memory can't generate an accurate simulation. Toward this end, even with a single computer the algorithm may limit the number of virtual users that can be added. For example, you may specify that a test add 50 virtual users every 1 minute, but if the computer you are using can't handle that at the moment, a smaller number of virtual users may be added.

Analyzing Load Test Results

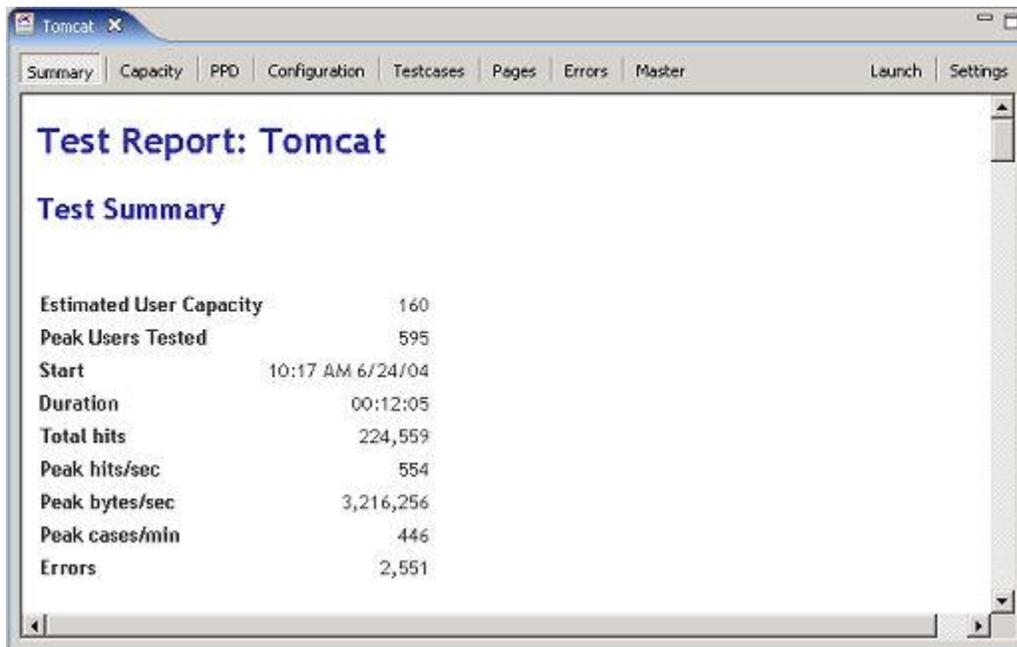
To analyze the results of a load test click on the Reports Button from the Load test Results View:



The report can also be accessed by a right-click on test results in the Navigator:



The [Load Test Report](#) view will be displayed:



The screenshot shows a web browser window titled "Tomcat" with a tab. The browser's address bar is empty. The page has a navigation menu with tabs: Summary, Capacity, PPD, Configuration, Testcases, Pages, Errors, Master, Launch, and Settings. The "Summary" tab is selected. The main content area displays the title "Test Report: Tomcat" and a sub-section "Test Summary". Below this is a table of test results.

Estimated User Capacity	160
Peak Users Tested	595
Start	10:17 AM 6/24/04
Duration	00:12:05
Total hits	224,559
Peak hits/sec	554
Peak bytes/sec	3,216,256
Peak cases/min	446
Errors	2,551

The contents of the report can be viewed in an external browser by clicking on the Launch Button, and from there it can be printed, saved for editing in an external program such as Microsoft Word, converted to a PDF, etc. The contents of the report are designed to be self-explanatory in that they include text surrounding the graphs and charts to explain their meanings. This extra text can be edited as needed for written or printed reports.

FAQs

General FAQs

Q: How do I report a bug?

A: You can access our [support-tracking system](#) directly or submit an issue directly from our product using the Support Request form from the *Help* menu. See the section on [Getting Help](#) for more information.

Q: When will you support other OSES besides Windows?

A: We hope to provide support for Linux, Solaris and OSX shortly after the 3.0 release. To help us decide which platform to support first, please vote for your favorite platform in our [issue-tracking system](#).

Q: Analyzer created a repository for me automatically - where did it go?

A: By default, repositories are stored in the [Workspace](#). If you cannot find it there, try re-opening Analyzer and hover the mouse over the repository in the Navigator - the tooltip will show the complete path of the repository file.

Q: I want to change where Analyzer stores my files and settings?

A: The files (by default) and settings are stored in the workspace. The [Workspace](#) section of the reference manual describes configuration of the workspace location.

Recording FAQs

Q: Why do I need to record some pages for Analyzer to analyze my website? Why can't it just scan my site and test all the pages?

A: Analyzer is designed for complex transactional websites that have strict performance goals and are likely to experience performance problems due to architectural, database or business-logic problems. Scanning a website for all the pages is impractical in these cases.

Our initial product surveys indicated that analyzing a website in the "spider" manner has little demand - but we are happy to be proven wrong! If you have this need, please tell us! You may vote for the feature request in our issue-tracking system (see the support section of our website and search for "scan website").

Q: When I record with Netscape (or Mozilla, Firefox etc), the pages do not look right (or might even have errors) in the Web Page tab of the Content viewer.

A: The embedded browser uses the default browser for the platform - on the Windows platform, the default browser is IE. Therefore, if your site returns content that is optimized for different browsers, the content displayed by the

embedded browser (IE) will be the content that was optimized for the browser used during recording, e.g. Netscape. The only solutions are: 1) record with IE, and 2) ignore the differences and errors in the embedded Web Page viewer.

Q: Why do all these *Security Alert* messages appear when I am recording or inspecting a testcase?

A: Because Web Performance products use "fake" server certificates during the recording process in order to decrypt secure pages from the server. See [these instructions](#) to silence the warnings.

Q: How do I record a site that uses Client Certificates?

A: See the [Client Certificates](#) section and the [Authentication](#) section.

Q: My browser is not recognized. Can I record testcases with it?

A: Yes, if the browser supports HTTP/S proxies. See the [Manual Browser Configuration FAQ](#) (next).

Q: How do I configure the browser manually for recording?

A: Follow these three steps:

1. Manually configure the ports that Analyzer will use for recording so that they will not change each time Analyzer starts. See the [General Settings](#) page for more help.
2. Configure Analyzer to start a custom browser when recording is started (*Browsers* section of the *Preferences* page). A custom browser may need to be configured if the browser was not automatically detected. Then the browser configuration should be set as the default browser. See the [Browser Settings](#) page for details.
3. Configure the browser to use Analyzer's recording ports as a proxy. This step is dependent on your browser - see the browser documentation for more information.

Playback FAQs

Q: How can I replay a testcase?

A: See [Replaying](#) section.

Q: How can I ensure my testcase was replayed correctly?

A: Manually, the pages can be inspected to see if the testcase replayed correctly. Select the replay in the [Testcase Editor](#) and then select the page to be inspected. The page will appear in the [Content View](#).

To automate this procedure, validators can be applied to each page to check the testcase in a automated fashion

1. Open the [Validators View](#)
2. Select the page to check in the [Testcase Editor](#)
3. In the Validators View, apply settings for size and/or content validation.

4. [Replay](#) the testcase
5. Open the [Errors View](#) - an error should appear in the table for any failed validators

Q: When I replay my recording, the value of some of the cookies are different. Why didn't Analyzer replay exactly what I recorded?

A: Analyzer is much more than a simple HTTP recorder/replayer. It simulates a real browser interacting with your server. This means that if the server sends new cookies, Analyzer will use them, just like a real browser does. As a result, Analyzer is compatible with sophisticated websites that require a user login and track session state using cookies.

Q: I want to change the username and password used in my testcase

A: See the [Authentication](#) section

Q: How do I replay with different users each time the testcase runs?

A: See the [Authentication](#) section

Q: How can I see which values are being used from a dataset during a replay?

A: The [Replay View](#) contains a tab that shows the current state of the datasets during a replay.

1. Open the [Replay View](#)
2. Select the *Datasets* tab
3. Replay the testcase.
4. If the testcase has very short think times between pages It may be helpful to *step* through the testcase one page at a time using the page-step button (see the [Toolbar](#) for the replay button descriptions)
5. The *Datasets* tab will indicate which datasets are in use by the Virtual User and what the values are for each field in that dataset row.

Navigator

The Navigator is used to view and manage test cases and datasets. Test cases and datasets are saved in a repository. Test cases are created by executing and saving tests with the Analyzer tool. Datasets are created by importing existing datasets or creating new datasets using the Dataset Editor. The repository is saved in a file with the .wpt extension.

Opening the Navigator View

The navigator view is opened by selecting *Window->Show View-> Web Performance Navigator* from the main menu.

Open, Close or Create a Repository

Each of these operations can be initiated from four places in the UI:

1. *Repository* menu.
2. Pop-up menu in the Navigator pane (right click to show)
3. Toolbar
4. *File* menu (standalone version only)

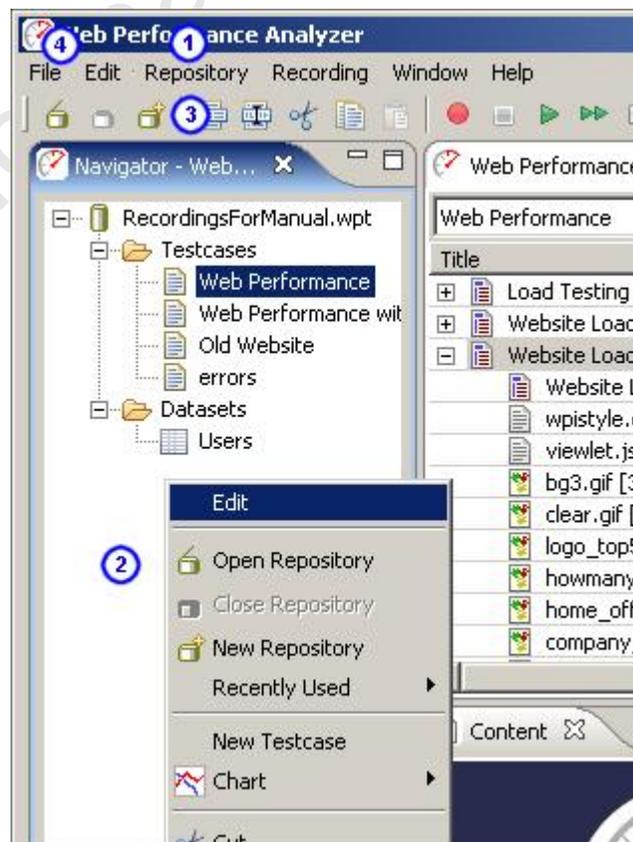
Opening

Multiple repositories can be opened at the same time using by holding the *Shift* key while selecting items in the list or tree.

The repository, test cases and datasets are displayed in the navigator when opened.

Closing

When a repository is closed, any open editor windows



associated with the repository are closed.

Creating

Before creating the repository, you will be prompted for the file name and location.

Recently Used Repositories

A shortcut to opening the recently used repositories is provided in the *Repository->Recently Used* and the pop-up menu in the Navigator.

Cut, Copy, Paste or Rename a Test Case

After selecting a Test Case in the Navigator, it can be cut, copied, or renamed. The paste capability is activated once a cut or copy has occurred. These actions are available in the following locations:

1. *Repository* menu
2. Pop-up menu in the Navigator pane (right click to show)
3. Toolbar

Cut

Any open editor window(s) associated with the Test Case are closed and the Test Case is removed from the list of Test Cases in the Repository.

Copy

Copy is used along with Paste to create a copy of the selected Test Case in the specified Repository.

Paste

Used in conjunction with Copy or Cut, Paste creates a new copy of the last cut/copied item in the Repository selected in the Navigator View.

Rename

A new name must be entered. Duplicate Test Case names within a Repository are not allowed.

Cut, Copy, Paste or Rename a Dataset

After selecting a dataset in the Navigator, it can be cut, copied, or renamed. The paste capability is activated once a cut or copy has occurred. These actions are available in the following locations:

1. *Repository* menu
2. Pop-up menu in the Navigator pane (right click to show)
3. Toolbar

Cut

Any open editor window(s) associated with the Dataset are closed and the Dataset is removed from the list of Datasets in the Repository.

Copy

Copy is used along with Paste to create a copy of the selected Dataset in the specified Repository.

Paste

Used in conjunction with Copy or Cut, Paste creates a new copy of the last cut/copied item in the Repository selected in the Navigator View.

Rename

A new name must be entered. Duplicate Dataset names within a Repository are not allowed.

Other Pop-up Menu items

Additional items on the Navigator's pop-up menu are available based on the item selected in the Navigator view. These are:

- Edit Testcase
 - Opens a [Testcase Editor](#) when a testcase is selected.

- Edit Dataset
Opens a [Dataset Editor](#) when a dataset is selected.
- Import Dataset
Opens the [Import Dataset Dialog](#) when a dataset or the Dataset Folder is selected.
- Reload Dataset
Opens the [Import Dataset Dialog](#) when a dataset is selected.
- Create a new Dataset
Opens the [New Dataset Dialog](#) when a dataset of the Dataset Folder is selected
- Opening a Chart
Creates a [Performance Trend Chart](#) (size or duration) when a testcase is selected.
- Record a new Testcase
Opens the Recording Dialog to start a new testcase when a testcase or the Testcase Folder is selected.

Notes:

- Every open repository loads the contents of each test cases into memory. Closing unused repositories will reduce the memory consumed by the test cases.

Testcase Editor

Testcase Editor

The Testcase Editor is used to view details of the testcase and the replays of the testcase. The Testcase Editor initially presents the test case as a tree of web pages, URLs. Opening the page (click the '+') will display the additional resources requested for the page (images, style sheets, etc.). When a new

testcase is recorded, a Testcase Editor window is opened and displays the transactions as they are recorded. The Testcase Editor can also display comparisons between the viewed content and a replay (or the original if a replay is viewed).

Title	Size	Duration	Status	URL
Load Testing Software - Web Performance, Inc.	86130	00:01.242	200	http://webperformanceinc.com/
Website Load Testing Software - Web Performance	94950	00:02.663	200	http://webperformanceinc.com/co
Website Load Testing Software - Web Performance	74933	00:01.011	200	http://webperformanceinc.com/pr
Web Performance Trainer Price List - Web Performa	70593	00:01.792	200	http://webperformanceinc.com/sa
Support - Web Performance Testing, Inc. [1]	80343	00:01.372	200	http://webperformanceinc.com/su
Support - Web Performance Testing, Inc. [2]	80315	00:01.172	200	http://webperformanceinc.com/su
Support - Web Performance Testing, Inc. [3]	91354	00:05.928	200	http://support.webperformanceinc
Demo Software Download Form - Web Performance	81578	00:11.086	200	http://webperformanceinc.com/dc
Website Load Testing Software - Web Performance	79784	00:04.316	200	https://webperformanceinc.com/lil
<forward>	1031	00:00.040	301	https://webperformanceinc.com/lil
Website Load Testing Software - Web Performa	13927	00:00.370	200	https://webperformanceinc.com/lil
wpistyle.css [9]	3518	00:00.030	200	https://webperformanceinc.com/w
viewlet.js [9]	2690	00:00.040	200	http://webperformanceinc.com/vik
bg3.gif [9]	4478	00:00.040	200	https://webperformanceinc.com/jir
logo_top5.gif [9]	7843	00:00.040	200	https://webperformanceinc.com/jir
howmany.gif [9]	4927	00:00.040	200	https://webperformanceinc.com/jir
home_off.gif [8]	1107	00:00.040	200	https://webperformanceinc.com/jir

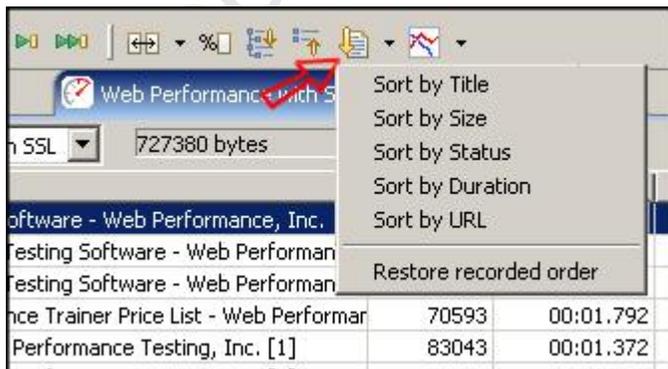
1. Type icon: An icon indicating the type of resource for this URL. There are icons for web pages (📄), text files (📄), images (🖼️), forwards (➡️) and errors (🚫).
2. Title: A logical name for the item. For web pages, the title will be extracted from the document. For other resources, a best-guess will be made, such as a filename. If the same resource appears multiple times, the titles will be numbered to avoid confusion.
3. Size: the size of the item (in bytes). For web pages, this is the total size of all items in the page.
4. Duration: the amount of time taken to load the item. For web pages, this is the total time taken to load all items on the page. The format is MM:SS:mmm (minutes, seconds, milliseconds).
5. Status: the HTTP code received in the response from the server.
6. SSL: displays a locked icon (🔒) if the transaction is with a secure site.
7. URL: the URL of the item, preceded by an icon representing the type of item (e.g. text, graphics).

8. Performance Goal warnings: If a performance goal failed for an item in the testcase, a warning icon (⚠) is displayed at the beginning of the column. Placing the mouse over the icon shows the cause of the performance warning.
9. Replay list: displays the replays that have been performed for this testcase. See the *Replay selection* section, below, for more details.
10. Editor menu: the pull down menu at the upper right corner of the Testcase Editor panel can be used to select comparisons, configure the testcase for replay, open charts, and modify the visible table columns.

update pic above to show (10) at the pulldown menu in the edit panel

Sorting

The Testcase Editor's table allows the user to sort the information in each column by either clicking on the column header or selecting the sort button on the toolbar and choosing which column to sort. Clicking the column header a second time or selecting the same column to sort reverses the sort direction. On the first sort, the title and URL columns are sorted alphabetically, the status column is sorted lowest to highest, and the duration and size columns are sorted highest to lowest. When a column is sorted, the transactions within each page are also sorted according to the sort selection.

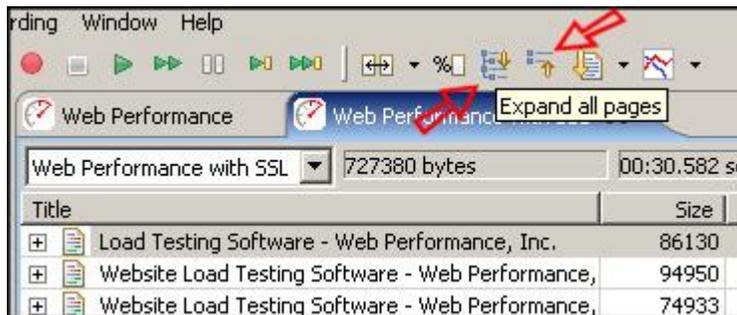


To return to the original view of the testcase, select the *Restore recorded order* selection at the end of the menu.

Expanding and Collapsing Web Pages

All of the Web Pages in a Testcase can be expanded to show all Transactions in the Testcase Editor by selecting the *Expand* button on the toolbar. The Web

Pages in the Testcase Editor can be collapsed to show only the main Web Page by selecting the *Collapse* button on the toolbar.



Editing Testcase Content

The Testcase Editor is used to view details of the testcase and the replays of the testcase. The Web Pages and HTTP transactions within the testcase can be moved, duplicated, or deleted using cut, copy, and paste. Performing these actions changes the structure of the testcase. This can limit the ability to accurately compare the testcase to other replays that have not been modified in an identical fashion. The editor displays a warning when performing an action that may invalidate comparisons. This warning can be suppressed by selecting the *Do not show this dialog again* option. To manually turn the warnings on or off, select the *Window->Preferences* option on the main menu, then select the *Web Performance->Testcase Editor* item. Select the checkbox next to *Display warning when operations performed that impact comparison data* item to change the setting.

The Testcase Editor is also used to modify the *think time* between Web Page requests in the testcase.

Menus and Shortcuts

The cut, copy, paste, undo and redo actions are available in two menus. A right-click context menu is available inside the Testcase Editor's View. The actions are also useable from the *Edit->* main menu. Standard keyboard shortcuts are enabled for the actions, these are listed in the following sections.

Cut

A Web Page or Transaction can be removed from the testcase using *Cut*. The keyboard shortcuts to cut an item are *Ctrl-X* and *Shift+Delete*. The item that is cut can be pasted back into a testcase until another cut or copy is performed.

Copy

A Web Page or Transaction can be duplicated in the testcase using *Copy* in conjunction with *Paste*. The keyboard shortcuts to copy an item are *Ctrl-C* and *Ctrl+Insert*. The item that is copied can be pasted back into a testcase until another copy or a cut is performed.

Paste

A Web Page or Transaction can be inserted into a testcase using the *Paste* action. The keyboard shortcuts to paste an item are *Ctrl-V* and *Shift+Insert*. The item that is pasted into the testcase is added at the next logical location following the item currently selected in the testcase. For example, pasting a Web Page while a transaction is selected adds the Web Page after the Web Page containing the selected transaction. It is possible to copy Web Pages and Transactions from one testcase to another by copying from the first testcase and pasting into the second.

Undo

The cut, copy, and paste actions can be undone (up to a maximum of 10 actions per editor). The keyboard shortcut to undo the last action is *Ctrl-Z*.

Redo

After an *Undo* is performed, the action can be redone using the *Redo* action. The keyboard shortcut to redo the last undo is *Ctrl-Y*.

Modifying think time

insert pic of the editor's menu with *Properties...* selected

To change the *think time* between web pages, either right click on the web page and select the *Properties...* item or press the pull down menu at the upper right corner of the Testcase Editor panel, and select *Properties...*

insert pic here of the *properties...* dialog

The think time is modified by entering the new value in the text field and pressing the *OK* button. The timing change can also be applied to all web pages within the testcase by checking the *Apply change to all web pages* item.

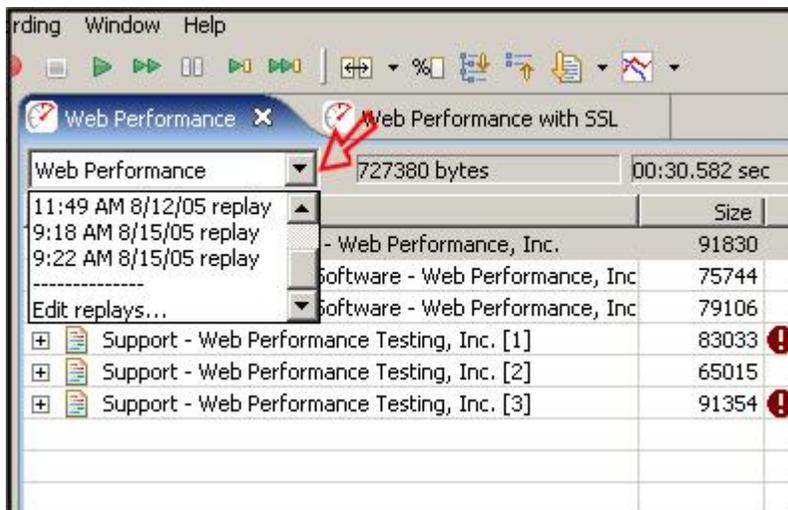
Working with Replays

Replay selection

Replays are created using the play options available under the *Recording* menu. For more information on creating replays, see the [Replaying](#) manual page. Note that any replay can be compared to any other replay or the original recording.

Although this section will only refer to replays, it always applies to the original recording as well.

If the Testcase being displayed in the Testcase Editor has been replayed, the pull down menu at the top of the Testcase Editor panel contains a list of the replays. Replays can be viewed by selecting the appropriate entry from the list. The replays associated with a Testcase can be deleted or renamed using the *Edit Replays* selection in the replay menu at the top of the editor panel.

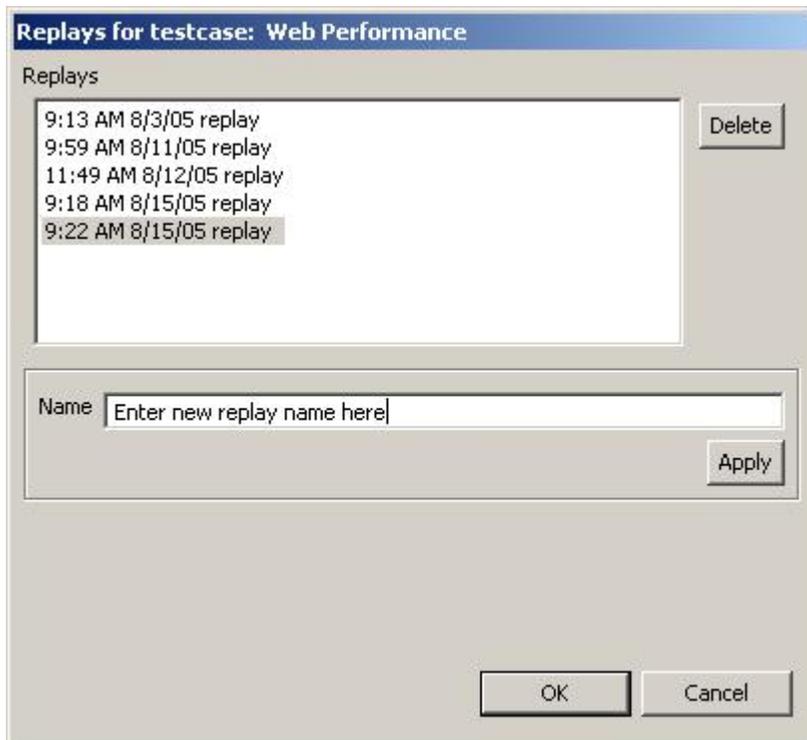


Modifying and Deleting Replays

To open the *Replay Editor*, select the *Edit replays...* item from the *Replay selection list*.

Once the Replay Editing dialog is opened, one or more replays can be deleted by selecting the replay(s) to delete and selecting the *Delete* button.

To rename a replay, select the entry in the list and modify the name in the text area below the list. When completed, select the *Apply* button to save the changes.

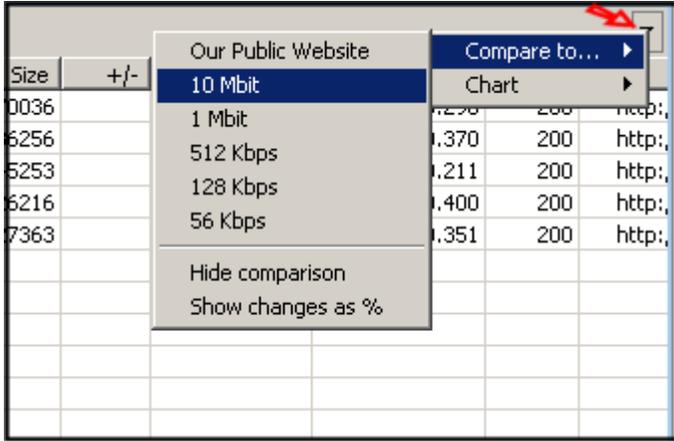


Comparing replays

The Testcase Editor can display the size and duration differences and changes in the Status between the displayed content and either a replay or the original recording. When a comparison is performed, the item being compared to the displayed content is shown at the top of the Testcase Editor panel to the right of the displayed item. The comparison is opened using one of the following:

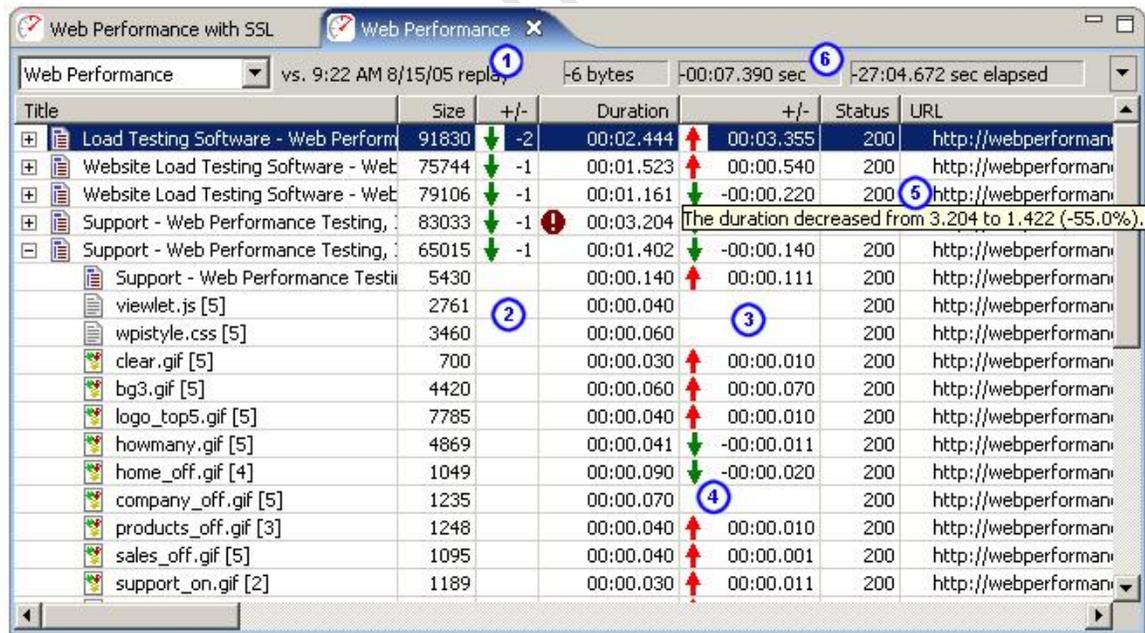
1. Press the pull down menu at the upper right corner of the Testcase Editor panel, select *Compare to...*, then select a replay (shown below)
2. Press the *Compare...* button  on the main toolbar, then select a replay.
3. Select *Edit->Compare to...* on the main menu bar, then select a replay.

for example:



Examining the differences

After selecting a replay to compare to, the Testcase Editor will change to display the results of the comparison. For example:



In this example, the following items have been added:

1. The comparison target has changed to reflect which replay is being compared to.
2. A column is added after the *Size* column that reflects the change in the page size
3. A column is added after the *Duration* column that reflects the change in the page duration
4. Icons in the new columns indicate if the performance has improved (↓) or degraded (↑).

5. Tooltips over the +/- columns indicate the magnitude of the change and the relative change as a %.
6. The change for the entire testcase is also displayed

Regardless of which replay is chosen first, the data will be compare in chronological order. For example, if the greater duration was encountered on the replay chosen via the *Compare To...* menu item, a duration increase will be displayed. Note that the *size* and *duration* displayed always correspond to the recording/replay selected in the dropdown list.

A change in the Status is indicated by a blue triangle icon. For more information about the change, placing the mouse over the icon in the table displays the full details for the comparison.

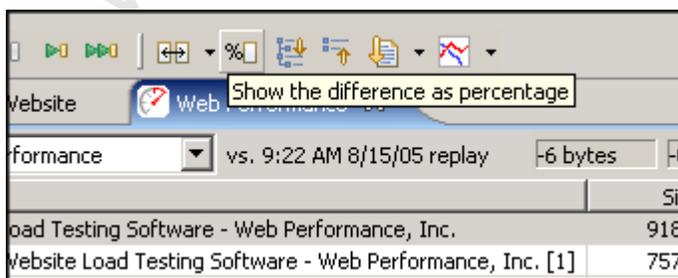
Viewing percentages

The size and duration differences are displayed as the numerical difference between the two values. The differences may also be displayed as a percentage difference. To view the difference as a percentage, there are three options:

- Press the pull down menu at the upper right corner of the Testcase Editor panel, select *Compare to...*, then select *Show changes as %*.
- Press the percentage button on the main toolbar.
- Select *Edit->Toggle percent view* on the main menu.

Note: once the Testcase has this setting modified, it will be remembered whenever this Testcase is opened.

for example:



Changing the % default

If preferred, the size and duration differences may be displayed as a percentage as the default setting.

To change the default setting to shows differences as a percentage, select the *Window->Preferences* option on the main menu. Select the *Web Performance->Testcase Editor* item. Select the checkbox next to *Display comparisons as*

percentage to set the application to show percentages as the default. Clear the checkbox to have the application show the numerical differences as the default.

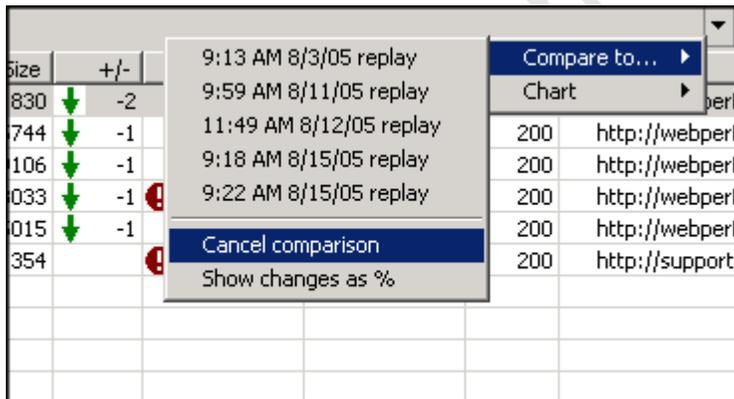
Note: Testcases that have been opened before will remember their difference setting and will override the default setting.

Cancel the comparison

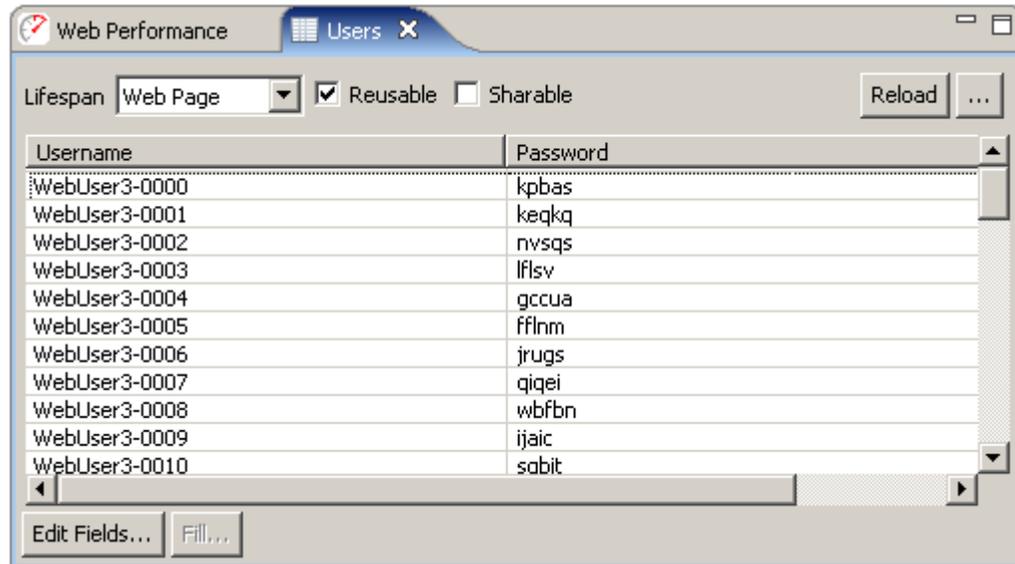
To cancel the comparison, use one of the following:

- Select *Compare to...* item from the Testcase Editor menu, then select *Cancel comparison*.
- Press the revert comparison button on the main toolbar.
- Select *Edit->Compare to...->Cancel comparison* on the main menu.

for example:



Dataset Editor



Dataset Configuration

Lifespan

The lifespan of the dataset defines how long the Virtual user will use values from the same row before fetching the next row of values. Note that if a testcase does not use any values from a dataset, no rows will ever be used.

- Virtual User - No matter how many times the testcase is executed, only one row from the dataset will be used..
- Testcase - A single row of values will be used for the duration of each testcase.
- Web Page - Each web page that uses values from the dataset will use a different row.
- URL - Each transaction (URL) that uses values from the dataset will use a different row.
- Single Use - Every time a dataset value is used, a different row will be used.

Reusable

If enabled, this setting will allow a Virtual User to start over at the beginning of the dataset when all the rows have been used. This could mean that a row is used more than once.

Sharable

If enabled, this setting would allow multiple Virtual Users to simultaneously use the same row from a Dataset. If a dataset is not reusable, it cannot be sharable.

Editing a dataset

To edit an entry in a dataset, double-click the cell and start typing. Press <ESC> to cancel the changes and <RETURN> to move to the next cell.

To add new rows, edit the last cell and press <RETURN>. A new row will be created with sample data - continue filling in new rows by typing new entries and pressing the <RETURN> key. Press the <TAB> key to finish editing the last new entry.

Reloading a dataset

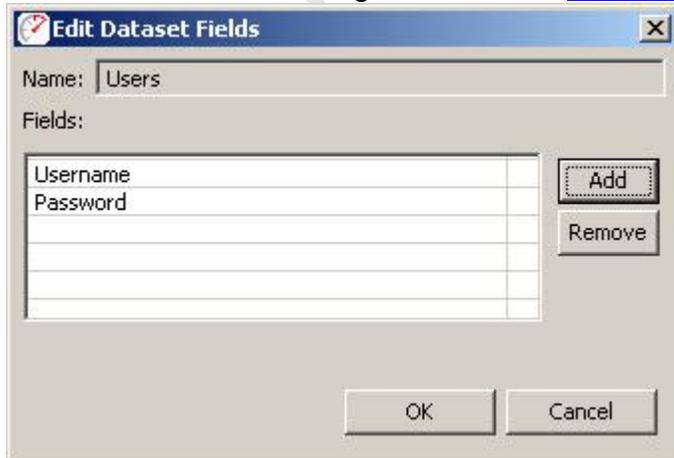
There are two options for reloading a dataset from an external file: automatic and manual.

The *Reload* button will attempt to automatically re-import a dataset using the settings that were originally used to import the dataset the first time. If an error occurs, or the dataset was not originally imported, then the manual method must be used.

The manual reload button (...) beside the *Reload* button will open the dataset import dialog. If the dataset was originally imported from a file, those settings will be preset in the dialog. If not, the dialog will have the settings last used to import a dataset.

Editing dataset fields

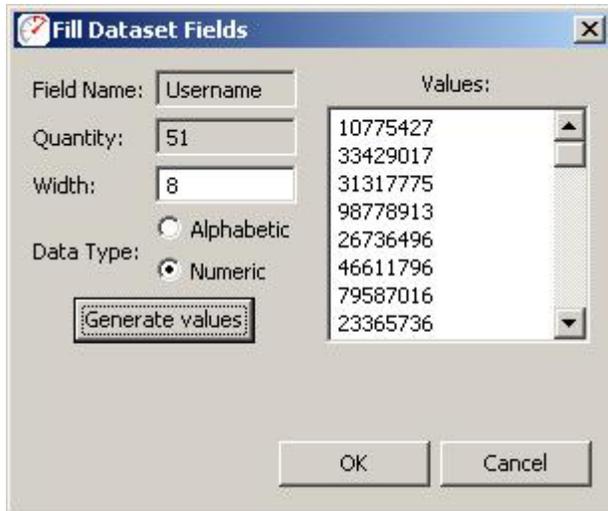
The *Edit Fields...* button will open the *Edit Dataset Fields* dialog, which is similar to the *New Dataset* dialog shown in the [Datasets](#) section.



This allows the creation or removal of fields in the dataset. A field may be effectively moved by deleting it and adding a new field at the end with the same name.

Filling fields with random data

The *Fill...* button allows the selected field to be filled with randomly generated alpha or numeric data. Select the field to be filled by pressing the field name (column heading) in the table.



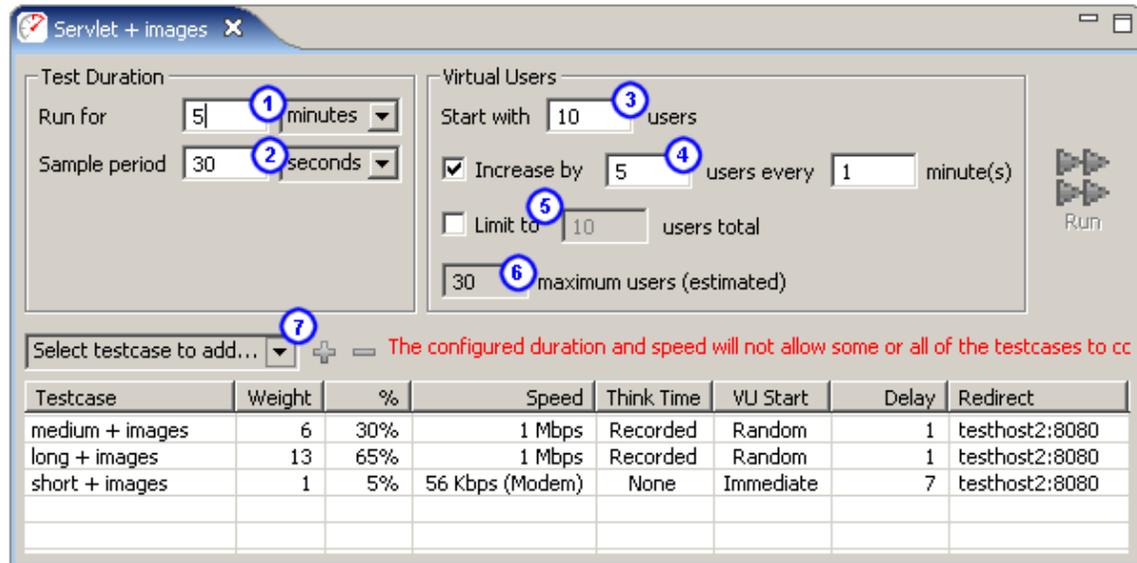
Note that the *Quantity* field will be automatically set to the total number of rows in the dataset and cannot be changed. The *Width* field defines how long each generated value will be. The *Data Type* chooses between alpha and numeric data. A preview of the data is available in the list at the right after pressing the *Generate Values* button. After generating values, pressing the OK button will save the values into the dataset.

Load Test Configuration Editor

The Load Test Configuration Editor is used to configure a load test. A new load test configuration is created by right-clicking on an existing load test or the Load Test folder in the Navigator View and selecting the *New Load Test Configuration* item. A new load test configuration initially displays the last used configuration or the application defaults if no load tests have been configured. To open the Load Test Configuration Editor for an existing load test configuration from the Navigator View, you may either double-click the configuration or right-click on the configuration and select the *Edit* item.

Configuring a Load Test

The Load Test Configuration Editor contains three major configuration sections: Test Duration, Virtual Users, and Testcases. While changing the configuration, if any fields contain invalid entries or any configuration errors are detected, a message is displayed immediately (shown below in red).



Test Duration

The load test duration ¹ can be in units of hours or minutes. The duration of the test should change depending on your testing goals. If you are just trying to get an idea of the speed of certain operations on your site, useful performance information can be gained for tests that are a few minutes long. You can then tweak parameters in scripts or machine configuration and see if it has an effect on performance. If, however, you are trying to stress your web site to see if anything breaks, you will want to run the test over a longer period of time.

The sample period ² is the length of time over which statistics will be sampled before saving the values. This value should be shorter for short tests, and longer for long tests. For example, if your test only lasts an hour, then having samples every 10 seconds makes sense. If, though, your test is intended to run overnight, then the sample period should be much longer, in the area of 5 minutes. This helps make the data easier to interpret. When running extended tests, large amounts of data are collected - which could cause the program to run out of memory and halt the test prematurely. As a rule of thumb: when running a test for multiple hours, you should have sample periods that are on the order of minutes, while short tests can handle sample periods as small as 5 seconds.

Virtual Users

The Load Test Configuration Editor allows you to specify the number of virtual users to simulate at the start of the test ³, and to optionally specify how many virtual users to periodically add to the test ⁴. You may also (optionally) limit the total number of virtual users ⁵. It is best to start with a low number of users and verify the correct operation of your server before performing more complicated

tests. Also note that the number of virtual users you can simulate is limited by the speed and memory of the load machine, so that the actual number of virtual users generated can be lower than the value in the *estimated* field .

Testcases

Testcases are added to the load test using the pull-down menu  located above the table listing all testcases in the load test. Select the desired testcase from the menu and click the '+' button to add the testcase to the load test. To remove a testcase from the load test, select the testcase in the table and click the '-' button.

Once a testcase has been added to the load test, the testcase can be configured by double clicking the appropriate entry in the table. The settings that can be modified are:

- **Weight:** Determines the number of users allocated to this testcase during the load test. For example, if you have two business cases set to 2 each, and the performance test starts out with 10 virtual users, 5 users will be assigned to each of the testcases. As the number of virtual users increases, they will be assigned to the testcases according to the percentages, keeping the correct ratio.
- **Speed:** Used to simulate the browser connecting to the web server over different types of network connections, from a 9.6kbps modem to a 100Mbps LAN. The parameters are in bits per second (and they include the two *stop bits* required for Modem communications).

This setting limits the amount of data the simulated user can read or write to or from the server. The result is a more accurate simulation of expected server load. Accurate simulation of network speed for each user also results in a more accurate simulation of resource usage on the server - especially open network connections. For example, if your application generates a 40K graph, the browser might spend a fraction of a second to read the graph when connecting via a LAN, but could take up to 13 seconds when the browser is connecting over a modem. Having the socket open for 13 seconds instead of a fraction of a section puts a greater burden on the server - and can significantly influence the resulting performance measurements.

- **Think Time:** There are two choices for this option, *none* and *recorded*. When *none* is chosen, the web pages in testcases are played back in sequence without waiting between pages. When *recorded* is chosen, the web pages are played back at the rate at which they were recorded. For example, if the user paused for 30 seconds between pages while recording the original testcase, the virtual user will pause at the same place for 30 seconds before replaying the next web page.

- **VU Start:** There are two choices for this option, *random* and *immediate*. When *random* is selected, virtual users do not start playing back at the same time. Instead, they are distributed over a one minute period. This option simulates a more realistic scenario - in which users start the visit to the web site in irregular intervals. When *immediate* is selected, all of the virtual users (for each incremental period) start simultaneously.
- **Delay:** A virtual user is assigned to play back a single testcase repeatedly. The delay setting is the number of seconds to delay between repeats.
- **Redirect:** All of the URLs contained in the testcase can be redirected at runtime to a different host. This enables you to run the recorded testcases against different hosts without re-recording them. If you have a testcase that contains multiple hosts, you should use the [Headers View](#) to change hosts (because this option redirects all requests within the testcase.)

If the application runs on a non-standard port, the port must be specified as well - use the syntax *host:port*. Example: *192.168.1.103:81*. Standard ports are 80 for HTTP and 443 for HTTPS.

Limits

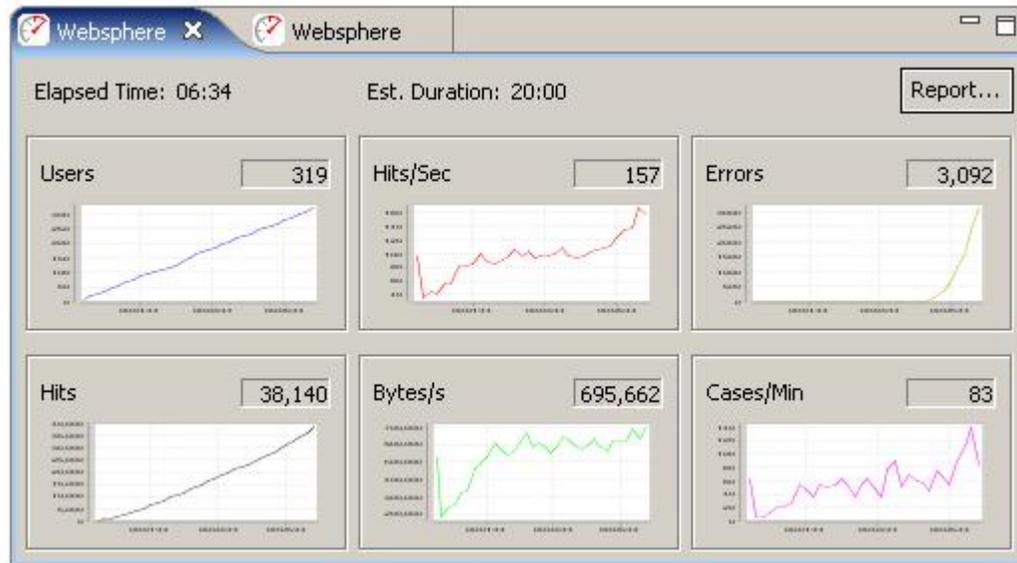
By default the starting number of virtual users is 50, and the maximum number of users to add is 50. When running larger number of virtual users generated by multiple computers these values may be low. In that case, edit the configuration file `system.properties` and modify the parameters `MaximumStartUsers` and `MaximumIncrementUsers`. The configuration file is usually installed in `C:\Documents and Settings\Username\WebPerformance`.

Running the Load Test

The load test cannot be run until the application detects there are no invalid entries or configuration errors. Once the load test configuration is valid, the *Run* button on the Load Test Configuration Editor is enabled. Selecting this button begins running the load test and opens the [Load Test Results View](#).

Load Test Results View

This view is activated when a load test is started to allow monitoring of the progress of the test while it is running. After the test is completed, the same view provides a summary of the most frequently used test parameters.



Elapsed time

While a test is running, this indicates the time elapsed since the test was started. If the test has completed, it indicates the total duration of the test.

Estimated Duration

This indicates the estimated duration of the test, based on the test configuration.

Report...

Opens the [Load Test Report](#).

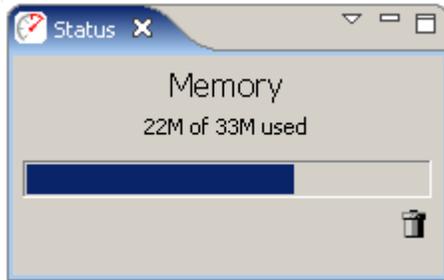
The remainder of the view displays numerical and graphical displays of 6 key performance statistics. During a test, these statistics will be updated periodically. After a test has completed, the charts will show the results for the entire test while the numbers will reflect the last sample collected.

Status View

The *Status View* provides detailed information about certain long-running operations, such as [Replaying a testcase](#). When no operation is in progress, it shows the current memory usage.

Memory status

In default mode, the memory status is displayed. The numbers displayed reflect the heap memory usage - which is the space the program has available for creating and manipulating data. It does not include the memory consumed by the application code.

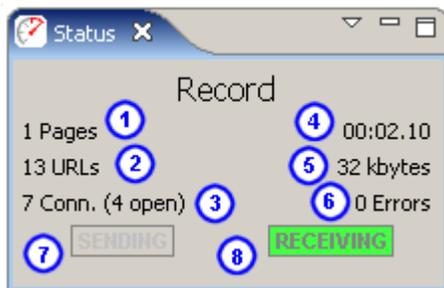


The  button in the corner runs the garbage collector to recycle memory that is no longer being used. Note that you are never required to push this button manually - the garbage collector automatically runs when needed. But some people really like pushing buttons, so we added one!

Record status

While recording, the *Status View* displays the current state of the recording:

1. number of pages recorded
2. number of URLs recorded
3. total open connections to the server
4. elapsed duration of the recording session
5. total bytes transferred (requests and responses, including HTTP headers)
6. number of network and HTTP errors encountered
7. sending status: active while a request is in progress
8. receiving status: active while a response is in progress



Shortly after a replay ends, the *Status View* will automatically return to displaying the memory status.

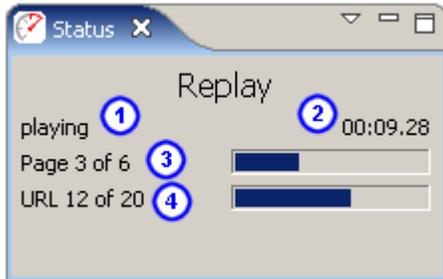
Replay status

During a replay, the *Status View* displays the current state of the replay:

1. replay status (playing, paused, thinking, stopped)

2. time (total replay time or remaining think time)
3. number of pages completed
4. number of URLs completed

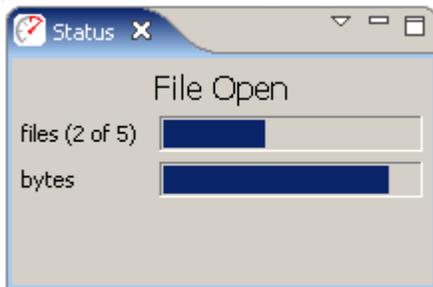
Shortly after a replay ends, the *Status View* will automatically return to displaying the memory status.



File opening status

While repository files are opening, the *Status View* will display the progress of the operation:

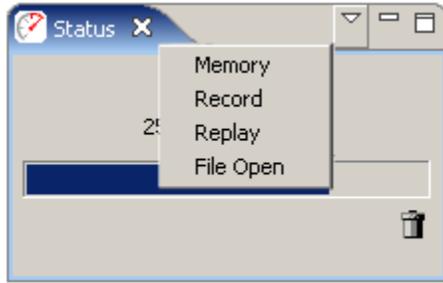
- number of files completed
- bytes read from the current file



Shortly after the files have been read, the *Status View* will automatically return to displaying the memory status.

Changing the display

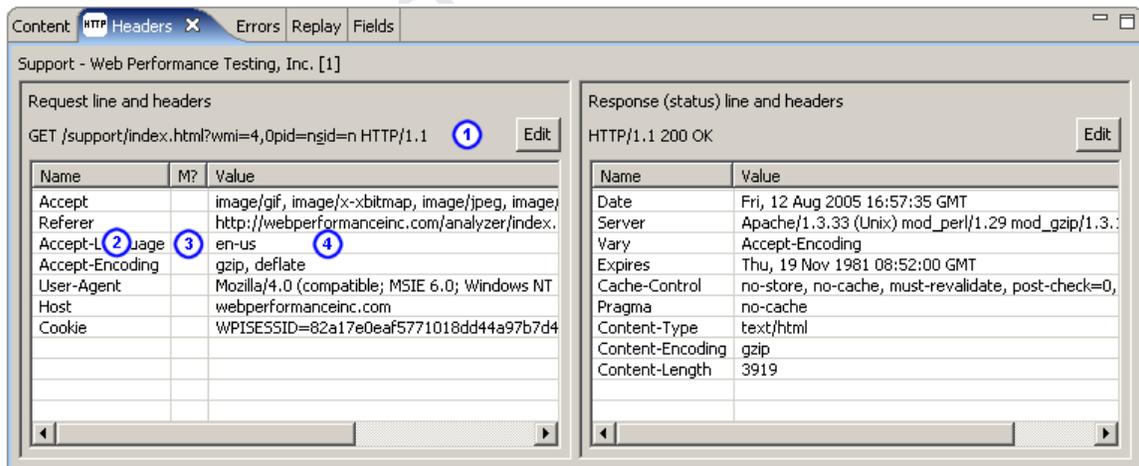
The *Status View* will typically select the best mode for displaying information relevant to the current operation. The mode may be manually selected using the drop-down menu at the top of the view..



Headers View

The Headers View displays HTTP start-line and headers for the request and response of the item currently selected in the Testcase Editor. The Headers View is opened by selecting *Window->Show View->Headers* from the main menu.

The title of the item being displayed is shown in the upper left portion of the Headers View. If any modifiers are present on the HTTP Request header fields, the icon for the field in the *Modifier* column is active.



1. request-line and corresponding *Edit* button
2. header name
3. modifier column - an icon here indicates the header has a modifier configured
4. header value

Editing the request-line (including URL parameters and path segments)

Pressing the *Edit* button (1) will open the *Edit HTTP Request-line/URL* dialog below, which allows editing of the entire request-line, including the URL path and query parameters.

1. HTTP method - GET and POST are most commonly used. Be very careful when changing the method - changes might cause errors when replaying the testcase.
2. HTTP Version
3. Entire URL path and query - Changes here will be reflected in the tables below immediately.
4. URL path elements - Each path element can be changed or configured with a

Edit HTTP Request-line/URL

Method: GET (1)

Version: HTTP/1.1 (2)

Path and Query: /support/index.html?wmi=4,0&pid=n&sid=n (3)

Path Elements

Name	Replace With
support	(4)
index.html	

Query Parameters

Name	Value	Replace With
wmi	4,0	RandomValues:shortnumber
pid	n	
sid	n	

Use:

Constant: 4,0 (6)

Dataset value: (7) DataSet: RandomValues Field: shortnumber

User variable: (8)

OK Cancel

- modifier
by
selecting
the
element
and using
the fields
at the
bottom of
the
dialog.
5. Query parameters - Each parameter can be changed or configured with a modifier by selecting the element and using the fields below. To rename a parameter, use the raw *Path and Query* field, above.
 6. Constant - Change the constant value for the selected item.
 7. Dataset value -

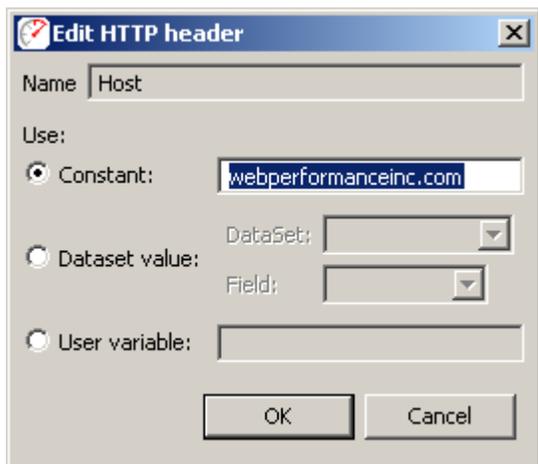
- Select a dataset and field for modification during a replay. A value from the dataset and field will be substituted for the existing value during the replay.
8. User variable - Similar to the Dataset value, above. The named value is extracted from the Virtual User's local variable store. This feature is still under development.

Editing Header Values

Any of the Request or Response header values can be changed by double-clicking in the *Value* column and typing the new value. Request headers may also be edited using the modifier configuration (below).

Configuring Modifiers on Headers

Modifiers can be added, changed, or removed from HTTP request headers using the *Edit HTTP header* dialog, which is opened by double-clicking on the modifier icon. This dialog is similar to the *Edit HTTP Request-line/URL* dialog above - see the description for fields 6-8.



Editing Status-line

The status-line in the HTTP response may also be edited by clicking the *Edit* button:



Content View

The *Content View* shows the content (body) of each HTTP message for the selected web page or URL. It has two sub-viewers for the request and response content. Each viewer has tabs for displaying different types of content. By default, they will automatically select the best viewer for the content type.

1. Content viewer mode selection buttons - these buttons control the visibility of the request content and response content viewers.
2. Title - shows the title of the selected web page or transaction

3. Content viewer lock - selecting this option disables the content type auto-selection mechanism. This allows the user to manually select which viewer to use for the selected content.
4. Hex mode - this button controls the formatting of content in the *Raw* content viewer. By default, it will display in *hex dump* format. The alternate is to dump the content as text formatted in the local character set.
5. Export buttons - these buttons can be used for export the request or response content.
6. Request viewer - displays the request content formatted for the selected content-type tab.
7. Response viewer - displays the response content formatted for the selected content-type tab

HTML/XML viewer

The HTML/XML viewers display the content in an embedded browser that renders the selected page from memory, including all images, style sheets, etc from the recording.



Text viewer

The *Text* tab displays other text resources (javascript, style sheets, etc) and the source HTML for a web page. If the response had either a Transfer-Encoding or Content-Encoding scheme applied (e.g. gzip, deflate), the text is decoded as needed.

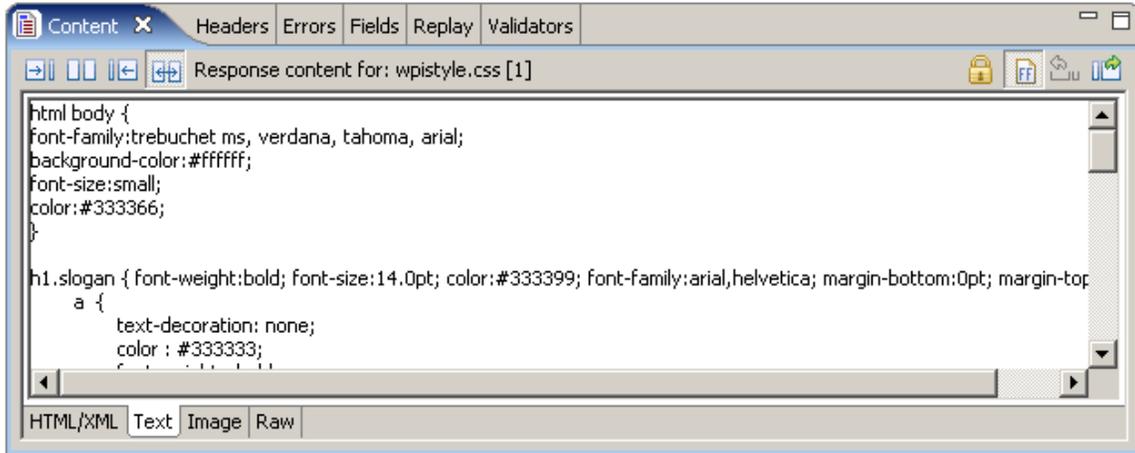
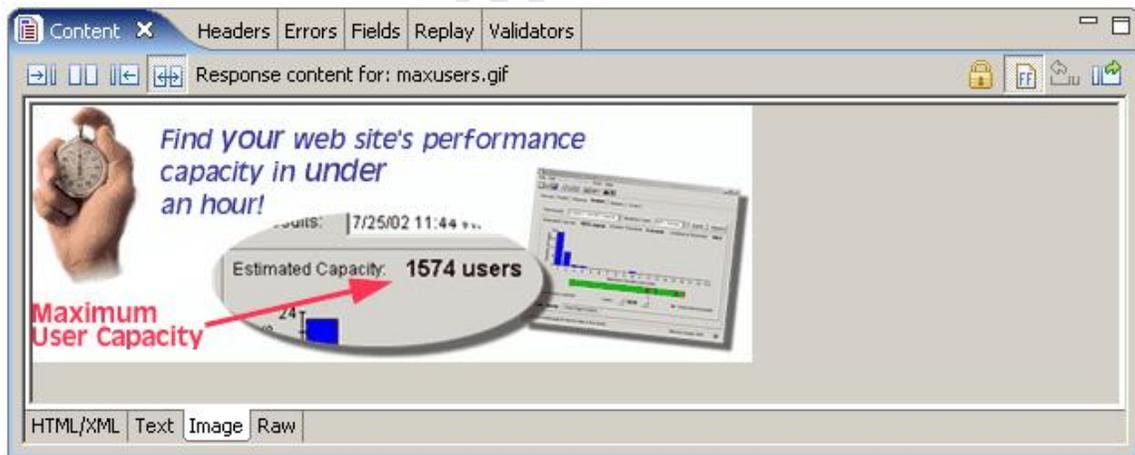


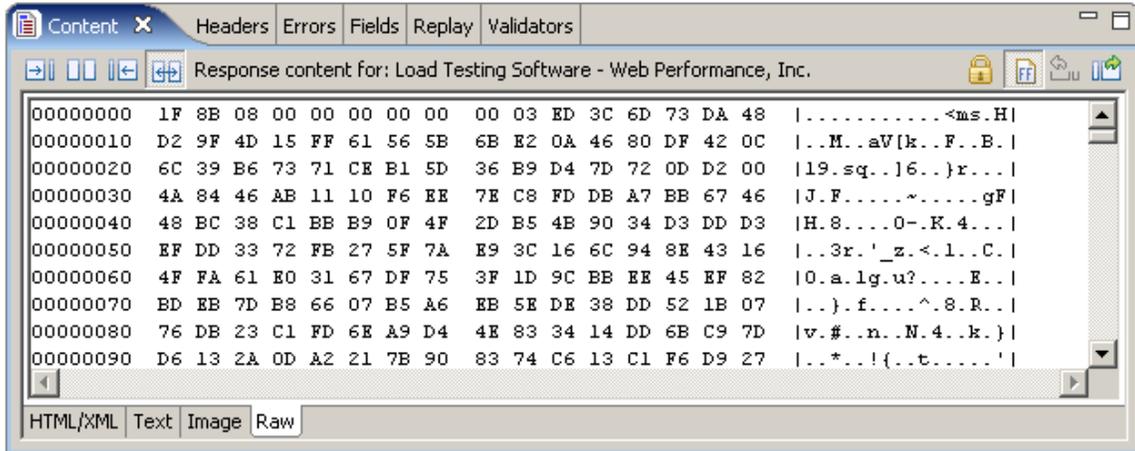
Image viewer

The image viewer displays common image formats, including PNG, GIF, JPEG, and BMP (on Windows).

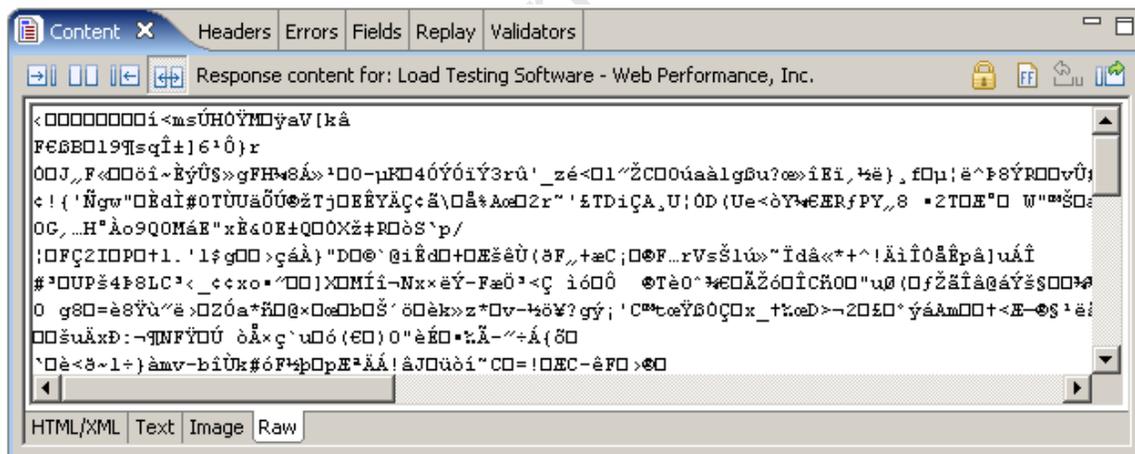


Raw viewer

The *Raw* tab displays the content exactly as it was received from the server. By default, the raw content is displayed in hex dump format:



Alternatively, the content may be displayed as text rendered from the default character set. To view this, de-select the *Hex Mode* button.



Exporting Content

The content may be exported using the *Export* buttons in the upper right corner of the view (see #5 in 1st screenshot).

Note that the content exported will be in the same format as the active view. For example, exporting a web page that was gzipped when sent by the server will export as the uncompressed HTML from either the web page view or the text view. When the raw view is active, the exported content would be in hex dump format if the *Hex View* was active or the raw compressed bytes if *Hex View* was not active.

Errors View

The Errors View displays errors found in the item selected in the Navigator or Editor. Errors can be obtained from testcases, testcase replays and load test results.

Opening the Errors View

The Errors View is opened from the menu *Window->Show View->Errors*.

Viewing Errors

All errors for a testcase are shown in the Errors View when any of the following items are selected:

- Testcase in the Navigator
- a Testcase Editor tab
- Web Page or URL in the Testcase Editor

All errors for a testcase replay are shown in the Errors View when any of the following items are selected:

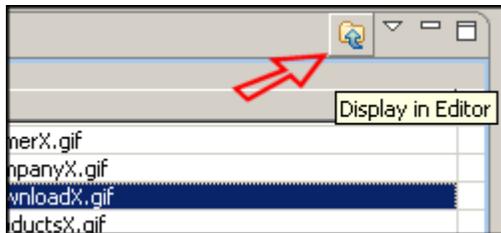
- a Testcase Editor tab, when the replay is displayed
- Web Page or URL in the Testcase Editor when the replay is displayed

All errors for a load test are shown in the Errors View when any of the following items are selected:

- Load test results in the Navigator
- a Loadtest Results Editor tab

Go to the URL for the error

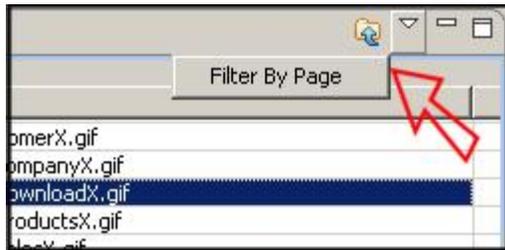
To locate the URL responsible for an error, select the error in the Errors View and press the *Display in Editor* button. For testcase errors, the corresponding URL is selected in the Testcase Editor. For loadtest errors, an icon is present in the description field if the data associated with the error was recorded, and selecting the button opens the Testcase Editor containing the URL with the error and the URL corresponding to the error is selected.



Filtering errors

When there are many errors in a Testcase, it can be helpful to only view the errors for the selected web page. This option can be enabled from the *Filter By Page* item in the Errors View menu, and is only available for testcase and replay

errors. When activated, this causes the Errors View to only show errors from the selected web page (or the page corresponding to the selected URL).



Viewing the Transaction Title and URL

The final column in the table contains the URL the error occurred at and placing the mouse over the text in that column displays the transaction title for that URL. The table can be modified to display the transaction title and show the URL for mouse over by selecting the *Show URL* or *Show Transaction Title* item in the Errors View menu.

Replay errors

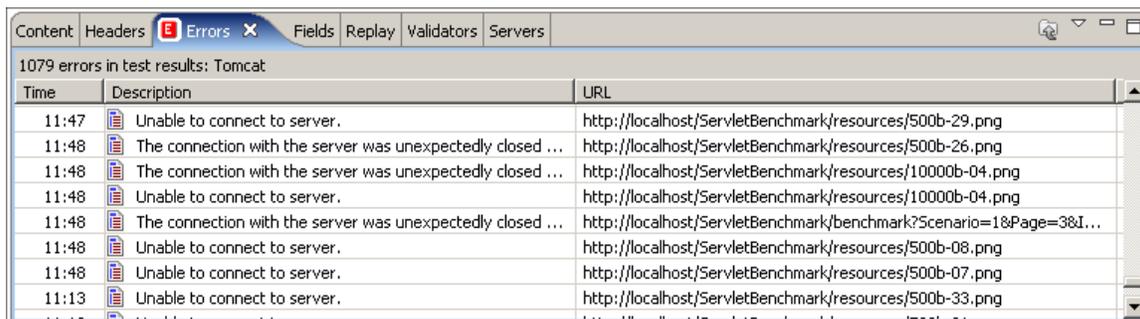
When a [replay](#) is selected, the view changes slightly to show errors encountered during the replay. During a replay, if the *Errors View* and *Testcase Editors* are active, it is dynamically updated as each page completes (except in fast-replay mode - then it is updated when the replay finishes).

Time	Description	URL
00:00:00	Reply size validation failed: the content size (20195) did not match the expected value of 20188	http://dell7/homepage404s.h
00:00:00	404 Object Not Found	http://dell7/graphics/buttons,
00:00:00	404 Object Not Found	http://dell7/graphics/buttons,
00:00:00	404 Object Not Found	http://dell7/graphics/buttons,
00:00:00	404 Object Not Found	http://dell7/graphics/buttons,
00:00:00	404 Object Not Found	http://dell7/graphics/buttons,
00:00:00	404 Object Not Found	http://dell7/graphics/buttons,
00:00:00	404 Object Not Found	http://dell7/graphics/buttons,
00:00:10	Content validation failed: the content (abcdefg) was not found on the page.	http://dell7/company.html

Loadtest errors

When a loadtest is running, the errors view is continually updated with any errors that occur during execution of the test. If the data associated with the error was recorded, an icon is present in the description field. Presence of the icon indicates that it is possible to go directly to the Testcase Editor to view the URL

corresponding to the error by selecting error in the table, then selecting the *Display in Editor* button.



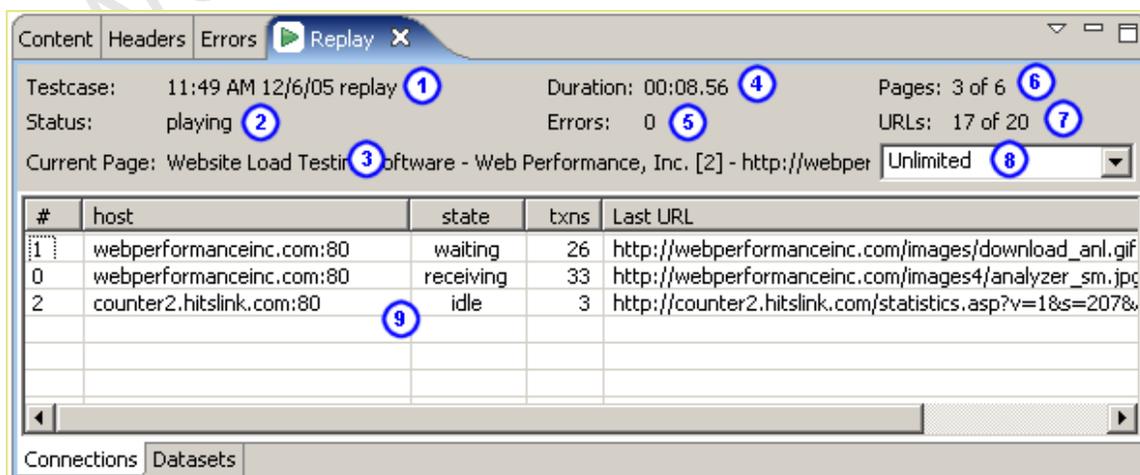
Replay View

The replay view allows you to monitor the status of a replay as it is performed. To open the replay view, select the *Window->Show View->Replay* selection from the main menu.

For details on performing replays, see the [Replaying](#) manual page.

note: By default, the *Replay View* is placed in the same window pane as the *Content View*. In order to see the pages as they complete and view the replay status information at the same time, it may be useful to move the *Replay View* to another window pane (see the [Navigating the UI](#) section for details).

Replay View Fields



1. name of the replay.
2. current activity: paused, stopped, playing, or thinking
3. title and URL of the current page

4. running time of the total replay duration (including think time)
5. total number of errors encountered during the replay
6. number of the current page and the total number of pages in the testcase
7. number of the current URL (on the page) and the total number of URLs in the current page
8. bandwidth limitations applied to the replay
9. additional info, depending on the selected tab

Connections tab

#	host	state	txns	Last URL
1	webperformanceinc.com:80	waiting	26	http://webperformanceinc.com/images/download_anl.gif
0	webperformanceinc.com:80	receiving	33	http://webperformanceinc.com/images4/analyzer_sm.jpg
2	counter2.hitslink.com:80	idle	3	http://counter2.hitslink.com/statistics.asp?v=1&s=207&
①	②	③	④	⑤

Connections Datasets

This table shows details about each connection established during the replay.

1. Connection number (starting at 0)
2. host name
3. connection state
4. number of transactions performed on the connection
5. current URL being processed

Datasets tab

This shows each of the dataset rows that is currently in use by the Virtual User (VU) during the replay.



In the above example, the Virtual User is using one dataset - *Users*. The dataset has two fields, *password* and *username*, and the currently selected row has values "123" and "dave" for those fields.

Menu actions

These actions are available from the Replay View menu:



Remember dataset position

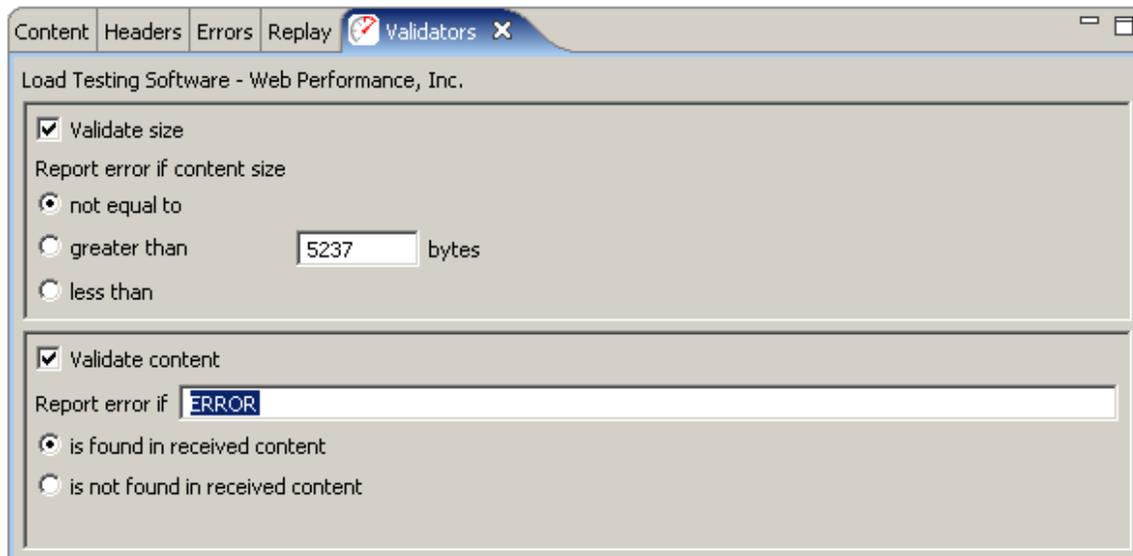
When a VU replays a testcase and it has modifiers configured to pull values from a dataset, the position of the row in the dataset is automatically advanced when the row is returned. This allows a testcase to be replayed multiple times with different data. This setting is on by default. Turning it off will cause the VU to start at the beginning of each dataset when the replay begins.

Reset dataset state

This action forces the VU to reset the next position of each dataset to the beginning. Rows currently in use are not affected.

Validators View

The *Validators View* shows the validators applied to an HTTP Transaction or Web Page. Validators can be configured to check the content size and/or a specific string within the content. The Validators View is opened by selecting *Window->Show View->Validators* from the main menu.



To view or configure validators for a transaction, open the Validator View and then select the transaction in the Testcase Editor. Any existing validators are displayed in the Validators View.

Note: changes in this view are made effective immediately.

Size validation

Select the *Validate Size* check box in the Validators View. The default value of the size field is the size of the recorded content. Change the size if desired, then select one of the three size validation options:

- *not equal to*
- *greater than*
- *less than.*

To disable size validation, de-select the *Validate Size* check box.

Content validation

Select the *Validate Content* check box in the Validators View. Enter the content to search for in the text area, then select either:

- *is found in received content*
- *is not found in received content.*

To disable content validation, de-select the *Validate Content* check box.

Fields View

The Fields View displays the form fields and URL query parameters found in the item selected in the Navigator or Testcase Editor. The Fields View can be opened from the menu *Window->Show View->Fields*.

Name	Type	#	M.	Value(s)	Transaction Title
failure	mixed	2		webperformance, http://webperformanceinc....	
failure	query	1		webperformance	statistics.asp [2]
failure	form	1		http://webperformanceinc.com//registererror....	<text>
gl	query	1		US	<image>
hl	query	1		en	<image>
je	query	4		true, Dataset: Users:Username	
je	query	1		true	statistics.asp [1]
je	query	1		Dataset: Users:Username	statistics.asp [2]
je	query	1		true	statistics.asp [3]
je	query	1		true	statistics.asp [4]
l	query	4		Dataset: Users:Username	
mkt	query	1		0	cc.gif
orderAmount	query	2		12, 0	

1. Name of the field
2. Type - URL query parameter (*query*) or form field (*field*). On parent nodes with more than one type of child, *mixed* will be displayed
3. Number of usages - how many times is this field used
4. Modifier applied? - if a modifier is applied to dynamically change this value during a replay, an icon is shown here. On parent nodes with children that both have and have no modifier applied, a *grey* version of the icon is displayed. Double-clicking this column invokes the Field Editor Dialog - the value can be changed and/or a modifier configured for this field.
5. Values of the field - If a field has multiple values, they are displayed separated by commas (,) and the tooltip will show multiple lines with one value per line. Note that only unique values are displayed (i.e. multiples of the same value are only displayed once).
6. Title / URL - Displays the title of the transaction this field is used on, with the URL in a tooltip. This may be reversed (show URL with title in tooltip) via the view menu (see Show URL, below).
7. Go to transaction button - when a field is selected in the table, press this button to show the related transaction in the testcase editor.

8. Edit button - use this button when one or more fields are selected to invoke the Field Editor Dialog.

Scope of fields displayed

The fields displayed depends on the item selected:

- All fields in the testcase are displayed if a Testcase is selected (in the Navigator or a Testcase Editor is selected but nothing is selected within the editor)
- The fields for all transactions in the page are displayed when a Web Page is selected in a Testcase Editor
- The fields for a single transaction are displayed when a single Transaction is selected in a Testcase Editor

Go to Transaction

To locate the Transaction containing a field, select the field in the Fields View and press the *Display in Editor* button. The corresponding Transaction is then selected in the Testcase Editor.

Editing fields

To change the constant value or configure modifiers on fields, the Field Edit Dialog can be opened by either:

- single field: double clicking on the Modifier column in the Fields View
- multiple fields: selecting the fields in the table and pressing the *Edit* button on the upper right portion of the Fields View

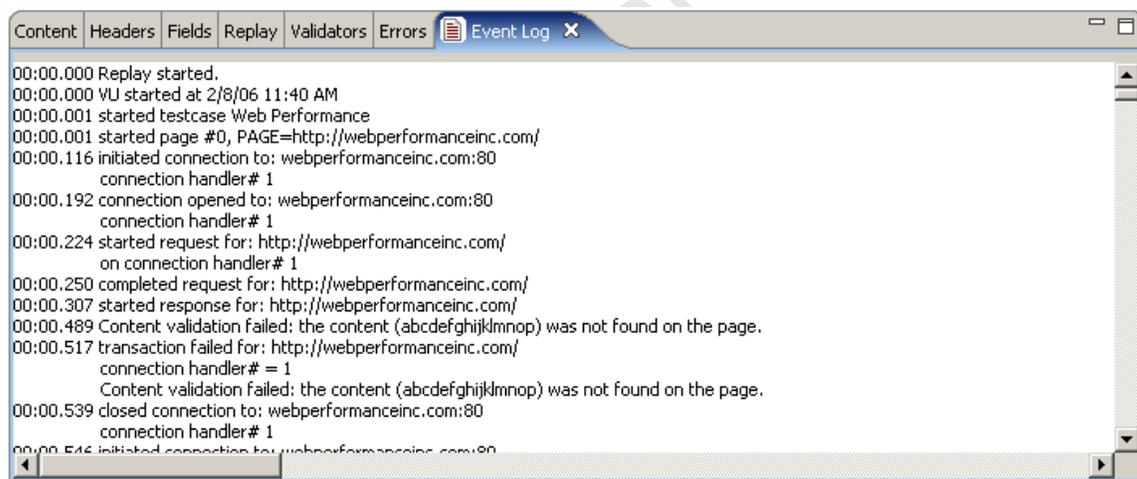
If multiple fields are selected, they will all be changed by this dialog. If they have different settings when the dialog is opened, the *Constant/Dataset/User* buttons will be initially de-selected - one must be chosen before editing can take place. For more information on using modifiers to customize a testcase, see the [Customizing a Testcase](#) section.

Show URL

Selecting the *Show URL* option from the Fields View menu will toggle the display of titles / URLs in the last column of the Fields View. When the URL is being displayed, the hover text for the entries in the URL column displays the transaction title. To revert to viewing the transaction titles in the table, select the *Show Transaction Title* option from the Fields View menu.

Event Log

The Event Log view provides a detailed log of connection and transaction events during a testcase [replay](#). The view can be opened from menu by choosing *Window->Show View->Other->Web Performance* and selecting the Event Log.

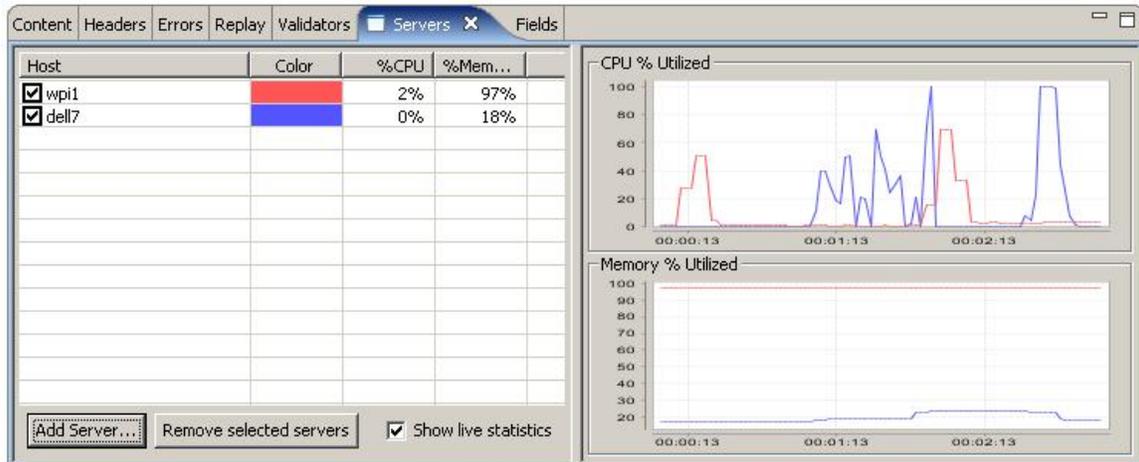


In order to activate the replay logging feature, the view must be activated before starting the replay. This means that it must have been opened at some point prior to starting the replay (it does not need to be visible at that time the replay starts). After the replay is complete, the log may be viewed at any future time by selecting the Event Log view and then selecting the replay in the testcase editor.

The log may be exported by clicking in the log area, copying the content (using Ctrl-A (select all) and Ctrl-C (copy)) and then pasting the log into the desired application.

Servers View

The Web Performance line of products are capable of monitoring two important statistics on your server(s): CPU utilization (%) and memory usage (%). You may make the appropriate configurations to configure these statistics through the Servers View.



When you open the Servers View, you will be presented with a list of servers which are presently configured for monitoring. The graph on the right side of the view displays information being actively observed from your server. The check box next to the host name may be used to toggle whether or not the particular server is being actively monitored.

To start monitoring a new server, simply press the *Add Server...* button.



Here, you must decide what style of server monitoring to use. Three styles are supported:

1. Windows to Windows - uses a built-in direct protocol to monitor the remote server (available only if both your server and the load testing workstation are running Windows).
2. Java-based server on Unix - Web Performance provides a WAR file containing a servlet that will provide the necessary server statistics. The provided WAR is supported on Solaris and Linux.
3. Custom monitoring (server script) - For unsupported platforms, a custom script may return the necessary server statistics in readable format (see script requirements later in this chapter).

For more information on setting up your server to be monitored, please see the [Server Configuration](#) section below.

Once a configuration has been selected, and the required fields filled in, pressing the "OK" button will attempt to connect to the server and verify that the server can be successfully monitored under normal conditions. Upon verification, the server will be added to the list of servers for statistics gathering during a load test.

Viewing Previous Results

The Servers View can also be used to display server statistics for a given series of load test results. In order to do this, simply uncheck the option to *Show live statistics*, and open a set of results. The view will update to display previously gathered information from the selected test. In order to switch back to the active configuration, simply re-check the *Show live statistics* option.

Server Configuration

Direct Windows monitoring

No server-side installation is necessary. However, the user running Web Performance Suite must have the appropriate Windows privileges to monitor the server. See your network administrator for details.

Note that you *must* log into the Windows Server from the client machine prior to beginning the test. The windows Direct monitoring is dependent on the Windows authentication established prior to the start of monitoring. For example, browsing a network share on the server that requires authentication will usually establish the required permissions.

The direct windows monitoring is the equivalent of using the Windows Performance Monitor (perfmon.exe) to monitor the % Committed Bytes In Use counter of the Memory performance object on the remote server.

UNIX server (with Java-based application server)

Install the WPIMonitor.war file in your server in the usual deployment folder. It will install a servlet named Monitor in the path /WPIMonitor/monitor.

If necessary, you may modify the deployment descriptor for the servlet as necessary for your environment. However, if you change the path required to access the monitoring servlet, then you must configure the monitoring within Web Performance Suite as a custom script installation and provide the full URL to the Monitor servlet.

Custom monitoring (server script)

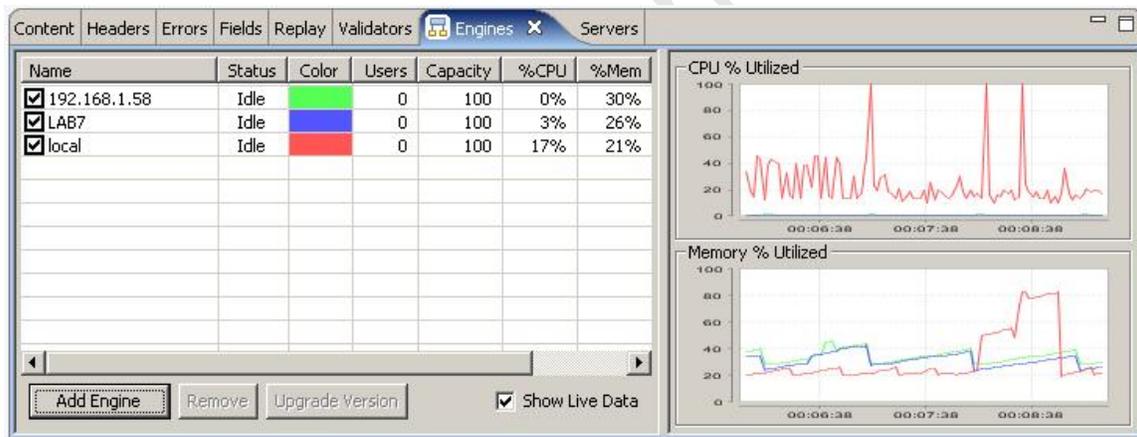
Web Performance can monitor any server via HTTP if a customized script is developed to return the server CPU% and memory% in the supported format. The following plain text content format is supported (MIME type text/plain):

```
version=1
CPU%=nnn
memory%=nnn
```

After writing and testing your script, enter the URL of the script into the URL field of the configuration dialog.

Engines View

The Engines View provides configuration options for each engine that will be used during a load test. In order to connect to a load engine, you will need a test computer with the load engine software installed. For more information, please consult the [Load Engines](#) section.



For active engines ready to or actively participating in a load test, the following information is displayed:

Name

The name of the computer that the engines is running on. If the value is *local*, this indicates the load engine embedded in the current computer is being used. This is not recommended if load engines are also being used.

Status

Indicates the state of the load engine. Possible values are "Offline", "Idle", "Running" and "

Color

The color used to represent this engine in the graphs on the right side of the view.

Active Users

The number of virtual users actually running.

Estimated Users

The total number of virtual users that it appears the machine could generate. Note that the estimated number usually is inaccurate at lower load averages, so that your computer very well may be able to generate a larger number of virtual users. This is because at low load averages the estimation is not as accurate as at a high load average. Also, the response of many computers is nonlinear, so that the load

average could hover at 20%, for example, and stay there while the number of virtual users climbs.

% CPU

The CPU utilization of the engine's computer, where 100% has all of the machine cycles being used. Note that on UNIX this value is greatly affected by background processes that are blocked, so even though a process isn't taking up any CPU time, if it is stuck in a disk wait or otherwise hung your load average will be higher. Use "ps", "top" or other programs to find and stop background processes that may be increasing the system load so that the full power of the computer is available for the performance test. Note that there is lag in getting the information from the operating system, so the value will not be exactly the same as the value displayed by other utilities.

% Memory

This measures how much of the memory allocated for internal buffers is actually in use. This number has no relation to any operating system specific information you might view using an OS utility such as Windows Task Manager. This value will go up and down during the performance test and could occasionally reach the *low* or *out of memory* values. It will slowly creep up towards the 80% mark when using large numbers of virtual users or when running the performance test for a long period of time. When the program absolutely has run out of memory you will see this value quickly climb into the 90% range every 30 seconds or so. When this happens, no more users will be added to the engine to prevent the program from running out of all memory and causing corrupt test results.

Version

The current version of the Load Engine software running on the engine.

Adding Engines

Many times, load engines are automatically detected and displayed in this view when the application is started, or when the load engine software is started. However, in some circumstances the engine may not be added automatically, and you will want explicitly connect to an engine. Alternatively, you may want the local controller to also participate in the load test, in addition to it's role of collecting statistics and distributing users. For either of these options you may select the *Add Engine* button.

In the Add Engine dialog, you may select to add the local controller as a load engine (only if it is not already selected as an engine), or select a remote engine. For remote engines, you may enter either the host name or the IP address for the engine. The port number used for engine communication defaults to 1099, and should only be changed if the remote engine has been explicitly configured to use another port. Once the *OK* button is pressed, Load Tester will attempt to connect to the engine, and verify that the engine is suitable for testing before adding it.

Selecting Engines

To select an engine to use during a load test (or to monitor live), select the checkbox next to the engine name. To disable an engine - un-check. Only those engines which are checked will be used during a load test. When selected, Load tester will attempt to verify that the engine is available for testing, and enable it if so.

Engines may also be removed entirely from the engine list by pressing the *Remove* button. Once an engine has been removed, it may be added back automatically if it or the controller is restarted.

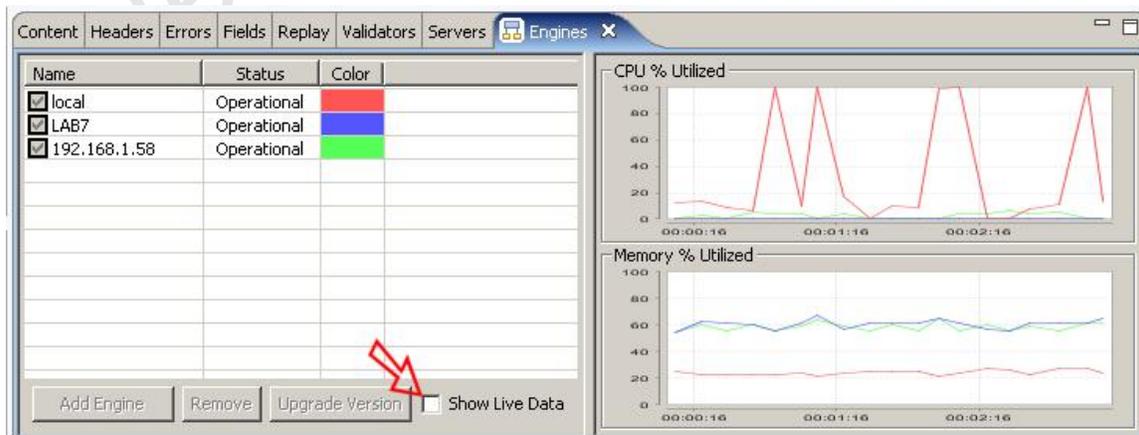
Upgrading Engines

If the engine is using a different software version, it cannot be enabled. If the software version of the engine is lower, selecting the engine will enable the *Upgrade* button. Press it to upgrade the software on the engine. Once the upgrade process has completed, the remote engine software will need to be restarted in order for the upgrade to take affect.

Viewing Historical Statistics

After a test has completed, selecting the test results will display the saved engine statistics from the selected test. Only the relevant columns will be displayed for historical data - which means none of the columns which show "live" data will be displayed.

Note that the live data option must be de-selected to view historical results.



Settings

General Settings

Recording

Page grouping

During recording, Analyzer uses several algorithms to determine what requests should be grouped into a page.

- *Referrer-analysis* uses the *Referrer* HTTP header in requests (along with content-type and status codes) to determine which transactions are web pages and which resources should be placed in which web pages.
- *Activity-monitoring* uses timing to group HTTP transactions into pages/groups. This analysis mode usually generates more logical groups when an application uses asynchronous methods to generate HTTP requests (e.g. AJAX-style applications). The *activity threshold* determines how much inactive time must elapse before a web page is considered "done". This algorithm also uses referrer-analysis where appropriate.

Recorder Ports & Browser Launching

Normally, the ports used by the Recorders internal proxy are selected automatically to avoid conflicts. However under some conditions, such as using browsers that cannot be configured automatically (custom browsers), it may be useful to set the ports manually.

The *Manually select recording ports* option is used to configure the Recorder to use the same ports every time it runs. If either of the port fields contains invalid values or unavailable ports, a warning message is displayed and the option cannot be enabled until valid ports are configured. For more information on manual browser configuration, see the [Manual Browser Configuration FAQ](#) page.

The *Launch default browser when recording* option controls the launching of the default browser when a new recording is started. Turning it off will cause no browser to be launched when a recording is started. This can be useful when an already-running browser is to be used for recording.

Dialogs

The *Start recording* dialog is displayed when a recording is started to set the testcase name and select the repository for the testcase. If this item is selected, the application does not prompt for this information, but generates a testcase name and selects the last logical repository for the testcase.

Replay

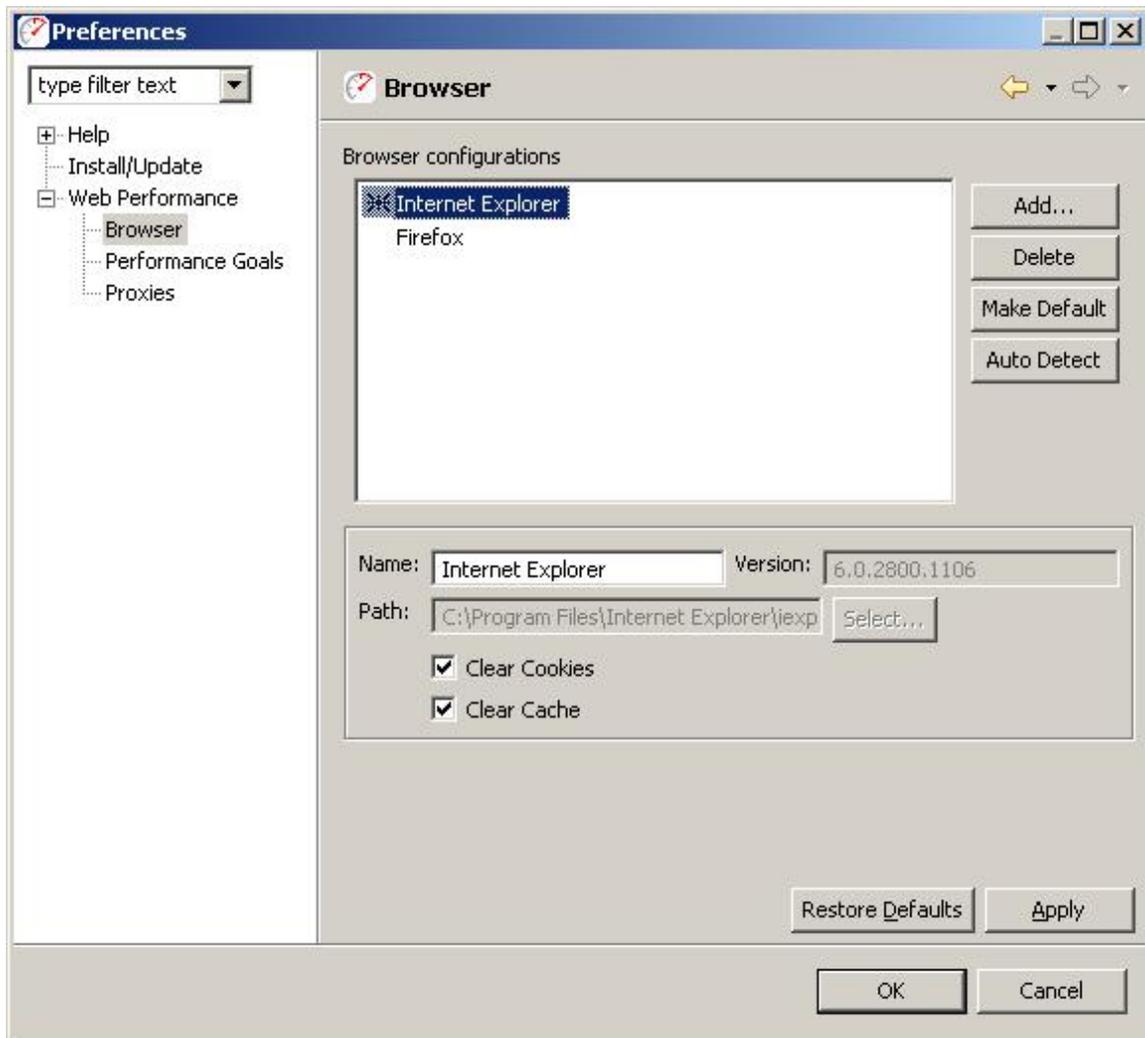
- Network errors can automatically be pruned from the testcase by selecting the *Remove network errors before replay* item. The replay is not allowed to continue until the network errors are either automatically or manually

removed. Turning this setting on allows the application to remove the network errors without displaying a warning dialog.

- Replays containing NTLM or Basic authentication should have the user identity configured before attempting to replay the testcase. In some scenarios, this behavior may need to be overridden. Changing the *Allow replay without authentication configured* item allows a replay to proceed even if the authentication is not configured.
- The Content View displays the Web Pages as they are completed during a replay, and the View is made visible in order to show the content. To prevent the Content View from automatically becoming visible during replay, change the *Activate content view during replays* item.

Browser Settings

The *Browser Settings* page is located in the *Web Performance* section of the *Preferences* dialog (*Window->Preferences* menu item). The preferred browser is normally chosen in the *Recording Configuration Wizard* when the first recording is performed. However, this setting can be customized and unsupported browsers configured in the *Browser Settings* page.



Default browser

The default browser is indicated by a mark (*) next to the browser name in the list. To select a different browser for recording, select the browser in the list and press the *Make Default* button. This browser will be configured and launched automatically during the [recording](#) process.

Restoring the auto detected browsers

To restore the auto detected browser information or to detect a recently installed browser, press the *Auto Detect* button.

Adding a custom browser

To add an browser that is not automatically supported, select the *Add* button to the right of the list of configurations. Enter a valid name and executable in the

lower right portion of the page. To save the new configuration, select the *Apply* button.

Note: Automatic configuration of the proxy settings are only provided for the auto-detected browsers. For custom browsers, the proxy configuration must be performed manually prior to recording and the recording ports must be specified on the *Web Performance Preference Page*. For more information on configuring a custom browser, see the [Manual Browser Configuration FAQ](#) page.

Modifying an existing browser

To change an existing browser, select the browser in the list. The lower right portion of the page displays the editable information. Make the changes as needed and select the *Apply* button. At any time before *Apply* is selected, the original information can be restored by selecting the *Restore Defaults* button.

Cache and Cookies options

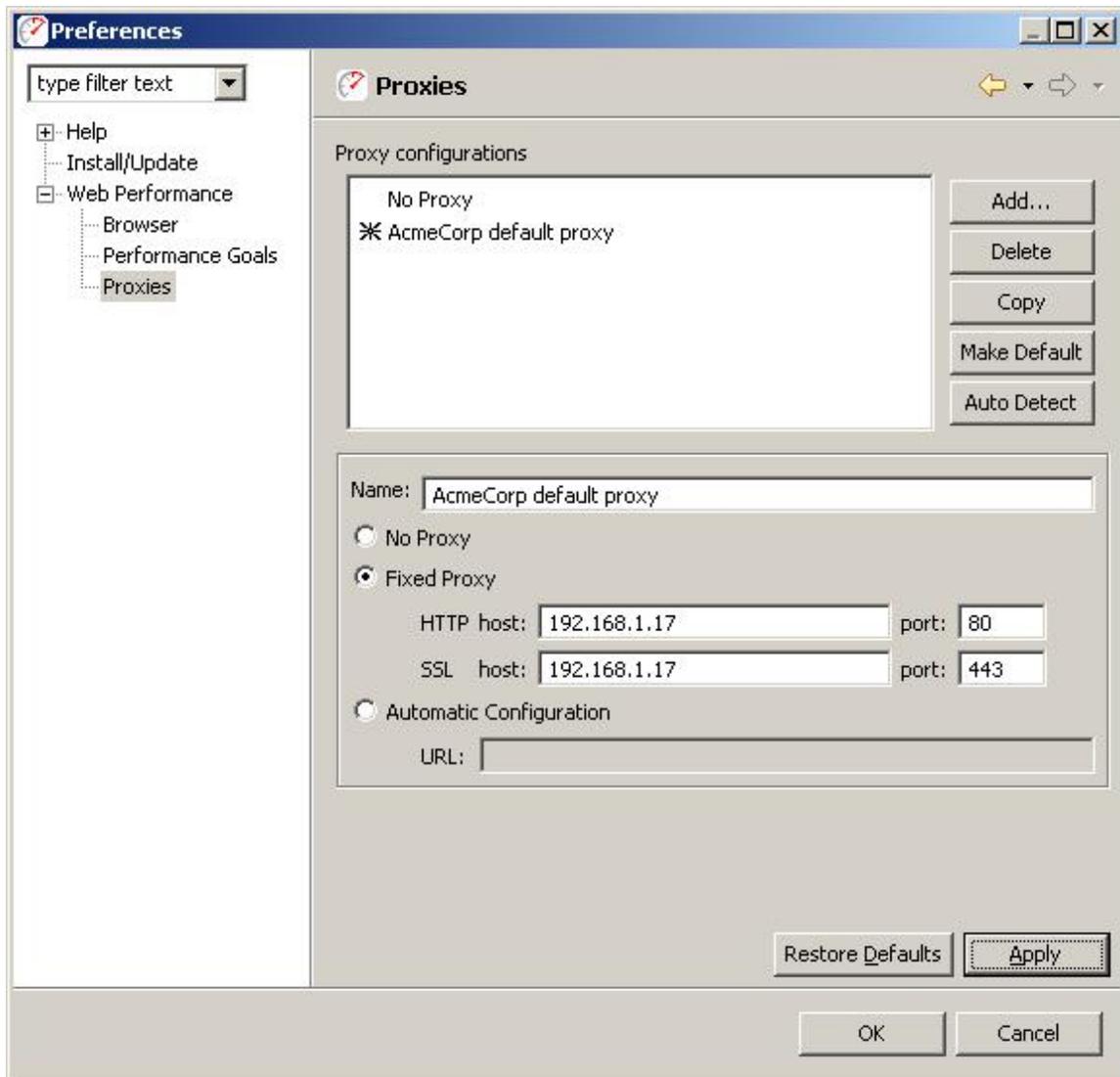
By default, the browser's cookies and cache are cleared prior to initiating recording and restored shortly afterwards. If this is not done, the recording would reflect the state of the browser's cache. For example, cached pages or images might never be requested from the server - which would result in a recording that did not reflect the true content of a web page.

Deleting a configuration

To delete a configuration, select the configuration in the list and select the *Delete* button. Note that at least one browser must be selected as the default browser and it may not be deleted.

Proxy Settings

The *Proxy Settings* page is located in the *Web Performance* section of the *Preferences* dialog (*Window->Preferences* menu item). Every attempt to automatically detected the proxy settings is made by the *Recording Configuration Wizard* when the first recording is performed. If this step fails or an alternate proxy configuration is desired, it can be customized in the *Proxy Settings* page.



Default proxy

The default proxy is indicated by a mark (*) next to the proxy name in the list. To select a different proxy for recording, select the proxy in the list and press the *Make Default* button. This proxy will be automatically configured for the browser when a recording is launched.

Restoring the auto detected proxy settings

To restore the auto detected proxy information or to detect changed proxy settings (for the computer or network), press the *Auto Detect* button.

Adding a new proxy

To add a new proxy, press the *Add* button to the right of the list of proxies. Enter a valid name and proxy information in the lower right portion of the page. To save the new proxy, press the *Apply* button.

To copy an existing proxy setting, select the proxy in the list and press the *Copy* button. To save the copy, press the *Apply* button

Modifying an existing proxy

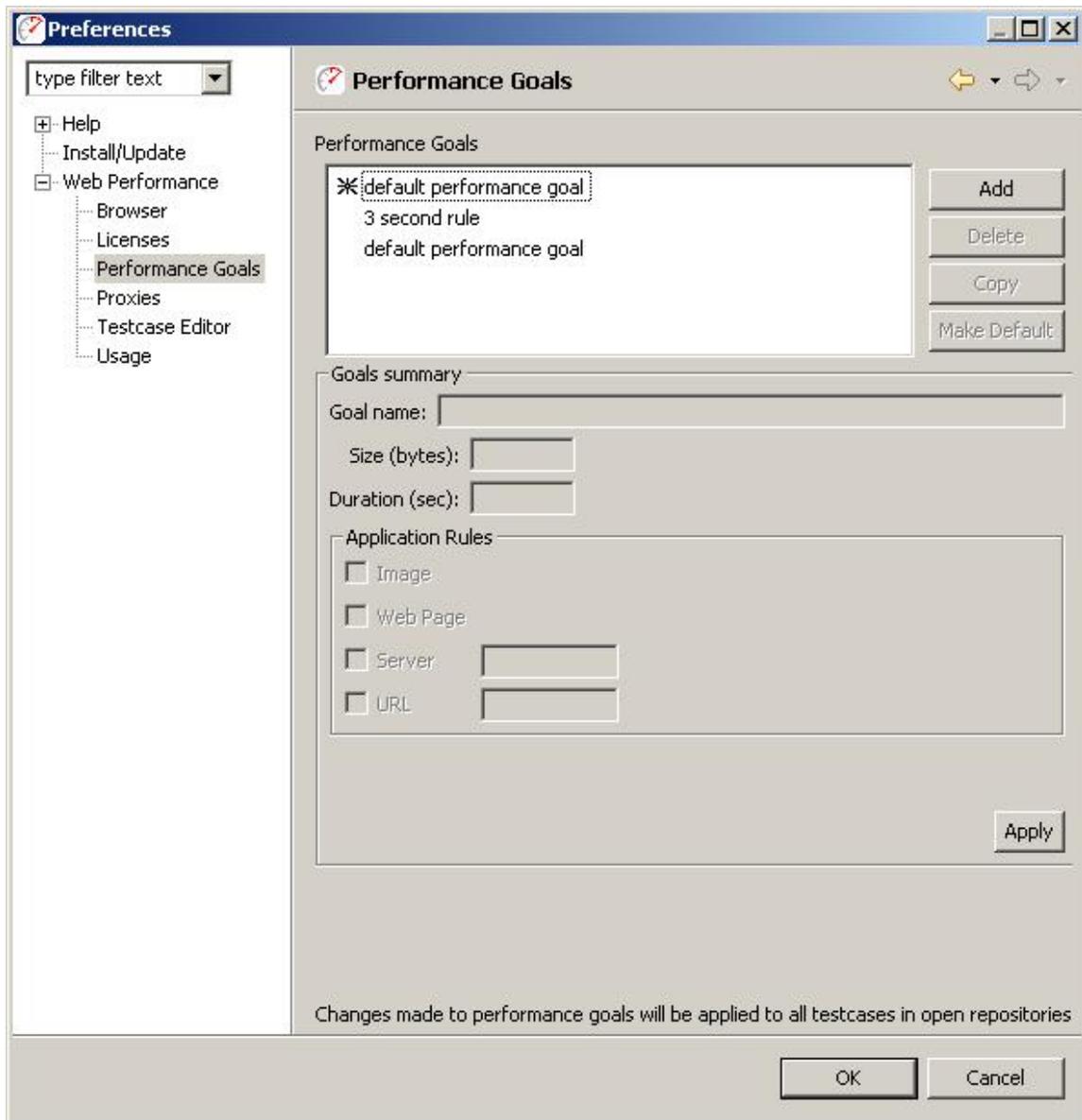
To change an existing proxy, select the proxy in the list. The lower right portion of the page displays the editable information. Make the changes as needed and press the *Apply* button. At any time before *Apply* is selected, the original information can be restored by press the *Restore Defaults* button.

Deleting a proxy

To delete a proxy, select the proxy in the list and press the *Delete* button. Note that at least one proxy setting must be selected as the default and it may not be deleted. If no proxy should used, select the *No Proxy* setting.

Global Performance Goals

The *Performance Goal* preference page allows changes to the global list of [performance goals](#) known to the application. The *Performance Goal* page is located in the *Web Performance* section of the *Preferences* dialog (*Window->Preferences* menu item).



A default performance goal is created by the *Recording Configuration Wizard*, if a default maximum page load time is provided during the wizard. This goal is named "default performance goal". Other goals may be created, edited, deleted or copied from the default using the buttons at the right and the *Apply* button.

Creating a performance goal

To create a new performance goal, select the *New* button. Enter a name for the goal and the appropriate values for size and/or duration (at least one of size or duration must be entered). Choose the application rules for the goal using the checkboxes in the *Application Rules* area. When completed, select the *Apply* button to save the new performance goal.

Default goals

Once a goal has been created, it can be set as a default goal by selecting the goal in the list and selecting the *Default* button. Default goals apply to all testcases in open repositories. Default goals will also be applied to all new recordings.

Application rules

The size or duration settings can be evaluated for any of the following:

- Image: a transaction that contains an image
- Web Page: a web page
- Server: a transaction or web page containing a specific server
- URL: a transaction or web page containing a specific URL

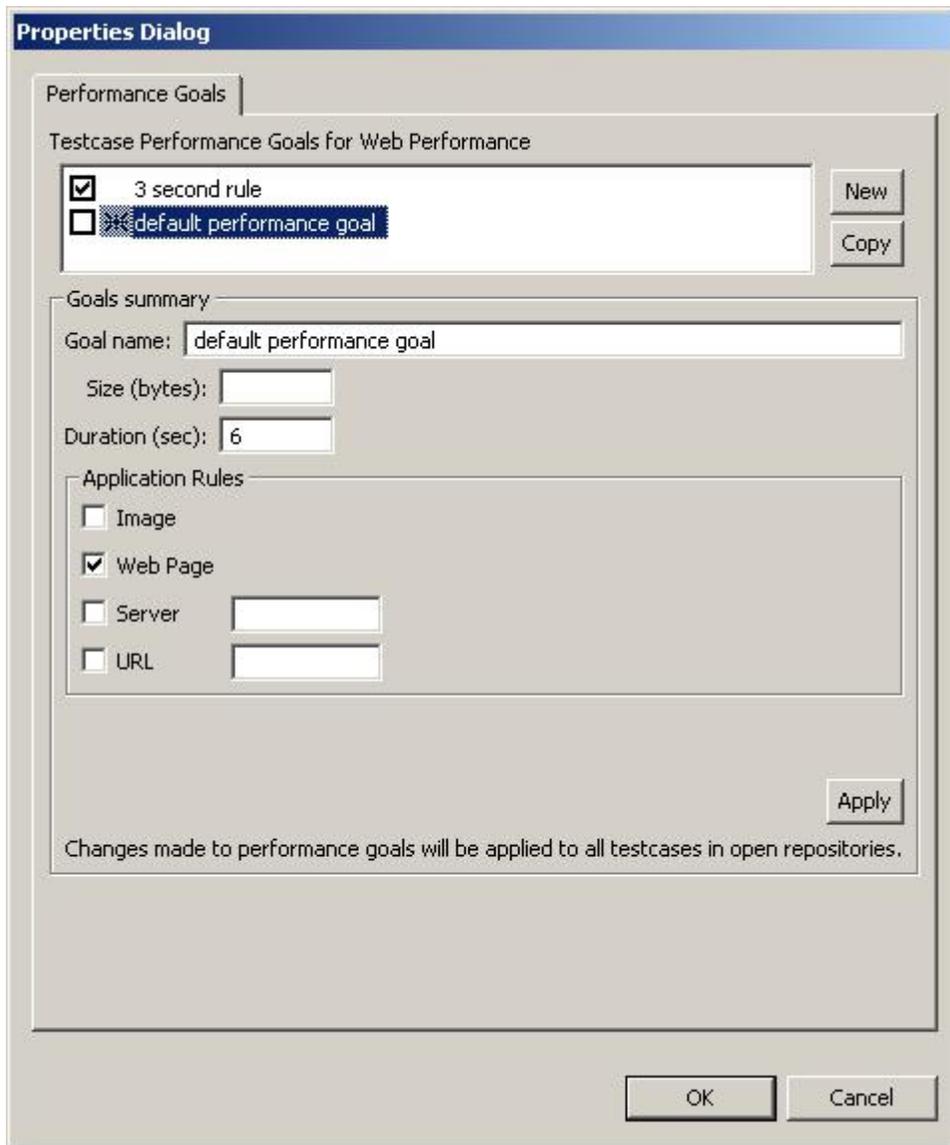
Notes

- Modifications to a performance goal are applied to all testcases containing that goal in any open repositories.
- When a repository is opened, any performance goals contained within testcases in that repository are added to the performance goals list and are usable on any testcase in the application.

For information on activating performance goals for a specific testcase, see the [Testcase Performance Goal Settings](#) section.

Testcase Performance Goals

The Testcase Performance Goal Settings Page allows changes to the performance goals applied to a testcase. The page is opened by either right clicking a testcase in the Navigator panel and selecting the *Properties* menu item or selecting a testcase in the Navigator panel and choosing the *Edit->Properties* menu item.



Creating a performance goal

To create a new performance goal, select the *Add* button. Enter a name for the goal and the appropriate values for size and/or duration (at least one of size or duration must be entered). Choose the application rules for the goal using the checkboxes in the *Application Rules* area. When completed, select the *Apply* button to save the new performance goal.

Modifying a performance goal

In the goal list, select the performance goal to be modified. Change the fields as needed and select the *Apply* button to save the modifications.

Copying a performance goal

In the goal list, select the performance goal to be copied. Select the *Copy* button to create a temporary copy of the selected goal. Make any changes to the size, duration, or application rules. To save the new goal, select the *Apply* button.

Applying a performance goal to the testcase

The performance goal is not evaluated against the testcase unless it is defined as a default (indicated by an asterisk in the list) or the checkbox in the list is selected. To add or remove a non-default performance goal from the testcase, select the checkbox and then select the *Apply* button.

Application rules

The size or duration settings can be evaluated for any of the following:

- Image: a transaction that contains an image
- Web Page: a web page
- Server: a transaction or web page containing a specific server
- URL: a transaction or web page containing a specific URL

Notes

Modifications to a performance goal are applied to all testcases containing that goal in any open repositories.

For information on deleting performance goals or creating default goals, see the [Performance Preferences](#) section.

License Management

Advanced features of Web Performance products are disabled until a license key has been installed. Evaluation license keys are available from the website - <http://webperformance.com>

ATTENTION: The license keys are encrypted binary files. Opening them with a text editor, or any program other than Web Performance products, will likely result in the display of some meaningless garbage. Calling us to complain will not change that. Please follow the directions for *Importing* license keys.

Managing license keys

To manage the installed licenses for Web Performance products, Select *Preferences* from the *Window* menu. Then select *Web Performance* and *Licenses* in the tree. Selecting an entry in the list displays the details of the license below.

Importing

License keys for Web Performance products usually arrive as an e-mail attachment. Detach the key and save it somewhere, such as your desktop. Then select the *Import* button in the license manager and provide the location of the key. After validating the license key, it appears in the list.

If the key cannot be imported because it is corrupted, it may be because the e-mail program treated the attachment as a text file rather than a binary file. Please consult your e-mail program's manual for instructions on detaching the file as a binary file.

Expired licenses

Expired licenses are indicated with (*expired*) in the license descriptions. You may safely delete these license keys with no effect on the program.

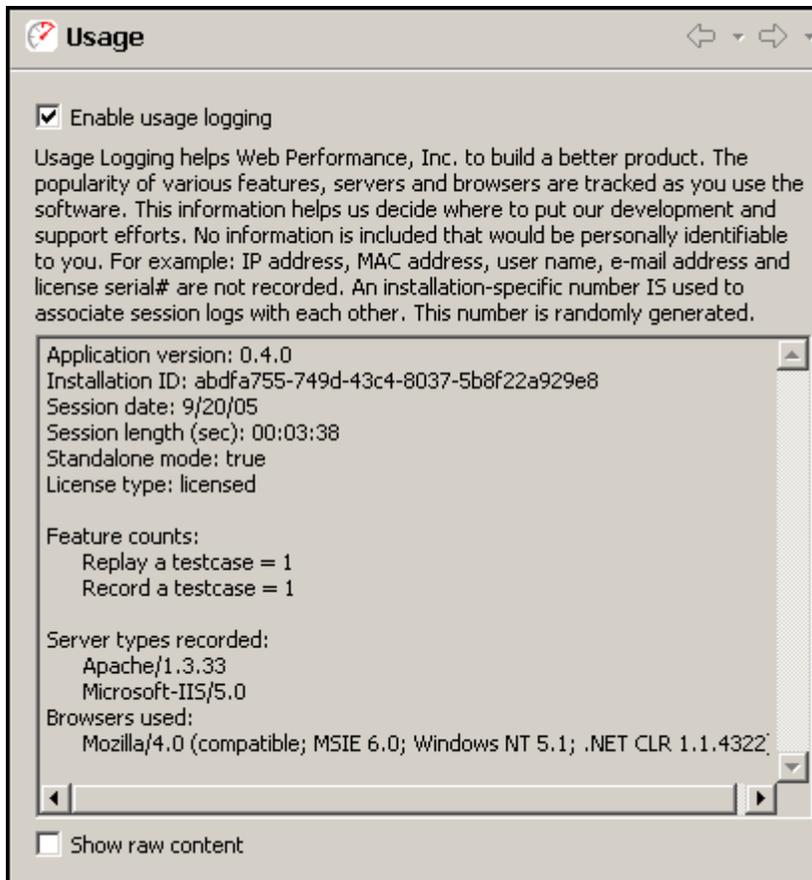
Disabled

When a license is shown as (*disabled*), it means that another copy of the same license key has been detected on a computer on the same network. As a result, both have been temporarily disabled. To remedy, remove the duplicate license key from one of the computers and restart both programs.

Usage Logging

Web Performance products have the ability to track the usage of various features and report those usage statistics to Web Performance, Inc. via the internet. This feature may be disabled by users with a valid license key via the *Window->Preferences->Web Performance->Usage* menu item.

The preference settings page shows the information that is collected so that a user may verify that private or personally-identifiable information is not submitted. An example of the preference page is show below, including an example of the information that would be submitted after a short session that included recording our website and replaying the testcase.



By default, the information is displayed in a user-friendly format. The *Show raw content* option may be selected to display exactly, byte for byte, what information is being submitted.

Configuring Memory Usage

If you receive an out of memory error, try the following to reduce memory usage:

- Close unused repositories.
- Delete unneeded testcases from open repositories.
- Close unused editors.
- Delete unneeded replays from testcases.
- Close unused views.
- Check your operating system to see if there are any background processes taking up memory.
- Try increasing the memory available to Web Performance Analyzer (See below).
- In the Status View, select the *Garbage Collection* icon.

Stand-alone program

The default setting for the maximum amount of memory allocated to Web Performance Analyzer is 64MB. The program will stop recording if this limit is reached. To increase this value, use the following steps:

1. Locate the file "*Web Performance Analyzer.lax*" in the directory where you installed the program.
2. Create a backup copy of the file.
3. Edit the file and change these lines to specify the desired memory amount:

```
lax.nl.java.option.java.heap.size.initial=64M
```

```
lax.nl.java.option.java.heap.size.max=64M
```

The values are the initial and maximum amount of memory that Web Performance Analyzer is allowed to use. You may increase it up to the maximum value of **free memory** you have available on your computer.

Eclipse Plug-in

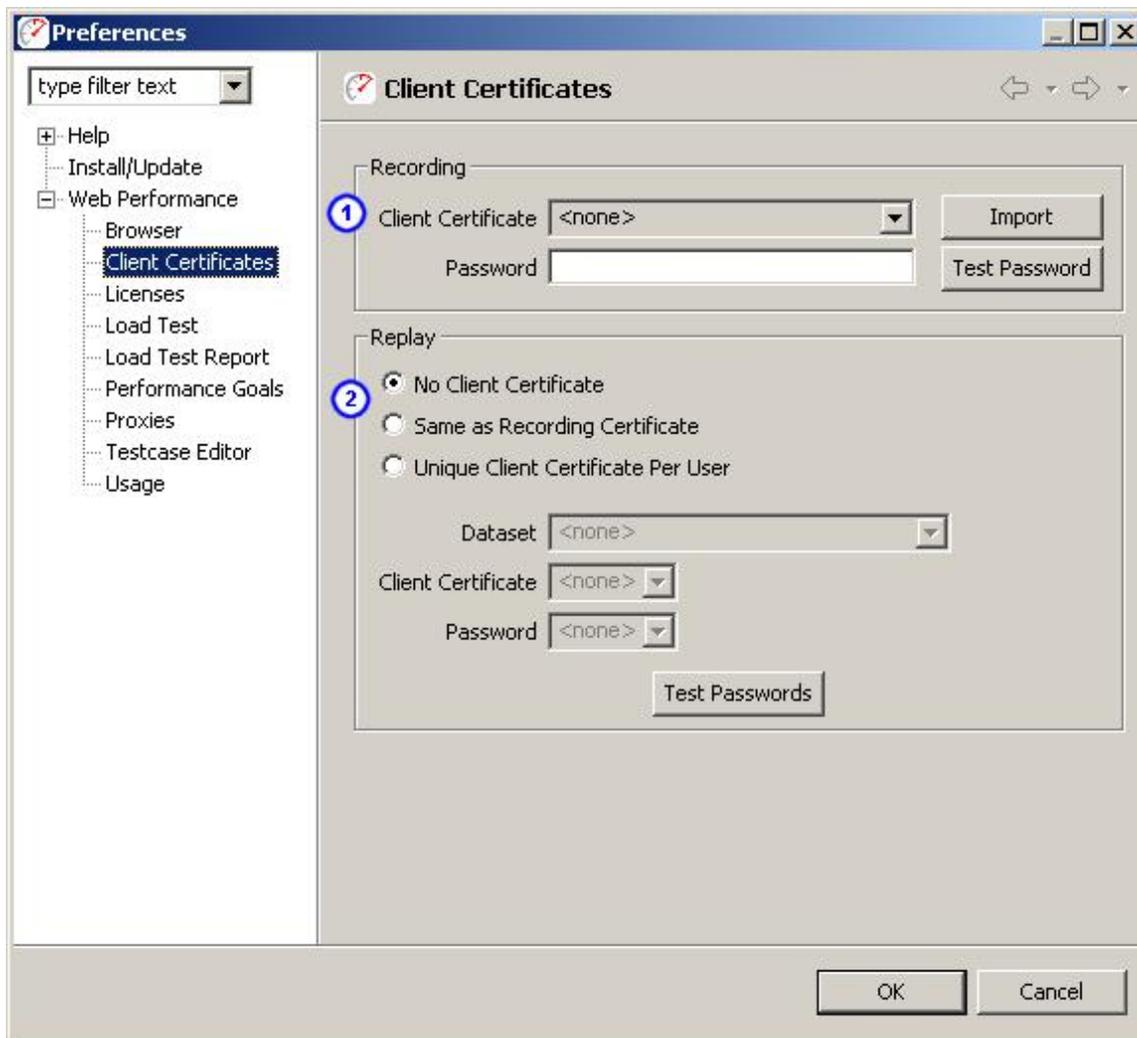
To increase the amount of memory available to Eclipse, start the application with the command line argument:

```
eclipse.exe [eclipse arguments] -vmargs -Xmx64M
```

The above sets the heap size to a maximum of 64M, modify the value as needed for your installation.

Client Certificates

The Client Certificates preference page determines what client certificates will be presented to the server during recording and playback.



Recording

The Recording section **1** determines which certificate will be presented during the recording process. The certificate(s) must be imported using the *Import* button. Certificates are protected by a password, use the *Test Password* button to test the password entered for the selected certificate.

Replay

The Replay section **2** determines which certificate will be presented by each virtual user during the replay of a testcase.

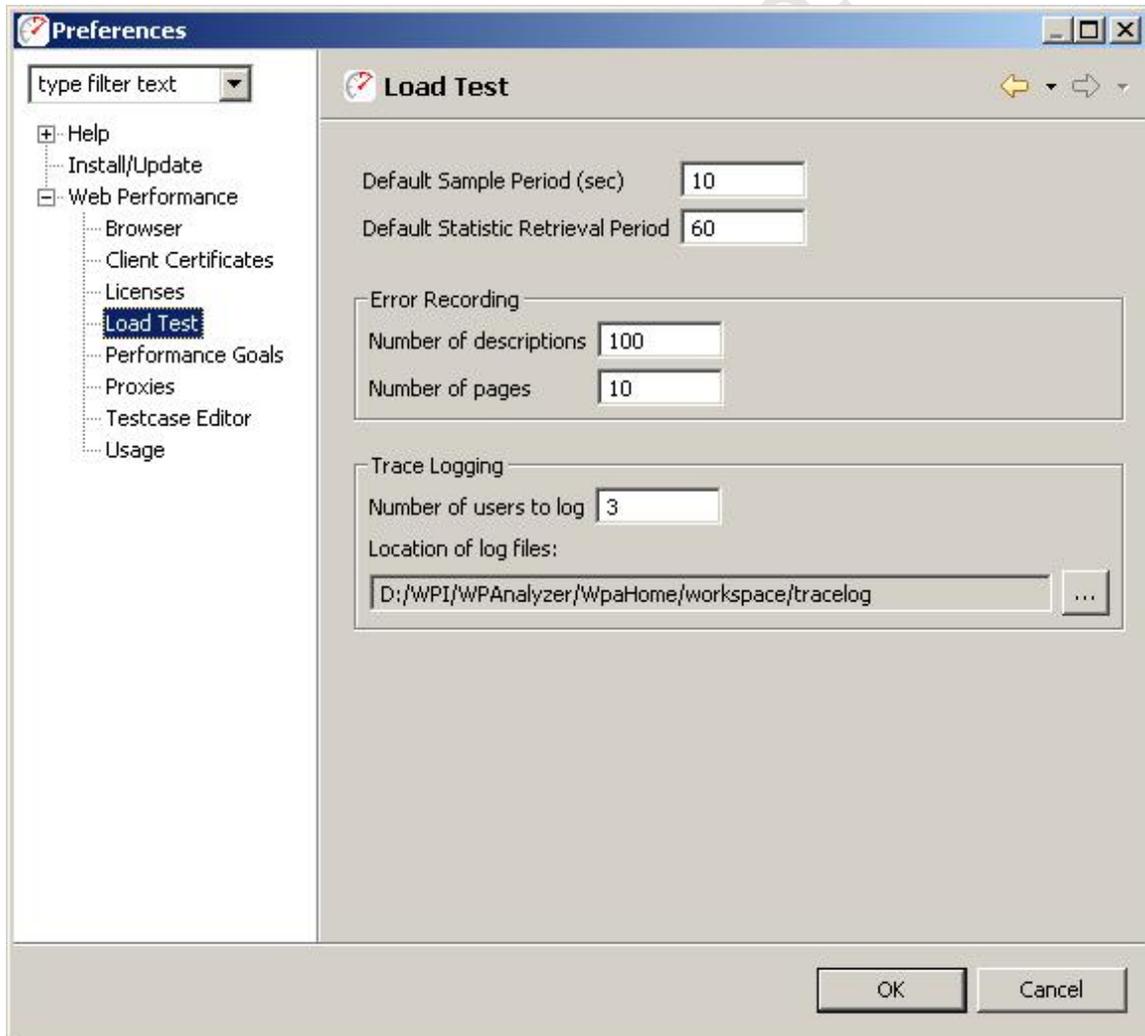
To configure each virtual user to use a different certificate, they must all be imported into Analyzer. In addition, a [dataset](#) must be created containing two fields:

1. filename
2. password

After creating the dataset, configure the *Dataset*, *Client Certificate* and *Password* fields. Use the *Test Passwords* button to validate the configuration. Depending on the number of certificates, testing the passwords could take several minutes.

Load Test Settings

The *Load Test Settings* page is located in the *Web Performance* section of the *Preferences* dialog (*Window->Preferences* menu item). The following pictures shows the default settings.



Default Sample Period

Specifies a default value for the frequency at which data statistics snapshots are recorded, in seconds. The default value will be used when a new load test configuration is created.

Default Statistic Retrieval Period

Specifies a default value for the frequency at which data is retrieved from load engines. The default value will be used when a new load test configuration is created.

Error Recording

Number of Descriptions

This setting limits the number of error description strings that are recorded during a load test (per testcase). Entering a large number may provide more information about errors when needed and it may also increase memory usage significantly.

Number of Pages

When an error is encountered during a load test, the web page which triggered the error is saved. This setting limits the number of times an error page will be recorded during a load test (per testcase).

Trace Logging

Detailed messaging between one or more virtual users and the servers in a load test can be saved for debugging purposes. To enable trace logging, select the *Enable* option. The number of users to save and the location of the log files can be specified once the option is enabled.

Charts & Reports

Performance Trend Chart

The Performance Trend Chart provides an easy way to spot trends in the performance of your web pages - are they getting faster or slower? Larger or smaller? By how much?

The chart is opened using one of the following:

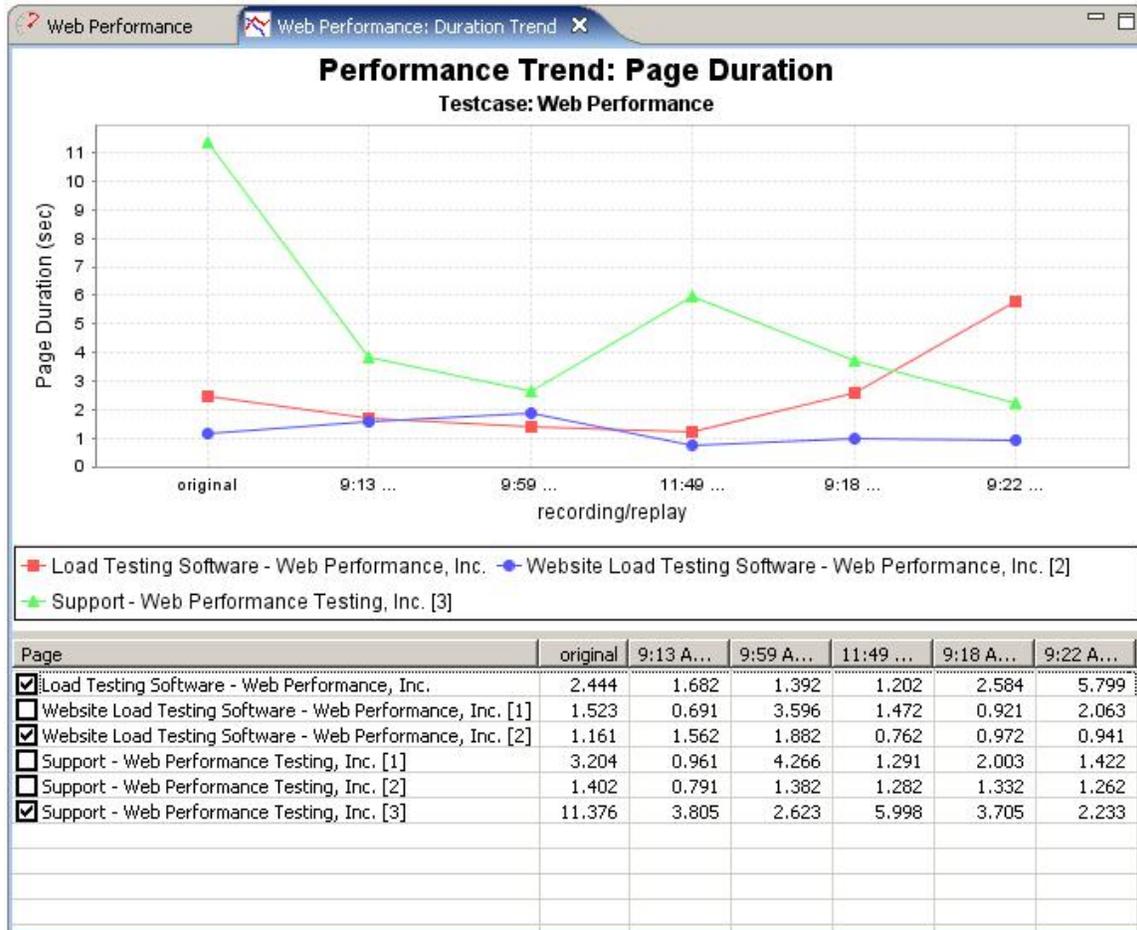
1. Press the pull down menu at the upper right corner of the Testcase Editor panel, select *Chart*, then select the type of chart (shown below).
2. Press the *Chart* button  on the main toolbar, then select the type of chart.
3. Select *Edit->Chart* on the main menu bar, then select type of chart.

The screenshot shows a table with columns for Size, Performance Trend (Duration), Performance Trend (Size), and URL. A 'Compare to...' menu is open, showing options for 'Performance Trend (Duration)' and 'Chart'. A red arrow points to the 'Compare to...' button.

Size	Performance Trend (Duration)	Performance Trend (Size)	URL
91830			formanceinc.com/
75744	00:01.523	200	http://webperformanceinc.com/prod
79106	00:01.161	200	http://webperformanceinc.com/anal
83033	00:03.204	200	http://webperformanceinc.com/supp
4734	00:00.090	200	http://webperformanceinc.com/supp

This example shows the chart for a testcase called "Web Performance".

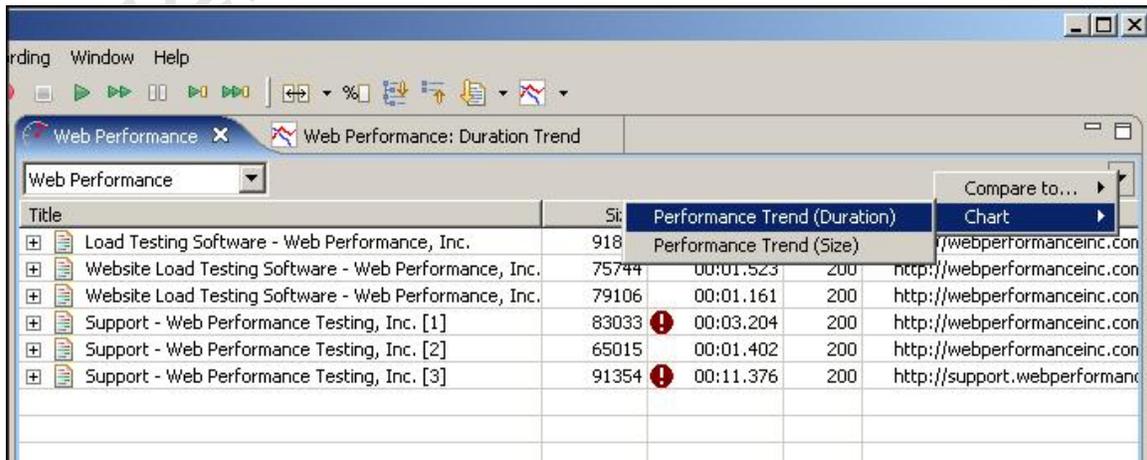
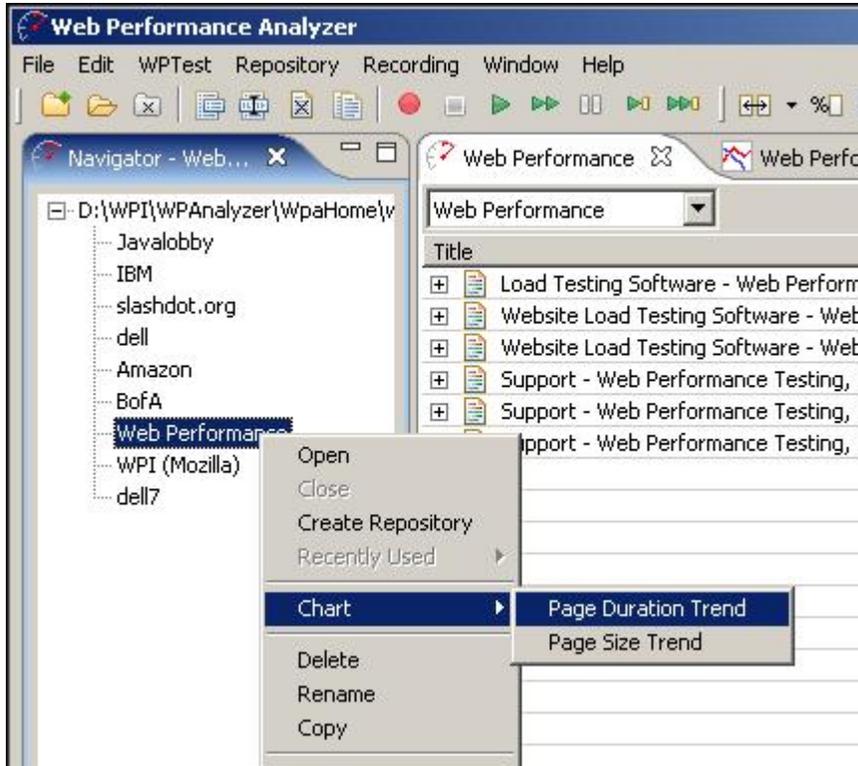
- The page durations are shown on the range axis (y-axis)
- The 5 replays and the original recording are shown on the domain axis (x-axis)
- In the table below the chart, the check boxes have been used to ignore the data from 3 pages



From the chart, you can see that the performance of one of our support pages has improved but our homepage has become slower.

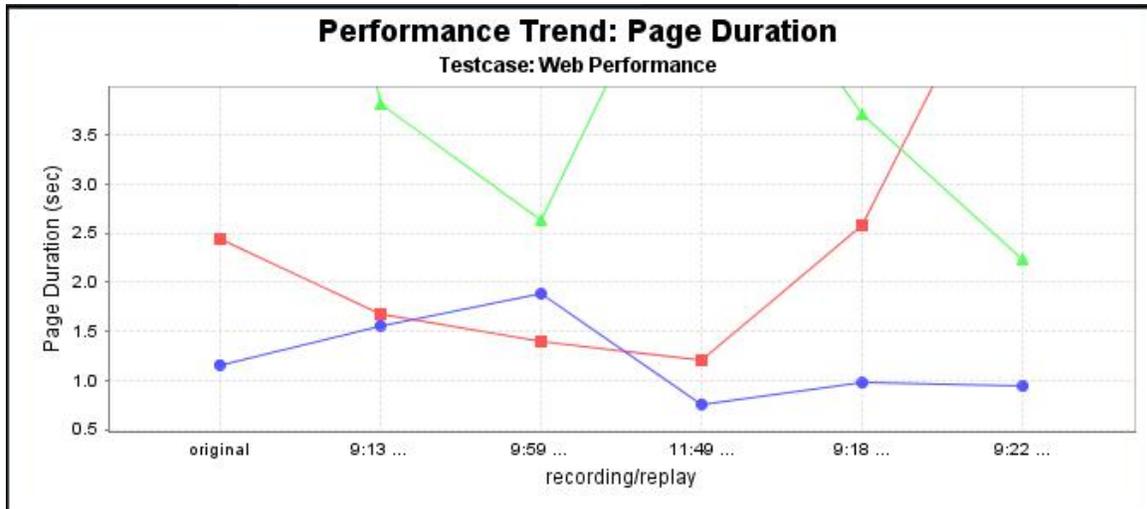
Accessing the charts

The charts can be opened from the context menu for the testcase as displayed in the navigator or from the menu in the upper-right corner of the testcase editor:



Zooming in

The chart can be adjusted to zoom in on specific parts on the range axis by dragging down over the area to be zoomed. This sample shows the result after zooming the range from 4.0 to 0.5 on the range axis:

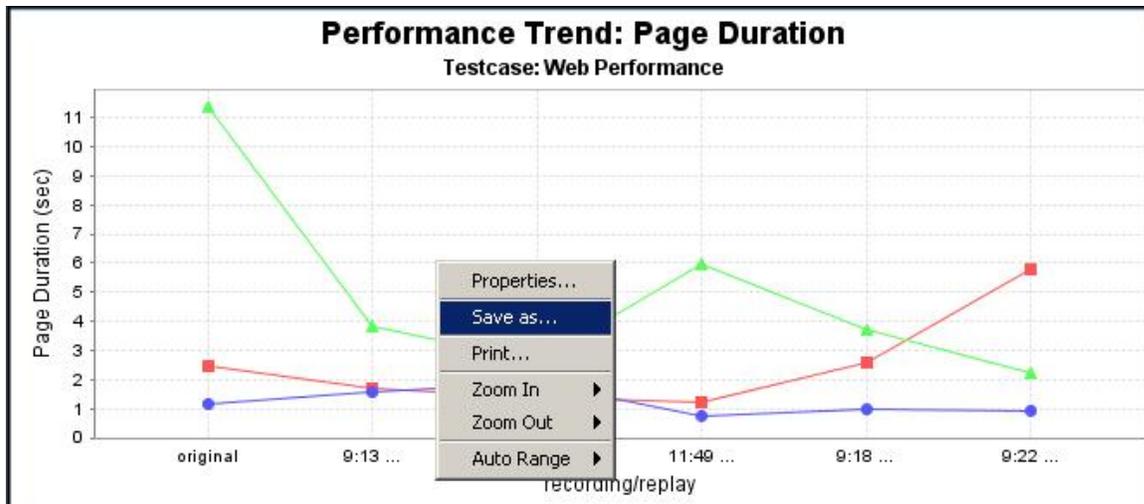


Return to the full-scale chart by dragging up in the chart area or selecting *Zoom Out* from the pop-up menu.

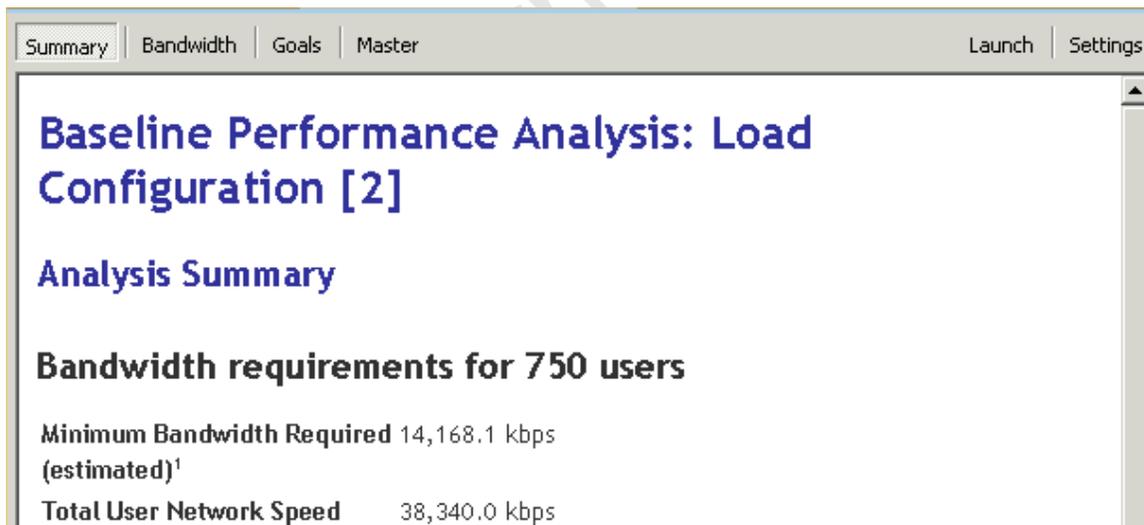
Selecting *Auto Range* from the pop-up menu will return the chart to the original layout.

Print and export

The chart may be printed or exported using the pop-up menu on the chart. Printing will automatically scale to the full size of the page. Most charts will most closely resemble what you see on the screen when the landscape orientation is used in the printer configuration options (platform and printer dependent). Exporting the chart will result in an image approximately the same size (in pixels) as the display on the screen.



Baseline Performance Report



The purpose of the Baseline Performance Report is to examine the performance of system with a single user in order to determine its performance when not under load. A great deal of the time web-based applications do not meet performance requirements with even a single user because the web page load times are not objectively measured.

The Analysis Summary gives an overall summary of the report's findings from the two other major sections which look at Performance Goals and Bandwidth Estimates.

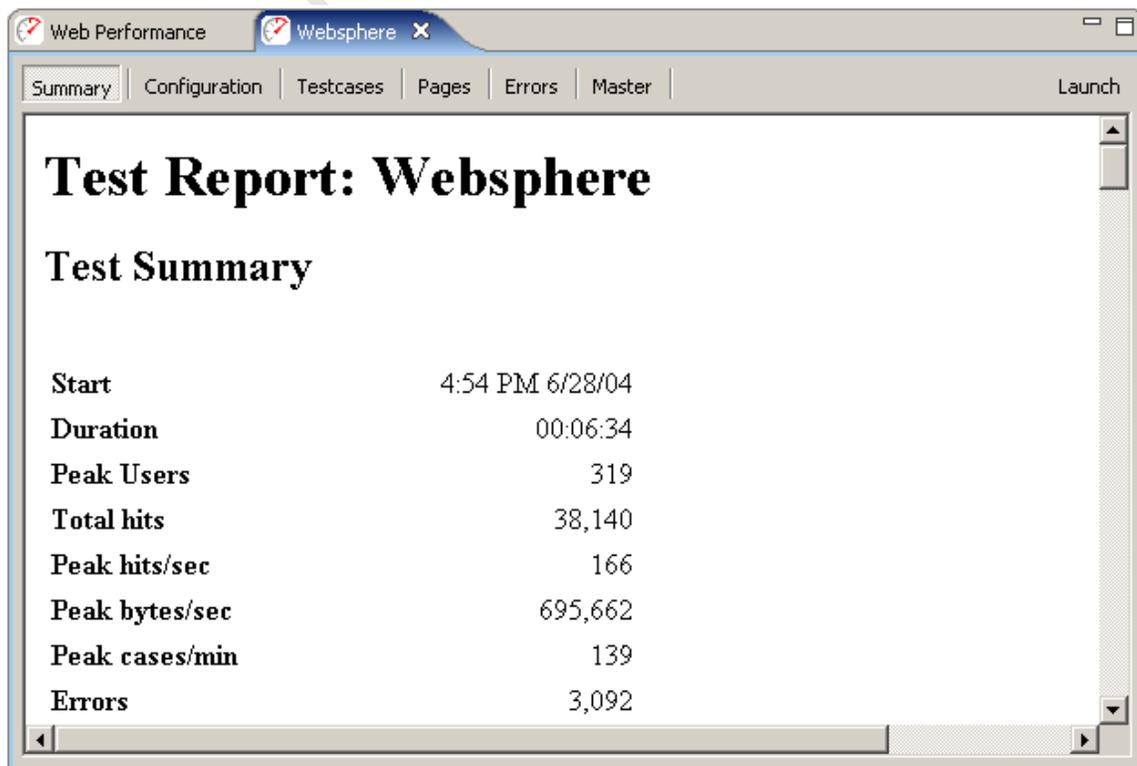
The Bandwidth report gives estimated values for the minimum and maximum bandwidth needed by the hosting company to support the specified number of users. It is a good place to start when planning the bandwidth that will be

required to performance the load test, and for capacity planning with the web hosting company. Of course, once a load test is performed real bandwidth data will be available.

The Goals section shows how many of the web pages will be estimated to meet or fail the performance goals given the test parameters. Of course these are just estimates and an actual load test will need to be run to get definitive answers. One of the most common sources of performance problems with web pages is designing the pages on a LAN with virtual unlimited bandwidth, which leads to large page sizes. When the pages are then viewed over a WAN, which is bandwidth limited, the pages can be much slower to view. This report uses the simulated bandwidth described in the Load Test Configuration Editor to estimate the effects of limited bandwidth on page load times.

Load Test Report

This is a comprehensive report summarizing the performance of a load test and detailing the performance of each page in the test. The report consists of several sections - as indicated by the buttons at the top of the view. When the report is opened (from either the [Navigator](#) or the [Load Test Results view](#)), the *Summary* section will be displayed. The *Master* report contains all of the other sections. The report may be viewed in an external browser window with the *Launch* button.



Printing and Exporting the report

Printing

The report may be printed by launching the report in a browser (using the *Launch* button) and printing from the browser. The currently selected section(s) will be displayed in the browser.

Saving as a file

Most browsers also provide a mechanism for saving the page to disk, including all the images. Internet Explorer also allows saving in the .mht web archive format (based on MIME), which combines the page and all images into a single file that can be easily distributed to other users.

Saving as a PDF

You may create a PDF of the report by launching the report in a browser and then printing to a PDF-enabled virtual printer. There are many available - some, such as [PDF Creator](#) are available free.

Manual Browser Configuration

Under the most common configurations, Web Performance software will automatically detect and configure the default browser for your platform (IE on Windows, Firefox on Linux/Unix, Safari on Mac OSX). When it does not, or the configuration is unusual, it may be necessary to configure the browser manually.

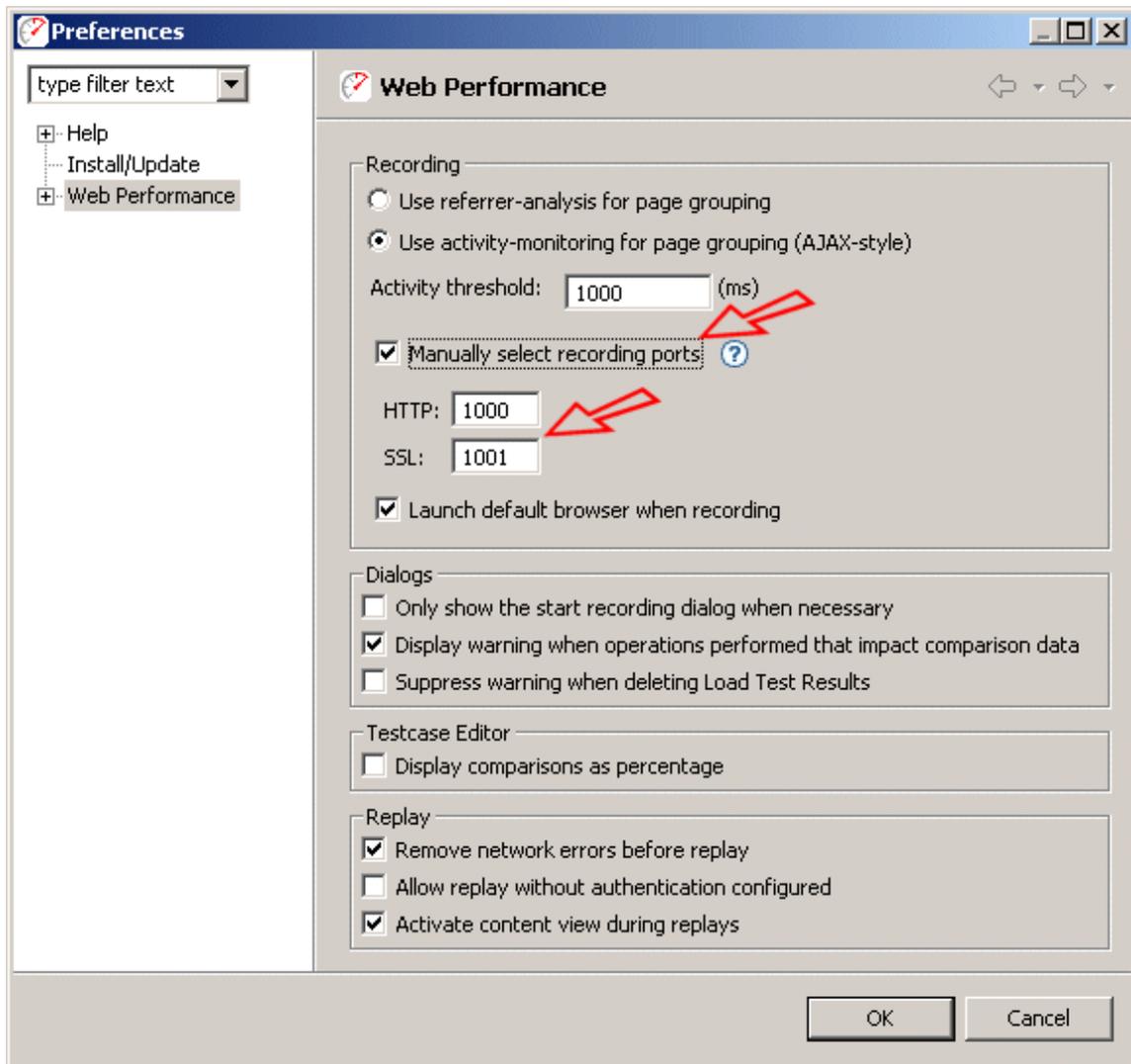
It may be possible to simply adjust the automatically detected browser settings in the [Browser Settings](#) and [Proxy Settings](#) preference pages. Additionally, unsupported browsers may be configured using those same preference pages. If this is not successful, then a fully manual configuration may be necessary. The following steps illustrate the process for IE and Firefox.

Step 1 - Configure recording ports

In order to manually configure the browser, Analyzer's internal recording proxy must be configured to use fixed port numbers (it normally chooses them automatically).

1. Select the *Preferences* item from the *Window* menu
2. Select the *Web Performance* item in the tree on the left
3. Turn on the *Manually select recording ports* option
4. If a warning message is displayed beside the port numbers, than the default ports are not available - you must enter different port numbers (they are automatically checked for availability)

5. Remember the selected port numbers - they need to be entered in the browser configuration later
6. Press the *OK* button



Step 2 - Configure the browser

The browser must now be configured to use the selected ports.

Manual configuration instructions are available for these browsers:

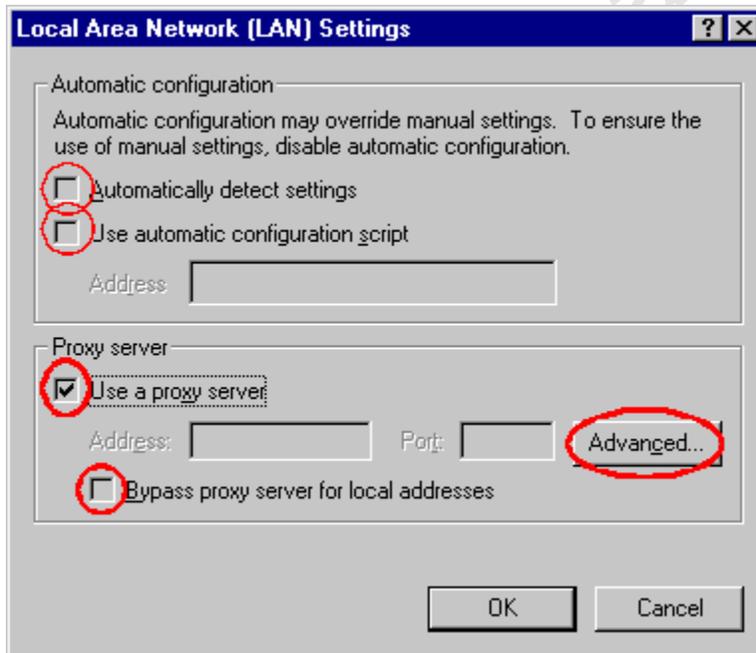
- [Internet Explorer](#) (6)
- [Firefox](#) (1.5) - also applies to Mozilla (1.x) and Netscape (6 and up)

For other browsers, the IE or Firefox instructions can be loosely followed, but deviations will be required when making the changes to the browser configuration. Consult the browser documentation where required.

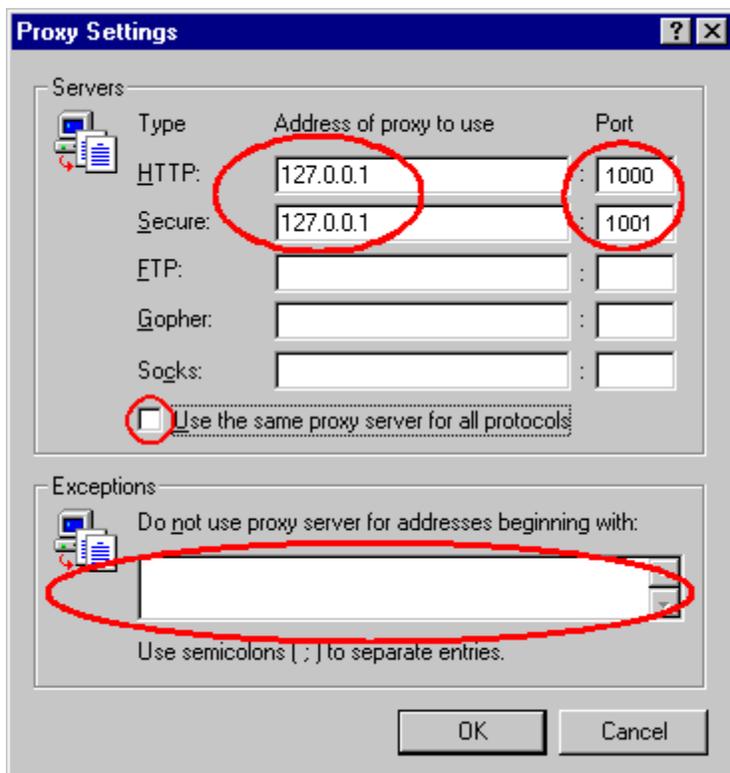
WARNING: these configuration changes will prohibit normal browsing when the Web Performance software is not running. These changes will need to be reversed to resume normal browsing. Be sure to write down or backup your settings to ensure they can be restored correctly.

Internet Explorer

- Open the *Internet Options* dialog by choosing the *Tools* menu and selecting the *Internet Options* item
- Push on the *LAN Settings* button to view the screen below

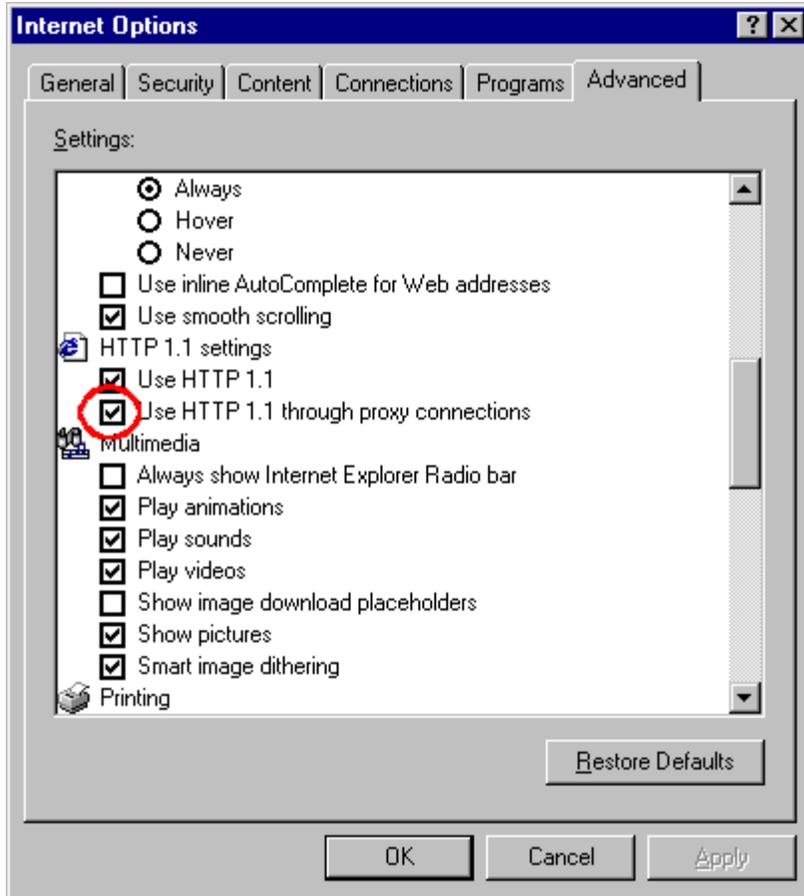


- In the *Proxy Server* section check the "Access the Internet using a proxy server" box
- Turn off the *Bypass proxy server for local (Intranet) addresses* option
- Turn off the *Automatically detect settings* option
- Turn off the *Use automatic configuration script* option
- Press the *Advanced...* button



- On the *Proxy Settings* dialog, IE must be provided with the address of the proxy server that will be listening for requests.
- In the *HTTP* fields enter "127.0.0.1" for the address and the HTTP port number configured in Step 1 for the port number
- Under certain configurations, you may have to try substituting the machine name "localhost" for the address "127.0.0.1"
- In the *Secure* fields enter "127.0.0.1" for the address and the SSL port number configured in Step 1 for the port number
- Note that the *Secure* line may not always be the 2nd line
- It is also important to clear any entries in the *Do not use proxy server for addresses beginning with:* field - these could prevent the browser from using the recording proxy
- press the *OK* button

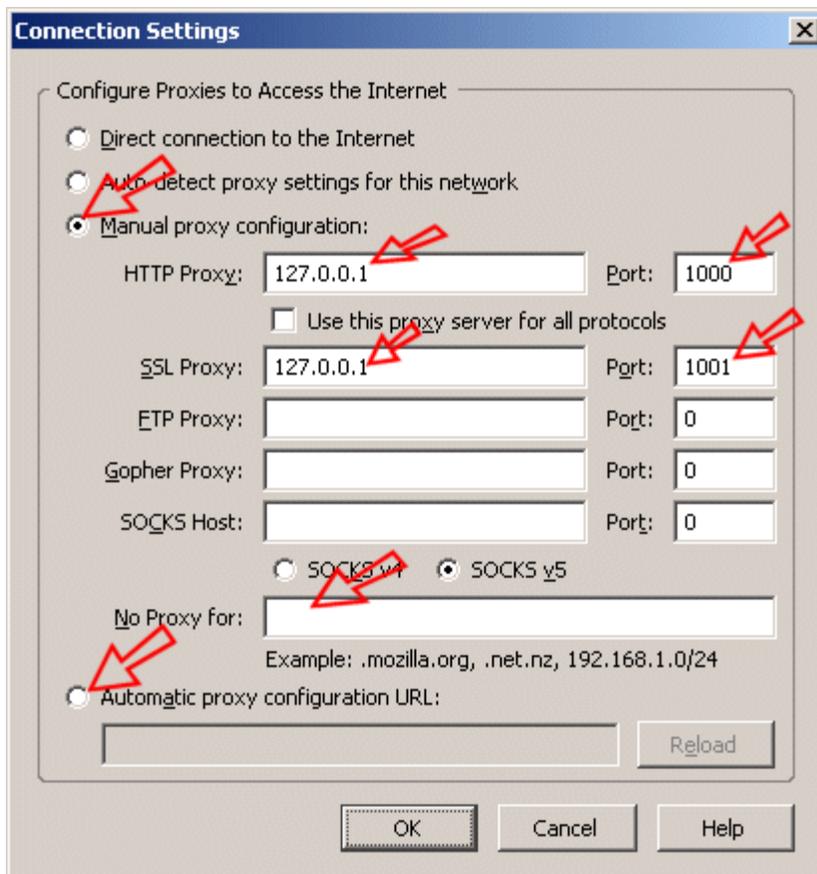
The final step in the browser configuration is to configure the HTTP connection for the browser for a proxy using the *Advanced* tab of the same Options Dialog. Make sure that the *Use HTTP 1.1 through proxy connections* option is turned ON.



- Push the OK buttons until you return to the browser
- Skip down to [Step 3](#)

Firefox

- Select the *Options...* item from the *Tools* menu (for Netscape/Mozilla, select the *Preferences* item from the *Edit* menu)
- Select the *General* section icon at the top and Push the *Connection Settings...* button (for Netscape/Mozilla, select the *Proxies* item in the *Advanced* section)



- Select the *Manual proxy configuration* option
- Enter the data as shown in the *HTTP proxy* and *SSL proxy* fields, substituting the port numbers from step 1
- Clear the *No Proxy for* field
- De-select the *Automatic proxy configuration URL* option
- Push the OK buttons until you return to the browser

Recommendation - manually switching the proxy configuration can be cumbersome. A Firefox extension called *SwitchProxy* makes the process much simpler - we recommend it! It is available from the Firefox extensions page.

Step 3 - Select proxy server

Finally, if a proxy server is required to access the applications to be recorded, it must be configured in the [Proxy Settings](#) preference page. If you do not know if a proxy is required - ask your network administrator.

When you have the necessary proxy information, use [these instructions](#) to add a new proxy configuration and make it the default setting.

Step 4 - Test the configuration

After these configuration steps are finished, press the refresh button in your browser to retry the diagnostic page. If the URL of the diagnostic page is no longer in the URL field, you may enter this:

```
http://webperformance.com/diagnostic/proxy_check.html
```

The browser should display a *Welcome* page indicating that the configuration is successful.

Workspace

The *Workspace* is the area on your computers disk where Analyzer stores settings, preferences and repositories (by default). The default location for the workspace is a new folder (named *WebPerformance*) in the user home folder. On Windows systems, this is usually in "C:\Documents and Settings\".

Changing the workspace location

The workspace can be moved to any location that can be accessed by the program, including network mounted drives. After moving the workspace, Analyzer will need to know where to find the workspace. In the installation folder (by default on Windows is "C:\Program Files\WPSuiteNN") there is a subfolder named *config*. In this folder there is a file named *config.ini*. Edit this file in a plain-text editor (e.g. notepad) and look for a line starting with "*osgi.instance.area*".

It should look like:

```
# set the workspace location
osgi.instance.area=@noDefault
#osgi.instance.area=workspace
#osgi.instance.area=@user.home\WPWorkspaceNN
#osgi.instance.area=C:\\Temp\\WPWorkspace
```

There are a number of options for this setting.

1. @noDefault - this allows Analyzer to choose automatically - it will use the folder described above by default.
2. workspace - simply entering a folder name will cause Analyzer to create a subfolder with the chosen name under Analyzer's installation folder. In this example it would result in "C:\Program Files\WPSuiteNN\workspace"
3. @user.home\WPWorkspaceNN - this will cause Analyzer to use a folder inside the user home folder with the specified name. In this example it would result in "C:\Documents and Settings\\WPWorkspaceNN".
4. The last option is to specify a fully qualified path to the folder where the workspace has been moved to. Note that on Windows systems, the

backslash (\) characters must be doubled since the backslash is the escape character for this configuration file.

Load Engines

The process of simulating hundreds of virtual users and collecting their statistics requires considerable computing power. In order to simulate enough users to fully load a sophisticated web server configuration, more than one computer will be required. Web Performance Suite allows multiple load engines to cooperate to generate large amounts of load.

The Web Performance Load Engine is a subset of the Web Performance Suite software that runs only the components required to run load tests. It has a limited user interface (text output only) primarily intended for reporting errors. Because there is no complex user interface running, a dedicated load engine makes most efficient use of computing resources for generating virtual users. It also allows the engine to run on headless workstations, such as a Unix platform without X Windows. Used in this way, one computer running Web Performance Suite will be used to control multiple Web Performance Load Engines. The engines will generate load against the web server and report the resulting statistics back to Web Performance Suite.

Installing a Load Engine

Installers for Linux, Solaris, and Windows are available on our [website](#). On non-Windows machines, the application can be installed using either the GUI or command-line interface. For example, to trigger a console installer from the command line for a Linux machine, the following command is used:

```
LoadEngine_Linux_3.0.bin -i console
```

Starting a Load Engine

Starting a load engine is similar to starting Web Performance Suite. On Windows platforms, there is a menu item labeled *Load Engine* in the same location as the item for starting Web Performance Suite. This starts a console window. When the load engine has finished initialization, a message appears in the console window reading *Load Engine started*. Entering *quit* in the console window stops the load engine and closes the window.

The load engine is started on Linux, Solaris and UNIX platforms using the installed shortcut or the startup script:

```
/usr/local/bin/WebPerformanceSuite/Load_Engine
```

Web Performance, Inc.

Index

I	
installation.....	1

Web Performance, Inc.