Web Performance Suite 3.5 Manual

Table Of Contents

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.

Installation

Supported Platforms

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

note: Installation of the Load Engine is supported on Windows, Linux and Solaris. Installation of the Server Agent is supported on Windows and Linux.

Installation Notes

- For Linux stand-alone installations, do not install the application at the top level directory (at "/"). There is a known issue with the install application when this directory is used.
- For any installation, ensure that the directory the application is being installed in is not a read-only directory to the users of the application. The application will not start properly when this occurs.
- For stand-alone installations, do not install Web Performance Suite and Web Performance Load Engine in the same directory.

Installation steps

1. Download the software from http://webperformanceinc.com/download
2. On Windows, run the installer and follow the wizard

3. On Linux/Unix with a GUI, run the installer and follow the wizard
4. On Linux/Unix from a console, run the installer with the "-i console" option.

Load Engine and Server Agent Installation

Load Engine installation instructions

Server Agent installation instructions

Getting Help

Integrated help system

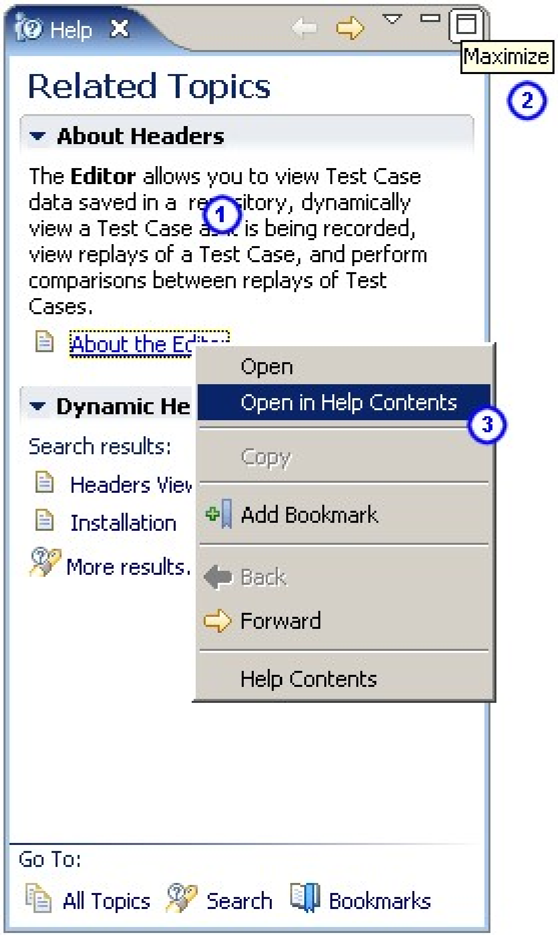
This page is part of the Quick Start Guide within the integrated help system. There are four books in the help system:

- 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.

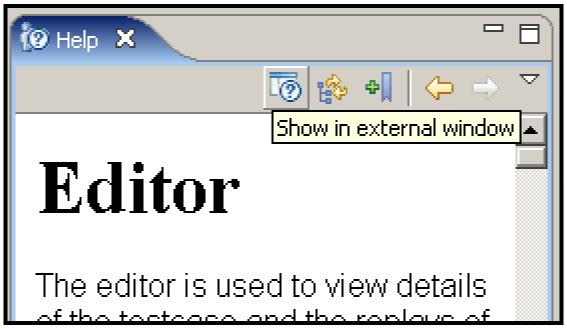
Help shortcuts

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



The **Editor** allows you to view Test Case data saved in a repository, dynamically view a Test Case as it is being recorded, view replays of a Test Case, and perform comparisons between replays of Test Cases.

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.



Editor

The editor is used to view details

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 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).

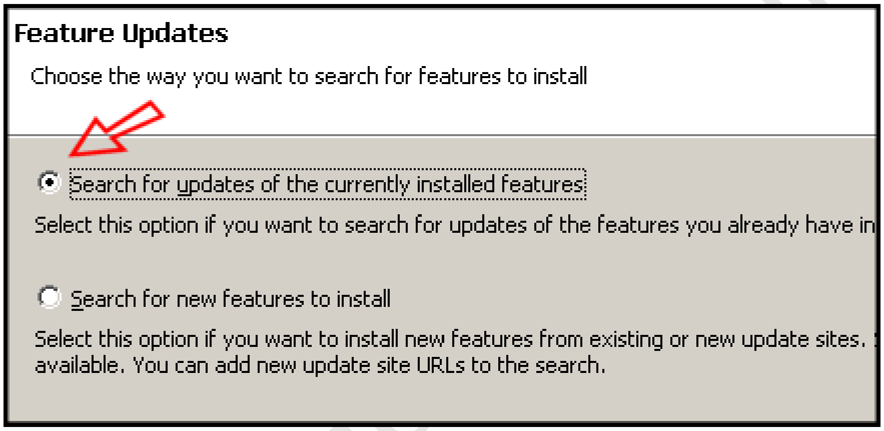
Updating the software

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.



Feature Updates

Choose the way you want to search for features to install

● Search for updates of the currently installed features

Select this option if you want to search for updates of the features you already have in

○ Search for new features to install

Select this option if you want to install new features from existing or new update sites. available. You can add new update site URLs to the search.

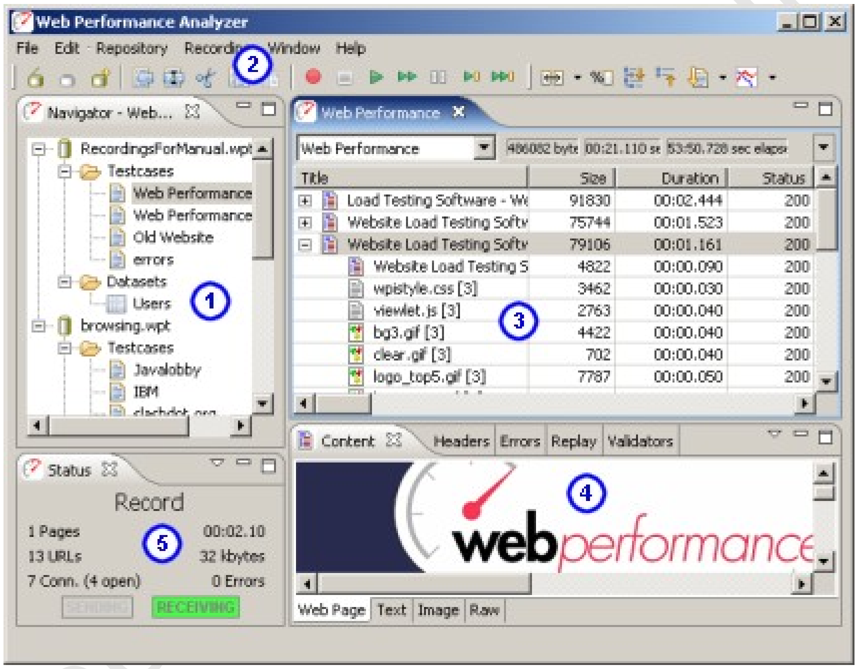
3. Follow the wizard to complete the installation.

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

For detailed information on downloading patch updates and upgrades, see the Managing software updates reference page.

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.



Title	Size	Duration	Status
Load Testing Software - We	91830	00:02.444	200
Website Load Testing Softv	75744	00:01.523	200
Website Load Testing Softv	79106	00:01.161	200
Website Load Testing S	4822	00:00.090	200
wpistyle.css [3]	3462	00:00.030	200
viewlet.js [3]	2763	00:00.040	200
bg3.gif [3]	4422	00:00.040	200
clear.gif [3]	702	00:00.040	200
logo_top5.gif [3]	7787	00:00.050	200

1 Pages 00:02.10
13 URLs 32 kbytes
7 Conn. (4 open) 0 Errors

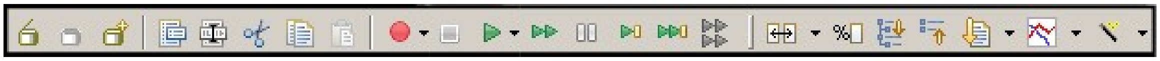
1. Navigator view
2. Toolbars
3. Editors and Charts
4. Detailed inspection views
5. Status view

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.

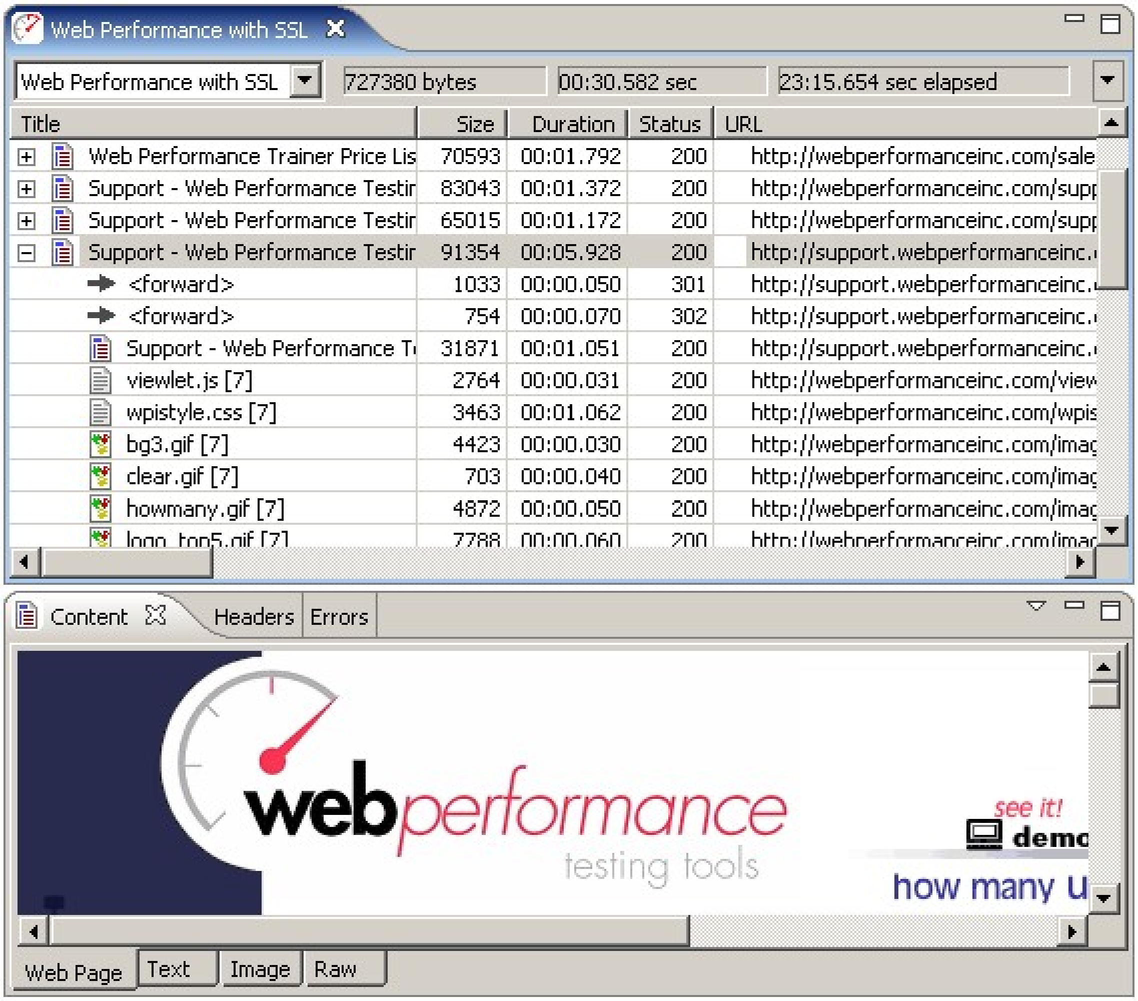
You may now review and inspect the contents of the recording in the Editor. The other views (Headers, Content, Errors, etc.) will updated to show additional information about the pages and URLs you select in the recording.

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.



Web Performance with SSL

Web Performance with SSL | 727380 bytes | 00:30.582 sec | 23:15.654 sec elapsed

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 Te	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/imag
clear.gif [7]	703	00:00.040	200	http://webperformanceinc.com/imag
howmany.gif [7]	4872	00:00.050	200	http://webperformanceinc.com/imag
logo_top5.gif [7]	7788	00:00.060	200	http://webperformanceinc.com/imag

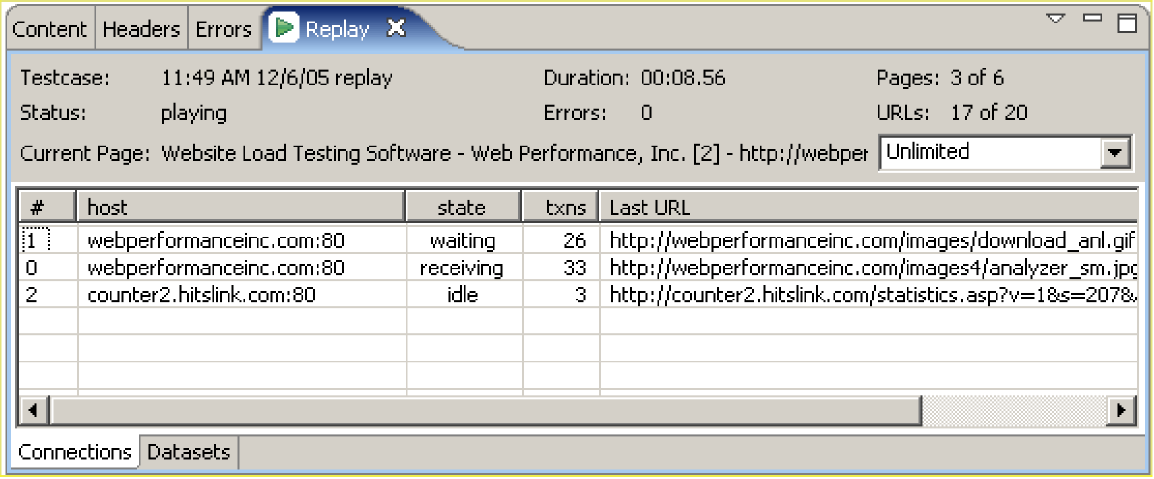
When a web page or URL is selected in the Testcase Editor, some views display details about the selected item. Web pages are displayed in an embedded browser in the Content View. Images and other text resources are displayed in text and image viewers, while other resources are displayed in the raw viewer. 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.

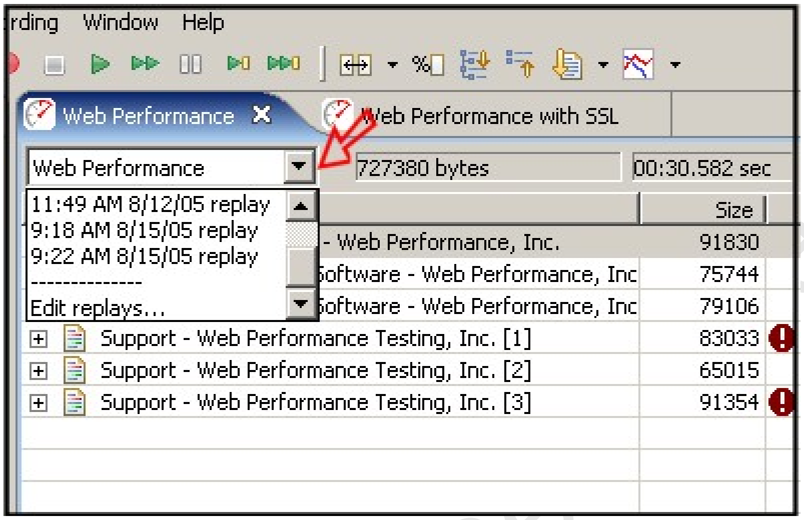


Testcase: 11:49 AM 12/6/05 replay
Duration: 00:08.56
Pages: 3 of 6
Status: playing
Errors: 0
URLs: 17 of 20
Current Page: Website Load Testing Software - Web Performance, Inc. [2] - http://webper
Unlimited

#	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.jpc
2	counter2.hitslink.com:80	idle	3	http://counter2.hitslink.com/statistics.asp?v=1&s=207&

Analyze the Performance Changes

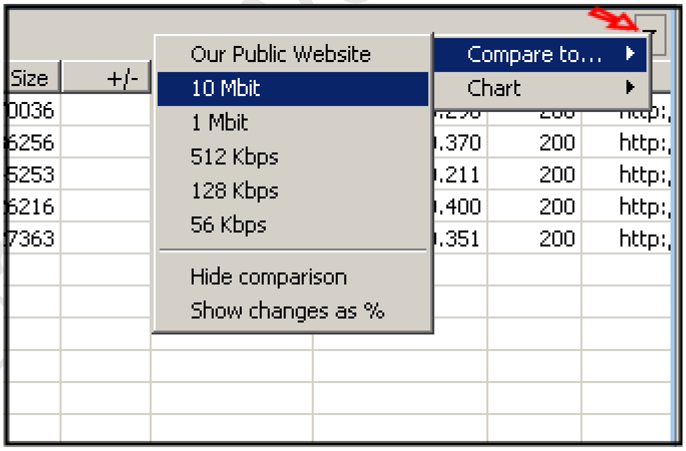
As soon a replay has been initiated, it appears in the *replay selection list*, as shown below. The testcase editor reflects the performance measurements of the selected replay for the testcase.

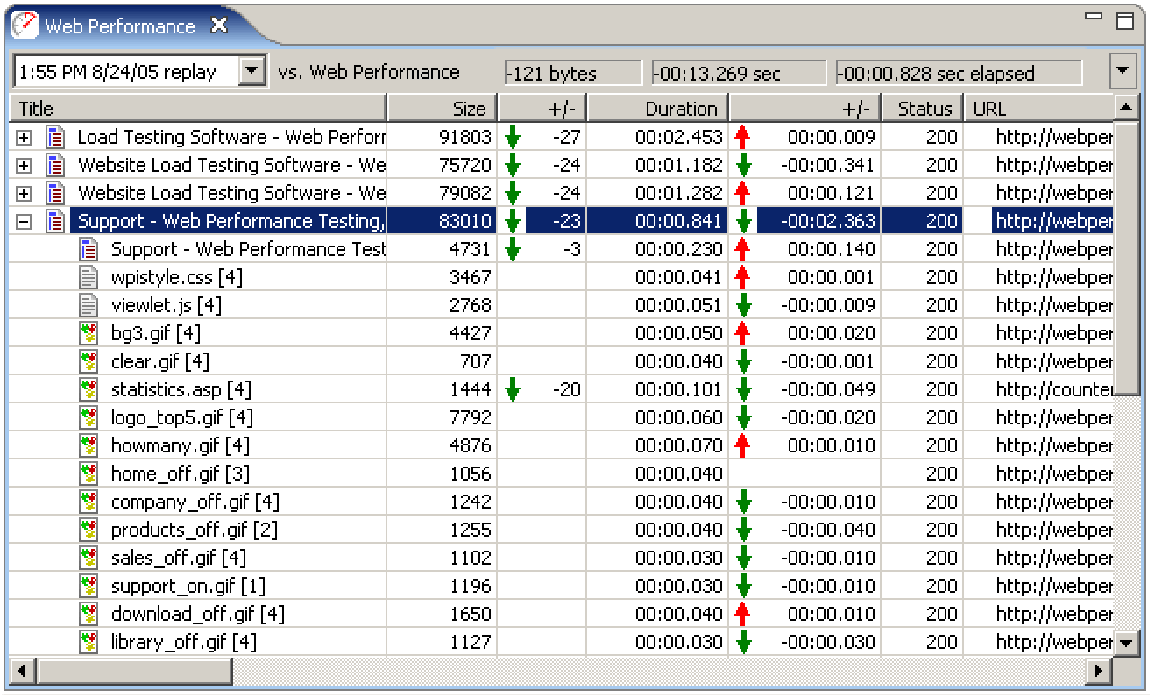


727380 bytes 00:30.582 sec

	Size
- Web Performance, Inc.	91830
Software - Web Performance, Inc	75744
Software - Web Performance, Inc	79106
Support - Web Performance Testing, Inc. [1]	83033
Support - Web Performance Testing, Inc. [2]	65015
Support - Web Performance Testing, Inc. [3]	91354

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.

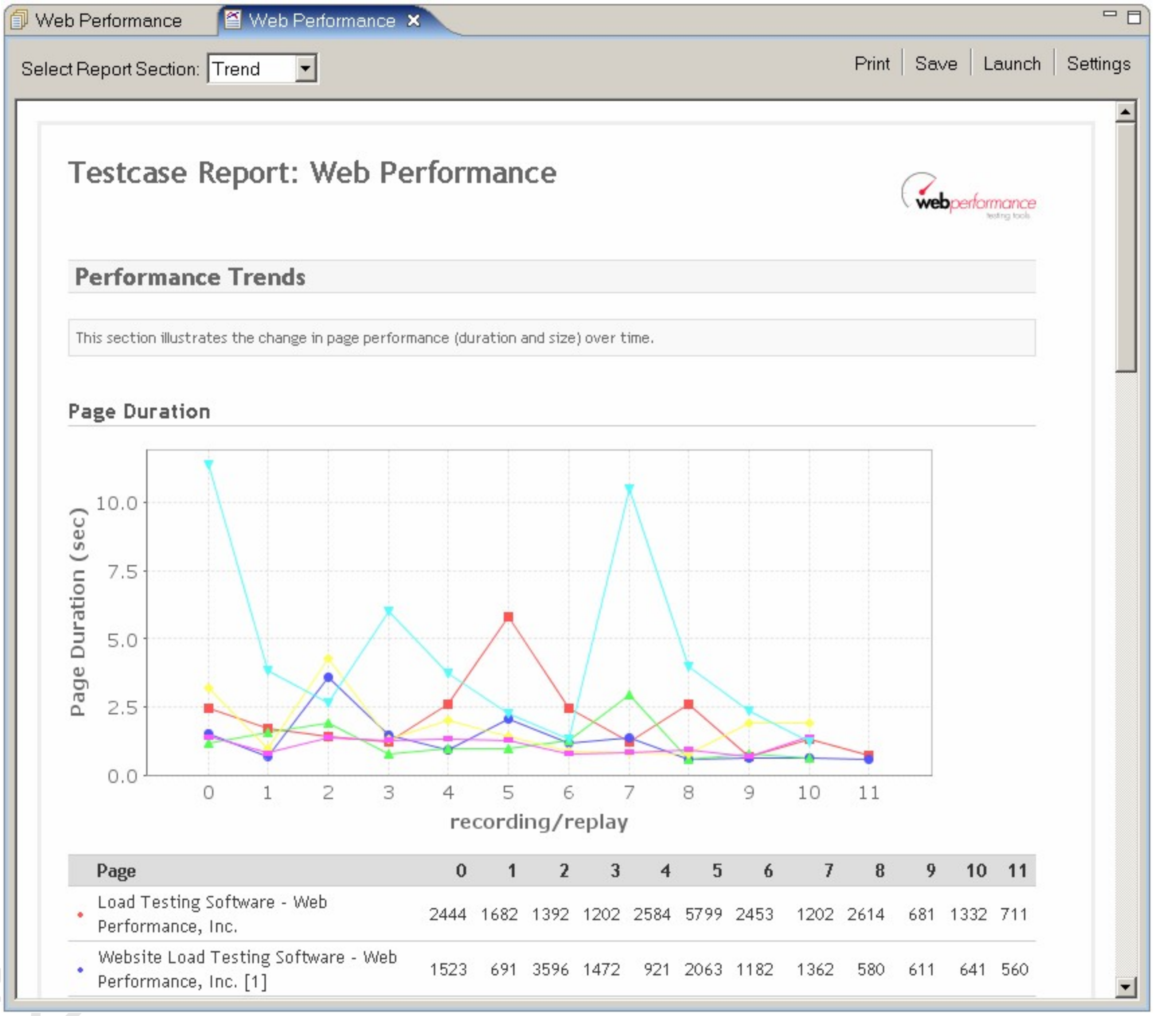




1:55 PM 8/24/05 replay ▾ vs. Web Performance | -121 bytes | -00:13.269 sec | -00:00.828 sec elapsed

Title	Size	+/-	Duration	+/-	Status	URL
⊞ Load Testing Software - Web Perforr	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 Testcase Report . For example, selecting the *Open Report* item from the *Testcase Editor* menu. Then navigate to the *Trend* section of the report.



Testcase Report: Web Performance

Performance Trends

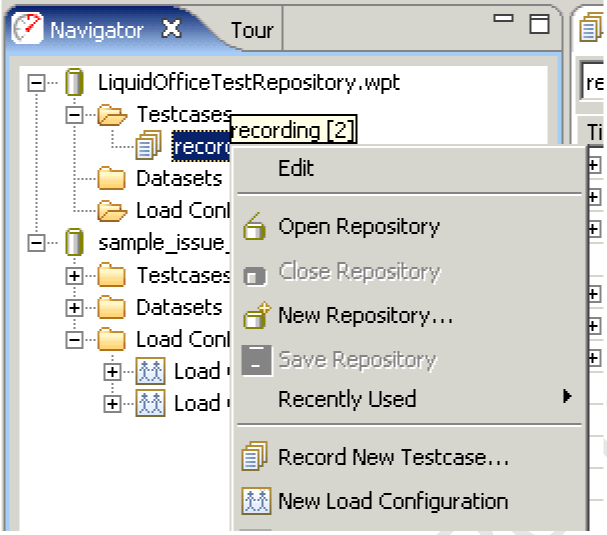
This section illustrates the change in page performance (duration and size) over time.

Page Duration

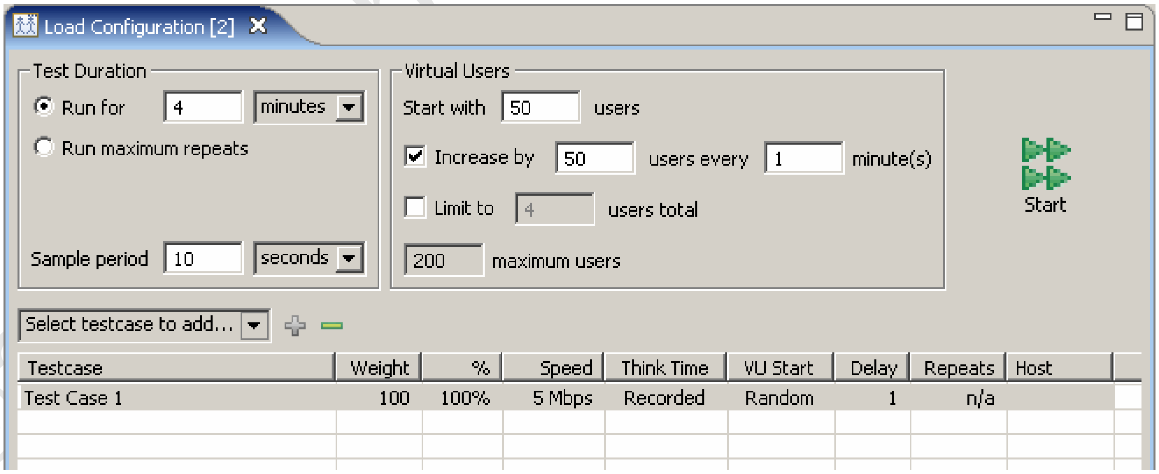
Page	0	1	2	3	4	5	6	7	8	9	10	11
● Load Testing Software - Web Performance, Inc.	2444	1682	1392	1202	2584	5799	2453	1202	2614	681	1332	711
● Website Load Testing Software - Web Performance, Inc. [1]	1523	691	3596	1472	921	2063	1182	1362	580	611	641	560

Run a Load Test

The first step is to create a Load Configuration. Select the testcase in the Navigator and, from the pop-up menu, choose *New Load Configuration*:



Fill in the Load Configuration options as desired:


Load Configuration [2]

Test Duration

● Run for 4 minutes

○ Run maximum repeats

Sample period 10 seconds

Virtual Users

Start with 50 users

☑ Increase by 50 users every 1 minute(s)

□ Limit to 4 users total

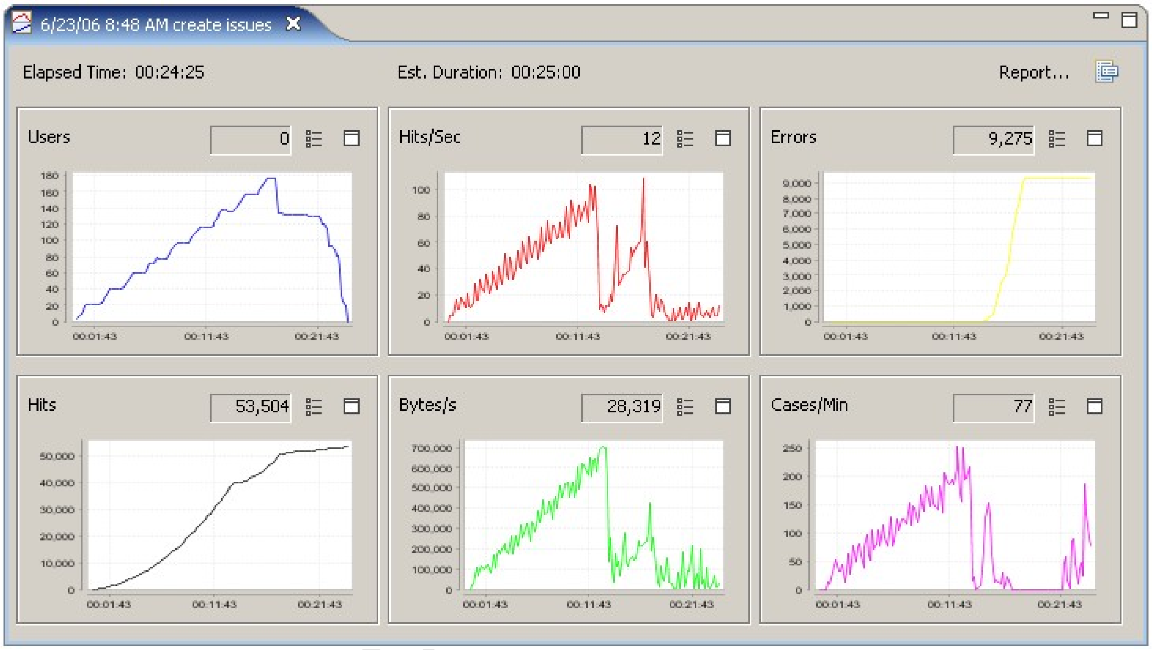
200 maximum users

Start

Select testcase to add...

Testcase	Weight	%	Speed	Think Time	VU Start	Delay	Repeats	Host
Test Case 1	100	100%	5 Mbps	Recorded	Random	1	n/a	

Once the configuration is acceptable, press the *Start* (⏩) button. The Load Test overview screen will appear while the load test runs:

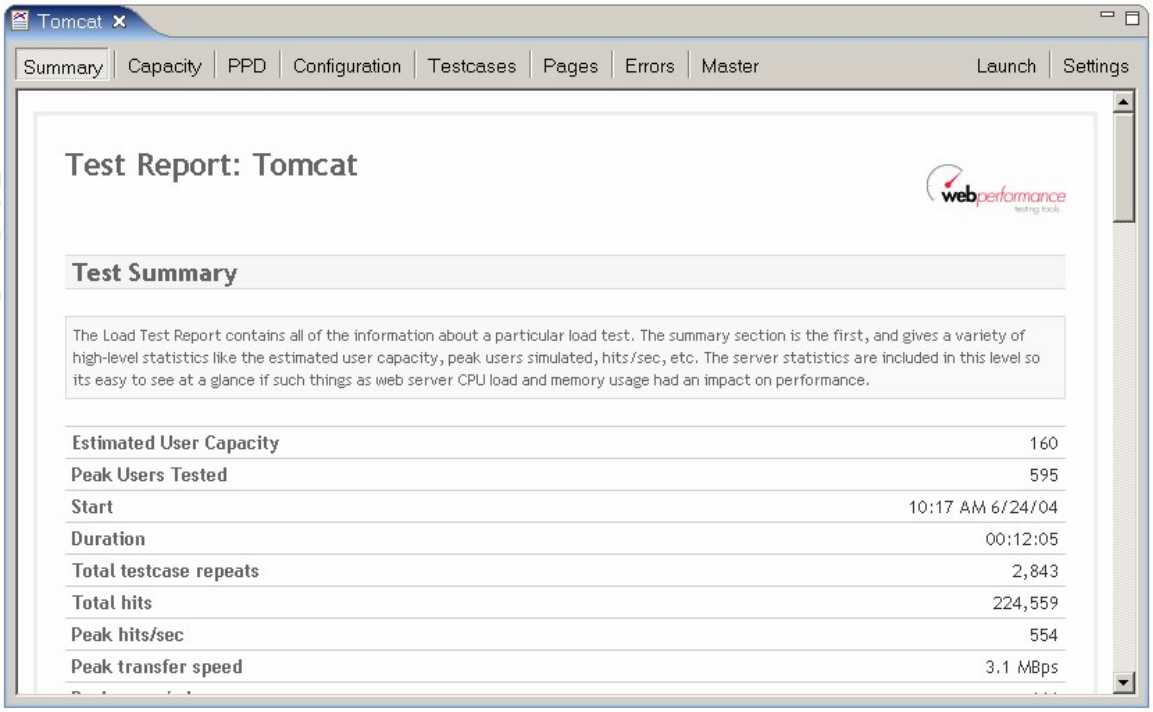


Elapsed Time: 00:24:25

Est. Duration: 00:25:00

Users	0
Hits/Sec	12
Errors	9,275
Hits	53,504
Bytes/s	28,319
Cases/Min	77

When the test is complete, press the *Report...* button to open the Load Test Report:



Test Report: Tomcat

Test Summary

The Load Test Report contains all of the information about a particular load test. The summary section is the first, and gives a variety of high-level statistics like the estimated user capacity, peak users simulated, hits/sec, etc. The server statistics are included in this level so its easy to see at a glance if such things as web server CPU load and memory usage had an impact on performance.

Estimated User Capacity	160
Peak Users Tested	595
Start	10:17 AM 6/24/04
Duration	00:12:05
Total testcase repeats	2,843
Total hits	224,559
Peak hits/sec	554
Peak transfer speed	3.1 MBps

You may also use the Errors View and Statistics View to get more detailed results of the load test.

Tutorials

Tutorials Index

Record and Analyze a Testcase

The Baseline Analysis tutorial (phase one) is the first step in any performance analysis. This tutorial describes how to configure Analyzer to use your browser to record your website. Then you see 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

The Test Configuration tutorial (phase two) describes 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 then describes 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 Large Scale Tests tutorial.

Web Services

Testing a Web Service presents a step-by-step example of testing a simple web service.

Advanced Configuration

The Advanced Configuration Guide demonstrates several techniques and product features for configuring testcases for web applications that use complicated session-tracking techniques or dynamic field naming conventions.

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:

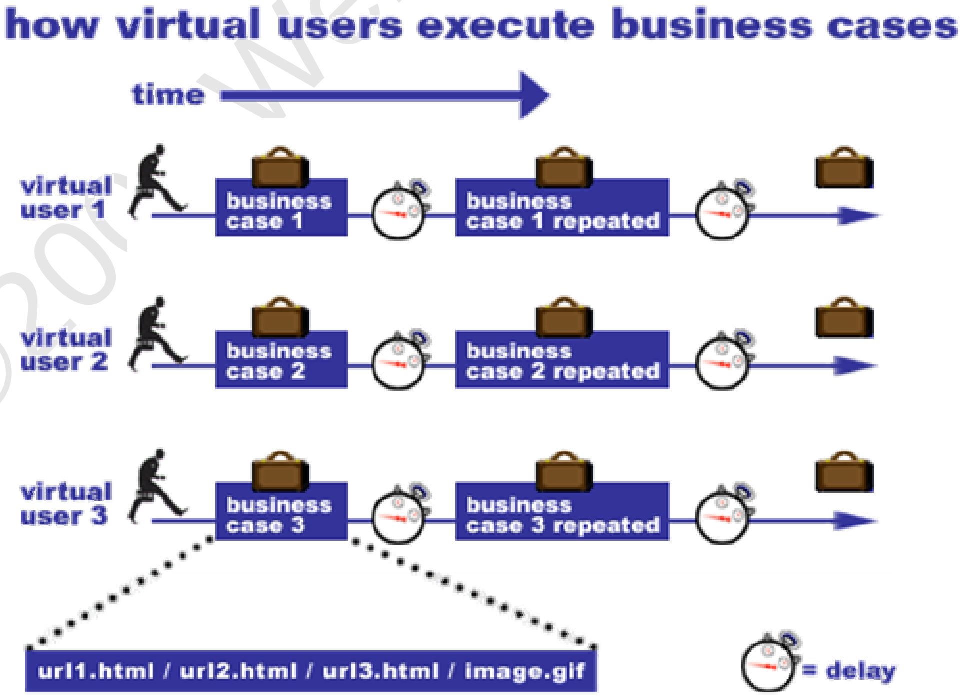
1. Baseline Analysis
2. Test Development/Verification

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.



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.

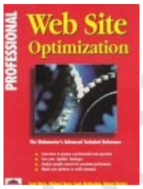
How many simultaneous virtual users you need to simulate depends on a number of factors, starting with how you express your web site's capabilities. Please refer to the hardware requirements for more information.

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:

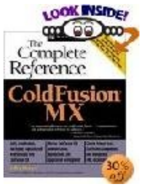


Professional Web Site Optimization by Scott Ware, Michael Tracy, Louis Slothouber, Robert Barker

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.

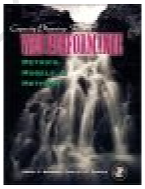


ColdFusion MX: The Complete Reference by Jeffry Houser
Although this book is specifically about ColdFusion, it does have a chapter on performance, and gives details how to both monitor and test the performance of a ColdFusion server. The basics of performance testing with Web Performance Trainer are 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.

WILEY
Testing Applications on the Web
Test Planning for Internet-Based Systems
Hung Q. Nguyen

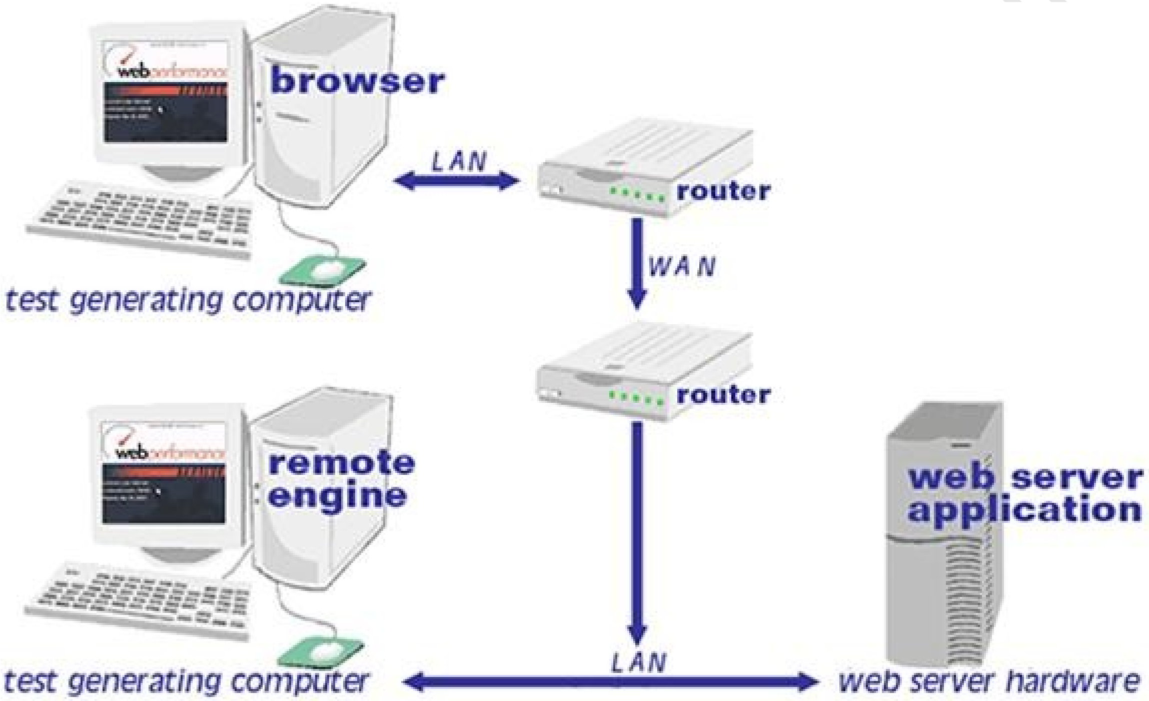
Testing Applications on the Web
Test Planning for Internet-Based Systems by Hung Q. Nguyen
As its title would suggest, this book is all about test planning for all types of internet based systems. Its chapters on performance testing give a lot of details about planning a performance test and analyzing 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.

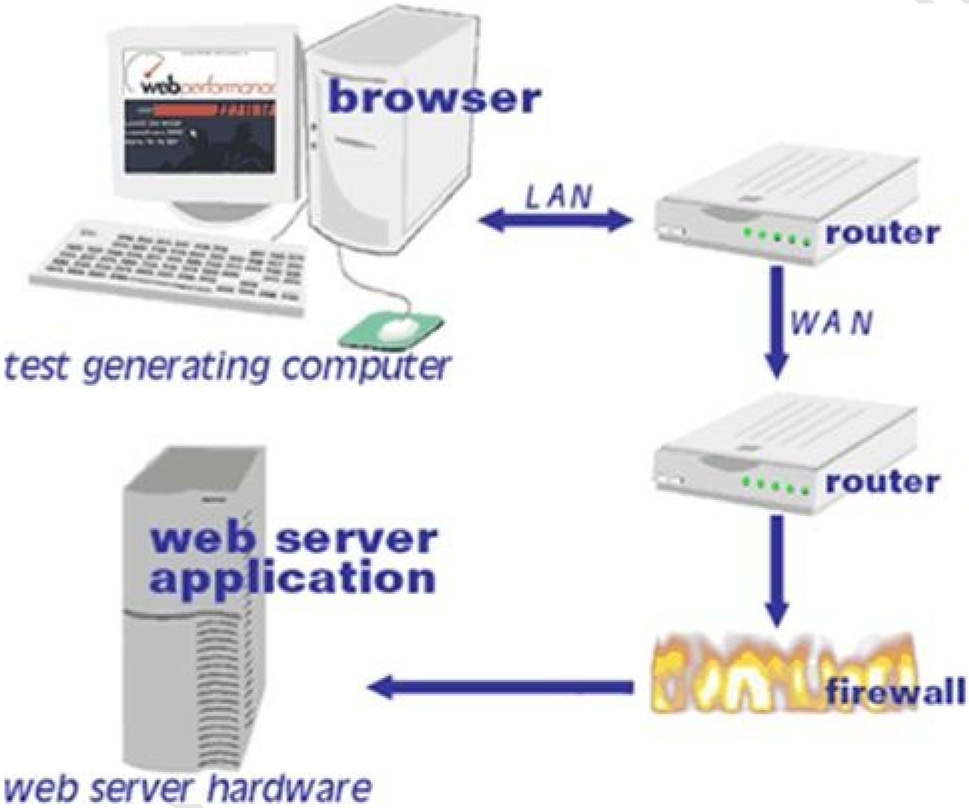
LAN vs. WAN

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.



LAN-based Load Generation

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.



WAN-based Load Generation

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

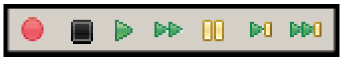
Using the base performance it is possible to estimate the bandwidth required to simulate any number of users, and is used as a ball-park figure to see if the available bandwidth is adequate before starting a load test.

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.

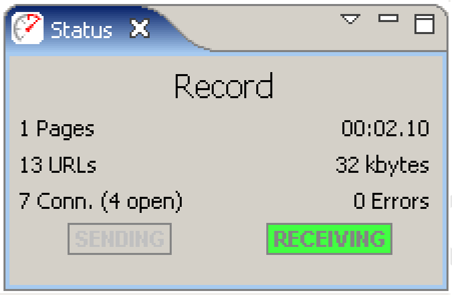
Browser and proxy configuration

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:



```
Record
1 Pages              00:02.10
13 URLs             32 kbytes
7 Conn. (4 open)     0 Errors
```

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.

note: the CER and PEM certificate files may be copied directly from the following folder (where <user> is your windows username) if the download links do not work:

C:\Documents and Settings\<user>\.webperformance

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.

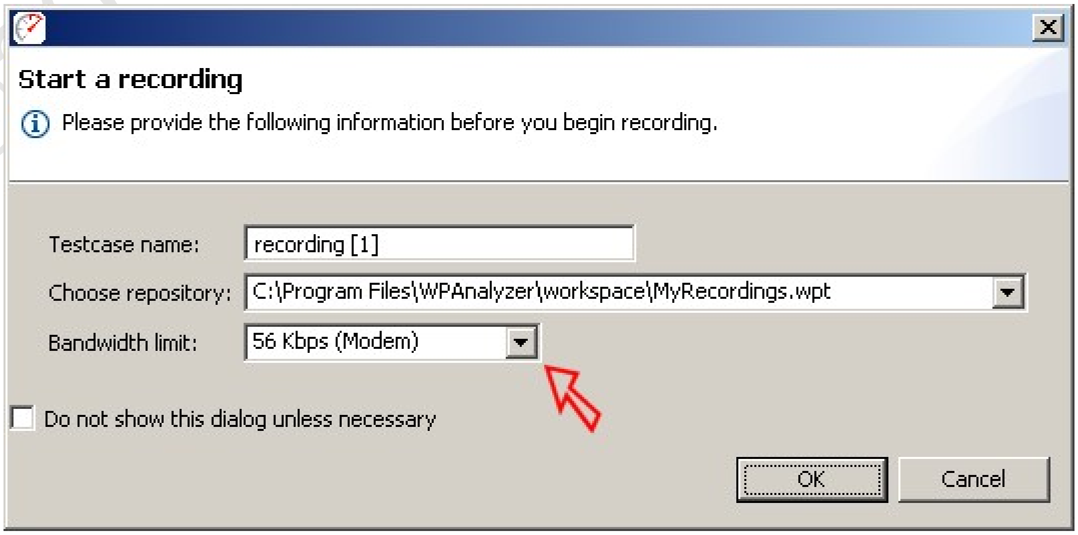
Bandwidth Simulation

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.



Start a recording

Please provide the following information before you begin recording.

Testcase name: recording [1]

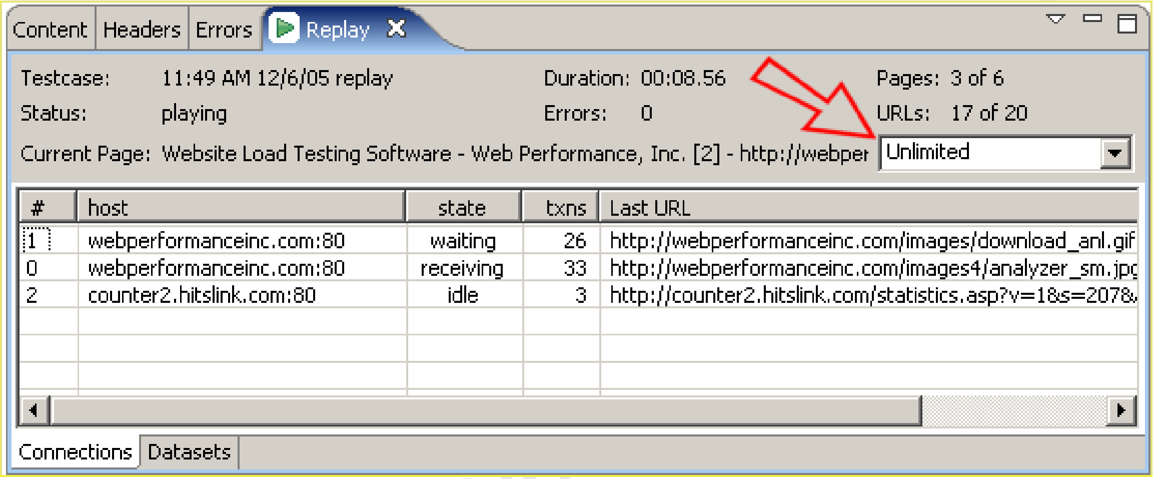
Choose repository: C:\Program Files\WPAnalyzer\workspace\MyRecordings.wpt

Bandwidth limit: 56 Kbps (Modem)

□ Do not show this dialog unless necessary

Replay

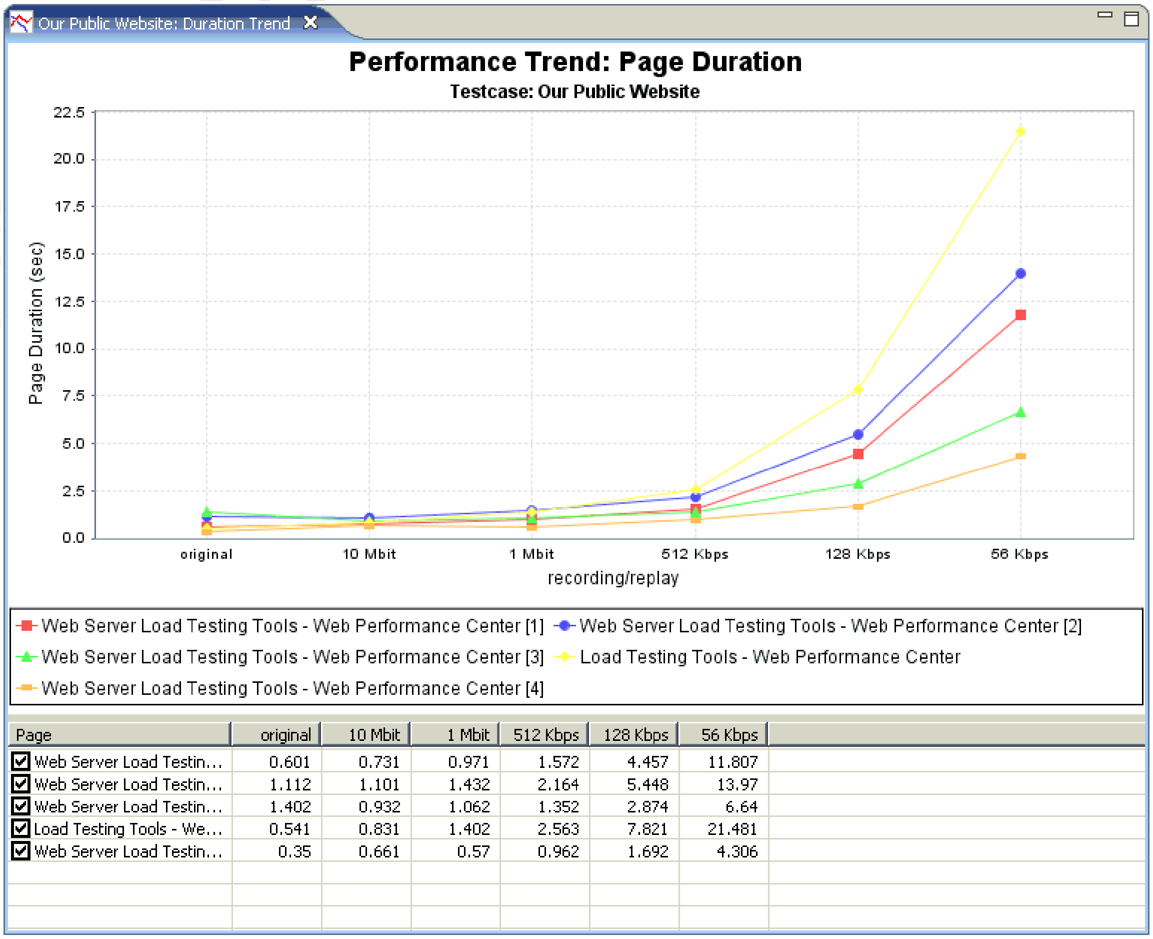
To replay a testcase using bandwidth simulation, open the *Replay view* and select the bandwidth from the list. The simulated bandwidth may be changed at any time during the replay.



Testcase: 11:49 AM 12/6/05 replay
Status: playing
Duration: 00:08.56
Errors: 0
Pages: 3 of 6
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&

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.


Our Public Website: Duration Trend
Performance Trend: Page Duration
Testcase: Our Public Website
Page Duration (sec)
22.5
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
original
10 Mbit
1 Mbit
512 Kbps
128 Kbps
56 Kbps
recording/replay
Web Server Load Testing Tools - Web Performance Center [1]
Web Server Load Testing Tools - Web Performance Center [2]
Web Server Load Testing Tools - Web Performance Center [3]
Load Testing Tools - Web Performance Center
Web Server Load Testing Tools - Web Performance Center [4]

Page	original	10 Mbit	1 Mbit	512 Kbps	128 Kbps	56 Kbps
☑ Web Server Load Testin...	0.601	0.731	0.971	1.572	4.457	11.807
☑ Web Server Load Testin...	1.112	1.101	1.432	2.164	5.448	13.97
☑ Web Server Load Testin...	1.402	0.932	1.062	1.352	2.874	6.64
☑ Load Testing Tools - We...	0.541	0.831	1.402	2.563	7.821	21.481
☑ Web Server Load Testin...	0.35	0.661	0.57	0.962	1.692	4.306

Analyze A Recording

Performance Goals

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

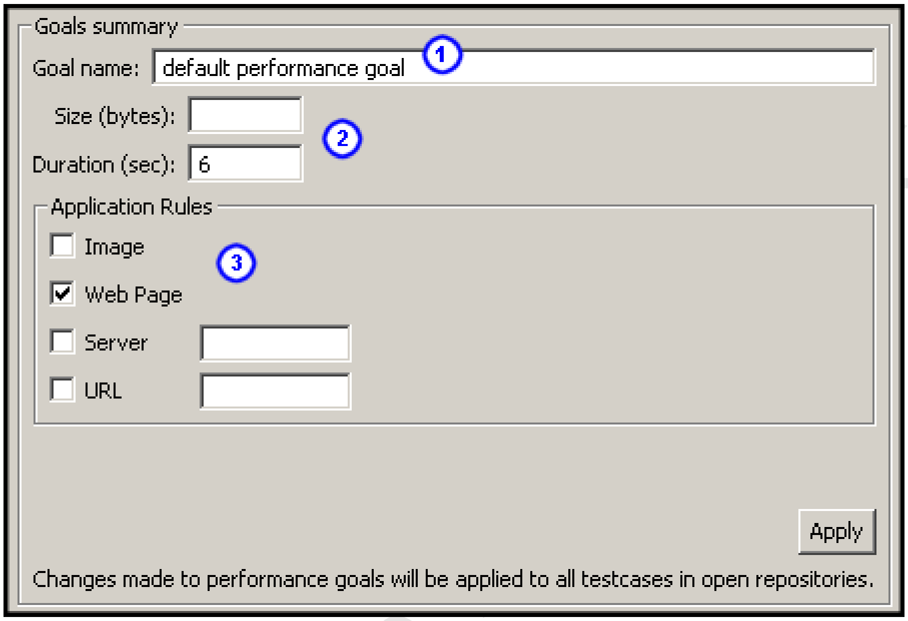
Evaluating Performance Goals

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

Size (bytes):

Duration (sec): 6

Application Rules

☐ Image

☑ Web Page

☐ Server

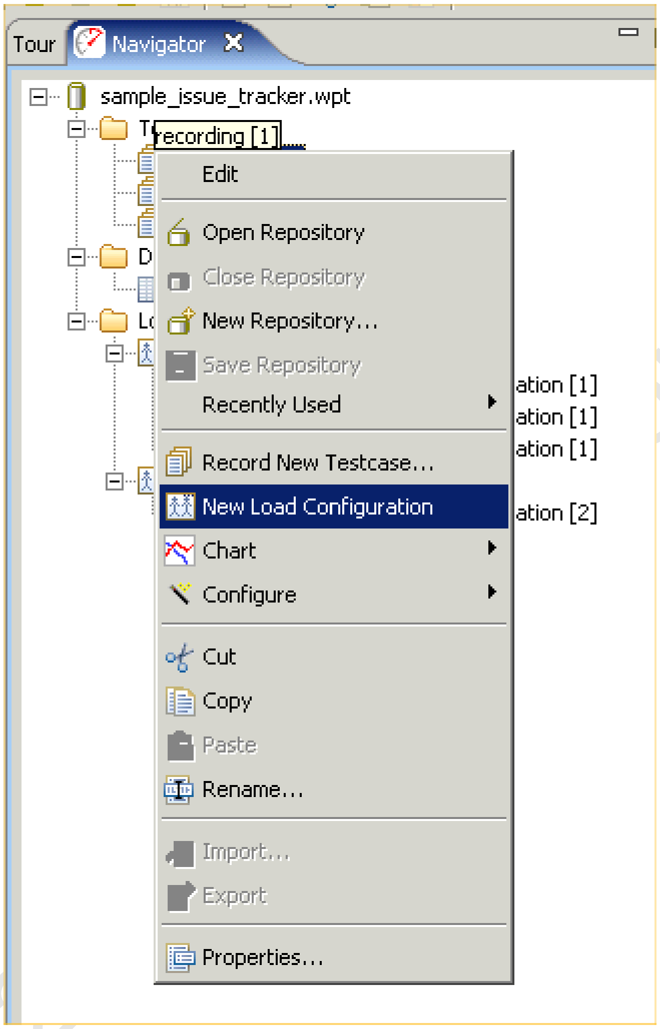
☐ URL

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

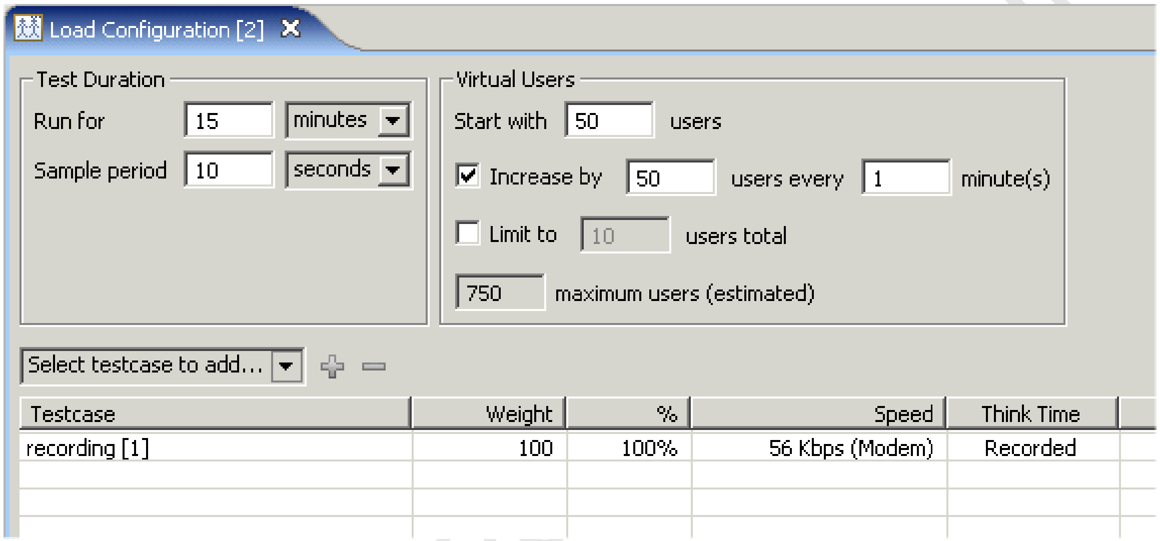
Baseline Analysis

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.



A new Load Configuration window will appear as shown below. This can be configured using the information in the Load Test Configuration Editor section of the Reference Manual.



Load Configuration [2]

Test Duration

Run for 15 minutes

Sample period 10 seconds

Virtual Users

Start with 50 users

☑ Increase by 50 users every 1 minute(s)

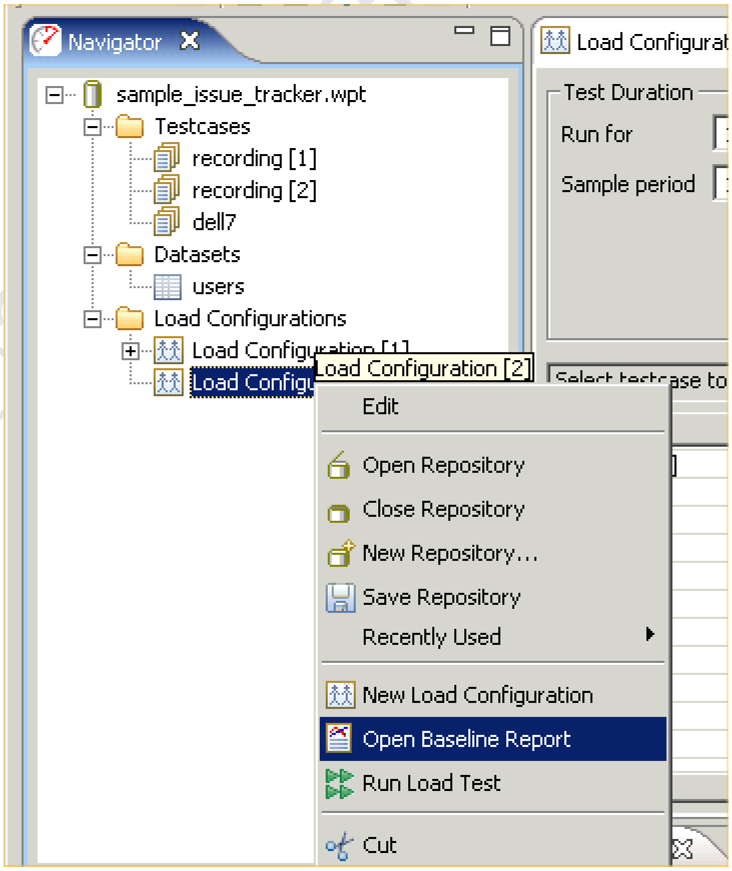
☐ Limit to 10 users total

750 maximum users (estimated)

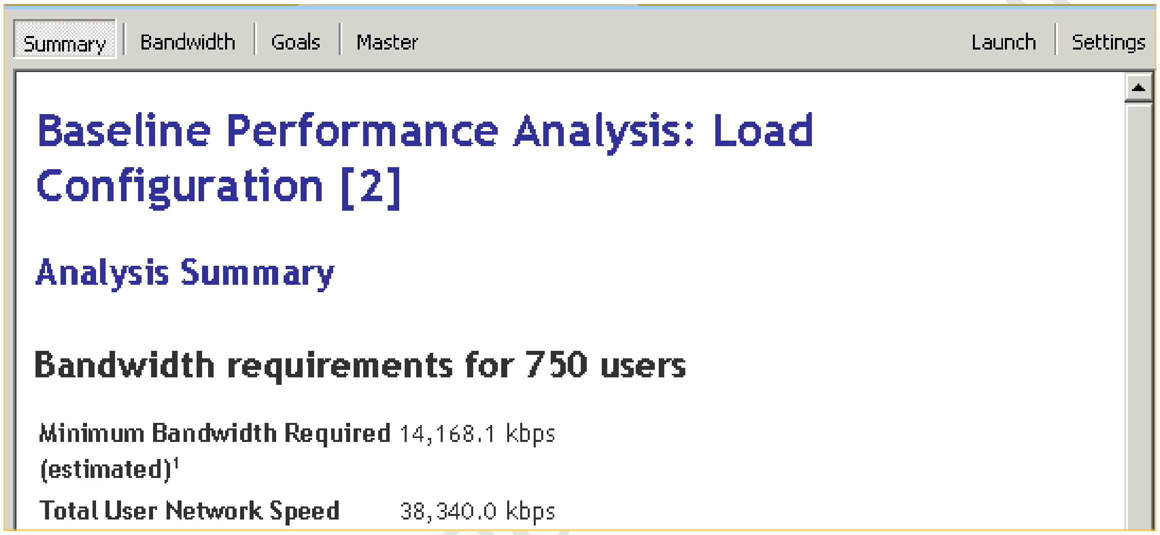
Select testcase to add...

Testcase	Weight	%	Speed	Think Time
recording [1]	100	100%	56 Kbps (Modem)	Recorded

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:



Baseline Performance Analysis: Load Configuration [2]

Analysis Summary

Bandwidth requirements for 750 users

Minimum Bandwidth Required (estimated)[1]	14,168.1 kbps
Total User Network Speed	38,340.0 kbps

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

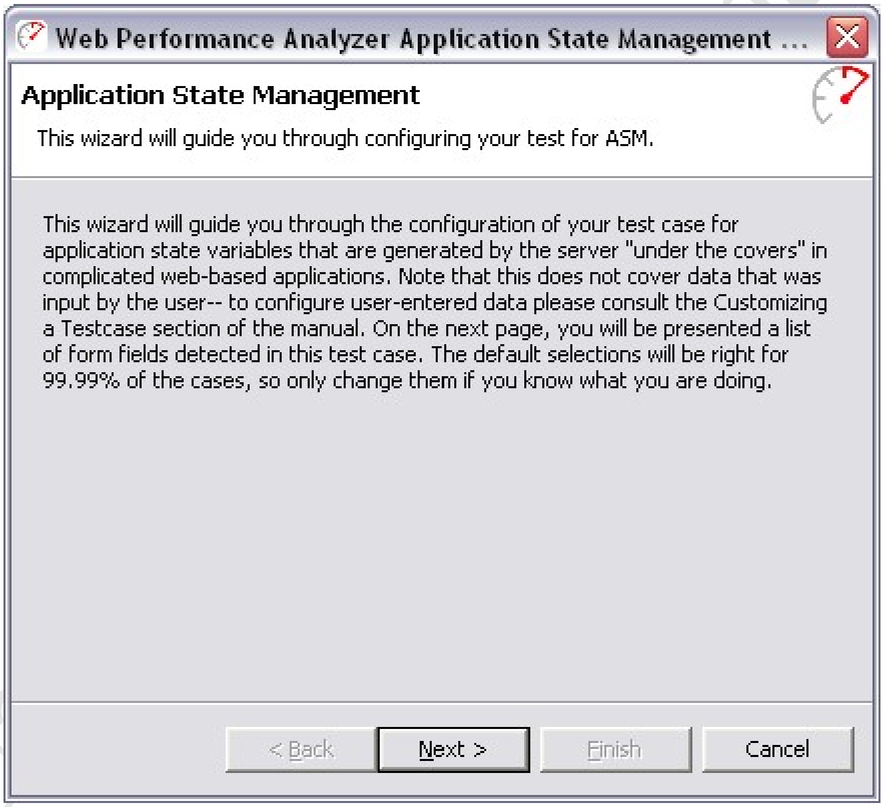
Application State Management

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

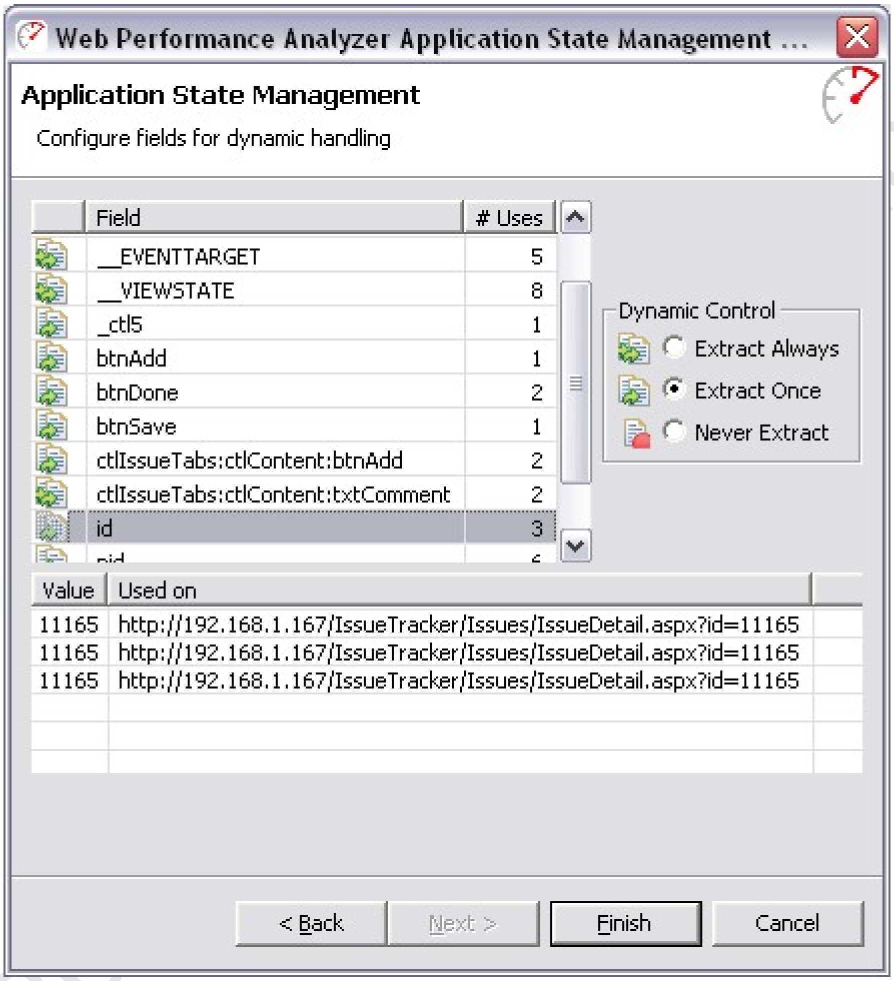


Application State Management

This wizard will guide you through configuring your test for ASM.

This wizard will guide you through the configuration of your test case for application state variables that are generated by the server "under the covers" in complicated web-based applications. Note that this does not cover data that was input by the user-- to configure user-entered data please consult the Customizing a Testcase section of the manual. On the next page, you will be presented a list of form fields detected in this test case. The default selections will be right for 99.99% of the cases, so only change them if you know what you are doing.

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:



Application State Management

Configure fields for dynamic handling

Field	# Uses
__EVENTTARGET	5
__VIEWSTATE	8
_ctl5	1
btnAdd	1
btnDone	2
btnSave	1
ctlIssueTabs:ctlContent:btnAdd	2
ctlIssueTabs:ctlContent:txtComment	2
id	3
nid	6

Dynamic Control

○ Extract Always
● Extract Once
○ Never Extract

Value	Used on
11165	http://192.168.1.167/IssueTracker/Issues/IssueDetail.aspx?id=11165
11165	http://192.168.1.167/IssueTracker/Issues/IssueDetail.aspx?id=11165
11165	http://192.168.1.167/IssueTracker/Issues/IssueDetail.aspx?id=11165

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 dynamically, 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 Web Performance Suite 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 pages in the replay, looking for error messages from the server. It may be useful 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.

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, 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
5. File uploads
6. Multipart related content, or raw content blocks

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 from the Fields View.

File Upload Fields

A file upload field is part of an HTTP POST where the application is sending a file to the server. These fields can be modified for data replacement using the Fields

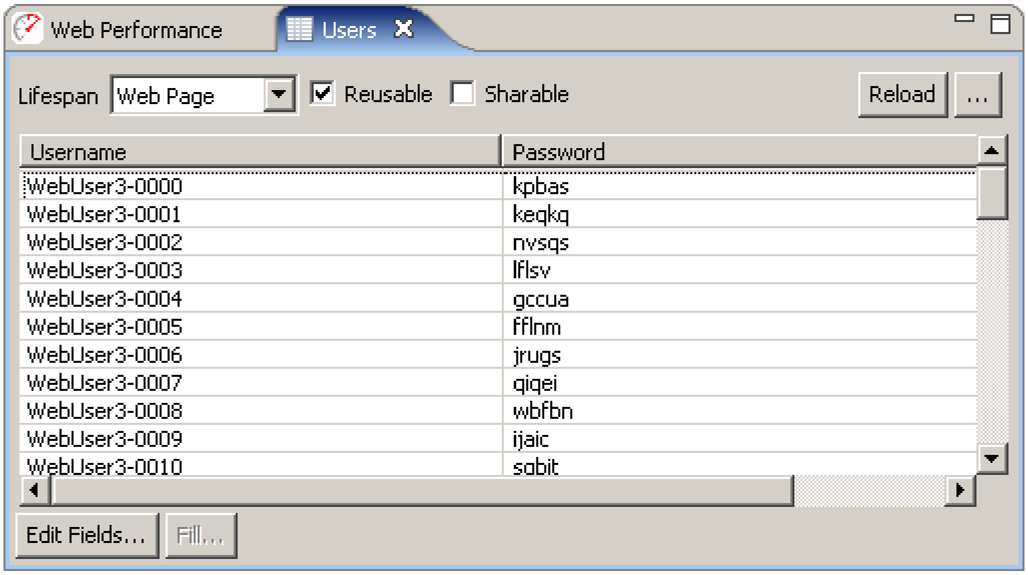
View. For more detailed information on setting up data replacement, see File Uploads.

Multipart related content, or raw content blocks
When the HTTP request contains the content-type multipart/related or post a single raw block of data, the part can be modified for data replacement using the Fields View. For more detailed information on setting up data replacement, see Part/Post Content.

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.

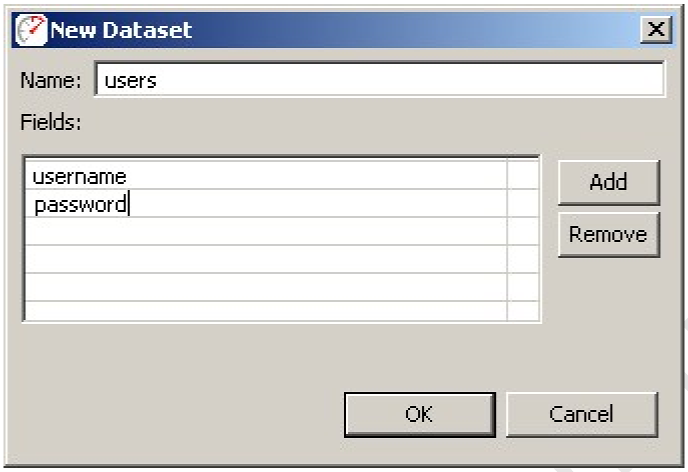


Lifespan Web Page ☑ Reusable □ Sharable

Username	Password
WebUser3-0000	kpbas
WebUser3-0001	keqkq
WebUser3-0002	nvsqs
WebUser3-0003	lflsv
WebUser3-0004	gccua
WebUser3-0005	fflnm
WebUser3-0006	jrugs
WebUser3-0007	qiqei
WebUser3-0008	wbfbn
WebUser3-0009	ijaic
WebUser3-0010	sqbit

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:



Name: users

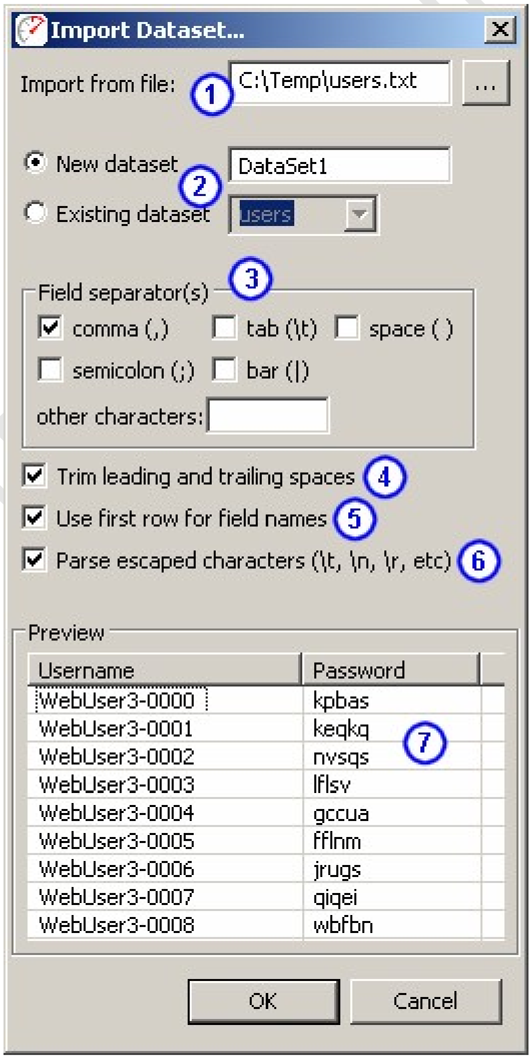
Fields:
username
password

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. If your file contains characters that are escaped, you may select this option to parse them as escaped characters. This is useful if you must import values which span multiple lines. Simply ensure that in your file, each row in the dataset appears on it's own line, and that line breaks within individual values are replaced with the characters "\r\n" (or an appropriate new line sequence for your application server). Once imported with the "Parse escaped characters" option, tooltips over each dataset value will display the complete value with line breaks.
7. 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.



Import Dataset...

Import from file: (1) C:\Temp\users.txt ...

● New dataset (2) DataSet1
○ Existing dataset users

Field separator(s) (3)
☑ comma (,) □ tab (\t) □ space ( )
□ semicolon (;) □ bar (|)
other characters:

☑ Trim leading and trailing spaces (4)
☑ Use first row for field names (5)
☑ Parse escaped characters (\t, \n, \r, etc) (6)

Preview (7)

Username	Password
WebUser3-0000	kpbas
WebUser3-0001	keqkq
WebUser3-0002	nvsqs
WebUser3-0003	lflsv
WebUser3-0004	gccua
WebUser3-0005	fflnm
WebUser3-0006	jrugs
WebUser3-0007	qiqei
WebUser3-0008	wbfbn

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.

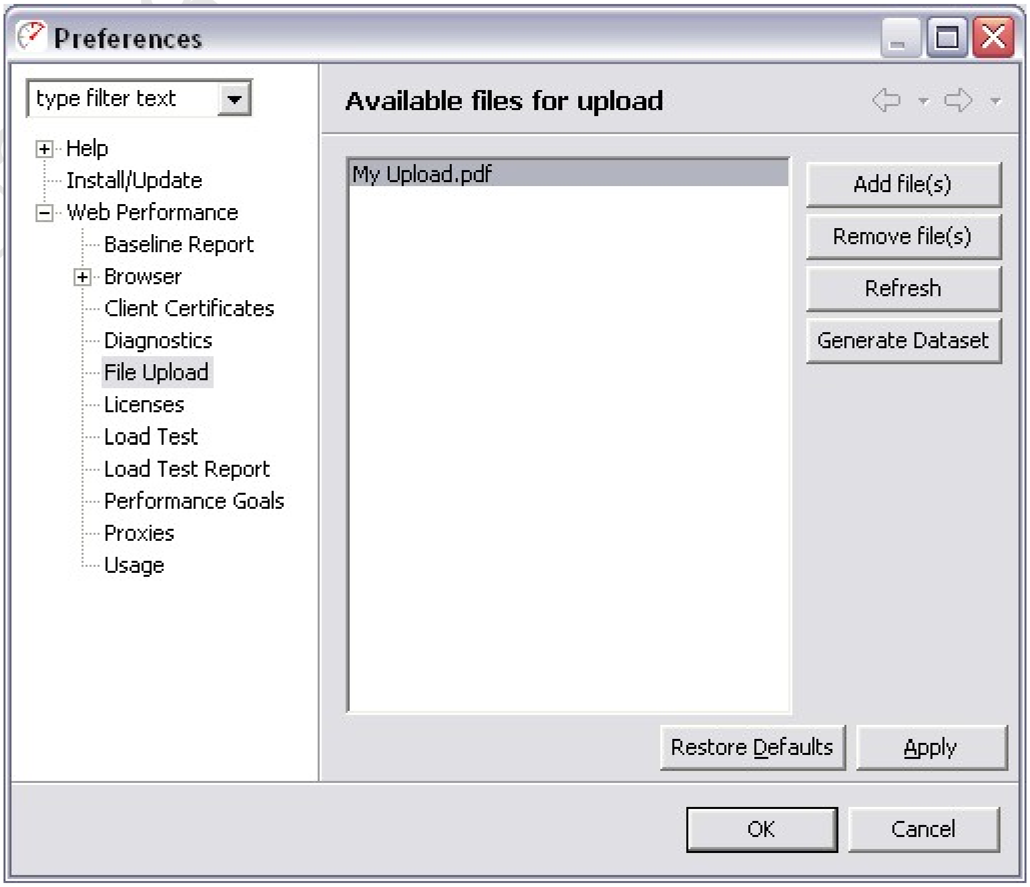
File Uploads

If the application includes an operation involving the upload of a file from the users computer to the server, it may be desirable to provide different files during a simulation. Configuration to simulate the upload of different data files during a load test is broken down into three steps:

1. Import the data files you would like to use during a load test into Load Tester.
2. Create a dataset of filenames specifying the specific files that will be used for the load test.
3. Create a modifier to have the Virtual User replace the recorded file content with one of the selected files during playback.

Importing files

A predefined set of files must be imported into Load Tester through the File Uploads preferences page. This page may be accessed from the menu: Window » Preferences&help; » Web Performance » File Uploads .


Preferences

Available files for upload

My Upload.pdf

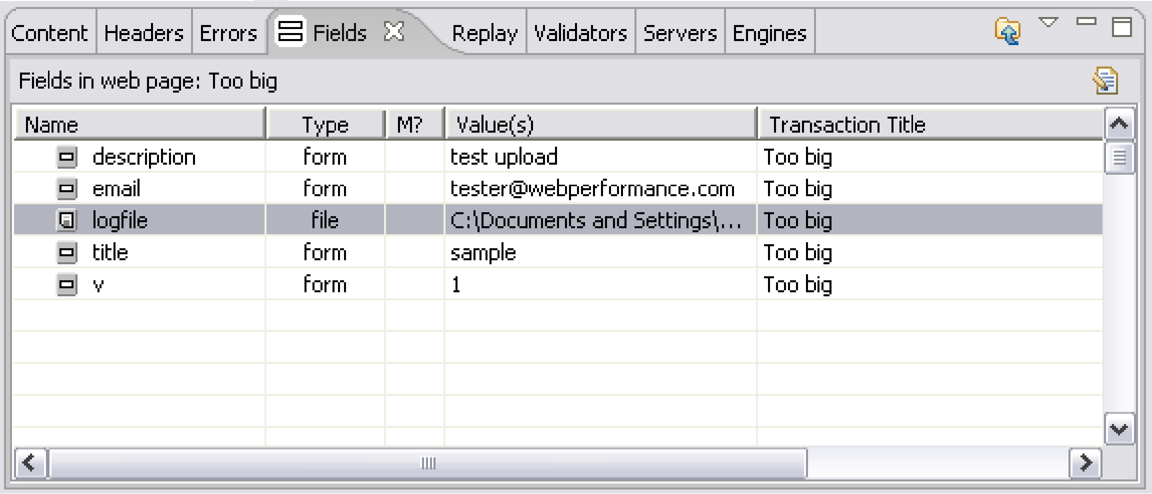
Once on the File Upload page, you may select the "Add file(s)" button to select files for Load Tester to keep available for use during a load test. Each file will be copied into a private location from which they may be synchronized with any remote load engines in use prior to a load test. If a file is changed after adding it, the "Add file(s)" button may be used to replace the copy used for load testing with the updated copy.

Creating a Dataset

The simplest way to create a dataset is from the same File Uploads preferences page described above. By selecting the "Generate Dataset" button, Load Tester will offer to create a dataset of all the files currently available using the repository and names specified.
Alternatively, if the intended load profile consists of different test cases where some files may be used in one test case and not another, then it may be necessary to customize a dataset to specify the subset of files that should be used. Please see Create a new dataset for more information.

Creating a Modifier

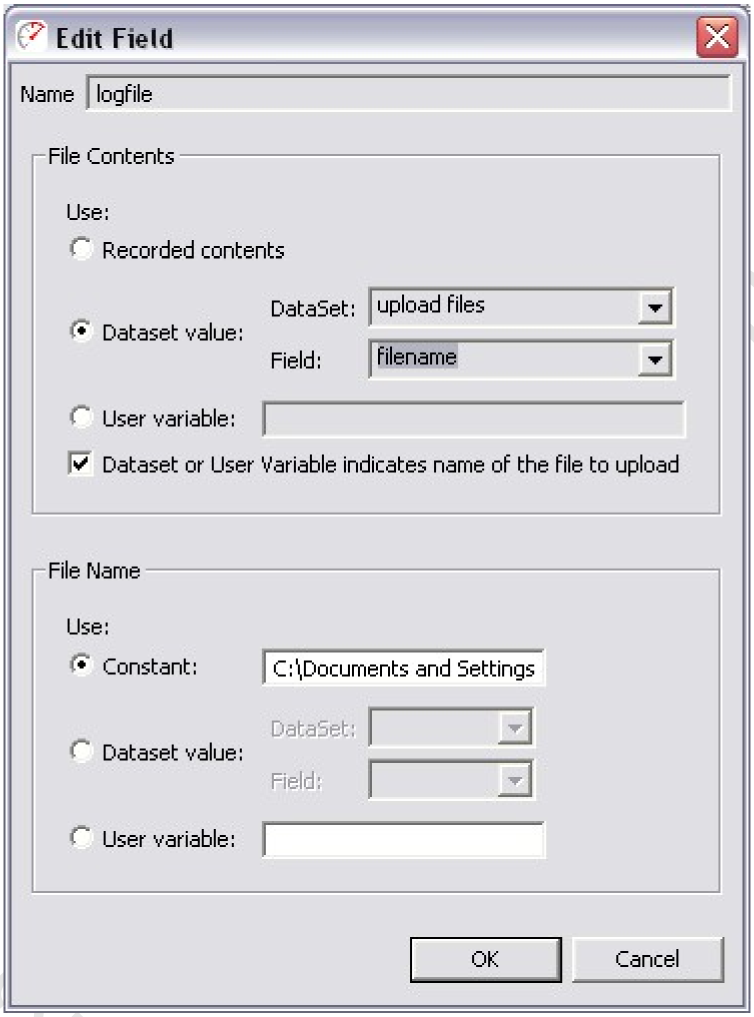
Once files have been successfully imported and a dataset prepared, we will be ready to proceed to the Fields View to locate a file upload to be updated.



Fields in web page: Too big

Name	Type	M?	Value(s)	Transaction Title
description	form		test upload	Too big
email	form		tester@webperformance.com	Too big
logfile	file		C:\Documents and Settings\...	Too big
title	form		sample	Too big
v	form		1	Too big

When editing the modifier of a file field, a dialog will be displayed giving the option to modify the file content, as well as the name of the file sent to the server.



Name: logfile

File Contents

Use:

○ Recorded contents

● Dataset value: DataSet: upload files; Field: filename

○ User variable:

☑ Dataset or User Variable indicates name of the file to upload

File Name

Use:

● Constant: C:\Documents and Settings

○ Dataset value: DataSet: ; Field:

○ User variable:

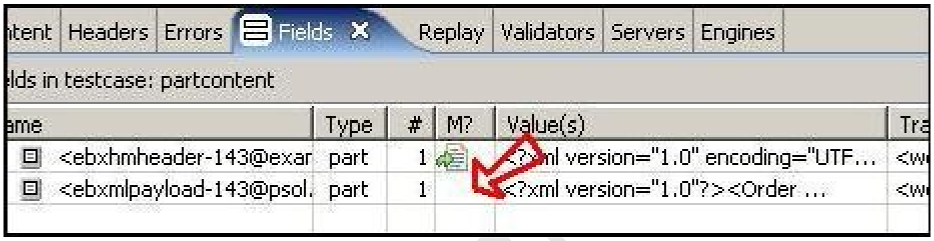
The "File Contents" section of the screen allows us to replace the actual file data that will be sent to the server during a load test. The default way to replace the data is to select the "Dataset value:" option, and select the dataset and field containing the names of files to be uploaded. The option specifying that "Dataset or User Variable indicates name of file to upload" should be checked.

Lastly, the File Name section of the screen can be used to adjust the name of the file sent to the server, if required. This may be the same dataset field as specified above if just the name of the file (excluding the path name) is sufficient, or an alternate dataset field containing a complete file name may be used.

Content Modifiers

Configuring replacement of part content is performed from the Fields View. If an HTTP request contains the content-type multipart/related, it is displayed in the Fields View with type *part*. For other styles of HTTP requests, such as those that post a single raw content block, that block will be displayed in the Fields View with type *post*.

Double clicking on either the modifier or value column opens a dialog where the replacement data is specified.



lds in testcase: partcontent

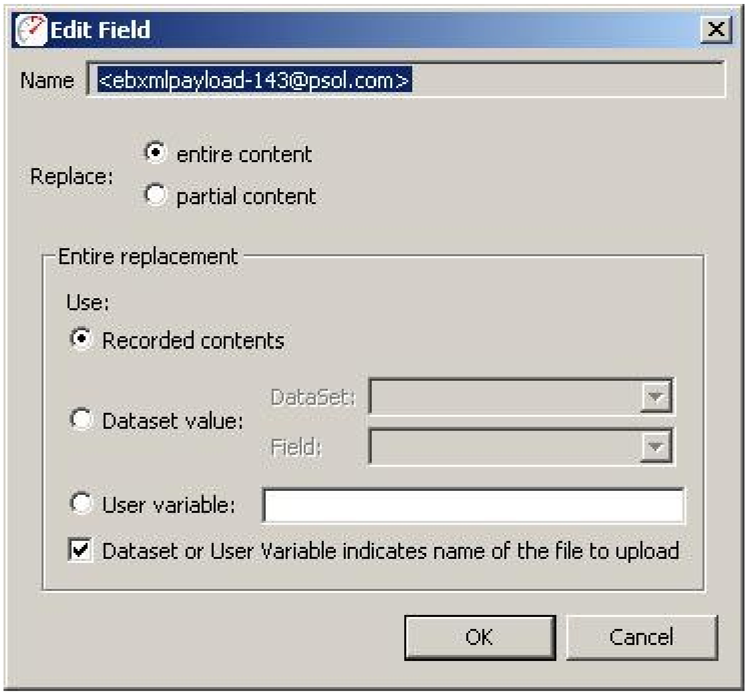
ame	Type	#	M?	Value(s)	Tra
<ebxhmheader-143@exar	part	1		<?xml version="1.0" encoding="UTF...	<w
<ebxmlpayload-143@psol.	part	1		<?xml version="1.0"?><Order ...	<w

The replacement configurations allowed are:

1. Replacing the entire content of the part from a file or dataset field.
2. Replacing multiple smaller sections of the part from datasets or user variables.

Entire content replacement
When replacing the entire content during a load test, the files to be used for data replacement can be specified in a dataset. The files must be imported into Load Tester through the File Uploads preference page. See the section on uploading files for a detailed explanation on using the preference page and creating a dataset.
Adding the modifier

Once your dataset is created, use the Fields View to open the dialog to edit the modifiers for the part:



Name <ebxmlpayload-143@psol.com>

Replace: ● entire content ○ partial content

Entire replacement

Use:

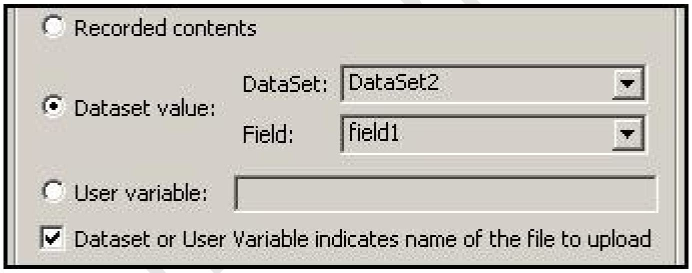
● Recorded contents

○ Dataset value: DataSet: Field:

○ User variable:

☑ Dataset or User Variable indicates name of the file to upload

Select the *Dataset value* button, and select the Dataset and field which contain the replacement information for this part. If the new values for each part are each in their own file, and configured through the File Uploads Preferences, then the option "Dataset or User Variable indicates name of file to upload" should be checked. Otherwise, if the Dataset contains exactly the values to be used during playback, this option may be left unchecked.



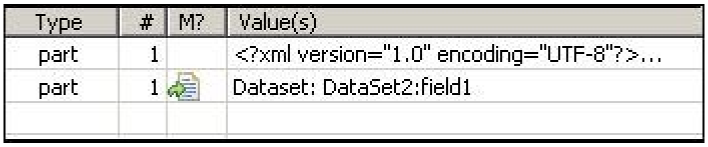
○ Recorded contents

● Dataset value: DataSet: DataSet2 Field: field1

○ User variable:

☑ Dataset or User Variable indicates name of the file to upload

After pressing *OK*, the new modifier is shown in the Fields View:



Type	#	M?	Value(s)
part	1		<?xml version="1.0" encoding="UTF-8"?>...
part	1		Dataset: DataSet2:field1

Removing the modifier

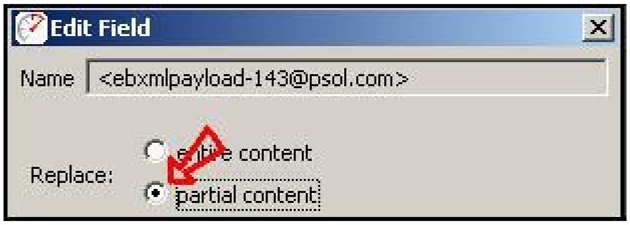
To remove the modifier, re-open the dialog from the FieldsView and select the *Recorded contents* option and press *OK* button.

Partial content replacement

If only sections of the part require data replacement, those modifiers are also configured from the Fields View.

Adding the modifier

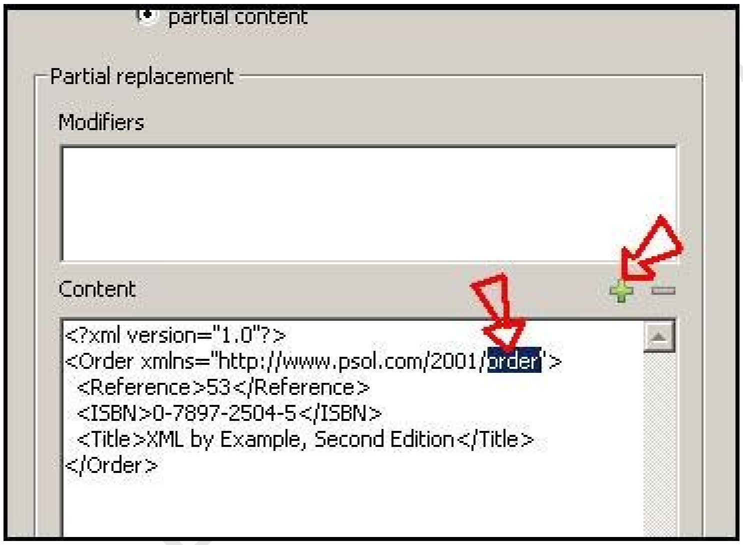
Open the modifer dialog and select the *partial content* button



Name <ebxmlpayload-143@psol.com>

Replace: ○ entire content ● partial content

The dialog changes to display the content. The top area shows modifiers configured on the part and the lower text area is used to select the sections of text to be modified. Select the section of text to be modified and click the + button to add a modifier.



◉ partial content

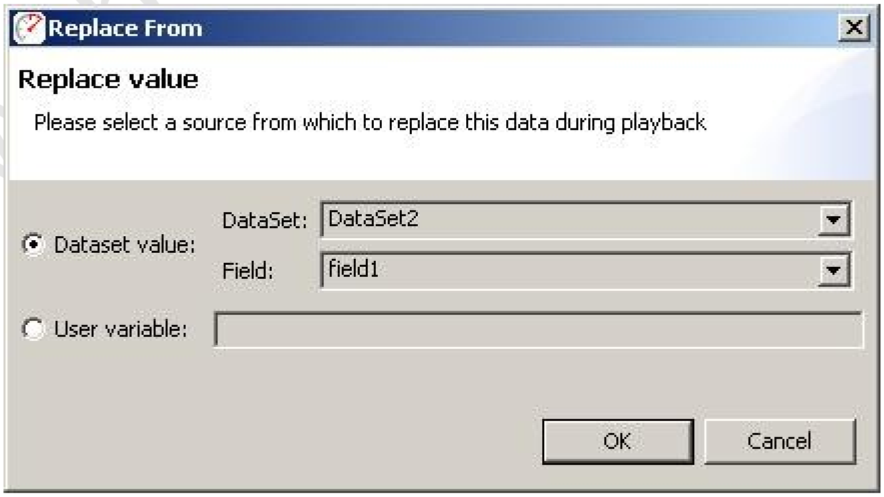
Partial replacement

Modifiers

Content

```
<?xml version="1.0"?>
<Order xmlns="http://www.psol.com/2001/order">
  <Reference>53</Reference>
  <ISBN>0-7897-2504-5</ISBN>
  <Title>XML by Example, Second Edition</Title>
</Order>
```

A new dialog is displayed where the dataset or user variable to use for the replacement must be specified. Make the selections on this dialog and press the *OK* button to create the modifier.



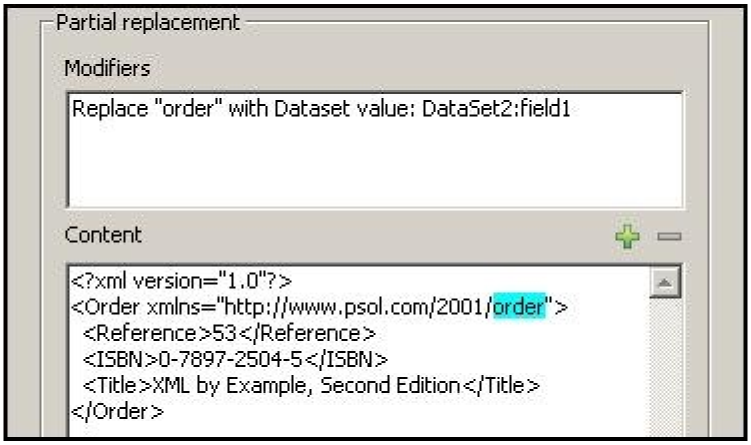
Replace value

Please select a source from which to replace this data during playback

◉ Dataset value: DataSet: DataSet2 Field: field1

○ User variable:

The modifier is now shown on the original dialog. Additional modifiers can be specified by repeating the previous step. Once all desired modifiers have been configured, press the *OK* button to add the modifiers to the



Partial replacement

Modifiers

Replace "order" with Dataset value: DataSet2:field1

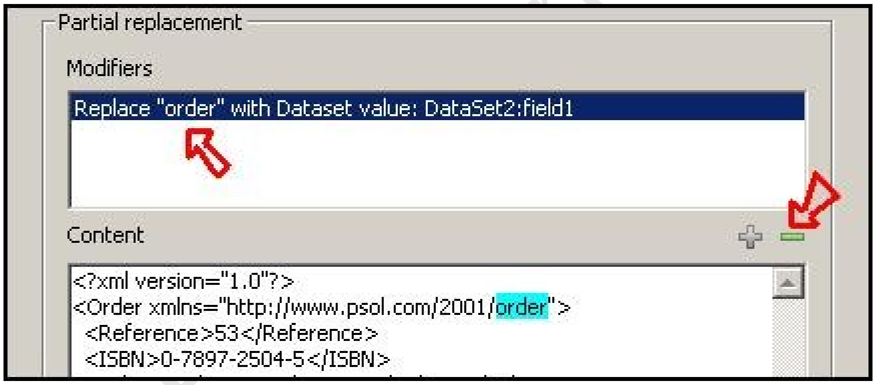
Content

```
<?xml version="1.0"?>
<Order xmlns="http://www.psol.com/2001/order">
  <Reference>53</Reference>
  <ISBN>0-7897-2504-5</ISBN>
  <Title>XML by Example, Second Edition</Title>
</Order>
```

testcase.

Removing a modifier

To remove a modifier, re-open the dialog from the FieldsView. Selecting the modifier to be removed in the list at the top of the dialog and select the - button. Press the *OK* button to make the change to the testcase.



Replace "order" with Dataset value: DataSet2:field1

```
<?xml version="1.0"?>
<Order xmlns="http://www.psol.com/2001/order">
  <Reference>53</Reference>
  <ISBN>0-7897-2504-5</ISBN>
```

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.

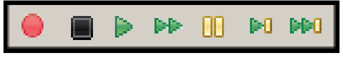
Configuration

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").

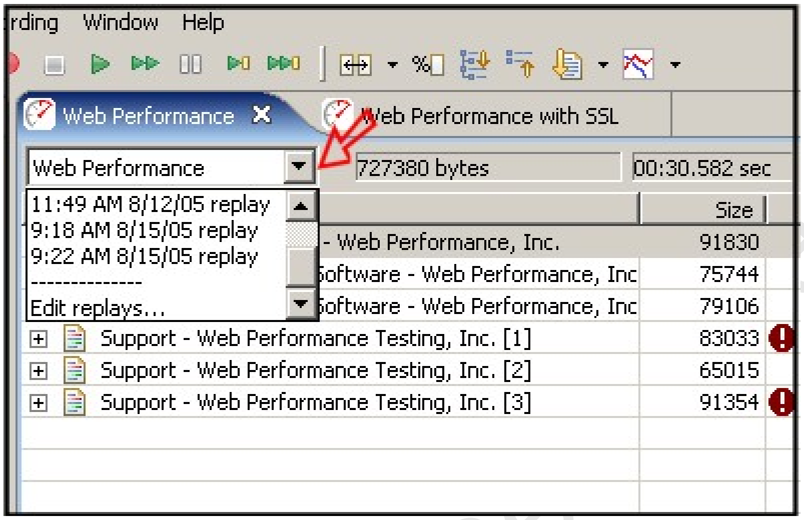
⏩ Fast Play - Replays the testcase without think time between pages.

⏸ 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.



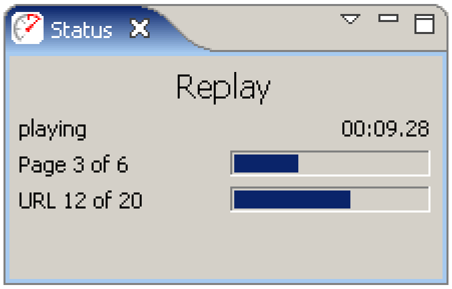
Web Performance ▾ | 727380 bytes | 00:30.582 sec

- 11:49 AM 8/12/05 replay
- 9:18 AM 8/15/05 replay
- 9:22 AM 8/15/05 replay
- --------------
- Edit replays...

	Size	
- Web Performance, Inc.	91830	
Software - Web Performance, Inc	75744	
Software - Web Performance, Inc	79106	
Support - Web Performance Testing, Inc. [1]	83033	!
Support - Web Performance Testing, Inc. [2]	65015	
Support - Web Performance Testing, Inc. [3]	91354	!

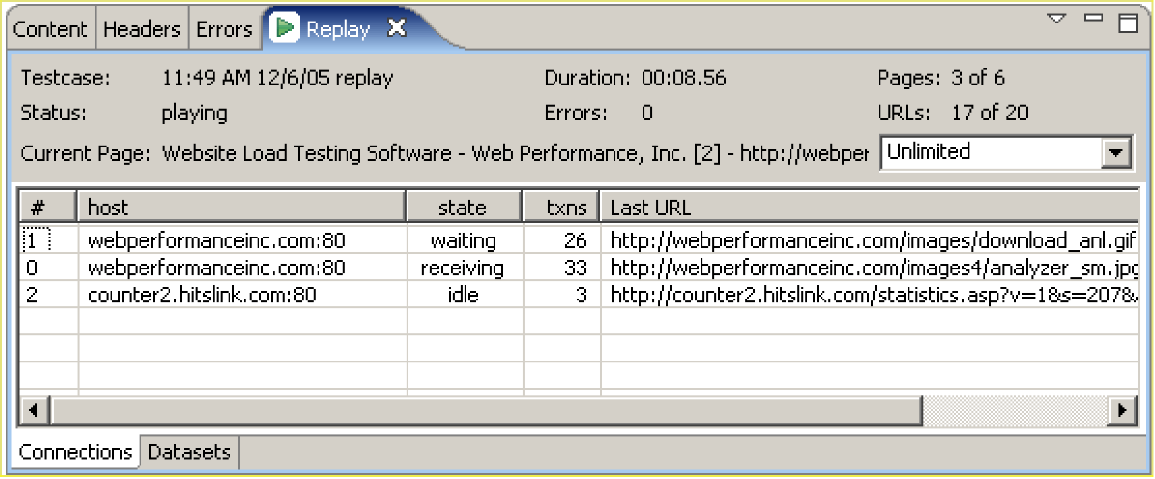
Replay status

The current status of the replay will be displayed in the Status View.



playing 00:09.28
Page 3 of 6
URL 12 of 20

More detailed information about a replay is available in the Replay View.



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&

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 for improvement, however, 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”.

Usernames & 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

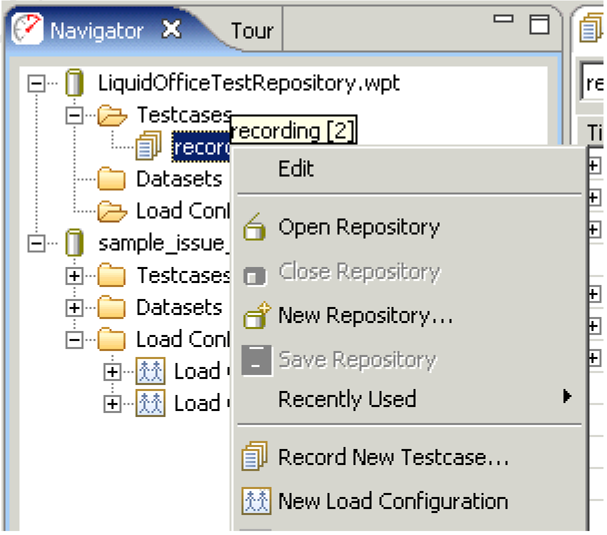
Deliverables

- A full performance report including a “how many users can your website handle” analysis.
- Recommendations for improving performance.

Load Testing

Configuring a Load Test

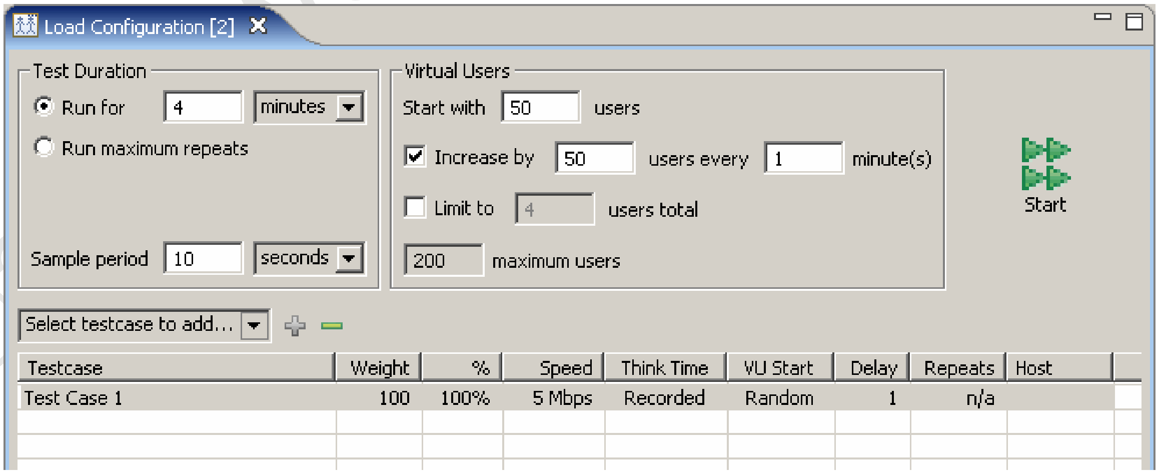
The first step in configuring a load test is to select a test case, and use the right-click menu to select New Load Configuration:



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 Duration
● Run for 4 minutes
○ Run maximum repeats
Sample period 10 seconds

Virtual Users
Start with 50 users
☑ Increase by 50 users every 1 minute(s)
□ Limit to 4 users total
200 maximum users

Testcase	Weight	%	Speed	Think Time	VU Start	Delay	Repeats	Host
Test Case 1	100	100%	5 Mbps	Recorded	Random	1	n/a	

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.

Alternatively, it is also possible to have a test stop after repeating a fixed number of times. This approach allows the test to continue running for as long as the server requires, until the test has been attempted at least as many times as specified in the limit (or until the test is stopped by the user).

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.

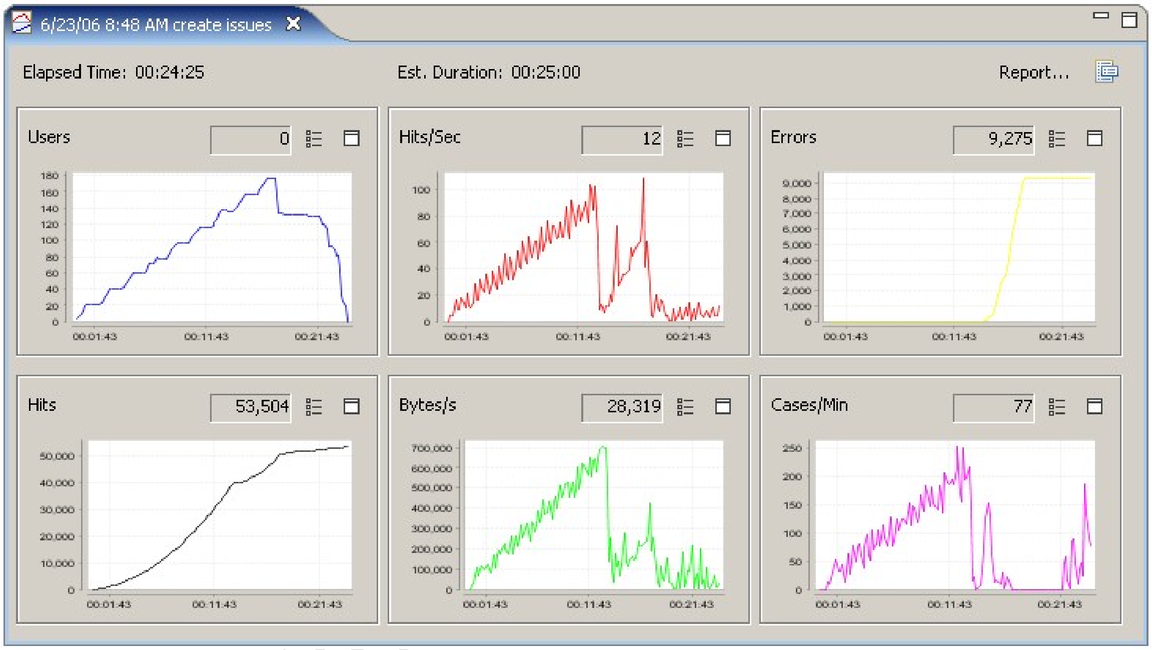
Sample Period

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 5 seconds.

For more information, please consult the section for the Load Test Configuration Editor.

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:



6/23/06 8:48 AM create issues

Elapsed Time: 00:24:25

Est. Duration: 00:25:00

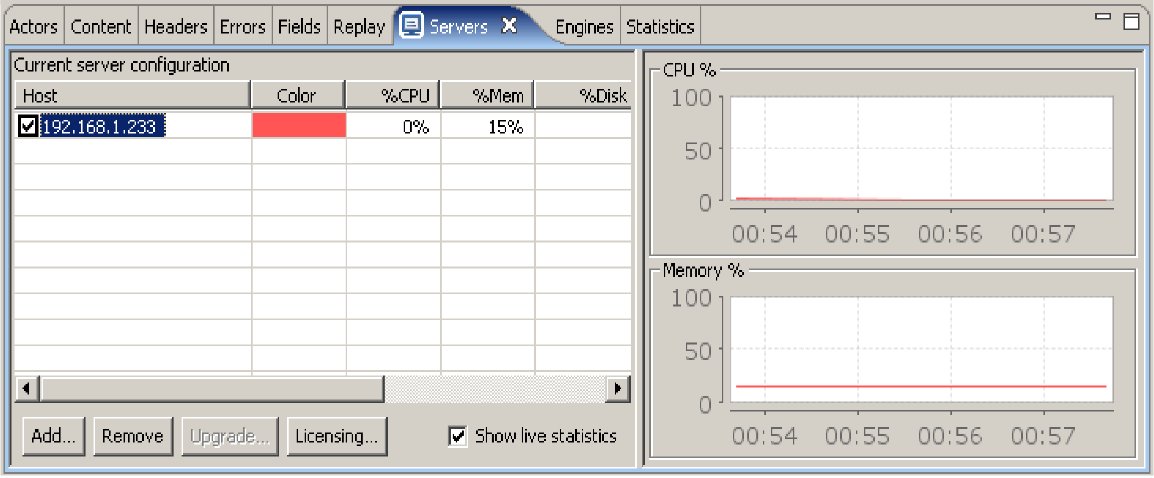
Report...

Users	Hits/Sec	Errors
0	12	9,275

Hits	Bytes/s	Cases/Min
53,504	28,319	77

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.

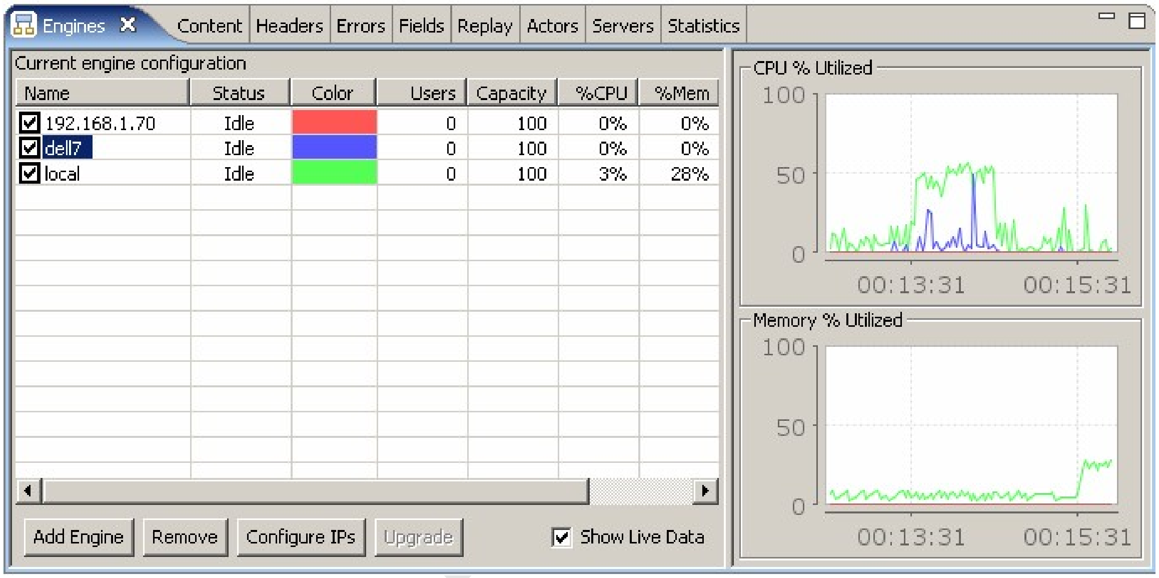
While the test is running you'll want to monitor the performance of your web servers using the Servers View:



Current server configuration

Host	Color	%CPU	%Mem	%Disk
☑ 192.168.1.233	🟥	0%	15%	

It is also important to monitor the performance of the computers generating the virtual users in the Engines View:



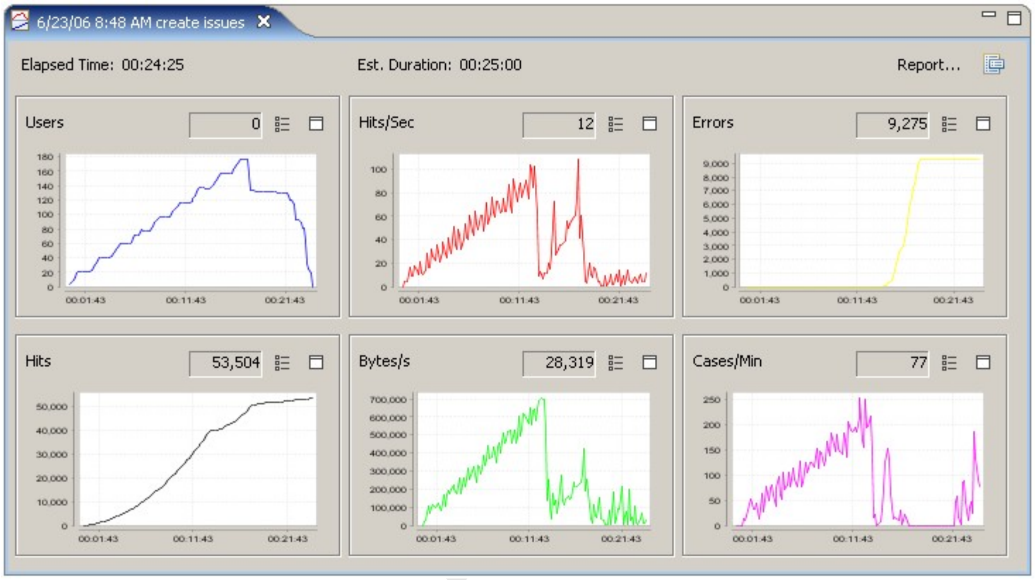
Current engine configuration

Name	Status	Color	Users	Capacity	%CPU	%Mem
☑ 192.168.1.70	Idle	🟥	0	100	0%	0%
☑ dell7	Idle	🟦	0	100	0%	0%
☑ local	Idle	🟩	0	100	3%	28%

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:



6/23/06 8:48 AM create issues

Elapsed Time: 00:24:25

Est. Duration: 00:25:00

Users: 0

Hits/Sec: 12

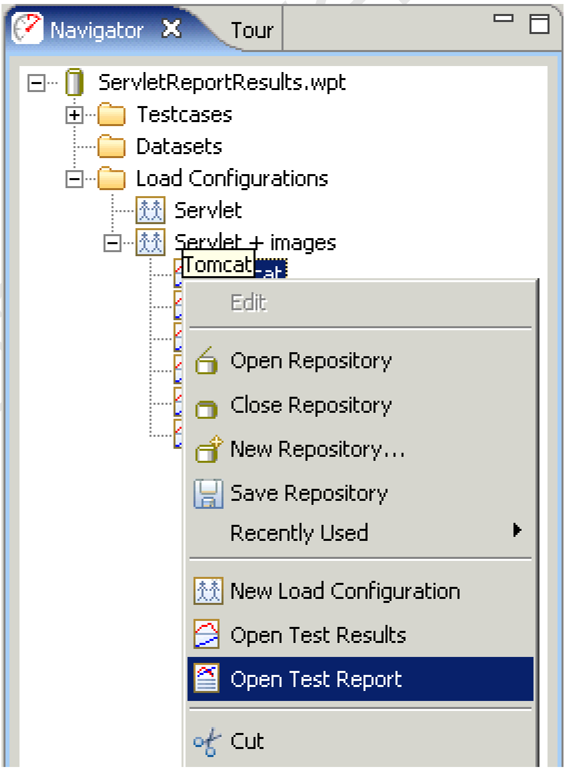
Errors: 9,275

Hits: 53,504

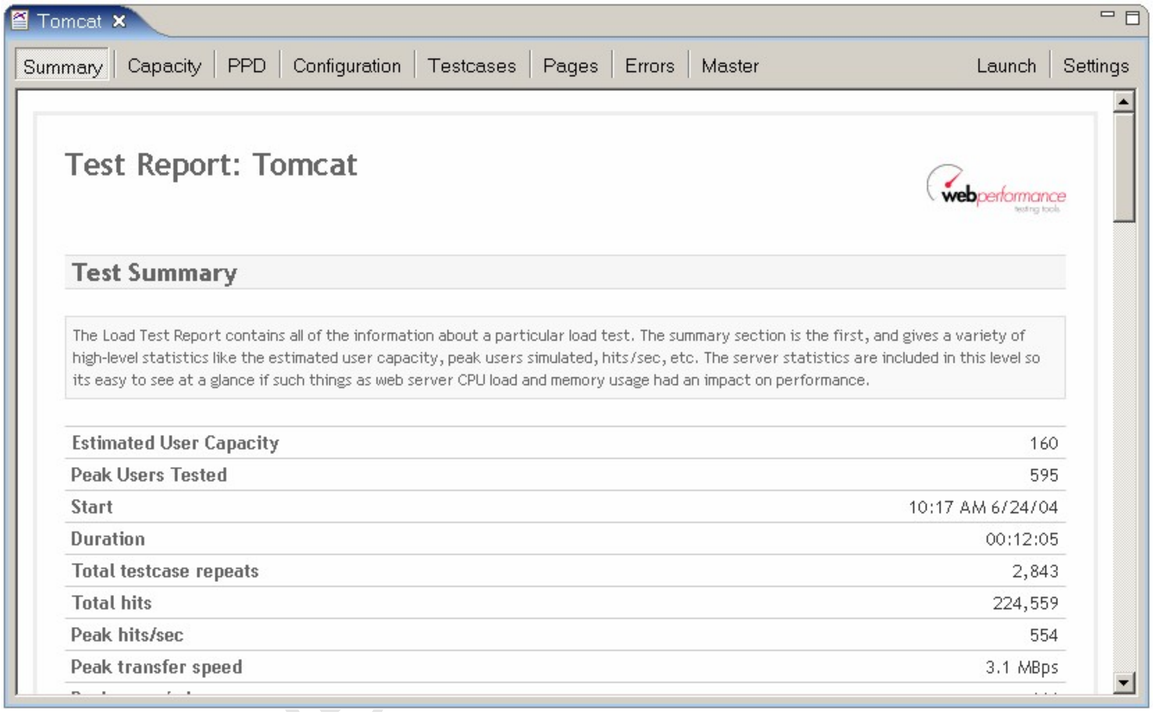
Bytes/s: 28,319

Cases/Min: 77

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



The Load Test Report view will be displayed:



Test Report: Tomcat

Test Summary

The Load Test Report contains all of the information about a particular load test. The summary section is the first, and gives a variety of high-level statistics like the estimated user capacity, peak users simulated, hits/sec, etc. The server statistics are included in this level so its easy to see at a glance if such things as web server CPU load and memory usage had an impact on performance.

Estimated User Capacity	160
Peak Users Tested	595
Start	10:17 AM 6/24/04
Duration	00:12:05
Total testcase repeats	2,843
Total hits	224,559
Peak hits/sec	554
Peak transfer speed	3.1 MBps

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 - 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.

Advanced Configuration Guide

Advanced Testcase Configuration

The Web Performance wizards have been developed to automatically configure testcases for the majority of web-based systems -- especially those based on popular application frameworks. However, some application frameworks and custom-coded applications use techniques that are not (yet) recognized by the wizards.

The goal of this tutorial is to help you determine what part of the testcase needs further configuration and demonstrate how to make the necessary changes.

Before beginning, it is important to note a few points:

- Determining exactly which part of the testcase is not yet configured correctly may require detailed knowledge of the application -- it is a good idea to get the application developers involved.

- If a working testcase is not achieved at the end of this tutorial, please contact Web Performance support for further assistance.
- When you achieve a working testcase, please consider submitting it to Web Performance support with a description of the configuration changes required to get it working. This will help us improve our automatic configuration wizards.

Overview

This process usually involves 3 steps:

1. Find the exact cause of the problem
2. Analyze what needs to be different in the underlying HTTP transactions to fix the problem
3. Make the necessary configuration changes in the testcase

The next three sections of the tutorial will describe these three steps in more detail and give some hints on how to accomplish them.

The remaining sections are examples of solving some specific problems with various combinations of the techniques described.

Finding the problem

When searching for the source of the configuration problem, you should start by using the Replay feature, rather than running a load test. When performing a replay, the software will save all the content received from the server (much like a recording), allowing you to review the content of each page and inspect the details of each HTTP transaction. This will be critical in identifying the problem.

There are cases where replays work but load tests do not. When a load test with only 2-3 simultaneous users fails, the cause almost always falls into one of these categories:

1. User identity or authentication - multiple simulated users are attempting to use the application using the same identity and the application does not support this.
2. Shared user data - the simulated users are attempting to operate on the same data. Either the operation is blocked by the application or the actions of one simulated user makes the actions of another simulated user erroneous or impossible.

When a load test succeeds with a small number of simultaneous users but fails when more users are running - the problem is almost always the application and/or the server. Don't be surprised - that's the point of load testing, right?

Finding the problem in a replay

In order to get the testcase configured correctly, you must identify the first transaction in the testcase where some symptom of the problem is displayed. This will frequently be a web page that:

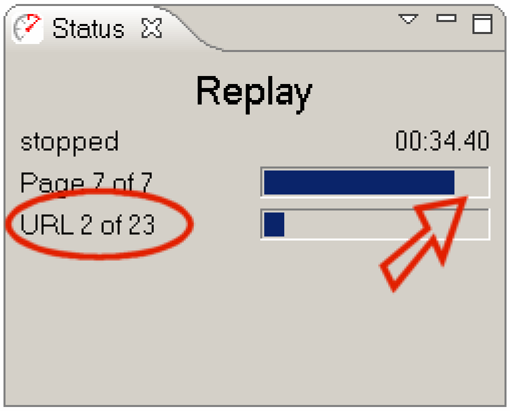
- causes a validation or extraction error in the Web Performance software
- displays the wrong content
- display an error message
- response has a different status code
- fails to return a valid response

Sometimes the test will run normally and indicate no errors but the testcases are not producing the desired results - e.g. a testcase that simulates purchasing an item does not generate any record of the purchase in the system.

Examples of the above situations:

Errors generated

When a replay cannot be completed successfully, the Replay View will indicate how far the replay progressed before errors were encountered:



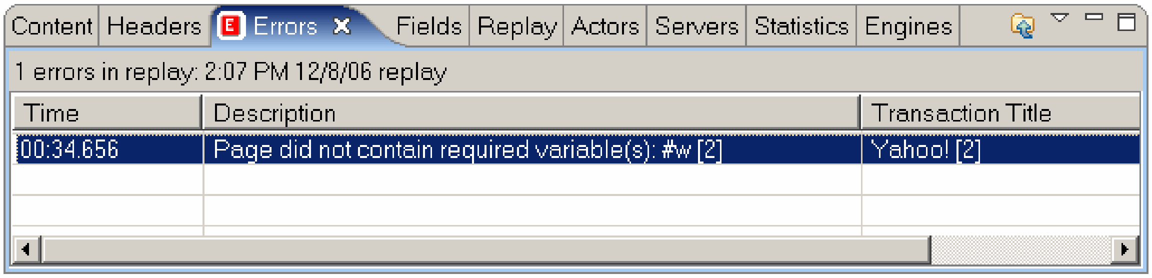
Replay

stopped 00:34.40

Page 7 of 7

URL 2 of 23

Checking the Errors View will usually provide more information about the problem:

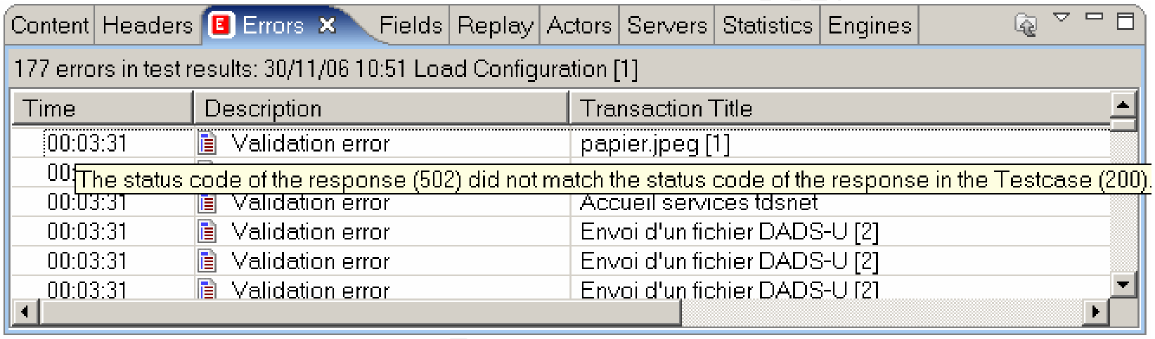


1 errors in replay: 2:07 PM 12/8/06 replay

Time	Description	Transaction Title
00:34.656	Page did not contain required variable(s): #w [2]	Yahoo! [2]

Different Status Code Returned

Automatic validation is performed on the status code of each transaction and in most cases it is expected to be an exact match to the recording. There are some exceptions that the logic will allow automatically. When a problem is detected, it will appear in the Errors View:



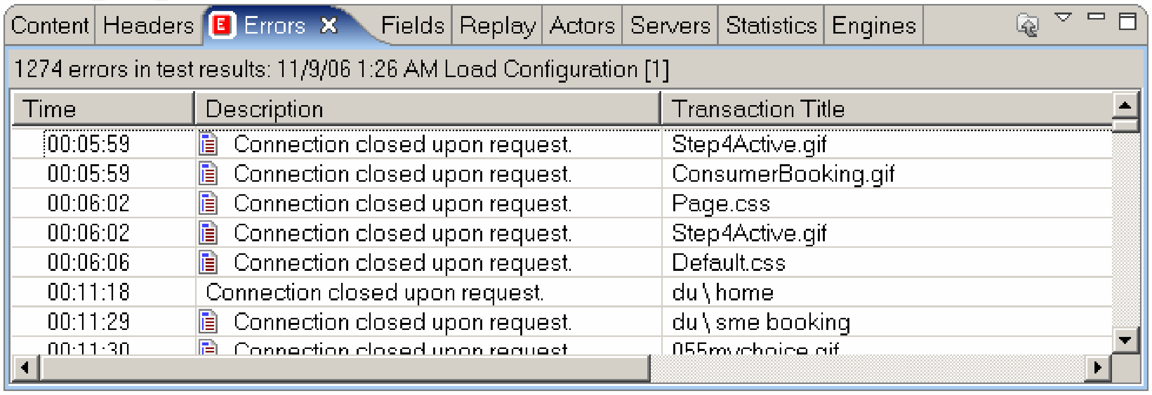
177 errors in test results: 30/11/06 10:51 Load Configuration [1]

Time	Description	Transaction Title
00:03:31	Validation error	papier.jpeg [1]
00		
00:03:31	Validation error	Accueil services tdsnet
00:03:31	Validation error	Envoi d'un fichier DADS-U [2]
00:03:31	Validation error	Envoi d'un fichier DADS-U [2]
00:03:31	Validation error	Envoi d'un fichier DADS-U [2]

The status code of the response (502) did not match the status code of the response in the Testcase (200).

Fails to Return a Valid Response

On occasion the error will be so serious that the server completely fails to return any response and closes the connection. Most of the time, this is caused by server overload, but occasionally it is caused by server application errors when unexpected inputs (due to a incorrect testcase configuration) are received by the server.



1274 errors in test results: 11/9/06 1:26 AM Load Configuration [1]

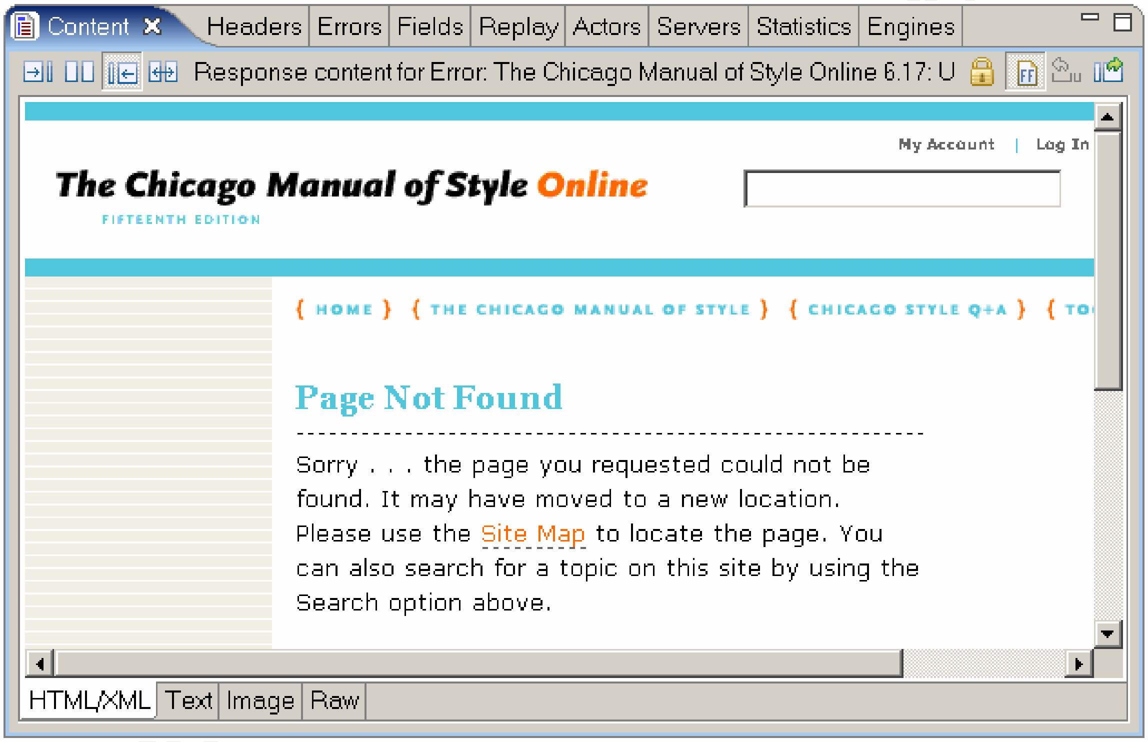
Time	Description	Transaction Title
00:05:59	Connection closed upon request.	Step4Active.gif
00:05:59	Connection closed upon request.	ConsumerBooking.gif
00:06:02	Connection closed upon request.	Page.css
00:06:02	Connection closed upon request.	Step4Active.gif
00:06:06	Connection closed upon request.	Default.css
00:11:18	Connection closed upon request.	du \ home
00:11:29	Connection closed upon request.	du \ sme booking
00:11:30	Connection closed upon request.	055mychoice.gif

Wrong Content Displayed

Inspect each page of the replay in the Content View to determine if the page appears to be correct. Each page will generally be a close (if not exact) match to the page from the original recording. Does the page look like the result of a different operation? For example - the login page is returned when another page is expected.

Error Message Displayed

In this example, the server returned a page indicating that the request could not be completed:



Response content for Error: The Chicago Manual of Style Online 6.17: U

Page Not Found

Sorry . . . the page you requested could not be found. It may have moved to a new location. Please use the Site Map to locate the page. You can also search for a topic on this site by using the Search option above.

Analyzing the required changes

This step will frequently require a more thorough understanding of the application then you may have needed before this point. It is often most expedient to enlist the help of an application developer at this point in the process.

The goal of this phase is to determine:

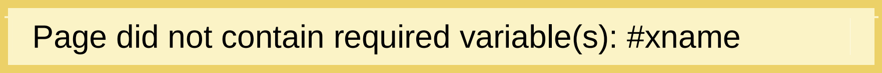
1. Where is the incorrect data?
2. What is the correct data to send?
3. Where does that data come from?

Step 1 - Where is the incorrect data?

More specifically - which part(s) of which transaction(s) does not contain the correct values?

You must start by identifying the transaction that is not configured correctly. More specifically, which HTTP request is not sending the correct data? In many cases, this will be the same transaction that displays the problem symptoms. If not, the transaction will be earlier in the testcase - you will have to track backwards in the application to find the source of the problem.

For example, if you have seen an error from the Web Performance software like this:



Page did not contain required variable(s): #xname

The related page is the first place that the Web Performance software has detected a problem, so this is the first place to look. Did this transaction receive the expected page in the response? If "yes", why was the variable not in the page? If "no", why was the correct page not returned? Was something in the request (URL query parameter, cookie, form field) incorrect? The answers to these questions, with the help of the application developer, should lead you to discover what piece(s) of data in the request(s) are in need of further configuration.

Step 2 - What is the correct data to send?

Once we have located the incorrect data - we can then determine what the correct values should be. In some cases this will be obvious. In other cases, the application developer may be able to help.

One of these two cases will apply:

1. The user would have provided this data
2. The server provides this data

> There are a few exceptions to the above:
>
> - Some data is randomly generated by scripts within the page. In this case, since the server will generally accept any value, we will consider it "user" data, even though the end-user did not actually enter this information.
> - With some testcases, you may find that the problem occurs in transactions that are not required for the testcase. The click-tracking URLs found on many large portal websites or e-commerce websites are a good example - the testcase does not required these URLs to succeed - and in many cases these URLs are undesirable for the testing process. You can use the Transaction Blocking features to eliminate these transactions from your testcase during the recording phase.

If #1 applies, then the values must be provided by you in the form of a Dataset. This data may be generated (such as random or sequential values) or imported into the dataset. The appropriate values is entirely dependent on your testcase and the details of the application implementation. Either the test designer or the developer will have to make that decision. The Datasets page describes the process for creating datasets and the Fields View page shows how to substitute data from a dataset into a field entered by the user.

Step 3 - Where does that data come from?

If the data did not come from the user, then it must have come from the server. The browser doesn't "make things up", so with the exception of randomly generated data in scripts, everything else must come from the server. The ASM wizard will automatically locate many of these. When it does not, you must locate that information and either:

1. Manually configure extractors and modifiers
2. Provide a custom rule that will help the ASM wizard find and configure this data automatically.

But first you must find it.

If the application developer cannot tell you where the data comes from, you can use the Content View (particularly the text tab) to inspect the source HTML of each page returned from the server during the recording. Note that many modern web applications receive significant amounts of their data in plain-text or XML formats that are retrieved by Javascript, rather than the source HTML of the pages. Be sure to look in those transactions, as well!

When you have located the data that causes the problem, then you can proceed to the next section to configure the Web Performance software to handle it.

Testcase Configuration

Once you have determined which piece of data is causing the problem, you can decide what to do about it.

The solutions generally fall into these categories:

1. Remove the entire transaction from the testcase
2. Ignore the field during automatic configuration
3. Find and configure the source of a dynamic field
4. Find and configure dynamically-named fields

Remove the entire transaction from the testcase

For complex or portal sites, there are often many transactions that come from systems that are not intended to be part of the test.

For example: tracking URLs (used for user activity) and URLs from advertising sites (for banner ads) should frequently be left out of the test altogether.

Any transaction that is not required for the correct operation of the testcase and comes from a server that is not being tested (or the URL has negligible performance impact) is a candidate for this technique.

> Note: The term *transaction* in these documents always refer to a single HTTP transaction, which corresponds to a single request/response between a web browser and a web server. It should not be confused with any concept of a transaction in the web application, such as making a purchase on an e-commerce site.

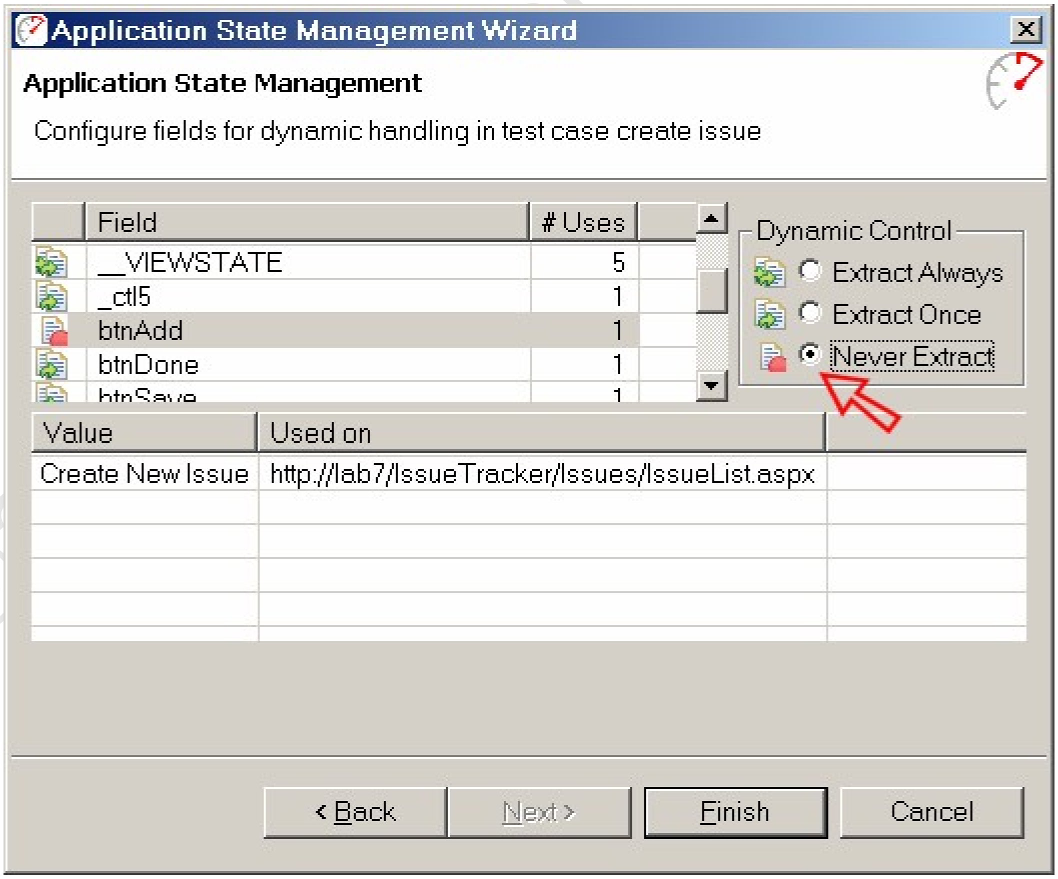
Once you have located the URLs or servers to be ignored, they can be removed from the testcase manually by deleting the undesired transactions. Be sure to run the configuration wizards again (especially the ASM wizard) to update the testcase configuration. Failure to do this can cause errors during replay. You may also use the Blocking preference page to block these transactions from being recorded and then re-record the testcase. Note that during the recording process, these transactions will still pass between the browser and server - they will simply be left out of the testcase.

Ignore the field during automatic configuration

Some fields are detected by the ASM logic as dynamic when they are not. Frequently there are fields that could change in various parts of the application, but not for the particular testcase being simulated. This causes extra work for the load testing tool and potentially misleading errors during a test if the fields do not appear in the page during a replay or load test.

The ASM wizard can be instructed to ignore these fields:

1. Run the ASM wizard
2. On the wizard page titled *Configure fields for dynamic handling in test case*, select each field to be ignored and select the *Never Extract* option for each field.
3. Complete the wizard



Application State Management Wizard

Application State Management

Configure fields for dynamic handling in test case create issue

Field	# Uses
__VIEWSTATE	5
_ctl5	1
btnAdd	1
btnDone	1
btnSave	1

Dynamic Control

○ Extract Always
○ Extract Once
● Never Extract

Value	Used on
Create New Issue	http://lab7/IssueTracker/Issues/IssueList.aspx

This will cause the ASM wizard to ignore these fields during configuration - as a result no attempt will be made to extract or modify them during a replay.

Once this technique has been determined to be useful for a field, you may use the Ignore Fields feature to permanently ignore a field during the ASM analysis.

Find and configure the source of a dynamic field

Some fields cannot be automatically be configured by the ASM wizard because it cannot find the source of the field. This is frequently the case when the fields are assigned a value by a mechanism other than using the *value* parameter in the field declaration tag (e.g. javascript) or for query parameters in URLs that are generated dynamically in the browser (e.g. javascript).

The next section, Configuring Dynamic Fields, shows an example of the configuration steps required to address this situation. The Web Service tutorial also demonstrates a solution to this problem.

Find and configure dynamically-named fields

Some applications use an architecture that results in the same fields having different names each time the testcase is performed. These fall into two categories:

1. Fields with name dependencies
2. Fully-dynamic file names

Case #1 has groups of fields where a portion of the field name is equal to the value of another field. For instance:

```
ITEM=123
FNAME123=Bob
LNAME123=Jones
ZIP123=44095

ITEM=456
FNAME456=Jane
LNAME456=Smith
ZIP456=76012
```

In the above example, the names of the FNAMExxx, LNAMExxx and ZIPxxx fields are *dependent* on value of the preceding ITEM field. These fields can be handled automatically using the Dynamically Named Fields feature, described later in this tutorial.

Case #2 has two variations:

1. The name of the field can be predicted in advance
2. The name of the field essentially random - it can not be known in advance

If #1 applies, then the field name can be supplied in a dataset and manually configured with a modifier in the Fields View.

If #2 applies, the testcase will require customization that is beyond the scope of this tutorial. Please use the Support wizard to send the testcase to the Web

Performance support system. The support personnel can help you arrive at a working configuration.

Configuring Dynamic Fields

Once you have determined what values need to go into which fields, and where it comes from, there are two ways to configure the testcase - manually and automatically.

- Manual configuration - This involves configuring a modifier and an extractor to move the data from a response to a subsequent request. This is a good way to handle simple cases and to validate a proposed solution before attempting to create an automatic configuration. However, if a similar change needs to be made for many transactions and/or for many similar data elements, this can be a lot of work.
- Automatic configuration - This involves creating a rule that the ASM Wizard can use to automatically locate the data and configure modifiers and extractors as needed when the wizard is run. This is a good way to handle cases where the same data needs to be extracted/modified in multiple transactions (such as session identifiers) or there are multiple fields that are expressed in the same manner. It can be a little more difficult then a manual configuration, but generally saves configuration time and effort in the long run.

Note that some situations can ONLY be configured manually and some can ONLY be configured automatically. Additionally, there are some cases where neither manual nor automatic configuration is possible via the techniques described here. Please submit your testcase to Web Performance support for assistance in these cases. You can use the Support wizard to help automate the process.

Extractors, Modifiers and User State Variables
There are a few important concepts to understand before proceeding. When a request requires dynamic data that comes from a previous response, there are two discrete steps:

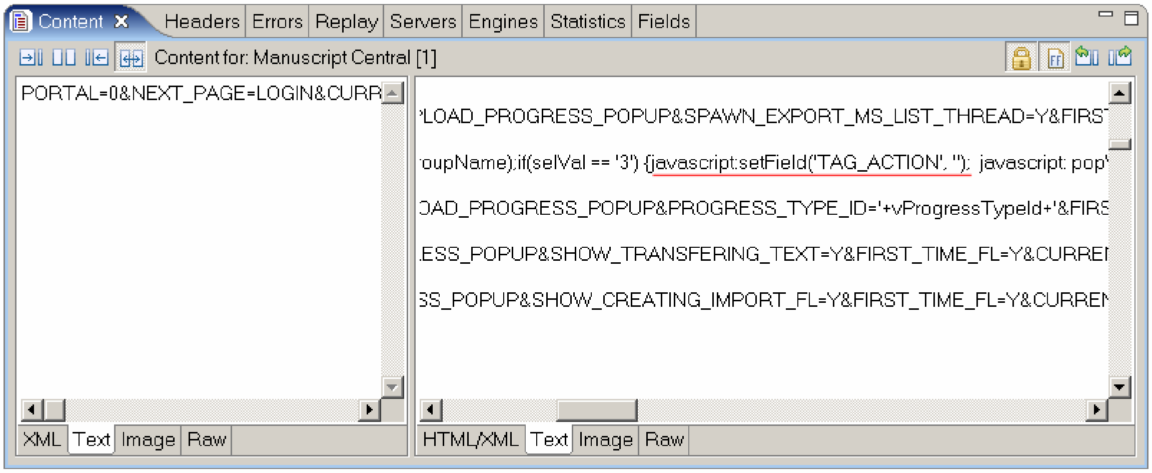
1. *extract* the relevant data from the response into a user state variable
2. *modify* the appropriate part of the request as it is written to the server to contain the value from the variable

Terminology:

Extractors operate on a response. They look for a specific piece of data in the response and place it into a User State Variable.
User State Variable is simply a *named* place to store some data. The User State Variables are (by default) reset each time the Virtual User restarts a testcase.
Modifiers operate on a request. They change a specific section of a request, as it is written to the server, to contain the value from a variable or dataset.

Manual Configuration

Here is an example of a pair of transactions that require manual configuration. The response of the first transaction contains some information that needs to be extracted and saved for use in the second transaction - the TAG_ACTION field. In this example, the javascript included in the source of the page will set the value of the TAG_ACTION field to the value supplied as the second parameter to the javascript setField() method. This value could be different each time the response is received - and must therefore be extracted dynamically during a replay in order to supply the correct value for the next transaction. The source of the response page is shown below:



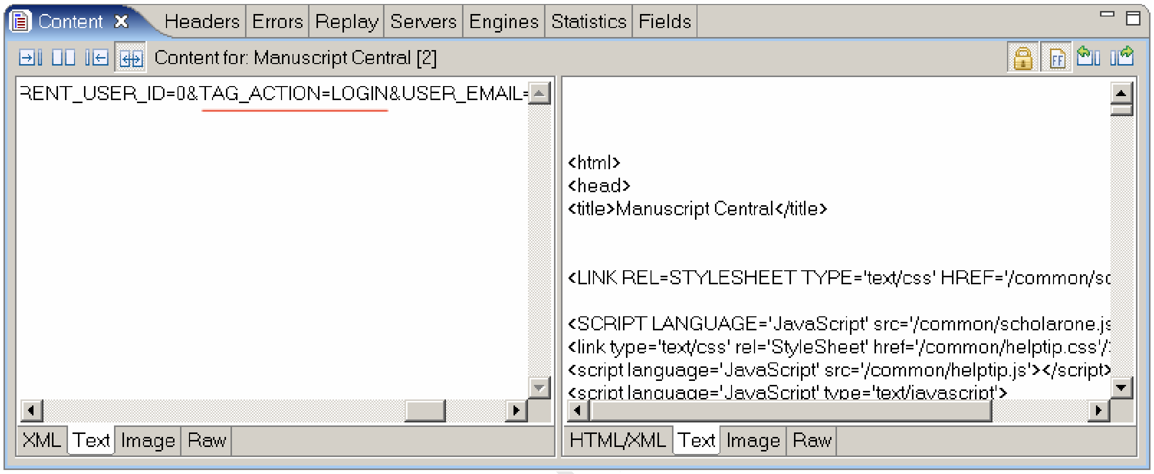
Content for: Manuscript Central [1]

```
PORTAL=0&NEXT_PAGE=LOGIN&CURR
```

```
'LOAD_PROGRESS_POPUP&SPAWN_EXPORT_MS_LIST_THREAD=Y&FIRST

oupName);if(selVal == '3') {javascript:setField('TAG_ACTION', '');  javascript: pop'

OAD_PROGRESS_POPUP&PROGRESS_TYPE_ID='+vProgressTypeId+'&FIRS

ESS_POPUP&SHOW_TRANSFERING_TEXT=Y&FIRST_TIME_FL=Y&CURREN

SS_POPUP&SHOW_CREATING_IMPORT_FL=Y&FIRST_TIME_FL=Y&CURREN
```

figure 1: transaction 1

The request of the second transaction should contain the dynamically extracted value (from the first transaction) instead of the value that was originally recorded. Note that if the value was supplied as the default value in the field declaration tag, the ASM wizard would automatically pick up that value, since that is the standard way to provide the value for fields that are not user entered.



Content for: Manuscript Central [2]

```
RENT_USER_ID=0&TAG_ACTION=LOGIN&USER_EMAIL=
```

```
<html>
<head>
<title>Manuscript Central</title>


<LINK REL=STYLESHEET TYPE='text/css' HREF='/common/sc

<SCRIPT LANGUAGE='JavaScript' src='/common/scholarone.js
<link type='text/css' rel='StyleSheet' href='/common/helptip.css'/
<script language='JavaScript' src='/common/helptip.js'></script>
<script language='JavaScript' type='text/javascript'>
```

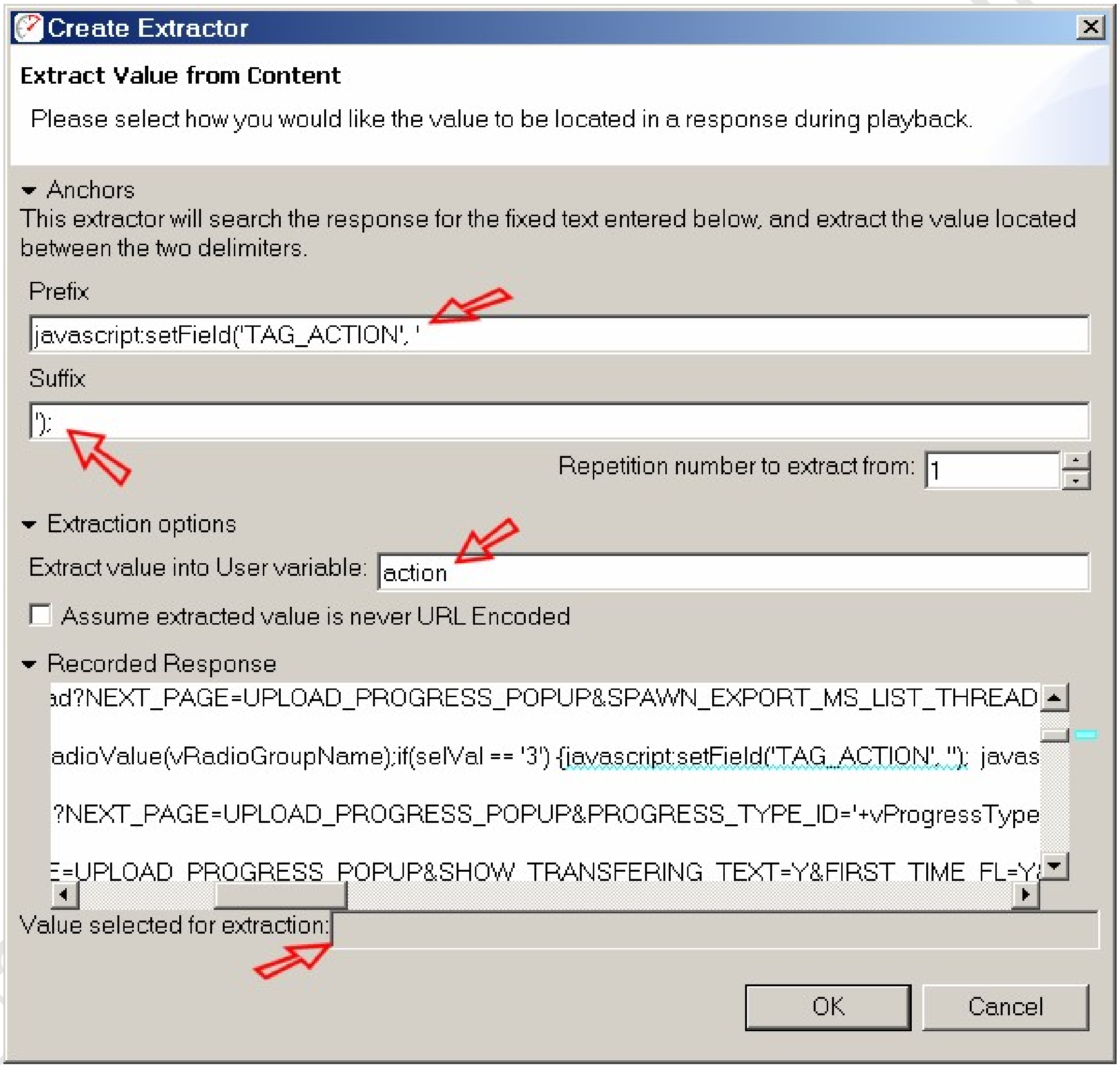
figure 2: transaction 2

Step 1 - Configure the Extractor

In order to reliably extract information from a response, we must be able to tell the extractor how to locate the data to be extracted. In this example it is easy, since the session identifier is surrounded by javascript:setField('TAG_ACTION', ' and ');.

1. Open the Actors view (Extractors tab) and select the transaction containing the data to be extracted (which must be in the response)
2. Press the *Add Extractor...* button (✚)
3. Complete the *Prefix* and *Suffix* fields
4. Enter the name of the user variable to store the session identifier in (e.g. action)
5. Verify that the correct value is shown in the *Value selected for extraction* field

When these items are complete, the dialog should look like the picture below. When it does, you can press the *OK* button to save the new extractor.



Extract Value from Content

Please select how you would like the value to be located in a response during playback.

Anchors

This extractor will search the response for the fixed text entered below, and extract the value located between the two delimiters.

Prefix: javascript:setField('TAG_ACTION', '

Suffix: ');

Repetition number to extract from: 1

Extraction options

Extract value into User variable: action

□ Assume extracted value is never URL Encoded

Recorded Response

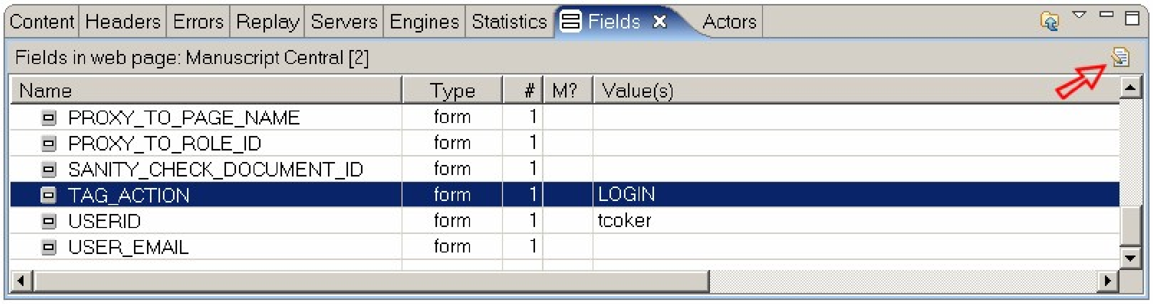
```
ad?NEXT_PAGE=UPLOAD_PROGRESS_POPUP&SPAWN_EXPORT_MS_LIST_THREAD
adioValue(vRadioGroupName);if(selVal == '3') {javascript:setField('TAG_ACTION', ''); javas
?NEXT_PAGE=UPLOAD_PROGRESS_POPUP&PROGRESS_TYPE_ID='+vProgressType
E=UPLOAD_PROGRESS_POPUP&SHOW_TRANSFERING_TEXT=Y&FIRST_TIME_FL=Y&
```

Value selected for extraction:

When replaying this testcase, the Virtual User will run this extractor after the response has been received. In this case, it will look for the target data (as configured above) and store it into a user state variable.

Step 2 - Configure the Modifier

In most cases, Analyzer will automatically parse the fields submitted with each request and they will be visible in the Fields view. Here is the TAG_ACTION field:

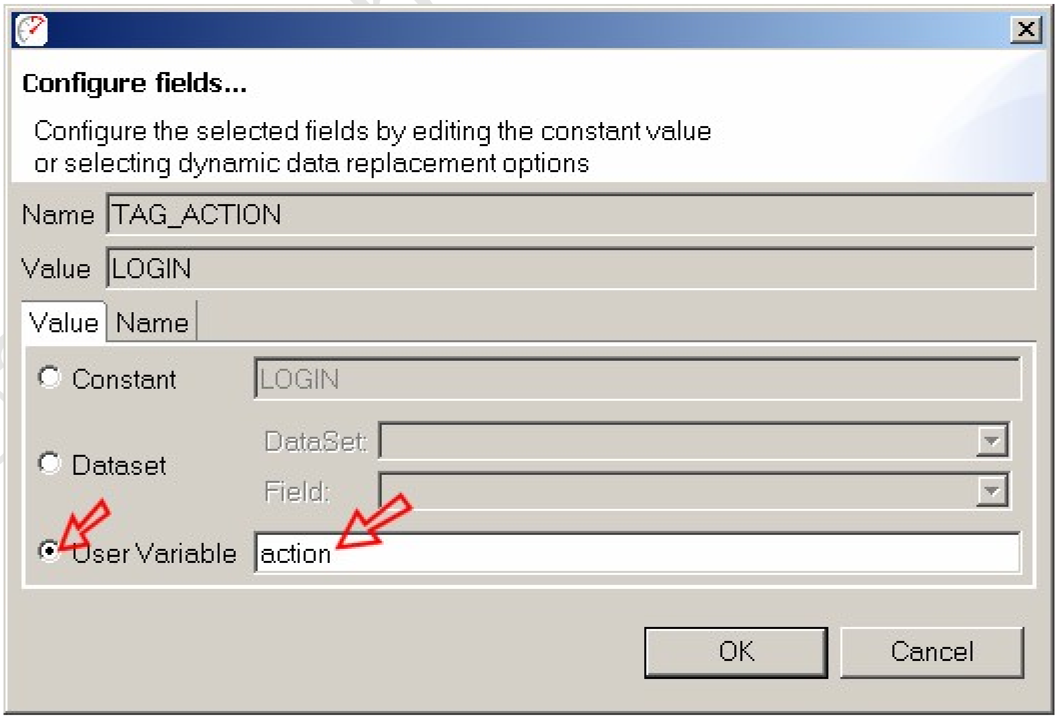


Fields in web page: Manuscript Central [2]

Name	Type	#	M?	Value(s)
PROXY_TO_PAGE_NAME	form	1		
PROXY_TO_ROLE_ID	form	1		
SANITY_CHECK_DOCUMENT_ID	form	1		
TAG_ACTION	form	1		LOGIN
USERID	form	1		tcoker
USER_EMAIL	form	1		

Selecting the field and pressing the *Edit* button (▤) or double-clicking the Modifier (M?) column for the field will open the Field configuration dialog. In this dialog, select the *User Variable* option for the field value and enter the name of the variable, as shown below.

Note that the variable name MUST be the same name used in the extractor configuration (above).



Configure fields...

Configure the selected fields by editing the constant value or selecting dynamic data replacement options

Name: TAG_ACTION

Value: LOGIN

○ Constant: LOGIN

○ Dataset — DataSet: — Field:

● User Variable: action

After these changes are complete, replay the testcase and verify that the new configuration changes have the intended effect. If the testcase still generates errors, be sure to verify that this particular field is now correct - there are often multiple fields that need customization before arriving at a successful testcase.

Once you have determined that the manual configuration works as intended, you may wish to automate the process by creating a custom detection rule, as described in the next section.

Automatic Configuration

The above example can be configured automatically in two ways:

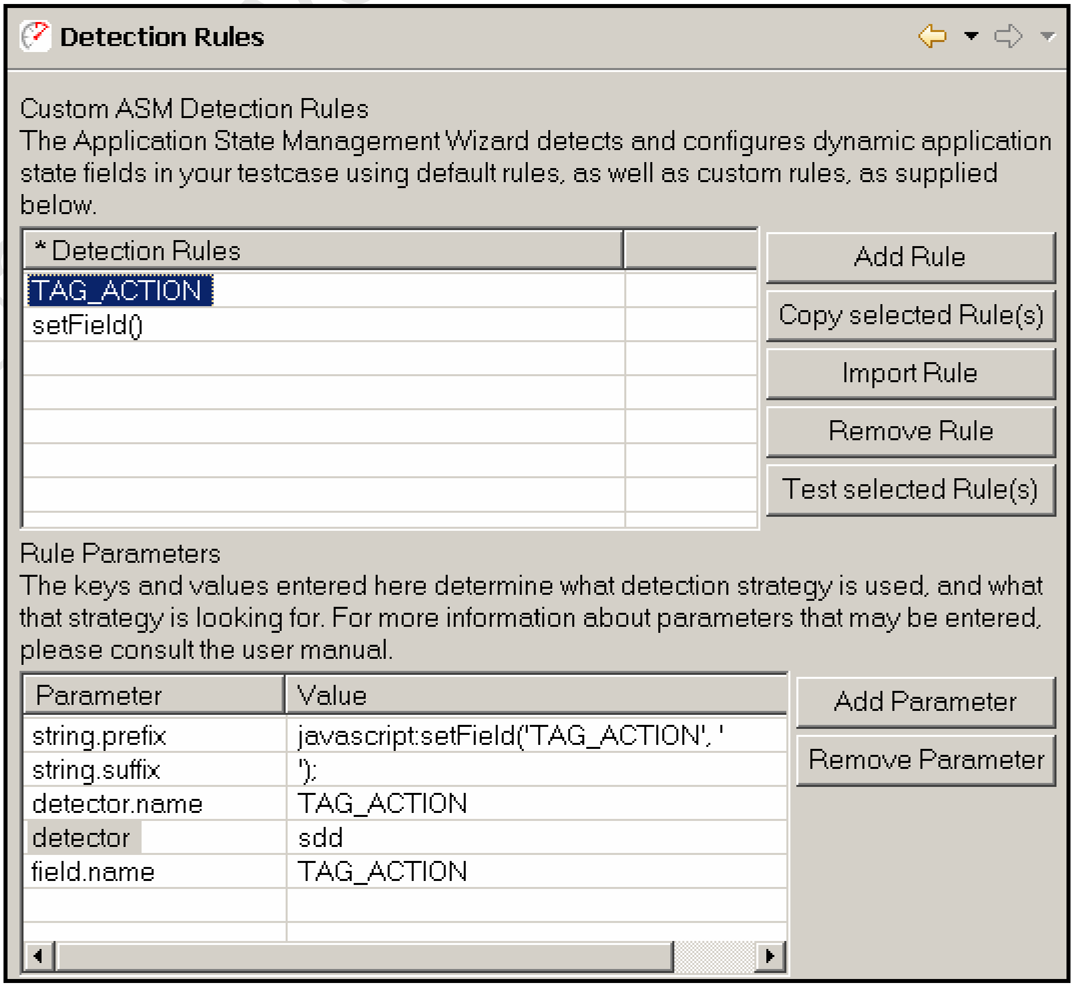
1. Single-field configuration - automatically configure this EXACT case, i.e. the TAG_ACTION field
2. Multiple-field configuration - automatically configure any field that matches this usage pattern

In both cases, you must create a *detection rule*. This rule describes how ASM should locate the value assignment for a given field name. More information on the configuration options for detection rules is available in the Custom ASM Configuration section.

Both configurations are performed in the Detection Rules preference page. Go to Window->Preferences->Web Performance->Application State->Detection Rules.

Single-field Automatic configuration

Add a detection rule with the parameters as shown below:



Custom ASM Detection Rules
The Application State Management Wizard detects and configures dynamic application state fields in your testcase using default rules, as well as custom rules, as supplied below.

* Detection Rules	
TAG_ACTION	
setField()	

Rule Parameters
The keys and values entered here determine what detection strategy is used, and what that strategy is looking for. For more information about parameters that may be entered, please consult the user manual.

Parameter	Value
string.prefix	javascript:setField('TAG_ACTION', '
string.suffix	');
detector.name	TAG_ACTION
detector	sdd
field.name	TAG_ACTION

These parameters will create a rule that looks for matching values between provided the *prefix* and *suffix* in web pages and attempt to match that value with the value of the named field (TAG_ACTION). Any places in the testcase where a match is found, the ASM wizard will automatically configure the extractor and modifier as demonstrated above.

Press the *Test selected Rule(s)* button to validate the configuration. Note that this step verifies that the configuration is valid but not necessarily correct. In other words, it checks that the combination of parameters provided is allowed and the parameter names are spelled correctly, but cannot verify that the rule will have the intended result.

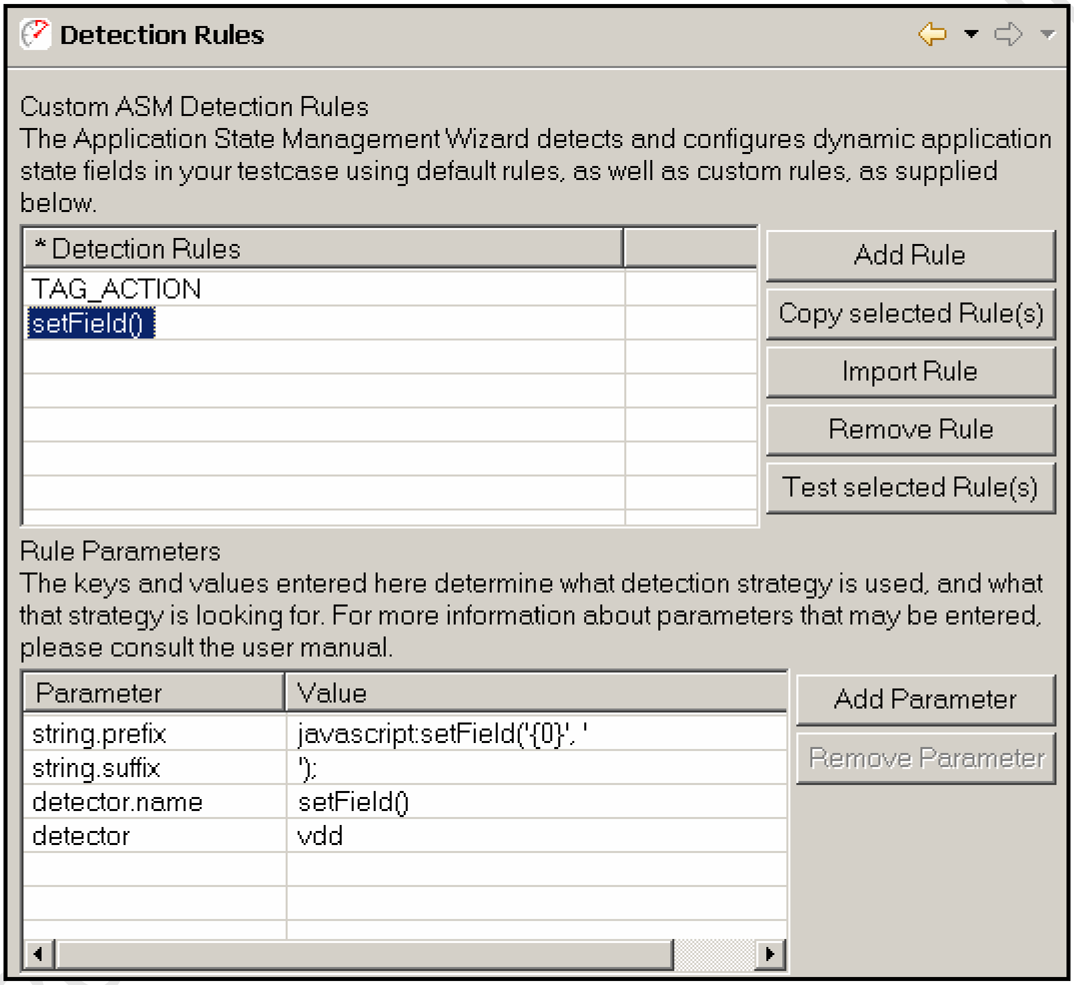
Accept the changes (*OK* button) and run the ASM wizard on the testcase by selecting the *Configure->Application State* item from the pop-up menu in the Navigator. Verify that the fields have been detected by the wizard as required.

Multiple-field Automatic configuration

Using this method, you will create a detection rule that will apply to fields with different names but share the same declaration format. For example, it would recognize the field value declarations for FIELD1 and FIELD2 in these lines:

```
javascript:setField('FIELD1', 'value1');
javascript:setField('FIELD2', 'value2');
```

Add a detection rule with the parameters as shown below:



Custom ASM Detection Rules
The Application State Management Wizard detects and configures dynamic application state fields in your testcase using default rules, as well as custom rules, as supplied below.

* Detection Rules	
TAG_ACTION	
setField()	

Rule Parameters
The keys and values entered here determine what detection strategy is used, and what that strategy is looking for. For more information about parameters that may be entered, please consult the user manual.

Parameter	Value
string.prefix	javascript:setField('{0}', '
string.suffix	');
detector.name	setField()
detector	vdd

These parameters will create a rule that is very similar to the rule created in the previous step. The primary difference is that instead of only employing this rule for a specific field name, it will apply the rule for any field for which it cannot find a match using the default rules. In addition, the name of the field will be dynamically inserted into the detection *prefix* and *suffix* so that the detection parameters can be specific as possible to avoid false positives. The field name will be substituted into the *prefix* and/or *suffix* in place of the **{0}** character sequence. Any places in the testcase where a match is found, the ASM wizard will automatically configure the extractor and modifier as demonstrated above.

Press the *Test selected Rule(s)* button to validate the configuration. Note that this step verifies that the configuration is valid but not necessarily correct. In other words, it checks that the combination of parameters provided is allowed and the parameter names are spelled correctly, but cannot verify that the rule have the intended result.

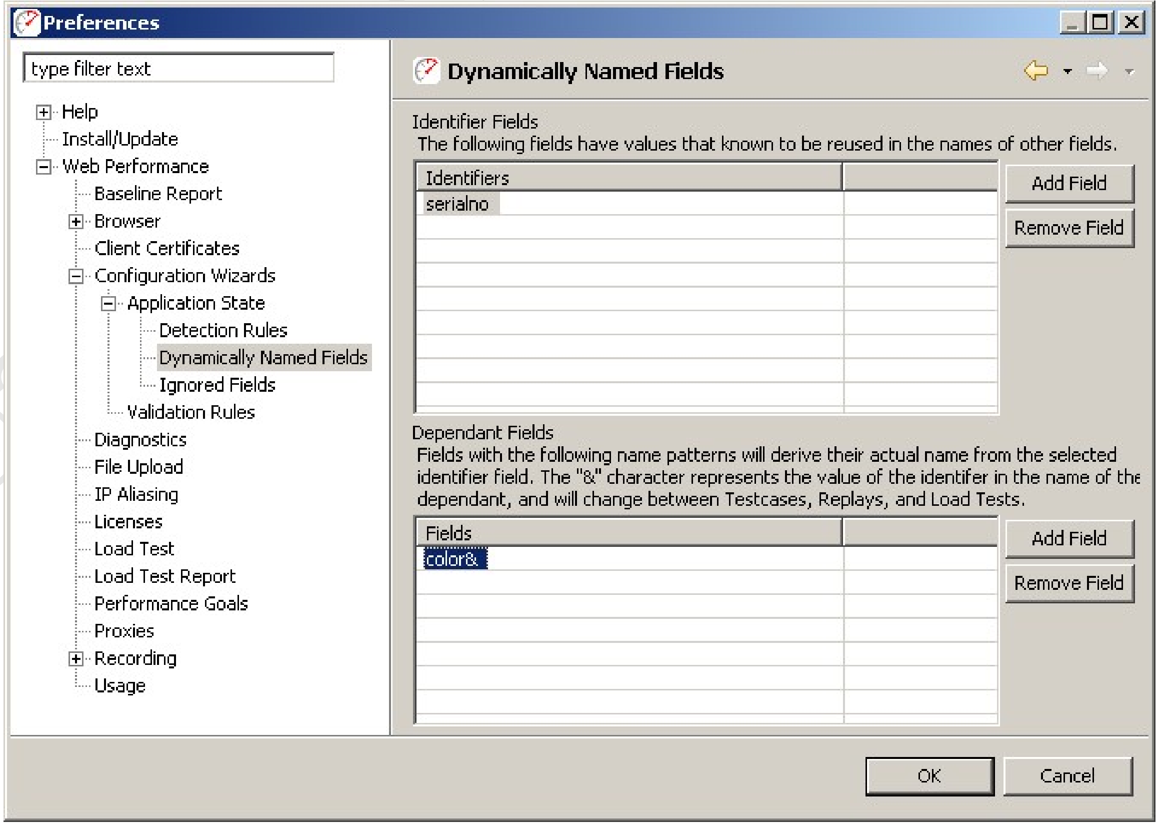
Accept the changes (*OK* button) and run the ASM wizard on the testcase by selecting the *Configure->Application State* item from the pop-up menu in the Navigator. Verify that the fields have been detected by the wizard as required.

Dynamically Named Fields

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 configure the "Dynamically Named Fields" preferences how to detect this behavior in your application. This option may be configured through a preference page, accessed by selecting Window→Preferences… and then selecting Web Performance→Configuration Wizards→Application State→Dynamically Named Fields.



Dynamically Named Fields

Identifier Fields
The following fields have values that known to be reused in the names of other fields.

Identifiers	
serialno	

Dependant Fields
Fields with the following name patterns will derive their actual name from the selected identifier field. The "&" character represents the value of the identifer in the name of the dependant, and will change between Testcases, Replays, and Load Tests.

Fields	
color&	

Configuring these fields is done in two phases. The first is to select the "Add Field" next to the "Identifiers" table, and enter the name of the field that identifies a value. In our example, the identifier is "serialno", whose value will be used later to identify the name of the next variable.
Next, select the field in the Identifiers table to display the dependant fields associated with it, and press the "Add Field" button next to the bottom "Fields" table to create a new dependant field. The name of the variable may be entered

here, replacing the dynamic part of the name with an ampersand (&). In our example, the color field would be entered as "color&".

The next time the Application State Management Wizard is run on a testcase, fields starting with the name "color", and ending their name with a value from the field "serialno" will be dynamically renamed when the testcase is replayed or run in a load test.

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 by adding their names to the "Fields" table:

- color&
- weight&_in_lbs
- &_assembly_date

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

It is also permitted for a dynamically named field to be associated with multiple identifiers. For example, consider another case:

```
itemid=123456789
checkbox123456789=checked
legacyid=123
checkbox123=unchecked
```

To configure this case, simply create to identifier fields:

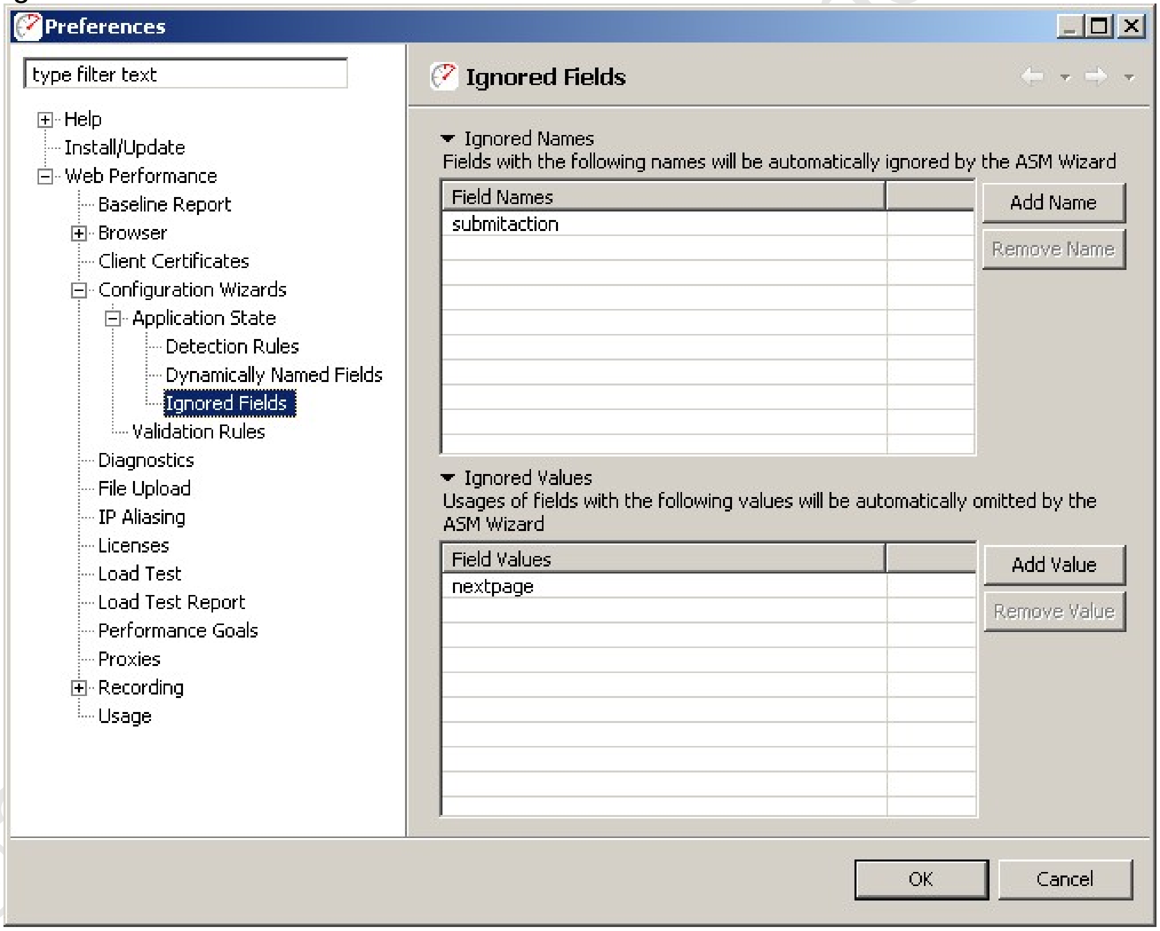
- itemid
- legacyid

Next, add the dependant field "checkbox&" to both identifier fields. The Application State Management Wizard will examine both uses of the "checkbox" fields, and only associate dependency when the name of the field corresponds to the value of the identifier. In this example, the wizard will associate the first "checkbox" field as being dependant on "itemid", and associate the second "checkbox" field as dependant on the field "legacyid".

Ignoring Fields in the Application State Management Wizard

The Application State Management 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 Application State Management Wizard offers ignore preferences in order to automatically ignore those fields which are not intended to be treated as dynamic. These preferences may be accessed by selecting Window → Preferences… and then selecting Web Performance → Configuration Wizards → Ignored Fields.



Ignored Fields

Ignored Names

Fields with the following names will be automatically ignored by the ASM Wizard

Field Names	
submitaction	

Ignored Values

Usages of fields with the following values will be automatically omitted by the ASM Wizard

Field Values	
nextpage	

This page contains two lists, one for omitting fields by name, and one for omitting specific uses of a field by their value. 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 Application State Management Wizard manually, or you could specify that this field always be automatically removed with either ignore list

Ignored Names	OR	Ignored Values
btnSubmit		submit

Be very careful not to include a blank entry unless you intend for the Wizard to treat blank values as fixed values as well. The next time you run the Application State Management Wizard, any usage with their name or value specified in one of the ignore lists will be automatically ignored or hidden by the wizard.

Load Testing a Web Service

Overview

The purpose of this tutorial is to illustrate the steps required to load-test a web service using Web Performance Load Tester. Although the demonstration service used in this tutorial is very simple, the concepts presented are directly applicable to more complicated services.

This tutorial is organized into 4 main steps:

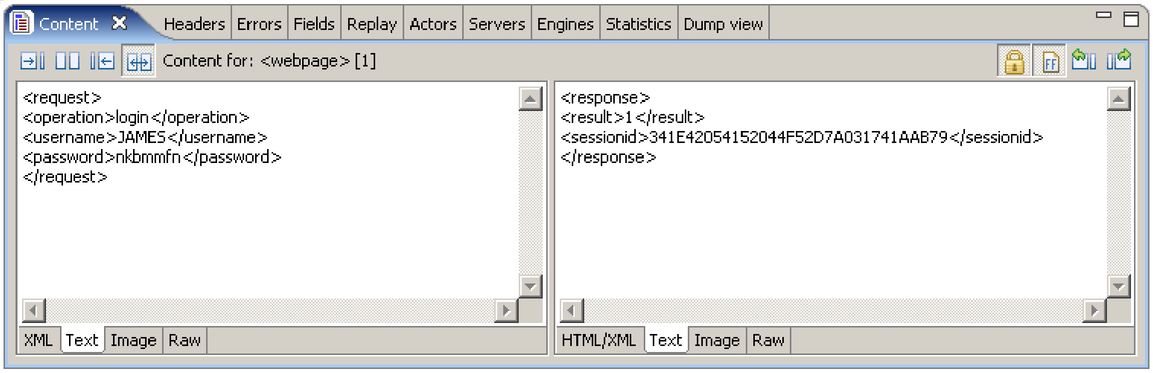
1. Create the testcase
2. Configure the testcase for replay
3. Configure the testcase for multiple users
4. Run test and analyze results

Introduction to the Bank web service

The service tested in this tutorial provides an interface to bank accounts. A similar service might be used, for example, to allow ATM machines in bank branches to access accounts in the central bank system. The operations used in this tutorial are *login* and *get-balance*. These operations use a simple pseudo-XML format for ease of readability.

Login transaction

The login transaction sends the username and password in a login operation. If successful, the response contains a session identifier that must be used for future transactions. The XML sent and received in this transaction look like this:



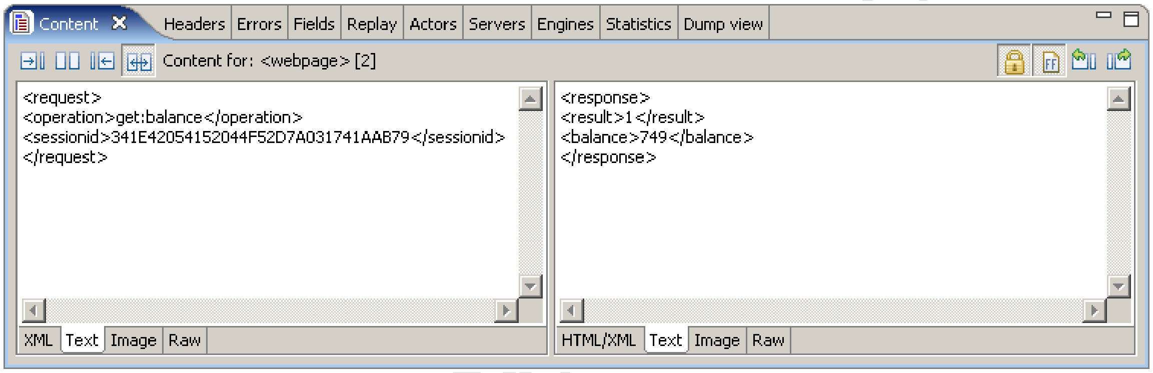
```
<request>
<operation>login</operation>
<username>JAMES</username>
<password>nkbmmfn</password>
</request>
```

```
<response>
<result>1</result>
<sessionid>341E42054152044F52D7A031741AAB79</sessionid>
</response>
```

> Note that the above screenshot shows the content as viewed in Analyzer's *Content* view.

Get-balance transaction

The get-balance transaction sends the session identifier and, if successful, receives the account balance (in cents). The pseudo-XML sent and received in this transaction look like this:



```
<request>
<operation>get:balance</operation>
<sessionid>341E42054152044F52D7A031741AAB79</sessionid>
</request>
```

```
<response>
<result>1</result>
<balance>749</balance>
</response>
```

In two example transactions show above, user "JAMES" logs into the system using the password "nkbmmfn". The result value "1" in the response indicates a successful login and provides a session identifier to be used in future transactions. He then checks his balance and finds that he has $7.49 in his account (749 cents).

Step 1 - Creating the testcase

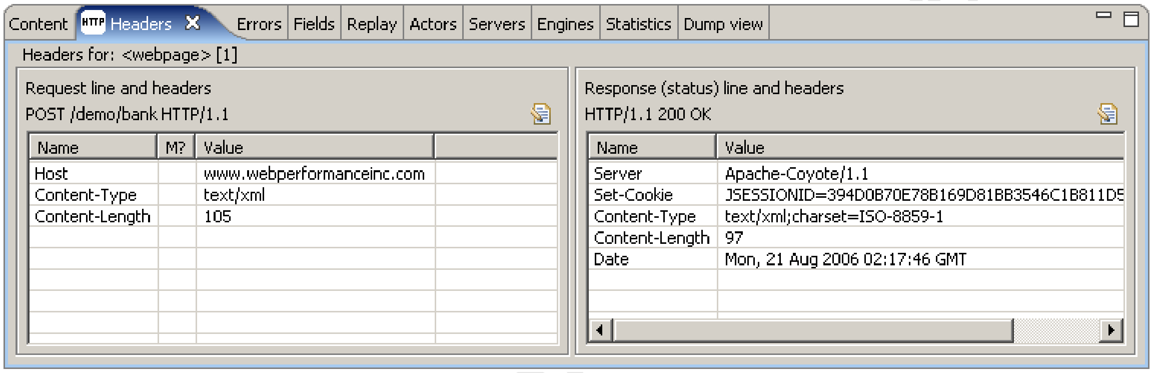
The first step in the load-testing process is to create the testcase to be simulated. The testcase is a collection of HTTP transactions (requests and responses). When testing a browser-based application, this step is usually achieved by using the browser to interact with the website in the same manner any user would - with the browser configured to use a recording proxy. This proxy records the HTTP transactions between the browser (client) and server and creates the testcase using these transactions. If you have a client for your web service and it supports the use of a proxy, this is the fastest way to get a testcase created.

Since many web services do not have this ability, we will demonstrate how to create the transactions from scratch.

Creating the login request

The first step is to create the content of the request - that is the pseudo-XML shown above. Paste this into your favorite plain-text editor and then save the file (e.g. login-request.txt). Next, note the length of the file - this length will be needed when we add the HTTP headers.

Step 2 involves putting the HTTP start-line and headers into the request. Below on the left are the start-line and headers used for this example request - the required headers may be different depending on the requirements of your service.



Headers for: <webpage> [1]

Request line and headers

POST /demo/bank HTTP/1.1

Name	M?	Value
Host		www.webperformanceinc.com
Content-Type		text/xml
Content-Length		105

Response (status) line and headers

HTTP/1.1 200 OK

Name	Value
Server	Apache-Coyote/1.1
Set-Cookie	JSESSIONID=394D0B70E78B169D81BB3546C1B811D5
Content-Type	text/xml;charset=ISO-8859-1
Content-Length	97
Date	Mon, 21 Aug 2006 02:17:46 GMT

Note that each part of the request headers will need modification as required by your service:

1. In the start-line, the path of the service will be different (/demo/bank)
2. The host should be changed to the hostname for the server your service resides on
3. The content-type could be different (text/xml is common)
4. The content-length must be changed to match the content you created for this request

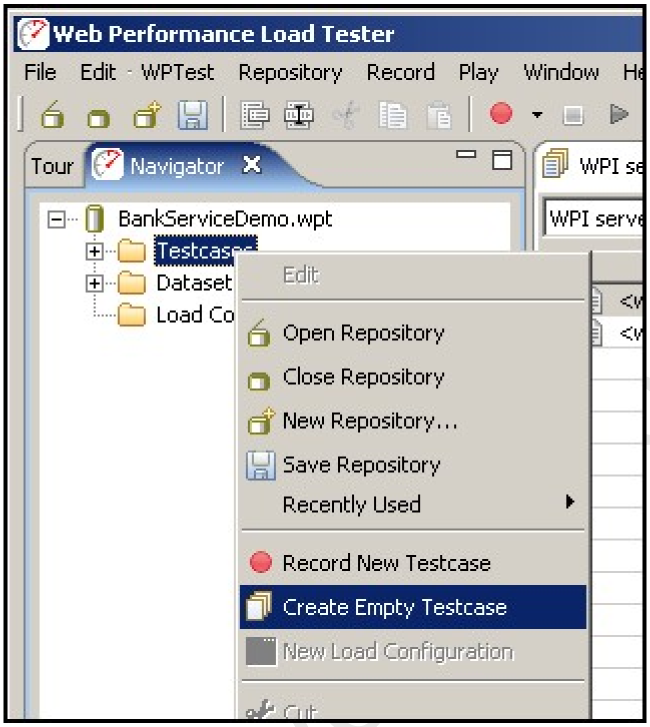
The text file used to create the request shown above contains this:

```
POST /demo/bank HTTP/1.1
Host: www.webperformanceinc.com
Content-Type: text/xml
Content-Length: 111

<request>
<operation>login</operation>
<username>JAMES</username>
<password>nkbmmfn</password>
</request>
```

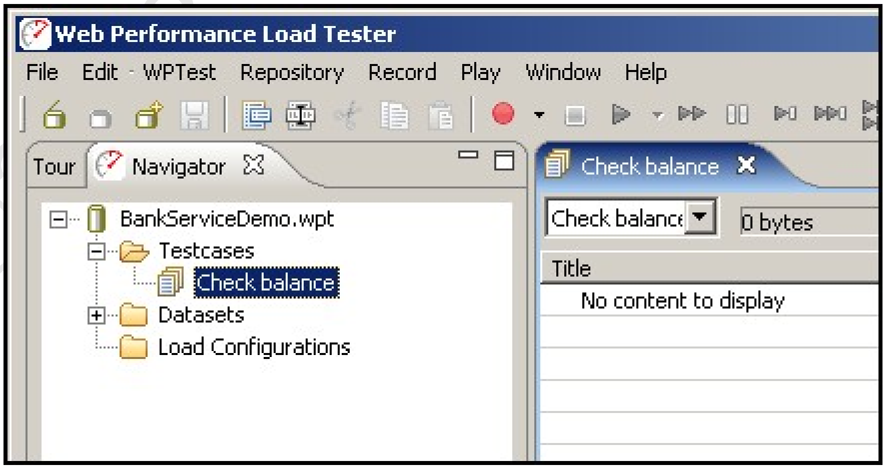
The other request and responses are created in a similar manner.

Once you have the 4 files created (a request and response for each of the 2 transactions), we can create the testcase.

In the Navigator, select the *Create Empty Testcase* item from the pop-up menu on the Testcases node to create a blank testcase where the transactions can be placed:

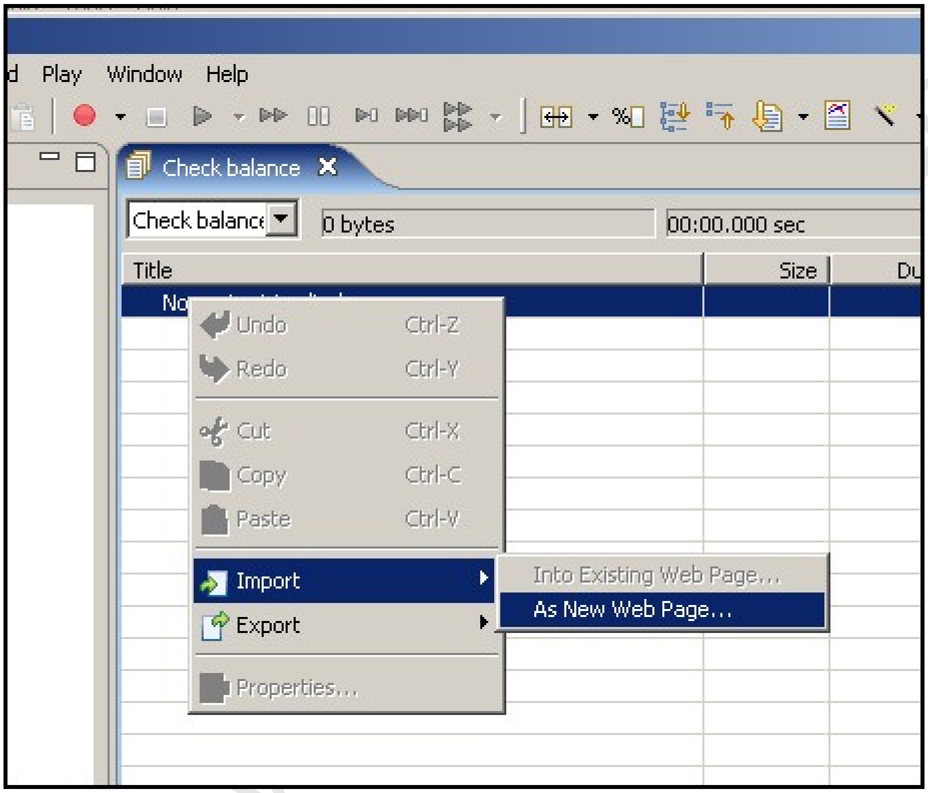


We now have an empty testcase (renamed "Check balance"):



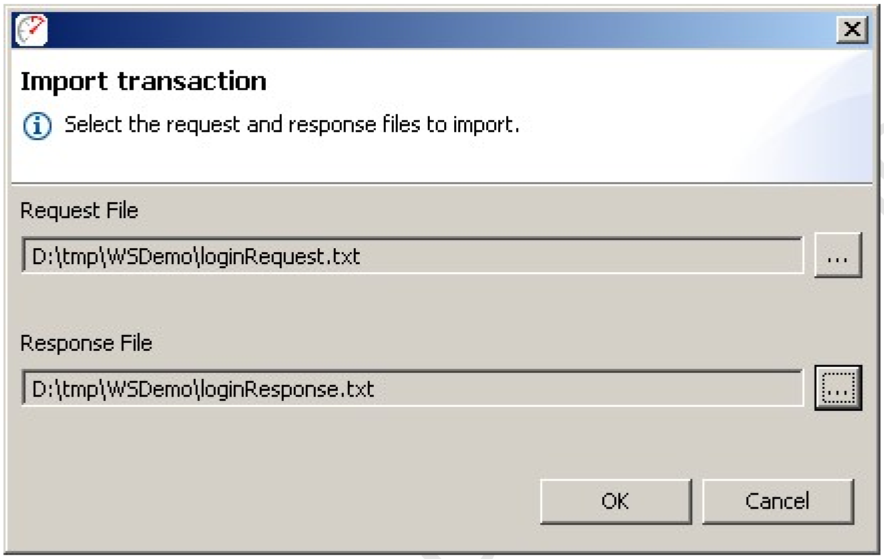
The next step is to create the two transactions required for the *Check balance* testcase.

Each transaction can be imported (request and response) using the *Import->As New Web Page* item from the pop-up menu in the testcase editor:



> Note that Analyzer groups transactions together in pages within a testcase. For a web service, there are no pages, so Analyzer will simply treat then as one-transaction pages. In complex services that issue several groups of transactions that are logically grouped and related, it can be useful to group the transactions together within a single page. Analyzer will calculate separate metrics for each transaction and for each page, which can be useful when analyzing load test results.

The request and response files are selected in this dialog:



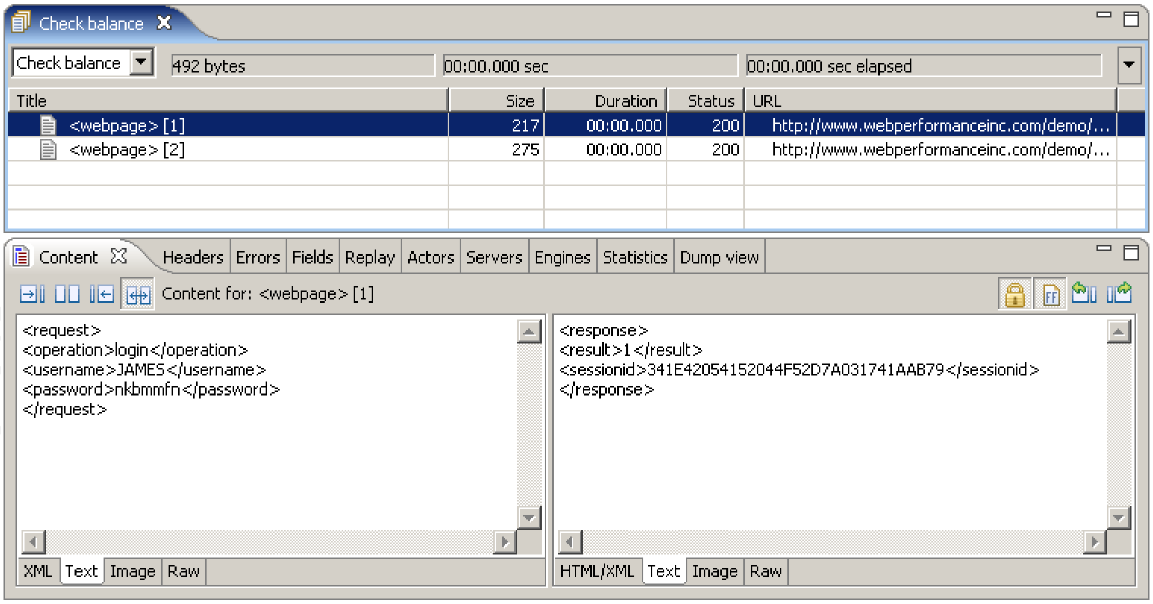
Import transaction

ⓘ Select the request and response files to import.

Request File: D:\tmp\WSDemo\loginRequest.txt

Response File: D:\tmp\WSDemo\loginResponse.txt

After importing both transactions, our testcase looks like this:



Check balance | 492 bytes | 00:00.000 sec | 00:00.000 sec elapsed

Title	Size	Duration	Status	URL
<webpage> [1]	217	00:00.000	200	http://www.webperformanceinc.com/demo/...
<webpage> [2]	275	00:00.000	200	http://www.webperformanceinc.com/demo/...

Content for: <webpage> [1]

```
<request>
<operation>login</operation>
<username>JAMES</username>
<password>nkbmmfn</password>
</request>
```

```
<response>
<result>1</result>
<sessionid>341E42054152044F52D7A031741AAB79</sessionid>
</response>
```

Step 1 is now complete.

> Note that the duration of the imported transactions is 0. Since these transactions have not yet been analyzed with a live server, the duration metrics cannot be determined. After a successful replay has been completed, consider *promoting* the replay to the base recording: Select the *Promote* button on the *Replay Properties* dialog - accessible from the replay drop-down at the top of the testcase editor.

Step 2 - Configuring session-tracking

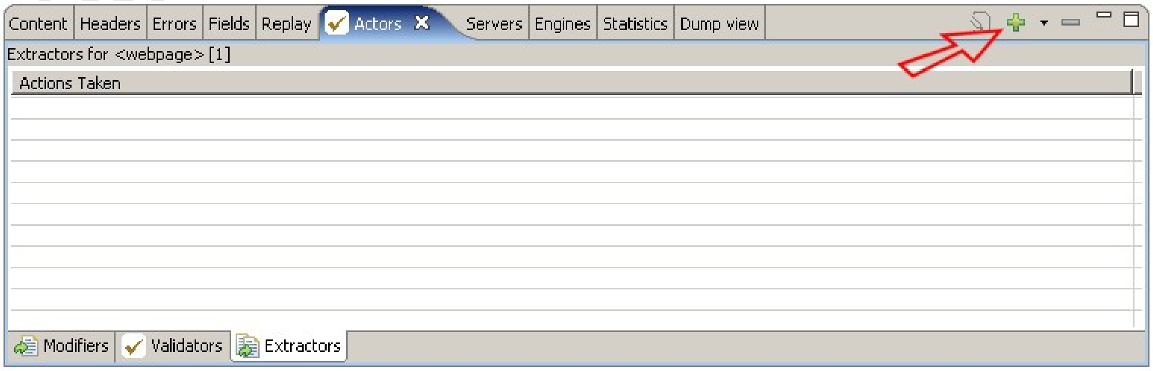
Before we can execute this testcase in the current form, we have to handle the session-tracking used by the service. We could replay the transactions exactly as we created them, but the service would fail on the second transaction, because the replay would send the session identifier that we put in the transaction. Since this session identifier has probably expired since we obtained it, the get-balance request needs to send the new session-identifier from the login response.

> Note that some services use cookies for session-tracking, much like a browser. If this applies to your service and you can record it using the recording proxy or you have imported *real* transactions captured in some other way, the cookie headers might be in place and will be handled automatically by Analyzer. In this case, this step may not be necessary.

Two steps are required to handle session-tracking for this testcase:

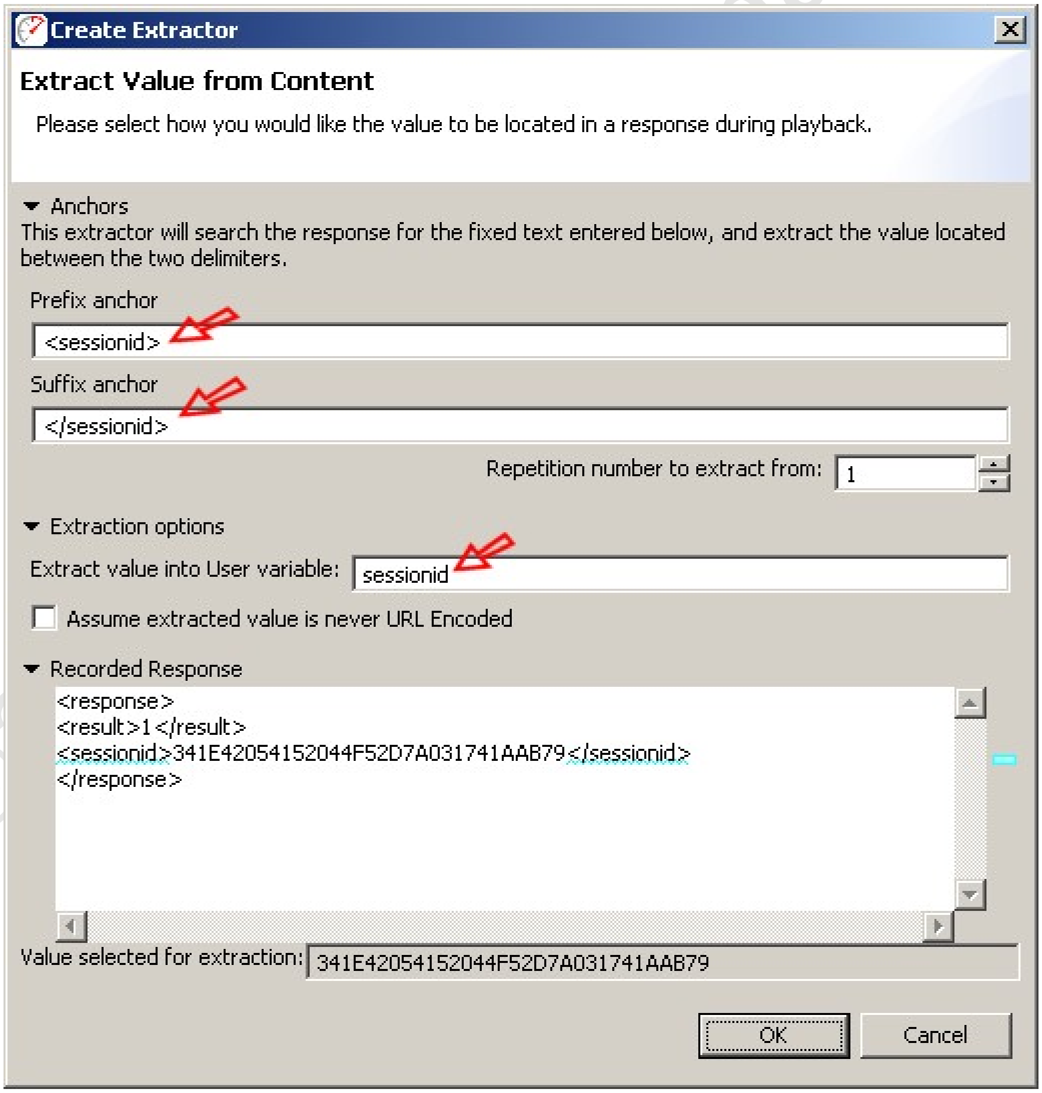
1. Extract the session identifier from the login response
2. Modify the session identifier sent in the get-balance request

To do this, activate the Actors view and select the login transaction in the testcase editor. The Actors view will initially appear as below (empty).



Press the *Add Extractor...* button (✚) to add a new extractor - the resulting dialog allows configuration of the extractor. In this case, we want to extract the session identifier which is located between the <sessionid> and </sessionid> tags (delimiters) in the response. As these values are entered in the *Prefix anchor* and *Suffix anchor* fields, the delimiters will be highlighted in the response content field in the lower third of the dialog. If either delimiter cannot be located, an error will be indicated at the top of the dialog. If both the prefix and suffix anchors are found, the target value for extraction will be displayed in the field at the bottom. Next we enter *sessionid* in the *Extract value in user variable* field. This will create a variable in the user state that will contain the session identifier when it is located by this extractor.

Note that if the delimiters appear several times and the first instance does not contain the desired value, the *Repitition number...* field may be changed to select the correct value.



Extract Value from Content

Please select how you would like the value to be located in a response during playback.

▼ Anchors

This extractor will search the response for the fixed text entered below, and extract the value located between the two delimiters.

Prefix anchor: <sessionid>

Suffix anchor: </sessionid>

Repetition number to extract from: 1

▼ Extraction options

Extract value into User variable: sessionid

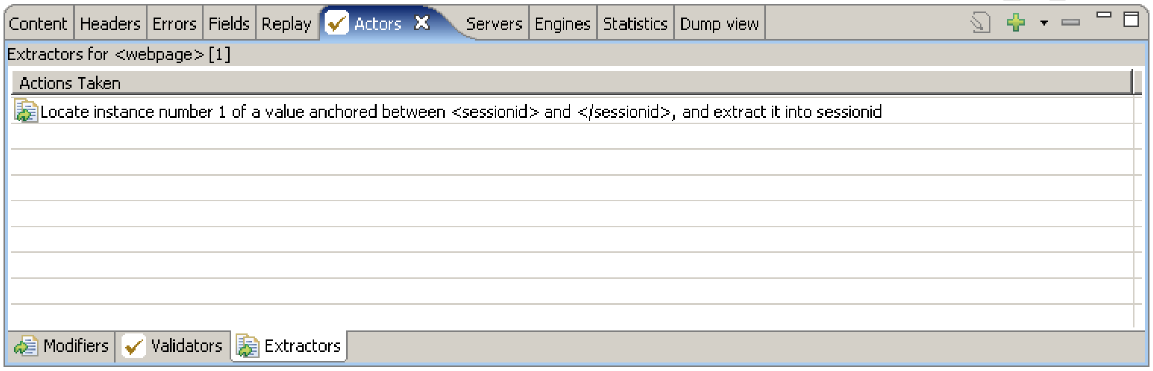
□ Assume extracted value is never URL Encoded

▼ Recorded Response

```
<response>
<result>1</result>
<sessionid>341E42054152044F52D7A031741AAB79</sessionid>
</response>
```

Value selected for extraction: 341E42054152044F52D7A031741AAB79

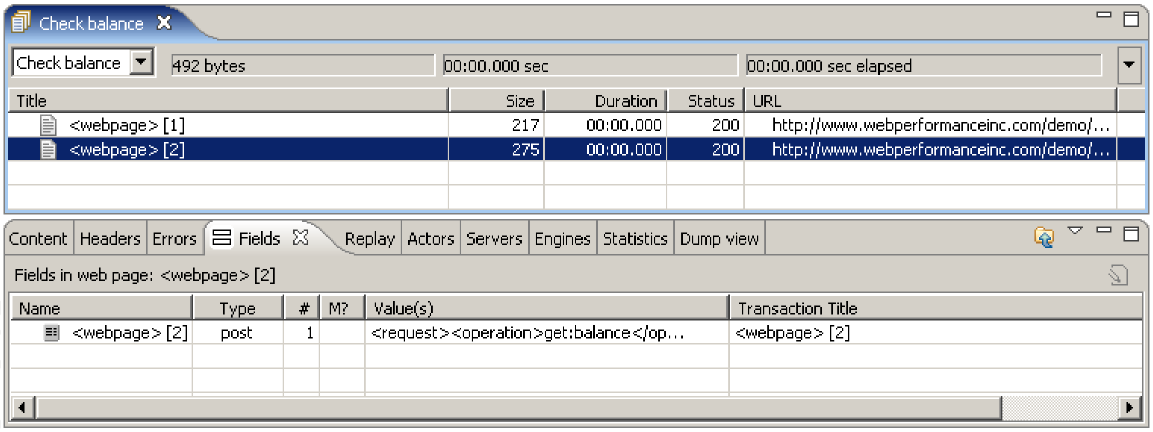
After pressing the *OK* button, the Actors view will show the Extractor like this:



Extractors for <webpage> [1]

Actions Taken
Locate instance number 1 of a value anchored between <sessionid> and </sessionid>, and extract it into sessionid

Once the session id has been extracted from the login response, it should be dynamically replaced in the get-balance request. This is accomplished by adding a modifier in the *Fields view*. Open the Fields view and then select the get-balance transaction in the testcase editor (2nd transaction). The fields in the request will be displayed as below.



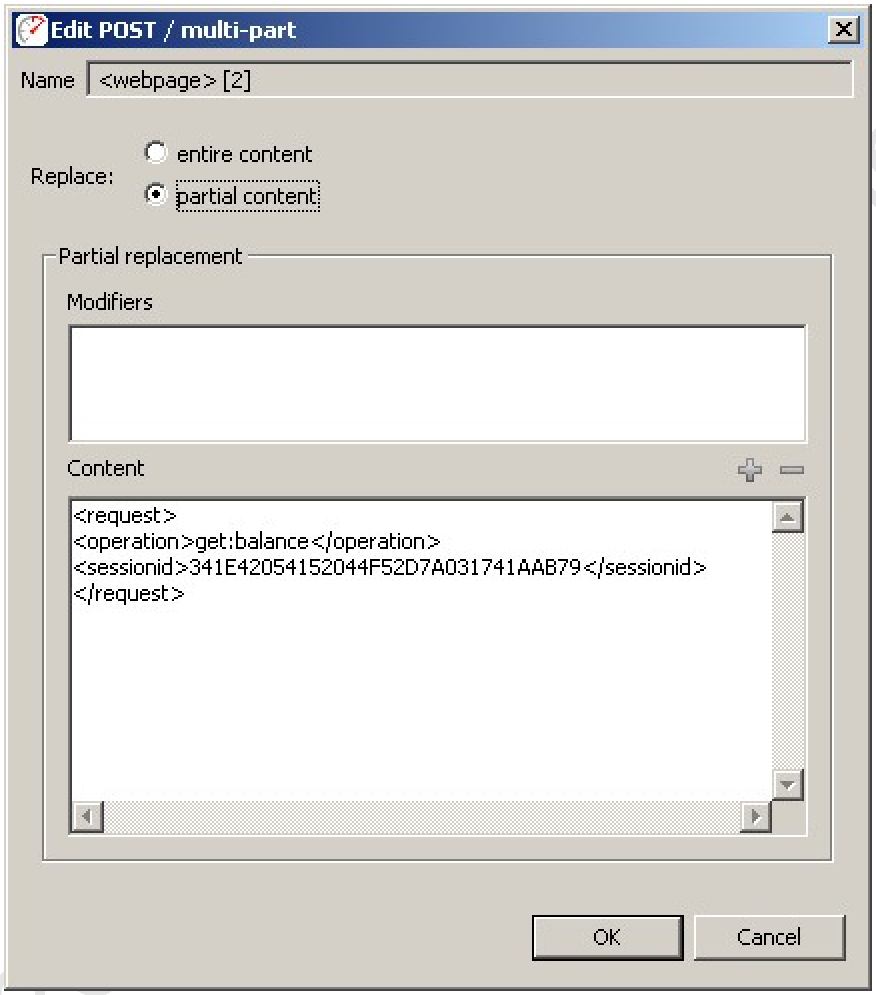
Check balance | 492 bytes | 00:00.000 sec | 00:00.000 sec elapsed

Title	Size	Duration	Status	URL
<webpage> [1]	217	00:00.000	200	http://www.webperformanceinc.com/demo/...
<webpage> [2]	275	00:00.000	200	http://www.webperformanceinc.com/demo/...

Fields in web page: <webpage> [2]

Name	Type	#	M?	Value(s)	Transaction Title
<webpage> [2]	post	1		<request><operation>get:balance</op...	<webpage> [2]

Since the request content in a web service can be any format, Analyzer presents it as a single *post* field. Double-clicking the *Modifier* column (M?) will open a dialog for adding a modifier. Since we only need to replace a small part of the request content, select the *Partial content* radio button. The dialog will now appear like this:

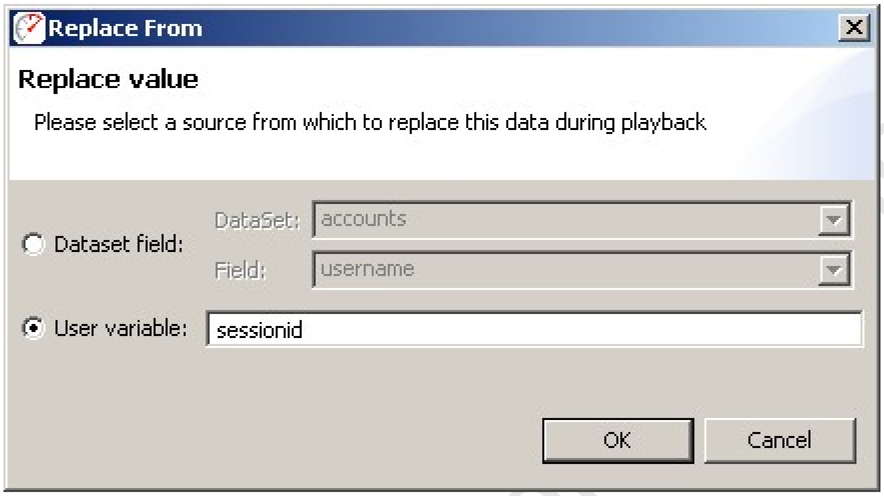


Name: <webpage> [2]

Replace: ○ entire content ● partial content

```
<request>
<operation>get:balance</operation>
<sessionid>341E42054152044F52D7A031741AAB79</sessionid>
</request>
```

Since no modifiers have been configured yet, the *Modifiers* list is empty. To add a modifier for the session identifier, select the session identifier value in the *Content* section and press the *Add* (✚) button. In the resulting dialog, the modifier should be configured to replace the recorded content with the User variable *sessionid* - which corresponds to the User variable that we created when configuring the extractor for the session identifier in the previous step. The configuration should look like this:



Replace From

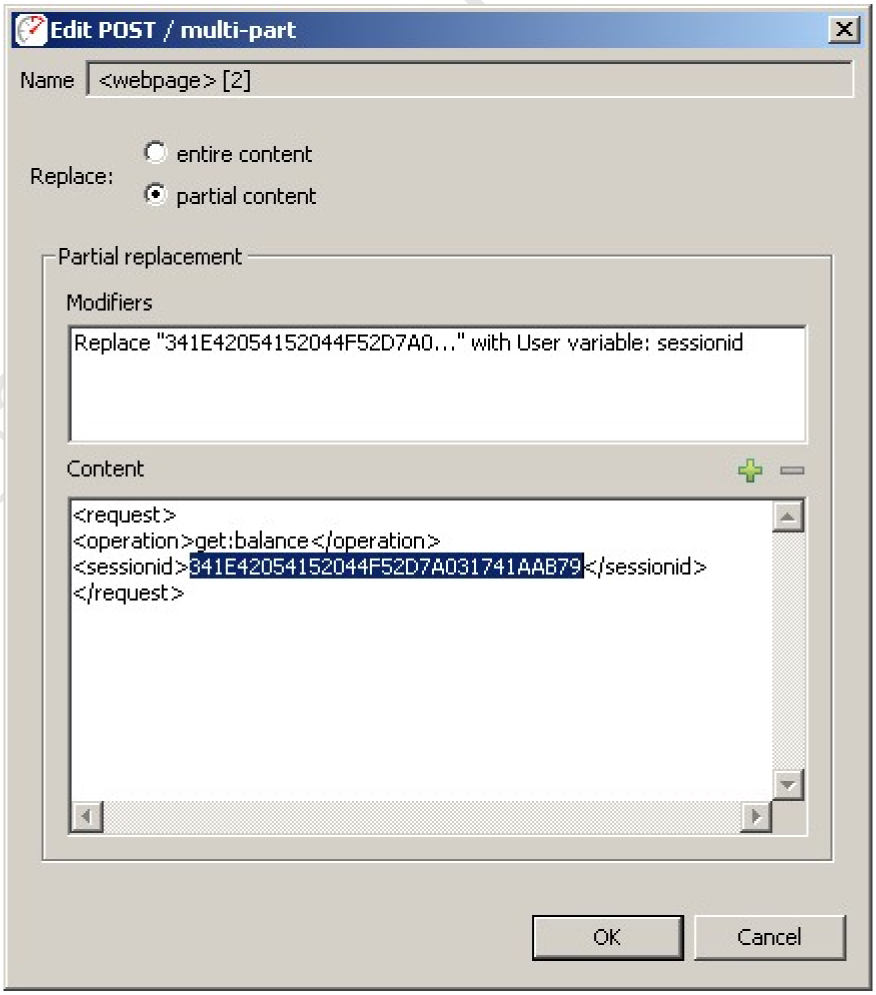
Replace value

Please select a source from which to replace this data during playback

○ Dataset field: DataSet: accounts Field: username

● User variable: sessionid

After pressing the OK button, the modifier will be added to the list:



Edit POST / multi-part

Name <webpage> [2]

Replace: ○ entire content ● partial content

Partial replacement

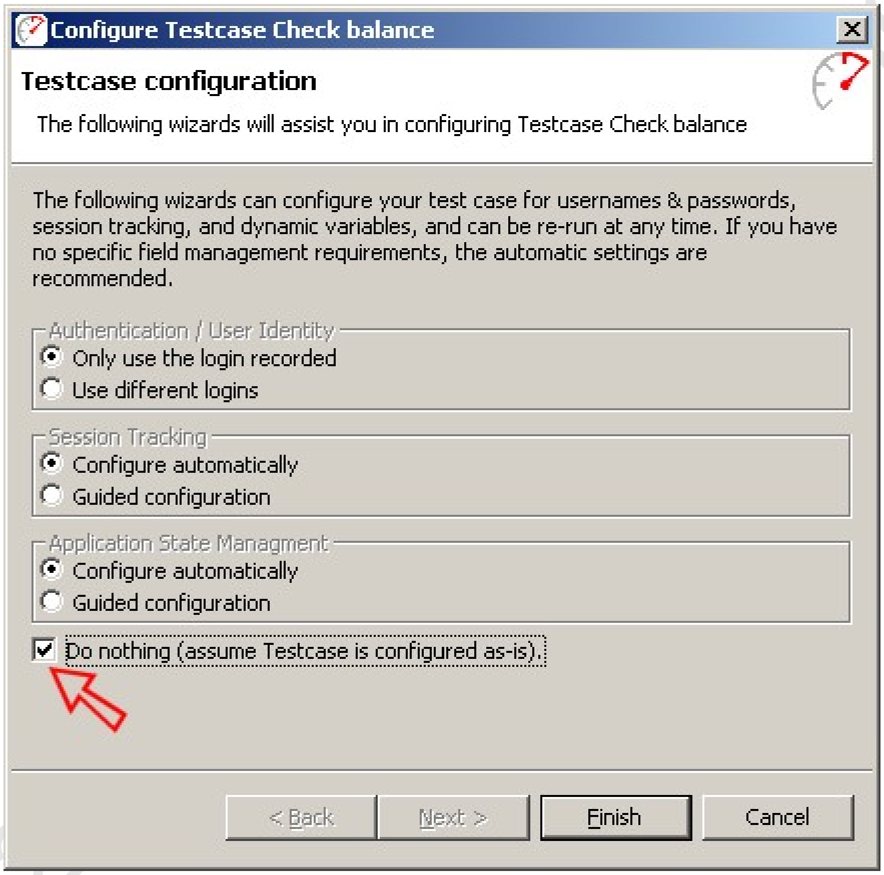
Modifiers

Replace "341E42054152044F52D7A0..." with User variable: sessionid

Content

```
<request>
<operation>get:balance</operation>
<sessionid>341E42054152044F52D7A031741AAB79</sessionid>
</request>
```

With session-tracking configured for the testcase, the testcase may be replayed. Pressing the Replay button (▶) in the toolbar will invoke the testcase configuration wizard. This wizard is very useful for complicated web applications but for simple web services it is frequently unnecessary. Select the *Do nothing...* option and press the *Finish* button.



Configure Testcase Check balance

Testcase configuration

The following wizards will assist you in configuring Testcase Check balance

The following wizards can configure your test case for usernames & passwords, session tracking, and dynamic variables, and can be re-run at any time. If you have no specific field management requirements, the automatic settings are recommended.

Authentication / User Identity

● Only use the login recorded

○ Use different logins

Session Tracking

● Configure automatically

○ Guided configuration

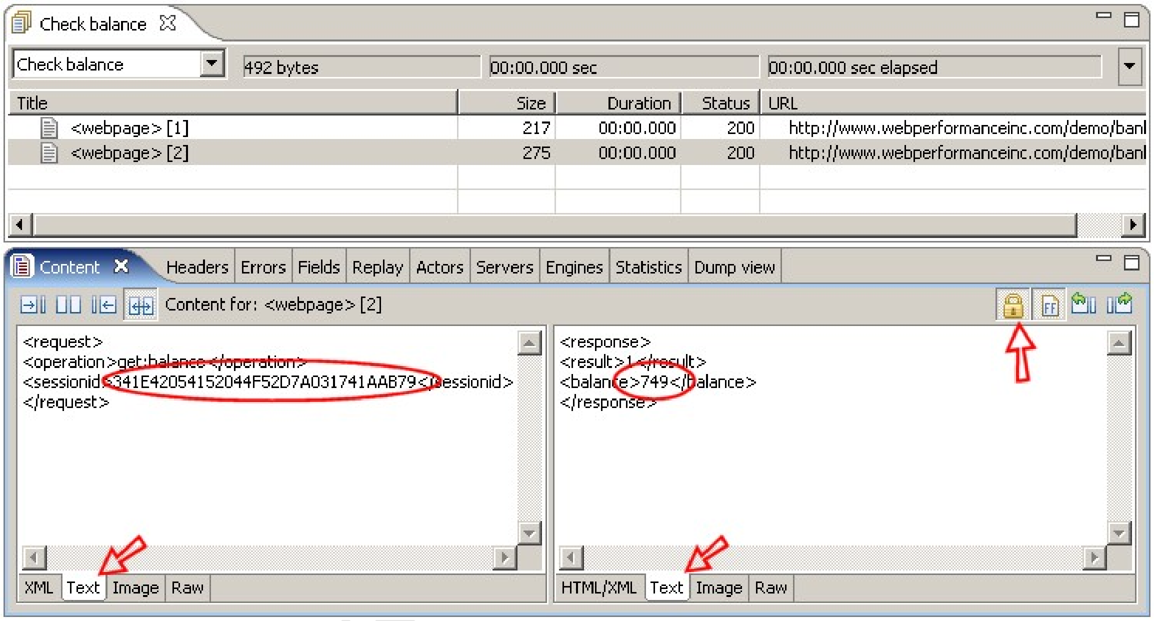
Application State Managment

● Configure automatically

○ Guided configuration

☑ Do nothing (assume Testcase is configured as-is).

After dismissing the Testcase Configuration wizard, the replay will begin. It will likely end very quickly, since it is a very simple and very short testcase. When it is complete, open the Errors view and then select the testcase to verify that no errors occurred during the replay. Then open the content view and select the get-balance transaction in the testcase editor to view the result:



Title	Size	Duration	Status	URL
<webpage> [1]	217	00:00.000	200	http://www.webperformanceinc.com/demo/banl
<webpage> [2]	275	00:00.000	200	http://www.webperformanceinc.com/demo/banl

Content for: <webpage> [2]

```
<request>
<operation>get:balance</operation>
<sessionid>341E42054152044F52D7A031741AAB79</sessionid>
</request>
```

```
<response>
<result>1</result>
<balance>749</balance>
</response>
```

Checking the value of the session identifier we can see that it is different from the original transactions we created - indicating that Analyzer was able to successfully extract the session identifier from the previous response and modify this request to send the correct session identifier. Also note that the balance returned is correct (we'll come back to this later in the tutorial). If you replay this testcase again, you should see a different session identifier with the same balance returned each time.

At this point we can declare success step 2 of this tutorial - we have configured the web service testcase so that it can be successfully replayed.

Note that since the Bank demo service does not use true XML, the XML viewers may not always format the content properly. In the screenshot above, the plain-text viewer has been selected for both the request and response content. Additionally, the tabs have been locked to prevent the view from switching back to the XML viewer each time a transaction is selected.

Step 3 - Configure the testcase for multiple users

The key difference between the simple replay completed in the last step and a load test is volume. A load test is simply a lot of replays, all happening at the same time. We could perform a load test with the testcase right now, if we wanted. But it would not be a very accurate simulation of the expected use-case of the system since the same person is not likely to be checking their balance simultaneously from multiple locations over and over again. Instead we would like the load tests to simulate many different users checking their balance.

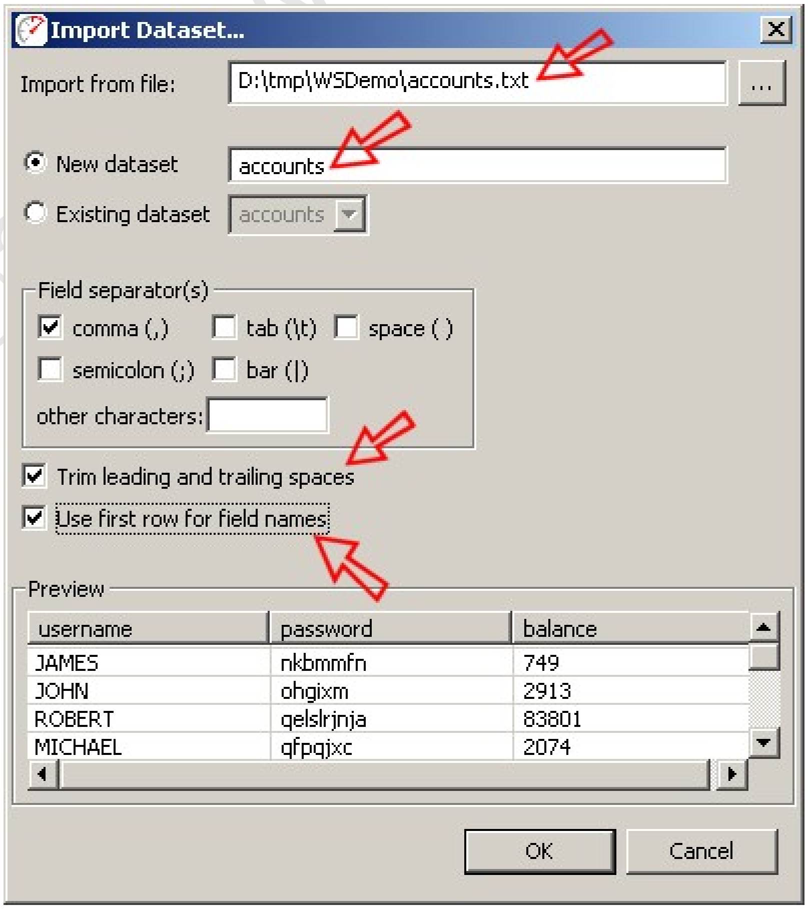
Creating test data

The first step in simulating multiple users is to create the list of users. For this tutorial, we will assume that the system is already populated with accounts and we have the list of user names, passwords and current account balances available to us.

Example users file:

```
username, password, balance
JAMES, nkbmmfn, 749
JOHN, ohgixm, 2913
ROBERT, qelslrjnja, 83801
MICHAEL, qfpqjxc, 2074
WILLIAM, nfbwhfh, 614
DAVID, hhjvksgue, 8365
RICHARD, rkipbo, 153
```

The next step is to import this data into Analyzer™. This is done from the *Dataset* node in the Navigator. Select the *Import...* item from the pop-up menu and complete the import dialog:



Import from file: D:\tmp\WSDemo\accounts.txt

● New dataset: accounts

○ Existing dataset: accounts

Field separator(s)

☑ comma (,) ☐ tab (\t) ☐ space ( )

☐ semicolon (;) ☐ bar (|)

other characters:

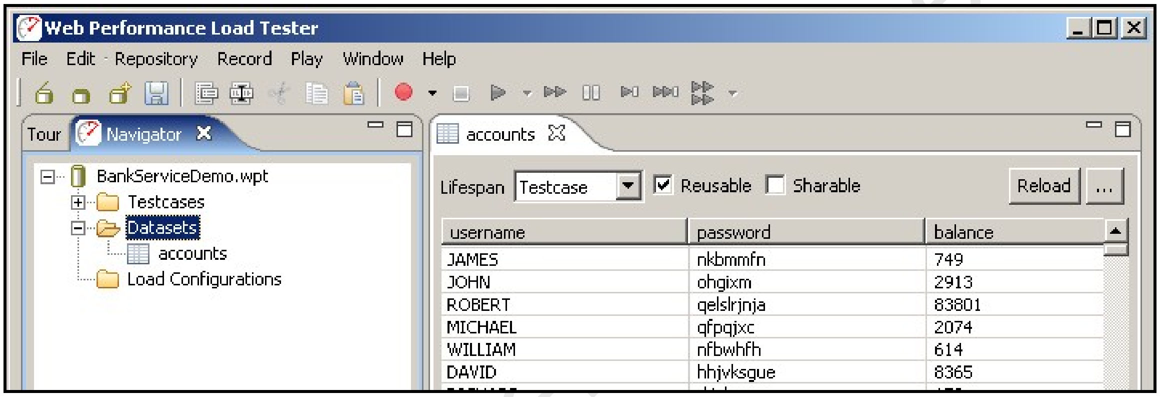
☑ Trim leading and trailing spaces

☑ Use first row for field names

Preview

username	password	balance
JAMES	nkbmmfn	749
JOHN	ohgixm	2913
ROBERT	qelslrjnja	83801
MICHAEL	qfpqjxc	2074

After importing the dataset, it will appear in the Navigator under the Datasets tree node and the dataset will open to show the imported data:



Lifespan Testcase ☑ Reusable ☐ Sharable

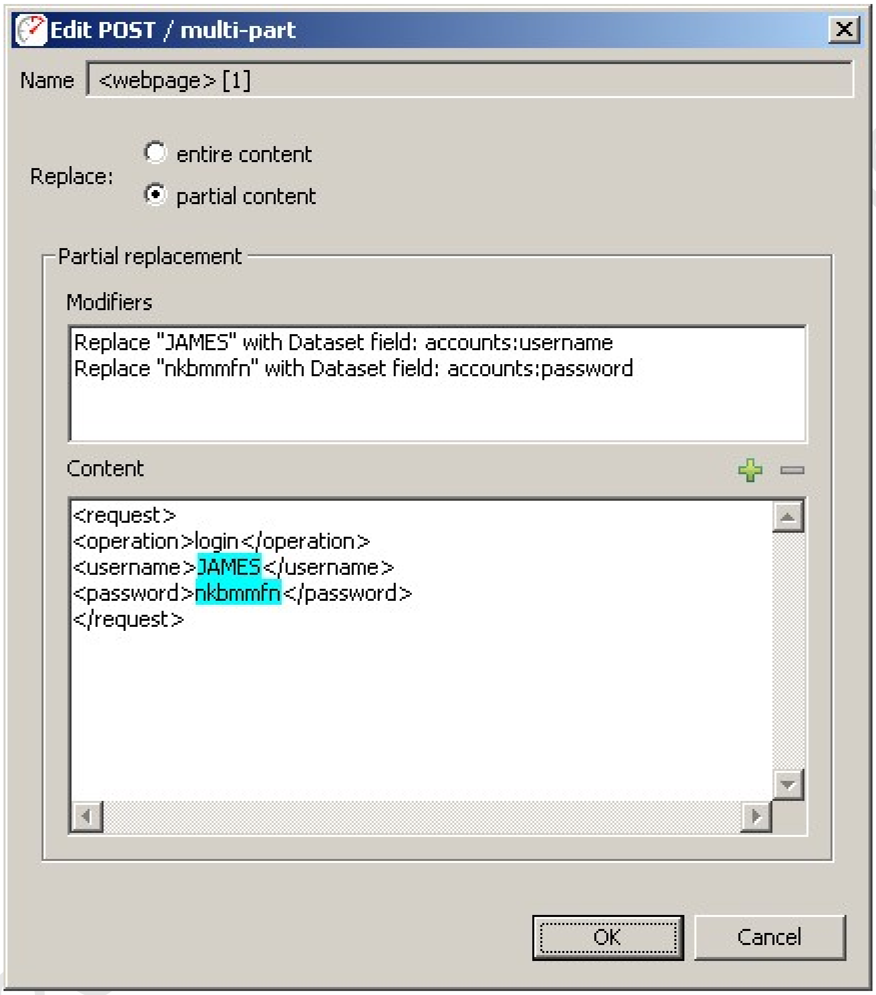
username	password	balance
JAMES	nkbmmfn	749
JOHN	ohgixm	2913
ROBERT	qelslrjnja	83801
MICHAEL	qfpqjxc	2074
WILLIAM	nfbwhfh	614
DAVID	hhjvksgue	8365

Note the 3 settings at the top of the dataset editor: Lifespan, Resuable & Sharable. The *Testcase* lifespan indicates that when a Virtual User (VU, or simulated user) selects a row of data from this data, it will use that row until the end of the testcase. When a dataset resuable, the row may be used more than once during a load test. When a dataset is not sharable, it can only be used by single VU at a time. The default settings, shown above, are usually correct for a dataset containing the user identity.

Customizing the testcase

Now that the user identity information is available, the testcase may be customized to use it. Two modifiers should be added to the first transaction to replace the username and password in the login request with values from the dataset. This configuration is similar to the configuration of the session-identifier described earlier in this tutorial:

1. Open the Fields view
2. Select the login transaction in the testcase editor
3. Double-click the modifier column (M?) and select the *partial content* radio button
4. Select the username in the *Content* text area, press the Add button (✚) and select the *accounts* dataset and *username* field.
5. Repeat previous step for the password



Edit POST / multi-part

Name <webpage> [1]

Replace: ○ entire content ● partial content

Partial replacement

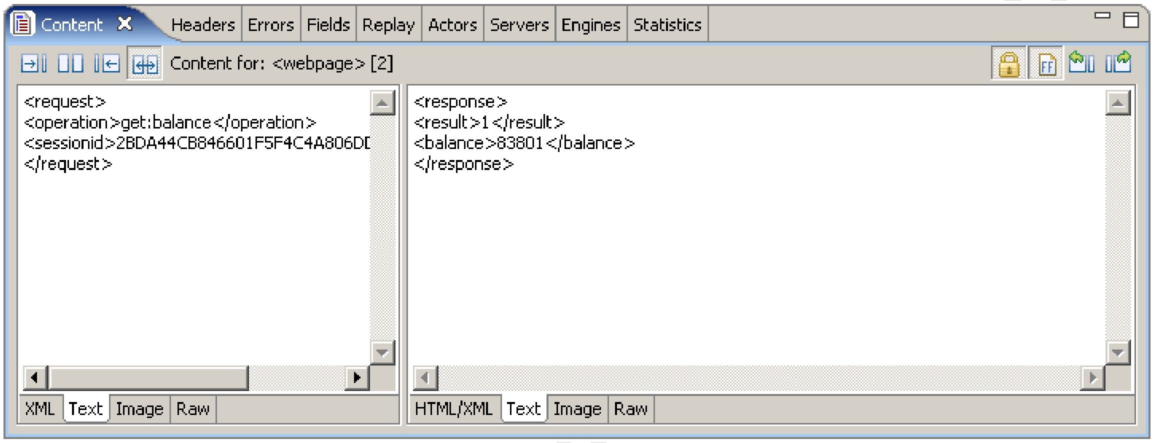
Modifiers

Replace "JAMES" with Dataset field: accounts:username
Replace "nkbmmfn" with Dataset field: accounts:password

Content

```
<request>
<operation>login</operation>
<username>JAMES</username>
<password>nkbmmfn</password>
</request>
```

The testcase is now configured to use a different user identity each time the testcase runs. Replaying the testcase, as described earlier, will allow you to easily verify this behavior. Each time the testcase is replayed, a different username and password are sent in the requests and a different account balance is returned. The following shows the get-balance transaction from the 3rd replay. You can see that the balance, 83801 matches the balance for that user in the dataset.



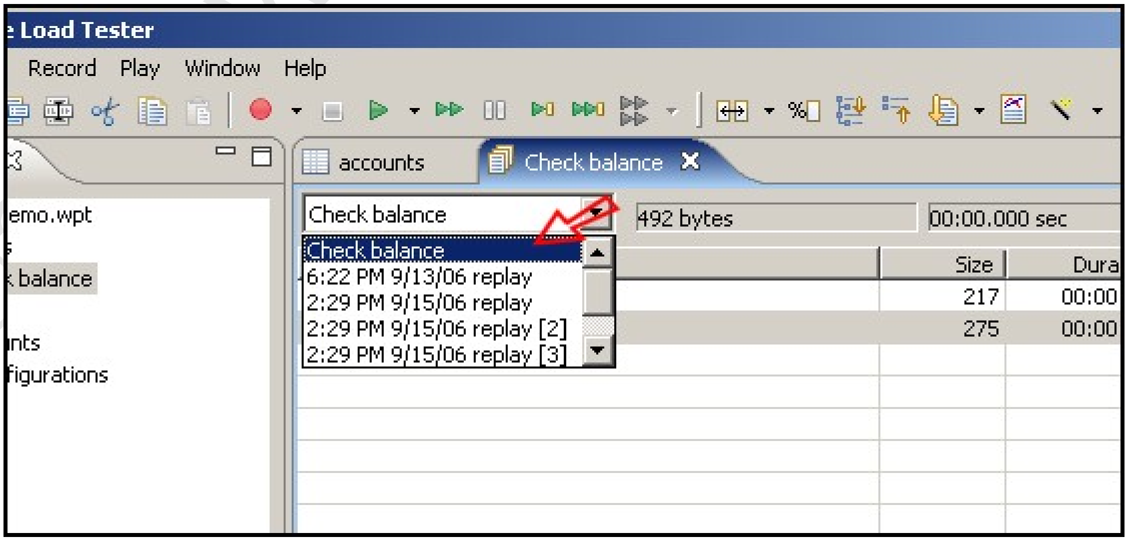
```
<request>
<operation>get:balance</operation>
<sessionid>2BDA44CB846601F5F4C4A806DI
</request>
```

```
<response>
<result>1</result>
<balance>83801</balance>
</response>
```

Configure validation

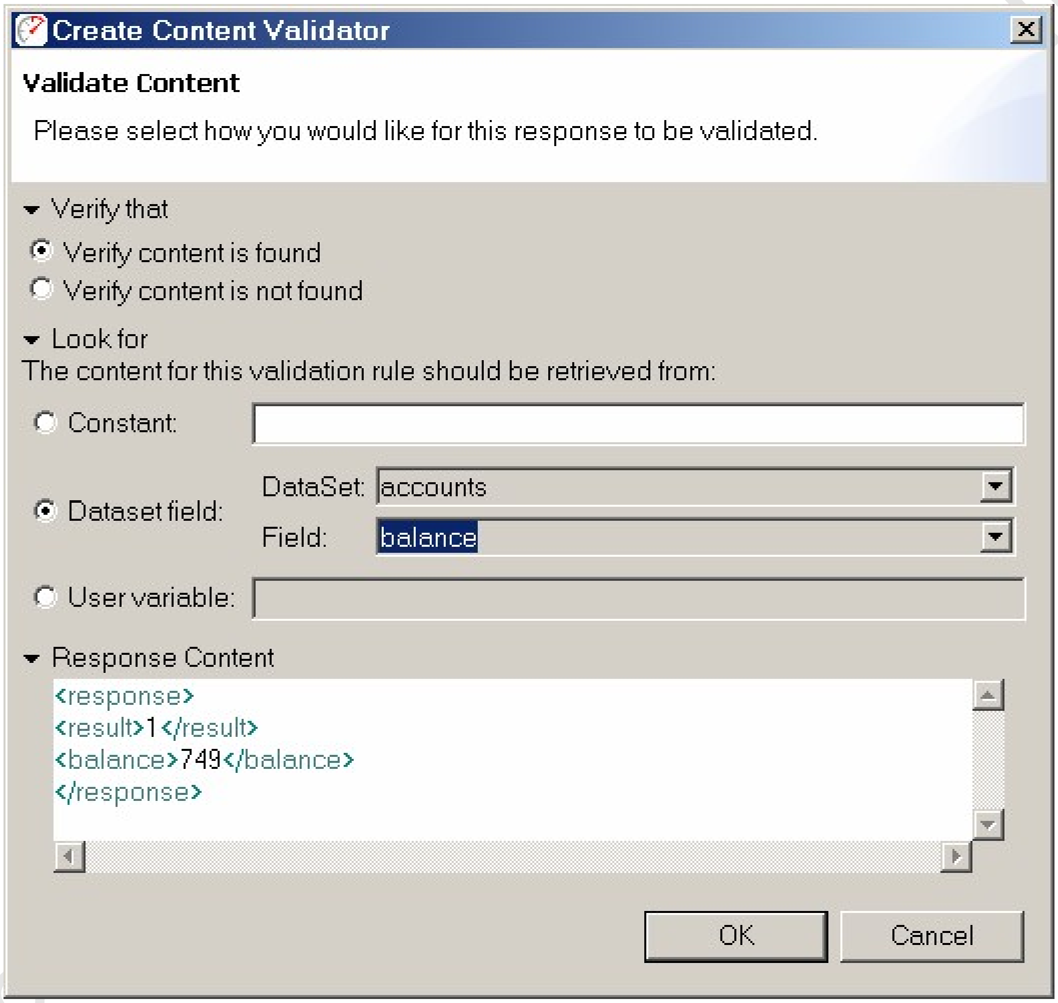
Validating the results in this way is easy for a few replays, but is not practical for a load test. It would be more efficient to have Analyzer automatically validate the balance returned in each testcase against the value in the dataset.

To configure a validator, first return to the original testcase in the editor:



Next, open the Actors view and then select the get-balance transaction in the testcase editor. Then select the *Validators* tab at the bottom of the Actors view. You will see the automatically-applied status-code validator already present in the list of validators. Press the add button (✚) to add a new validator.

Configure the validator to look for the account balance from the dataset in the response content by selecting the *Dataset field* and radio button and then select the *balance* field in the *accounts* dataset, as shown below:



Create Content Validator

Validate Content

Please select how you would like for this response to be validated.

Verify that

● Verify content is found

○ Verify content is not found

Look for

The content for this validation rule should be retrieved from:

○ Constant:

● Dataset field: DataSet: accounts Field: balance

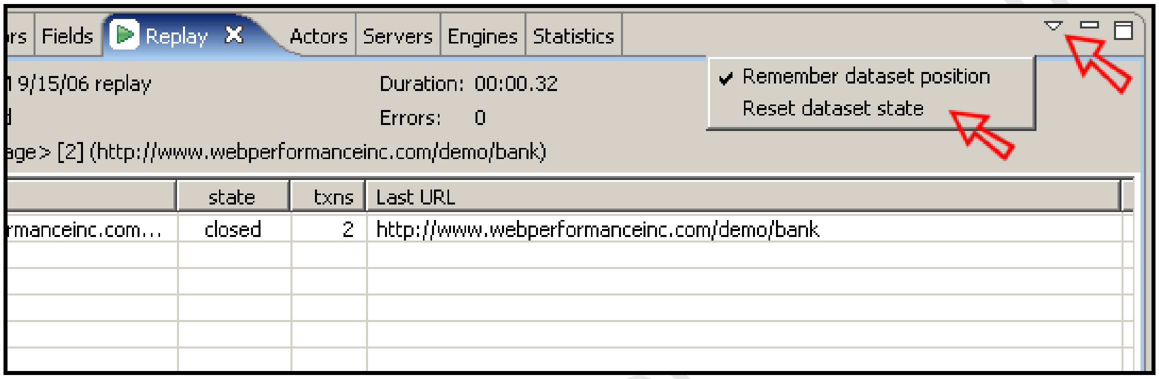
○ User variable:

Response Content

```
<response>
<result>1</result>
<balance>749</balance>
</response>
```

After applying the validator, replay the testcase again and check the Errors tab to verify that no errors were encountered. There should be no errors because our dataset accurately reflects the system being tested.

To see what the error looks like from the validator, the dataset will have to be changed to purposely have wrong data in it. Open the *accounts* dataset and change the value of the first entry in the *balance* column (double-click a cell to edit it). Before replaying, we will need the force the replay mechanism to reload the dataset - to do this, open the Replay view and select the *Reset Dataset State* item from the drop-down menu on the right side of the view:



1 9/15/06 replay

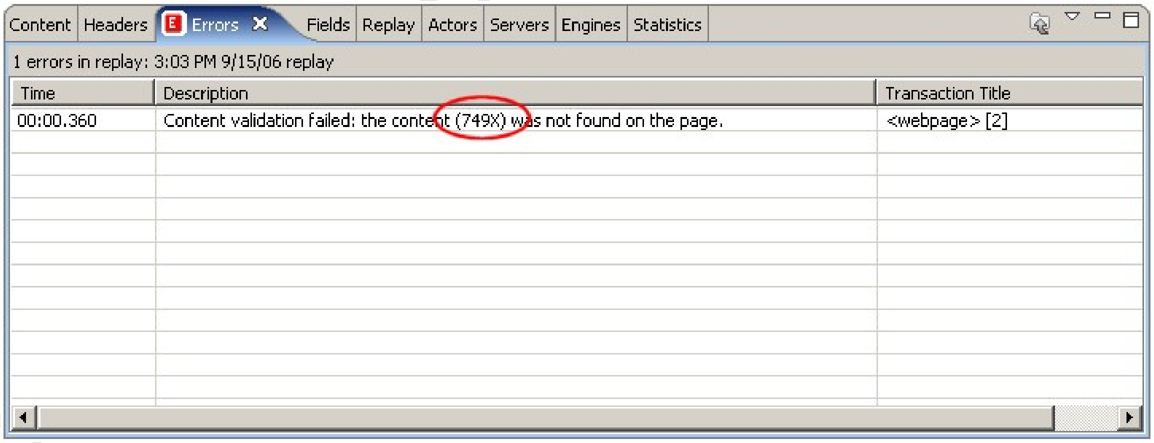
Duration: 00:00.32

Errors: 0

age> [2] (http://www.webperformanceinc.com/demo/bank)

	state	txns	Last URL
rmanceinc.com...	closed	2	http://www.webperformanceinc.com/demo/bank

Replay the testcase again and then open the Errors view. The validation error displayed indicates that the account balance that I entered in the dataset (749X) could not be found in the response content for the get-balance transaction:



1 errors in replay: 3:03 PM 9/15/06 replay

Time	Description	Transaction Title
00:00.360	Content validation failed: the content (749X) was not found on the page.	<webpage> [2]

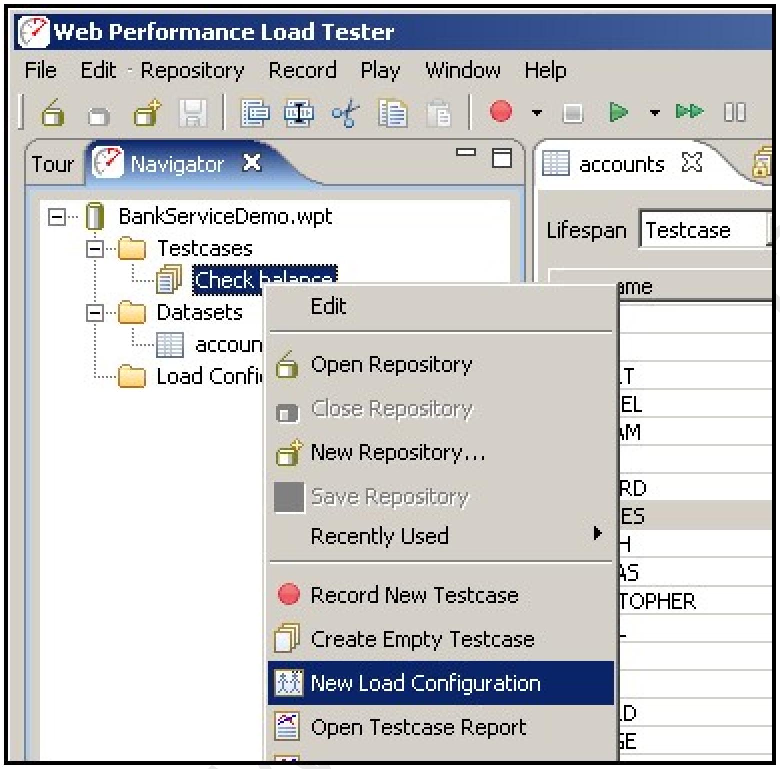
Don't forget to change the dataset value back to the correct value before moving on!

Step 3 is now complete - we can move on to running a load test.

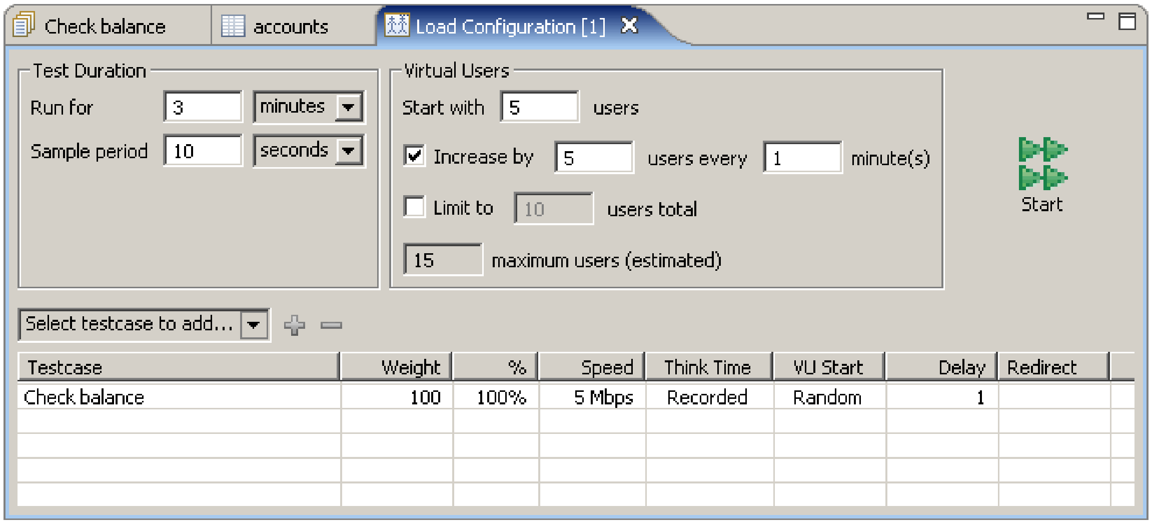
Step 4 - Run load test and analyze results

Creating the load configuration

After recording, configuring and verifying each testcase, the next step towards a load test is to create a load configuration. Select the *Check Balance* testcase in the Navigator and then select the *Create Load Configuration* item from the pop-up menu:



The default settings, shown below, should be fine for demonstration purposes.



Test Duration

Run for 3 minutes

Sample period 10 seconds

Virtual Users

Start with 5 users

☑ Increase by 5 users every 1 minute(s)

☐ Limit to 10 users total

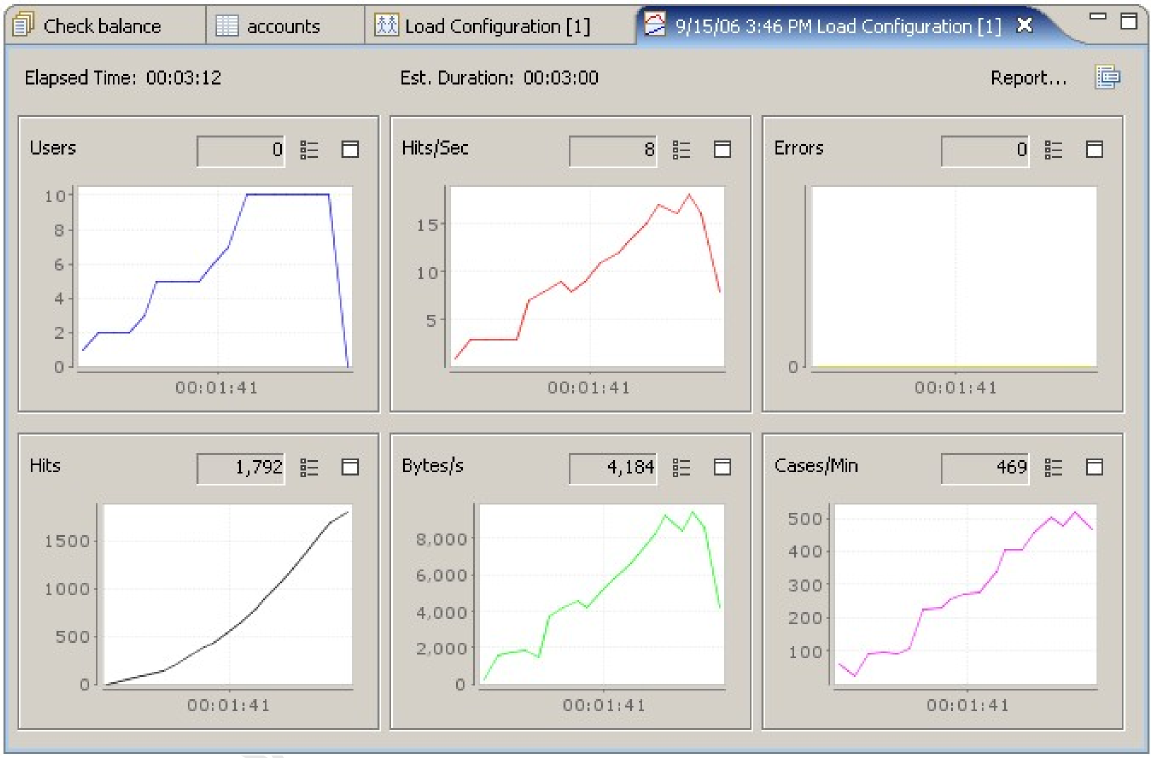
15 maximum users (estimated)

Testcase	Weight	%	Speed	Think Time	VU Start	Delay	Redirect
Check balance	100	100%	5 Mbps	Recorded	Random	1	

Running the load test

At this point, the hard work is done! Push the *start* button to start the load test.

The test will add 5 VUs each minute - if you are using a demo license, it will stop at 10 VUs for the last minute of the test. When the test is complete, the results screen will look something like this:



Elapsed Time: 00:03:12

Est. Duration: 00:03:00

Users 0

Hits/Sec 8

Errors 0

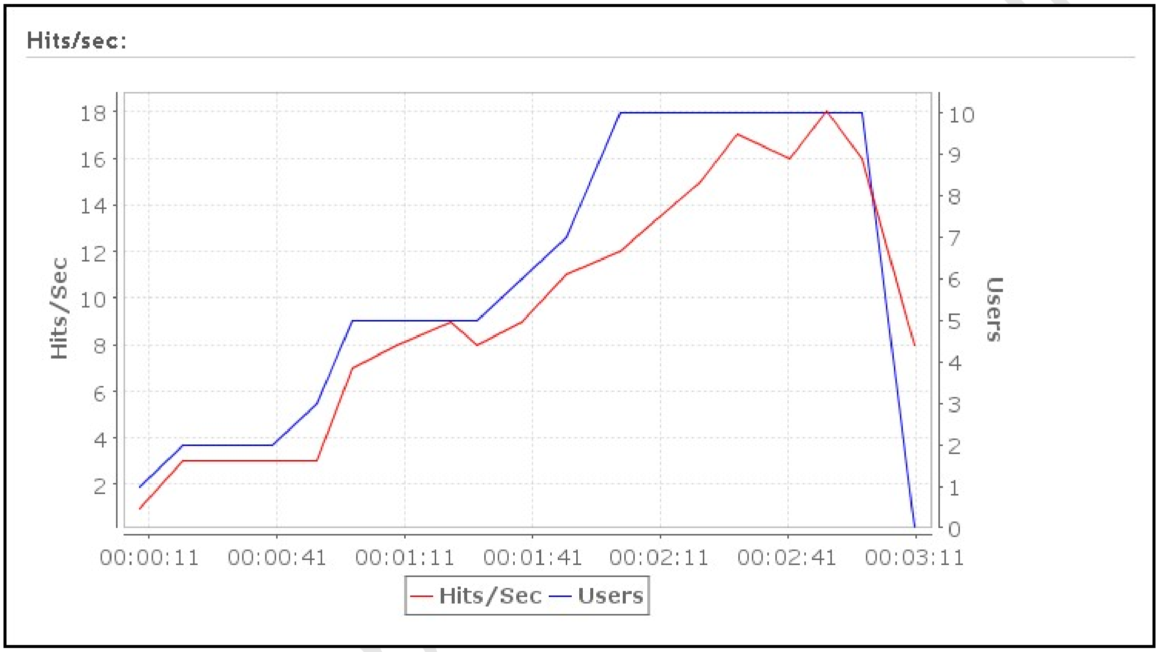
Hits 1,792

Bytes/s 4,184

Cases/Min 469

Analyzing the Results

The test results screen picture above, provides some basic metrics for monitoring the progress and state of the test - most notably the number of Users, Hits/Sec and Errors. For a more detailed analysis open the test report: press the *Report...* button on the results screen. In the first section of the report, you will see overall statistics presented on charts such as the one pictured below. This chart shows that the total Hits/sec scaled linearly as the number of users increased. This is a key measure of server performance and this test did very well!



A large amount of information is available in the test report at varying levels of detail. For a good example of a load test report, see the website: http://www.webperformanceinc.com/load_testing/reports/

Summary

In this tutorial we have demonstrated the basic process for testing a web service:

1. create the testcase
2. configure the testcase to handle session-tracking mechanism and test using the replayer
3. configure the testcase to simulate multiple user identities and test
4. configure the testcase to validate the result of the testcase
5. create a load configuration
6. run the test
7. analyze the results

Good luck!

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: I cannot record - what next?
A: Follow our Recording Troubleshooting Guide

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 question).

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.

Q: I have URLs in my testcase from a server that I do not wish to test.
A: If the URLs are not important to the testcase (such as images or links for link tracking, click analysis etc), they can be deleted from the testcase using the *Cut* item in the pop-up menu for the transaction. These URLs can be added to the URL blocking list - see the Blocking Undesired Transactions section of the manual.
If all the URLs for a particular server (host name) should be ignored, you can use the Host name blocking list, also described in the Blocking Undesired Transactions section.

Testcase Analysis FAQs

How can I determine the total size of a web page, including images and scripts?

1. Record the pages
2. In the Editor, check the *Size* column.
3. Expand the page in the tree to see the sizes of individual resources on the page

How can I examine the HTTP headers my server is returning?

1. Record some pages from your server
2. Open the Headers view
3. Select the page or URL of interest in the Editor

How can I see the cookies my browser is sending to a web server?

The cookies are sent between browser and server in the *Cookie* and *Set-Cookie* headers. See the Headers HowTo.

How can I determine if my web pages are getting faster or slower?

Follow these steps in the Quick Start Guide:

1. Create a recording
2. Replay a testcase
3. Analyze the Performance Changes

How can I find the slowest pages on my website?

1. Record the pages of your website
2. In the Editor, click the *Duration* column to sort the recording by page duration

How can I find errors on my website?

1. Record the pages of your website
2. Open the Error view

How fast will my web pages be over a modem?

There are two ways to answer this. If you have not already created a recording of the pages of interest:

1. Start a new Recording
2. On the *Start Recording* dialog, selected the desired modem/network speed.
3. Inspect the web page durations in the Editor

If you already have a recording of the pages, you can replay it with a specific network speed this way:

1. Record the pages of your website

2. Open the Replay View and set the Bandwidth Limit to the desired modem speed
3. Replay the testcase
4. Inspect the web page durations in the Editor ...or...
5. Open a Performance Trend chart for the testcase to see the difference in speed of each page plotted on a chart

How can I find parts of my website that do not match my performance goals?

1. Record the pages of your website
2. Configure one or more performance goals
3. Inspect the replay in the Editor - failed goals will be indicated by the ❗ icon.

How can I export report data to other formats?

Each data table has a link at the bottom titled *export to csv*. Clicking this link will invoke a dialog for saving the data. When viewing the report in an external browser, clicking the link will show the data in the browser. Most browsers have a function for saving a link content rather than navigating to it. In IE, the context menu item "Save Link As..." will perform this function.

Testcase Configuration FAQs

How can I change the URL recorded for a Web Page or transaction?

1. Open the Headers View
2. Select the desired Web Page or transaction
3. Edit the URL

How can I change a header in a transaction?

1. Open the Headers View
2. Select the desired Web Page or transaction
3. Edit the Header

How can I change the testcase to send different values in place of form fields or query parameters?

1. Open the Fields View
2. Select the testcase in Navigator
3. Locate the field(s) in the Fields View
4. Single values (or duplicate identical values) can be edited in-place by double-clicking the table cell

5. Multiple unique values can be changed to all have the same value by opening the Edit Field dialog (*Edit* button) and then entering a *Constant Value*.

How can I change the testcase to send different values in a form field on each replay?

1. Create or import the desired values in a dataset
2. Configure modifiers on each field/parameter in the Fields View by opening the Edit Field dialog (*Edit* button)
3. Each time the testcase is replayed, the next value from the dataset will be used (depending on the dataset configuration). To reset the dataset to the beginning, select the *Reset dataset state* item from the Replay View menu.

How can I change the username and password used in the testcase?

See the Authentication Section.

How can I create custom transactions or testcases without recording them?

Each transaction may be imported one at a time from an existing file.

How can I repeat part of a testcase?

Open the testcase properties dialog (*Properties...* item from the pop-up menu on the testcase in the *Navigator*). You can select the start-point and end-point for looping within the testcase. When the Virtual User runs the testcase, it will start from the beginning and proceed to the configured end-point (Run-To Page). Then it will restart at the start-point (Restart-At Page) and continue. When the load test is stopped, it will then run to the end of the testcase.

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.

Q: How can I replay every testcase in a repository
A: Select the *Advanced Replay...* option from the Replay toolbar button (▶) drop-down menu and select the "All testcases..." option.

Q: How can I replay testcases at night?
A: Select the *Advanced Replay...* option from the Replay toolbar button (▶) drop-down menu and select the "Schedule for..." option.

Q: How can I run replays in an automated process?
A: See the Command Line Tools section

Load Testing FAQs

Q: During a load test, I see many of these errors: "Unable to establish a connection to the server" What does this mean?

A: It means that the Virtual User failed to connect to the target server when initiating a request for a URL. If it happens with only a single user running then it could be a network or DNS problem. If the error does not occur at the beginning of a test but becomes more frequent as the test progresses, then the problem is frequently caused by a server configuration mistake. Check the server configuration settings for the maximum number of concurrent connections. This number should be at least double the maximum number of Virtual Users being simulated during the test.

Q: During a load test, I am seeing errors that indicate a 302 status code was received from the server when it was expecting a 200 (or some other code). Is this a problem?

A: The 302 status code indicates a redirect (forward) to another URL. When this is not expected during a load test, the most common cause is a session-tracking or authentication problem. When most servers detect an invalid session or invalid authentication, the will frequently return a redirect to the application's login screen. If you have not already run the Testcase Configuration Wizard (which normally runs automatically), you should run it from the Navigator view using the pop-up menu for the testcase in question.

If you have run the wizard and still encounter this error, you must determine what kind of session-tracking and authentication is being used by the application. If possible, it would also be helpful to find the exact cause of the condition that is causing the application to return the unexpected 302 - checking the server logs or discussing the problem with the application developers may help determine the cause. When you have this information, you may submit a support request (Help->Support Request) and send the test results to our support system.

Q: After a load test, Load Tester displays the message "Missing samples from engines." What does this mean?

A: When using remote engines during a load test, there may be times when the load engine is not able to promptly communicate with the controller. This can be caused by the engine approaching it's CPU, memory or network bandwidth capacities or by network congestion between the engine and the controller. In this case, the engine is usually able to continue performing in the load test, but the summary graphs may show subtle "dips" where data from a specific engine is unavailable.

Q: How can I see more detailed statistics from my load test results?
A: Activating the Statistics view and selecting the test results will display the detailed statistics for the test. Navigation within the view allows display of summary, testcase, server, engine, page and URL statistics. The detailed

statistics may also be exported from the *Export* item in Navigator pop-up menu for the selected test results.

Q: When running a load test with load engines on Windows, I see "Engine operating system out of network sockets" errors. What does this mean?
A: By default, when a Windows socket is closed by an application, Windows waits 240 seconds before closing it. During this time, the port that the socket used is placed in the TIME_WAIT state and cannot be reused. Windows also has a restriction on the highest port number that can be used (5000) when an application requests any available user port from the system.
Therefore, the Windows socket defaults may not accommodate heavy TCP/IP traffic very well without some tuning.

To alleviate the problem, we recommend changing the Windows default values of two registry keys. You must use the Windows registry editor (regedit) to make the changes, so be very careful. Make the following changes:

- Reduce the time-out wait time for closed sockets. Change the value of the registry key HKEY_LOCAL_MACHINE\System\CurrectControlSet\services\Tcpip\Parameters\TcpTimedWaitDelay

from its default of 240 to a smaller number. The valid range it 30-300 (decimal).

- Increase the maximum port number available to applications. Change the value of the registry key HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Tcpip\Parameters\MaxUserPort

from its default 5000 (decimal) to a higher number. The valid range is 5000-65534 (decimal).

For more information on these two registry keys, you can search for TcpTimedWaitDelay and MaxUserPort on the Microsoft Technet and Knowledge Base.

Q: When running a load test with load engines on Unix (Linux/Solaris), I see "Engine operating system out of network sockets" or "too many files open" errors. What does this mean?
A: In Unix environments, a file handle is used for each open network connection (socket). The number of resources allocated for these handles is controlled by the *ulimit* setting.

To see the current setting, run *ulimit -n*

To change the setting, run *ulimit -n NNNN* where NNNN is the new value.

Choosing a new value is dependent on the nature of your testcase, the number of users being simulated and other factors. We usually increase it by a factor of

10 over the system default, since load testing is considerably more network intensive than the default installation of most OSes expect.

Note that the above solution is not permanent - it will only affect the current session. For a more permanent solution, add a line like the following to the */etc/security/limits.conf* configuration file (may be different depending on the Linux/Unix distribution. You will need to logout/in before this change will take effect.

```
  <username> hard nofiles NNNN
```

Q: My load test generated the following errors, what does it mean?

```
Page did not contain required variable(s): #variable1, #variable8,
#variable23
```

A: It means that the software expected to find certain variables in the pages returned from the server - because they were there in the recorded testcase. The most common causes of this error are:

1. An unexpected page was returned from the server, and this page is *very* different from the recorded page, due to an error condition, such as improper authentication.
2. The page returned from the server is correct, but the field is missing from that page.
3. The page returned from the server is correct, but the software incorrectly identified the field to be retrieved or has a name that changes for each iteration of the testcase.

Viewing the error page in the content view should help determine if #1, #2 or #3 is the case:

1. The authentication of the testcase may not be configured properly - see the Authentication section for more information.
2. The ASM wizard tries to find every possible field that may be passed from one page to the next in order to accurately simulate a testcase. Many of these fields may not be necessary or may be the same for every iteration of the testcase. In this case, re-running the ASM wizard and removing those fields from consideration may alleviate the error while still allowing the testcase to be simulated correctly.
3. Ask the developer of the web-application if this field is needed within the context of this particular testcase. If it is, enter a support request so we may help you with advanced configuration to handle this situation.

Q: How can I run a load test at night?

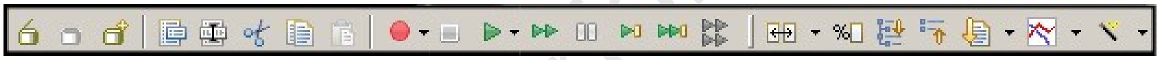
A: Select the *Schedule test...* option from the Loadtest toolbar button (⏩) drop-down menu and select the "Schedule for..." option.

Reference Manual

Views

Toolbar buttons

The toolbars provide quick access to frequently-used operations.



Repository

Open an existing repository
Create a new repository
Close the selected repository

Edit

Edit properties for selected item
Rename the selected item
Cut the selected item
Copy the selected item
Paste the selected item

Record & Replay

Start a new recording (use the drop-down menu to access advanced recording options)
Stop a recording or replay
Start replaying a testcase (use the drop-down menu to access advanced replay options)
Start replaying a testcase, without pausing for page think times
Pause the replay
Replay the next URL and pause
Replay to the end of the next/current page and pause
Start a load test

Editor

Compare the replays of a recording
Show differences in comparison as a percentage
Expand all pages in the testcase tree
Collapse all pages in the testcase tree

Sort the testcase
Chart the performance of the testcase

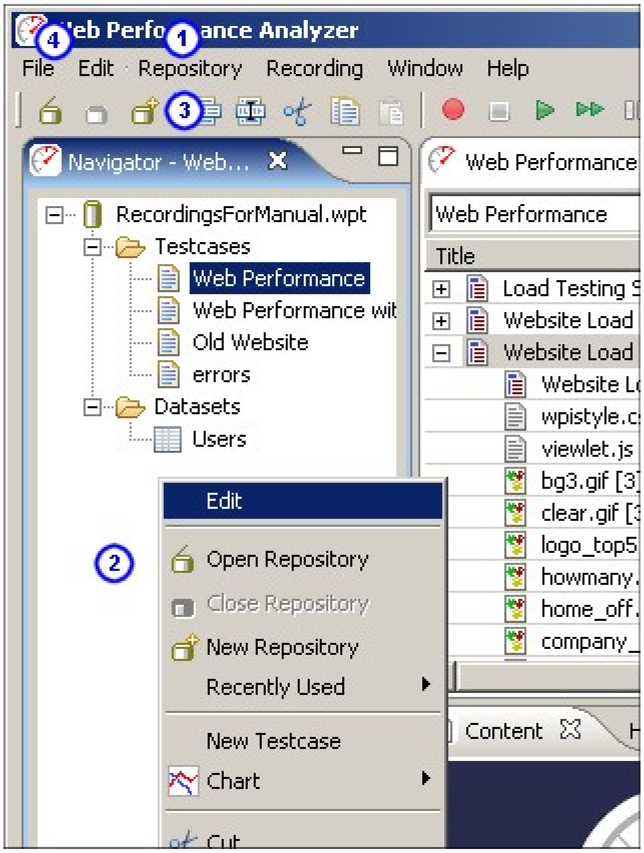
Navigator

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.

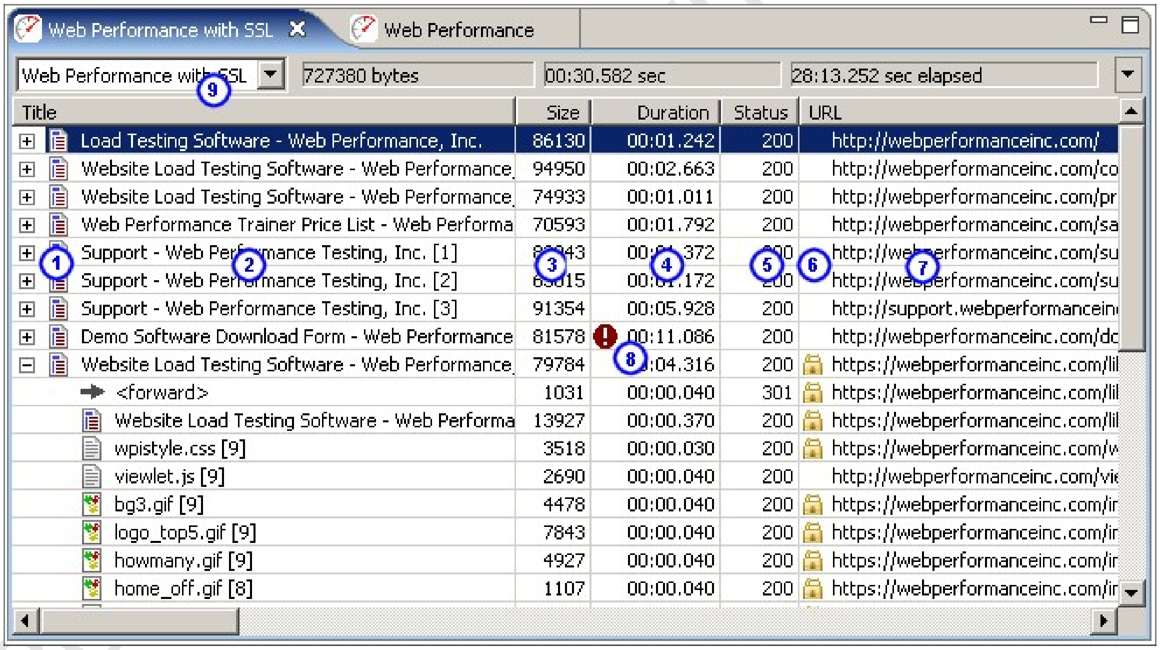
Notes:

- 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).



Web Performance with SSL | 727380 bytes | 00:30.582 sec | 28:13.252 sec elapsed

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]	[illegible]43	00:[illegible].372	[illegible]00	http://webperformanceinc.com/su
⊞ Support - Web Performance Testing, Inc. [2]	[illegible]015	00:[illegible].172	[illegible]00	http://webperformanceinc.com/su
⊞ Support - Web Performance Testing, Inc. [3]	91354	00:05.928	200	http://support.webperformancein
⊞ Demo Software Download Form - Web Performance	81578	! 00:11.086	200	http://webperformanceinc.com/dc
⊟ Website Load Testing Software - Web Performance	79784	[illegible]: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/vie
bg3.gif [9]	4478	00:00.040	200	🔒 https://webperformanceinc.com/ir
logo_top5.gif [9]	7843	00:00.040	200	🔒 https://webperformanceinc.com/ir
howmany.gif [9]	4927	00:00.040	200	🔒 https://webperformanceinc.com/ir
home_off.gif [8]	1107	00:00.040	200	🔒 https://webperformanceinc.com/ir

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 (E).

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.

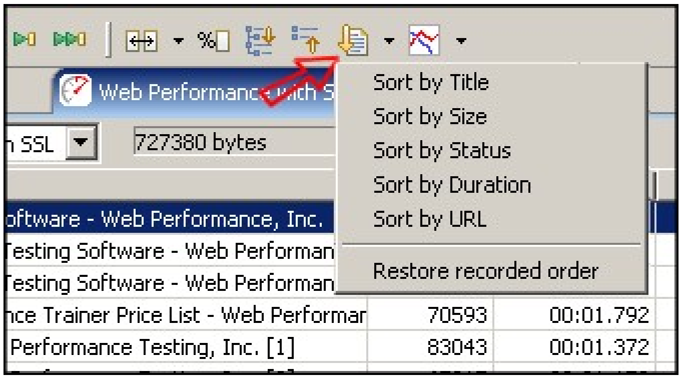
10. Editor menu: the pull down menu at the upper right corner of the Testcase Editor panel can be used to select comparisons, configure the testcase for replay, open charts, and modify the visible table columns.

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

Additionally, the think-time between web pages may be displayed. See the section on Changing the visible columns for details.

Sorting

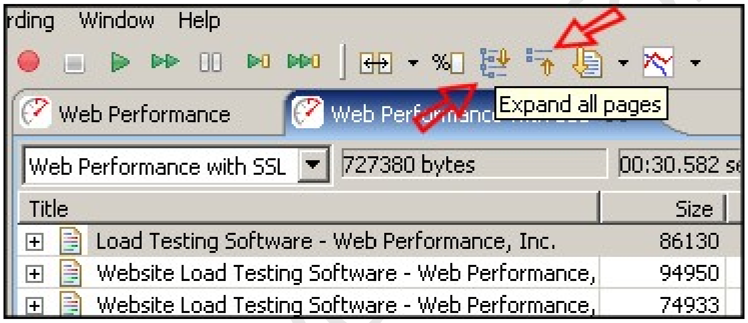
The Testcase Editor's table allows the user to sort the information in each column by either clicking on the column header or selecting the sort button on the toolbar and choosing which column to sort. Clicking the column header a second time or selecting the same column to sort reverses the sort direction. 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.


Web Performance with S
h SSL
727380 bytes
Sort by Title
Sort by Size
Sort by Status
Sort by Duration
Sort by URL
Restore recorded order
oftware - Web Performance, Inc.
Testing Software - Web Performan
Testing Software - Web Performan
nce Trainer Price List - Web Performar 70593 00:01.792
Performance Testing, Inc. [1] 83043 00:01.372

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.



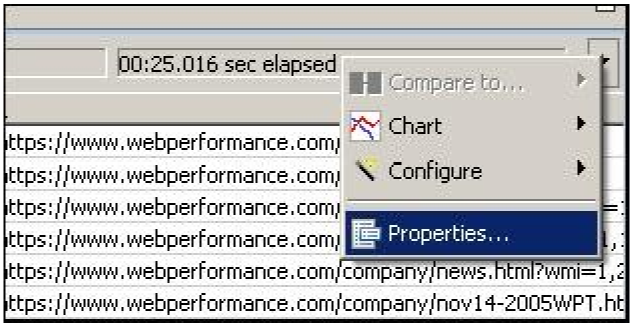
Web Performance with SSL | 727380 bytes | 00:30.582 se

Title	Size
Load Testing Software - Web Performance, Inc.	86130
Website Load Testing Software - Web Performance,	94950
Website Load Testing Software - Web Performance,	74933

Changing the visible columns

The Testcase Editor's table allows you to specify the columns you wish to view. To change the visible columns:

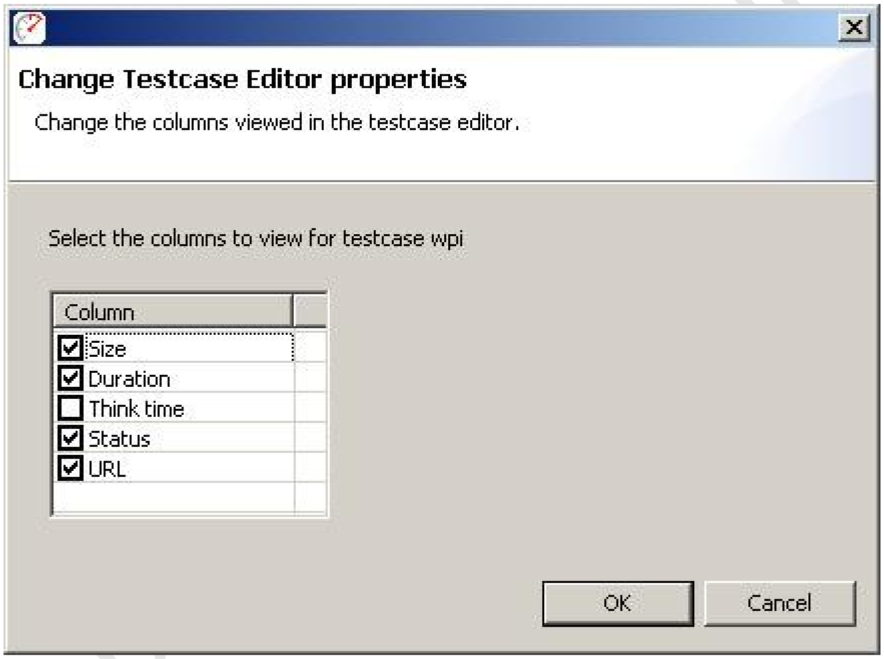
Open the Testcase Editor's menu and select the *Properties* item.



00:25.016 sec elapsed

ttps://www.webperformance.com/
ttps://www.webperformance.com/
ttps://www.webperformance.com/
ttps://www.webperformance.com/
ttps://www.webperformance.com/company/news.html?wmi=1,2
ttps://www.webperformance.com/company/nov14-2005WPT.ht

Check the columns to display and press the *OK* button.



Change Testcase Editor properties

Change the columns viewed in the testcase editor.

Select the columns to view for testcase wpi

Column
☑ Size
☑ Duration
☐ Think time
☑ Status
☑ URL

Editing Testcase Content

> Note that any changes to the testcase may cause the automatic configurations applied by the ASM wizard to be invalid, causing errors during replay. After any edits, run the ASM wizard to ensure a valid configuration. Also note that any changes to the testcase could make it impossible for the ASM wizard to correctly analyze the testcase, again resulting in replay errors.

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. To view the keyboard shortcuts available for actions within the Testcase Editor, press *Ctrl+Shift+L.*

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

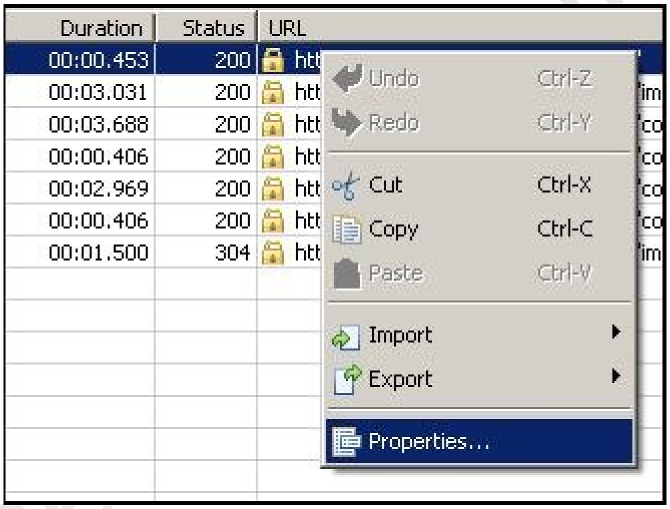
The cut, copy, and paste actions can be undone (up to a maximum of 10 actions per editor). The keyboard shortcut to undo the last action is *Ctrl-Z*.

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

To change the *think time* between web pages, right click on the web page and select the *Properties...* item.



Duration	Status	URL
00:00.453	200	htt
00:03.031	200	htt
00:03.688	200	htt
00:00.406	200	htt
00:02.969	200	htt
00:00.406	200	htt
00:01.500	304	htt

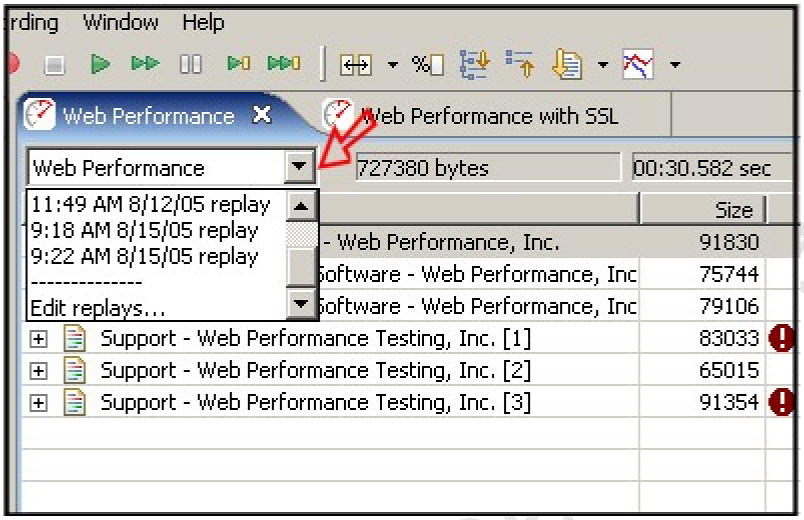
The think time is modified by entering the new value in the text field and pressing the *OK* button. The timing change can also be applied to all web pages within the testcase by checking the *Apply change to all web pages* item.

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.



727380 bytes

00:30.582 sec

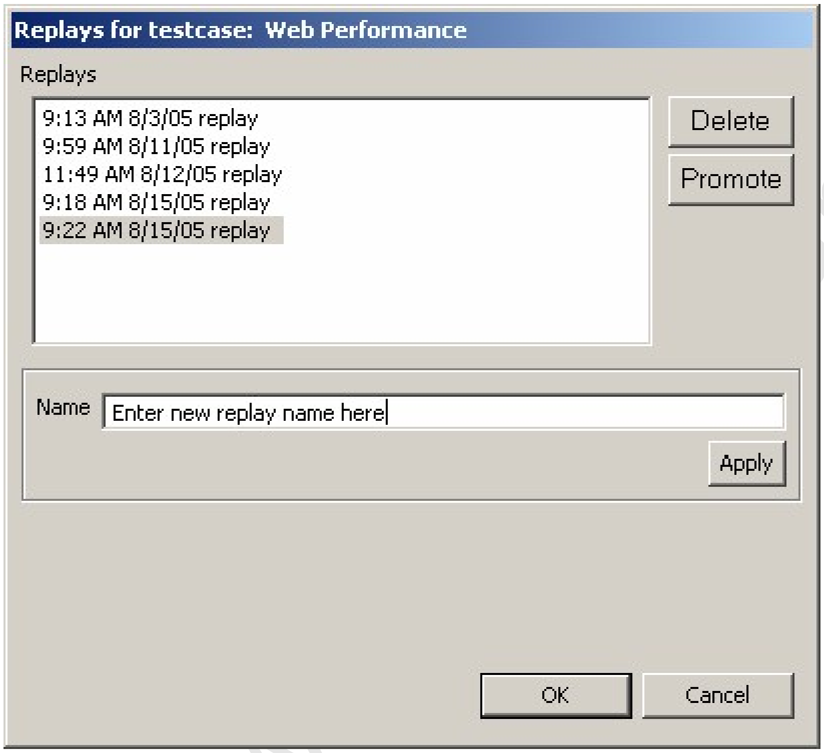
	Size
- Web Performance, Inc.	91830
oftware - Web Performance, Inc	75744
oftware - Web Performance, Inc	79106
Support - Web Performance Testing, Inc. [1]	83033
Support - Web Performance Testing, Inc. [2]	65015
Support - Web Performance Testing, Inc. [3]	91354

Renaming, Deleting and Promoting 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.


Replays for testcase: Web Performance

Replays

9:13 AM 8/3/05 replay
9:59 AM 8/11/05 replay
11:49 AM 8/12/05 replay
9:18 AM 8/15/05 replay
9:22 AM 8/15/05 replay

Name: Enter new replay name here

The *Promote* button will cause the selected replay to be *promoted* to the position of the original recording. This will result in several changes to the structure of the testcase:

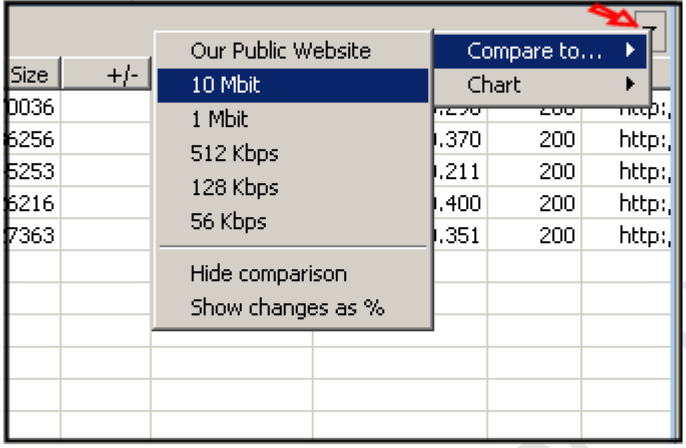
1. All user-defined actors from the original recording will be copied from the base testcase (original recording) to the replay.
2. All replays, except the promoted replay, will be deleted.
3. The original recording will be deleted.
4. The promoted replay will become the base testcase and will, for all intents and purposes, become the original recording.
5. Automatically-applied configurations will be cleared - replaying the testcase or using in a load test will require completion of the Replay Configuration wizards (authentication, ASM, etc).

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:

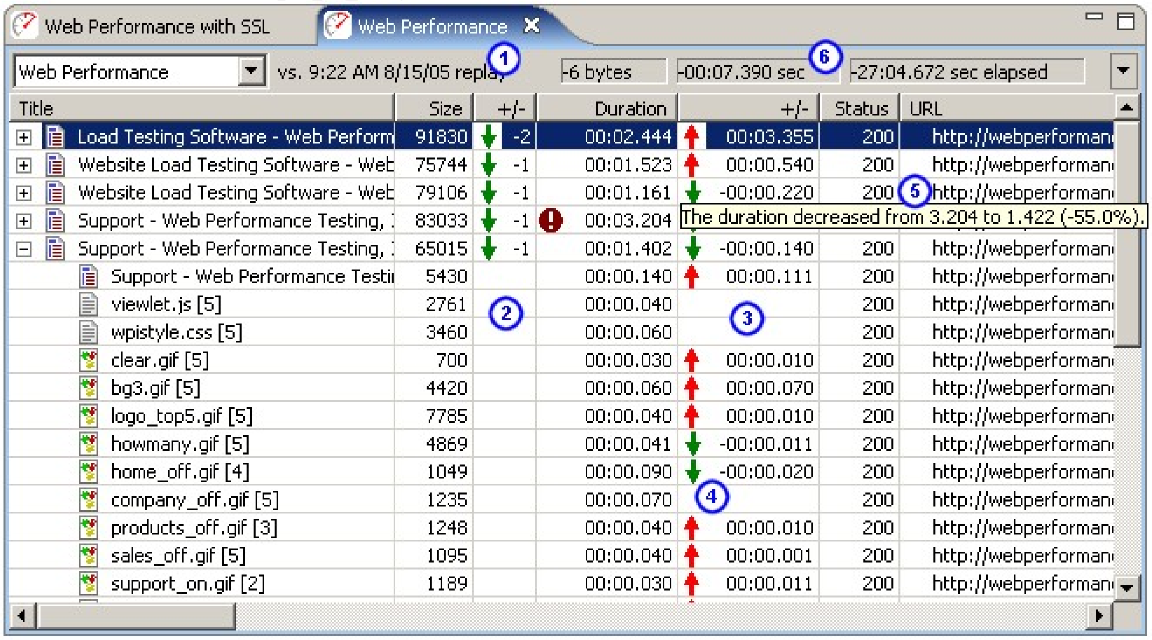
1. Press the pull down menu at the upper right corner of the Testcase Editor panel, select *Compare to...*, then select a replay (shown below)
2. Press the *Compare...* button ⇔on the main toolbar, then select a replay.
3. Select *Edit->Compare to...* on the main menu bar, then select a replay.

for example:



Examining the differences

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



Title	Size	+/-	Duration	+/-	Status	URL
⊞ Load Testing Software - Web Perform	91830	↓ -2	00:02.444	↑ 00:03.355	200	http://webperforman
⊞ Website Load Testing Software - Web	75744	↓ -1	00:01.523	↑ 00:00.540	200	http://webperforman
⊞ Website Load Testing Software - Web	79106	↓ -1	00:01.161	↓ -00:00.220	200	http://webperforman
⊞ Support - Web Performance Testing,	83033	↓ -1 ❗	00:03.204			
⊟ Support - Web Performance Testing,	65015	↓ -1	00:01.402	↓ -00:00.140	200	http://webperforman
Support - Web Performance Testi	5430		00:00.140	↑ 00:00.111	200	http://webperforman
viewlet.js [5]	2761		00:00.040		200	http://webperforman
wpistyle.css [5]	3460		00:00.060		200	http://webperforman
clear.gif [5]	700		00:00.030	↑ 00:00.010	200	http://webperforman
bg3.gif [5]	4420		00:00.060	↑ 00:00.070	200	http://webperforman
logo_top5.gif [5]	7785		00:00.040	↑ 00:00.010	200	http://webperforman
howmany.gif [5]	4869		00:00.041	↓ -00:00.011	200	http://webperforman
home_off.gif [4]	1049		00:00.090	↓ -00:00.020	200	http://webperforman
company_off.gif [5]	1235		00:00.070		200	http://webperforman
products_off.gif [3]	1248		00:00.040	↑ 00:00.010	200	http://webperforman
sales_off.gif [5]	1095		00:00.040	↑ 00:00.001	200	http://webperforman
support_on.gif [2]	1189		00:00.030	↑ 00:00.011	200	http://webperforman

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.

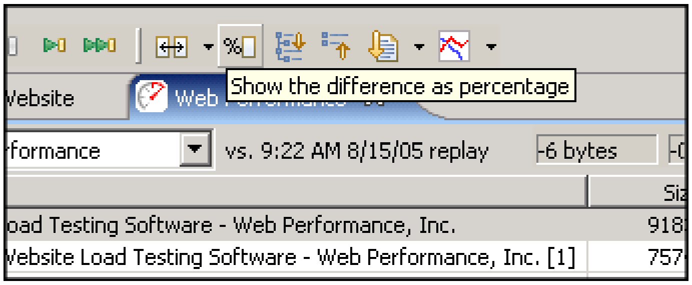
Viewing percentages

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:


Show the difference as percentage
Vebsite
Web
formance
vs. 9:22 AM 8/15/05 replay
-6 bytes
Siz
oad Testing Software - Web Performance, Inc.
918
Vebsite Load Testing Software - Web Performance, Inc. [1]
757

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.

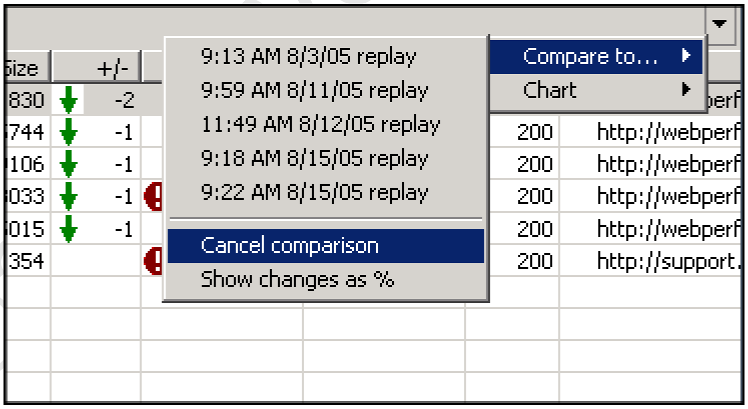
Note: Testcases that have been opened before will remember their difference setting and will override the default setting.

Cancel the comparison

To cancel the comparison, use one of the following:

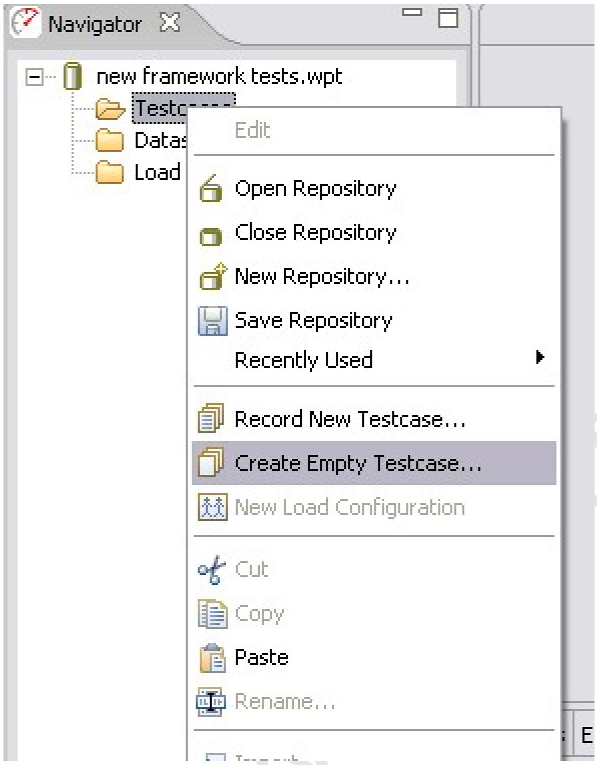
- Select *Compare to...* item from the Testcase Editor menu, then select *Cancel comparison*.
- Press the revert comparison button on the main toolbar.
- Select *Edit->Compare to...->Cancel comparison* on the main menu.

for example:

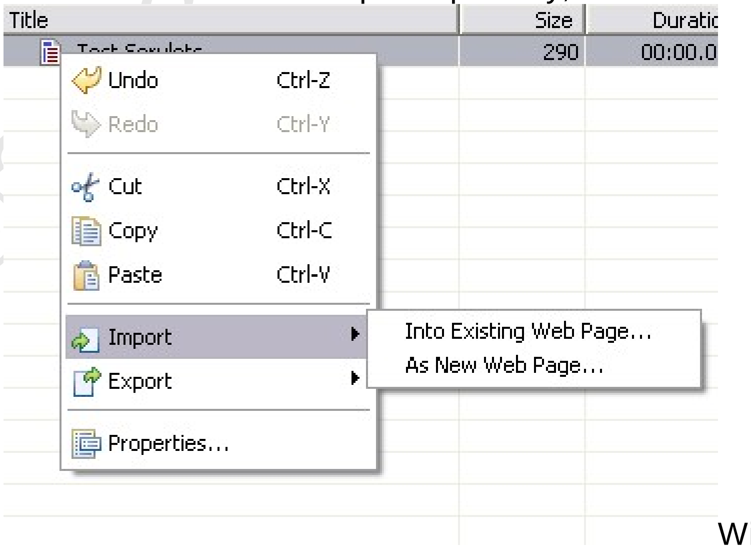


Importing Transactions

Occassionally, there may be a scenario where the normal process of recording a transaction is not appropriate. This sort of case could arise when attempting to test a Web Service, with a client does not support manual configuration of it's proxy servers. In this event, individual transactions may be created and imported into Web Performance Suite in order to create the desired testcase.
In order to import a transaction, a valid HTTP request and response pair will need to be created. This may be accomplished either through a network diagnostic utility, or another application capable of creating the exact transaction content given the testing environment. The file format should be a plain text file, containing the raw HTTP message, without any further information.



If the imported transactions are intended to be imported into a new testcase, then you may right-click on the "Testcases" node of an open repository, and select "Create Empty Testcase…".

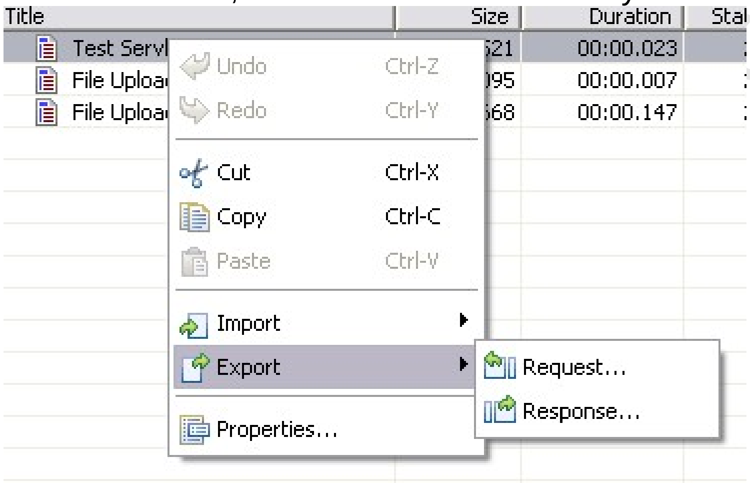


With a Testcase Editor open, you may right-click on any Web Page within the editor and select Import » Into Existing Web Page…, or right-click anywhere in the editor and select Import » As New Web Page… to create a new Web Page for your transaction.

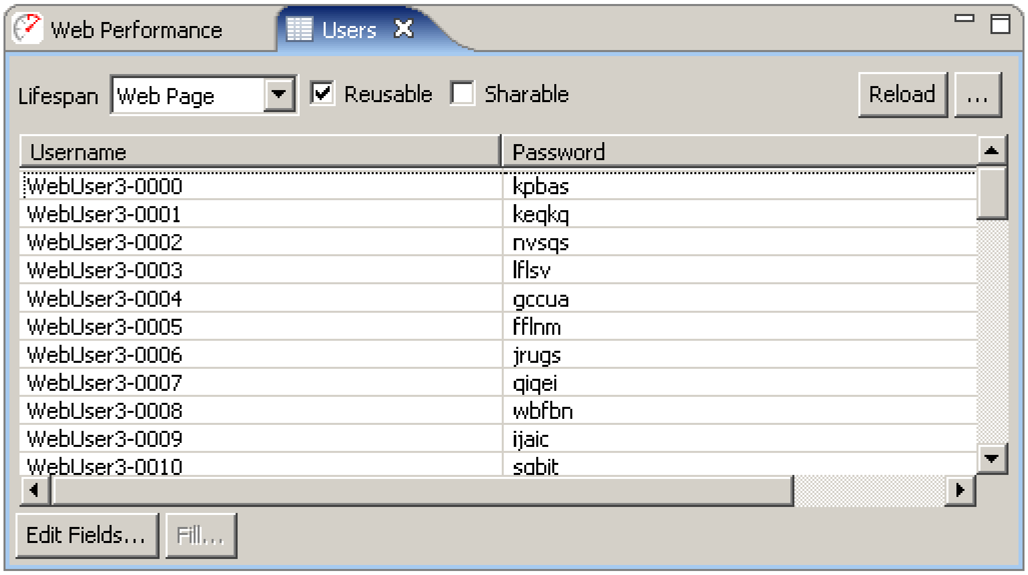
With the Import Transaction dialog open, simply select the locations of the appropriate request and response files, and then press "OK". The transactions will then imported into Web Performance Suite, for use during Replays and Load Testing.

Exporting Messages

Web Performance Suite is fully capable of also exporting raw transaction data from a Testcase Recording or Replay. To do this, simply open the recording or replay in a Testcase Editor, and select the transaction you would like to export data from. Right click on the transaction, and select "Export". You may export either the request or the response to a single file. From there, simply select the location and name for the file, and press "OK". A new file will be created with the complete message saved.



Dataset Editor



Lifespan: Web Page ☑ Reusable ☐ Sharable

Username	Password
WebUser3-0000	kpbas
WebUser3-0001	keqkq
WebUser3-0002	nvsqs
WebUser3-0003	lflsv
WebUser3-0004	gccua
WebUser3-0005	fflnm
WebUser3-0006	jrugs
WebUser3-0007	qiqei
WebUser3-0008	wbfbn
WebUser3-0009	ijaic
WebUser3-0010	sgbit

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.

Some examples of the correct settings are described below. *Virtual user*, *testcase* and *web page* are the most commonly used settings. The lifespan settings that are not mentioned are rarely used. If you are not sure why you would need it, then you probably don't.

Examples

User Identity

For a dataset containing usernames and passwords, the lifespan should usually be configured as *testcase.* If each virtual user should repeat the testcase using the same identity each time, the lifespan should be set to *virtual user*. Any other lifespan setting will probably be inappropriate for user identity datasets.

Form fields

For a dataset containing values that a user will enter into fields on a web page form, the lifespan should could be set to either *testcase* or *web page*. If the dataset contains values for *more than one* page, it should be set to *testcase*.

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.
Note that if a dataset is not reusable, the load test will encounter errors and terminate when all the rows have been used.

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. Note that if a dataset is not sharable, the dataset must have more rows than the number of Virtual Users in the load test.

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.

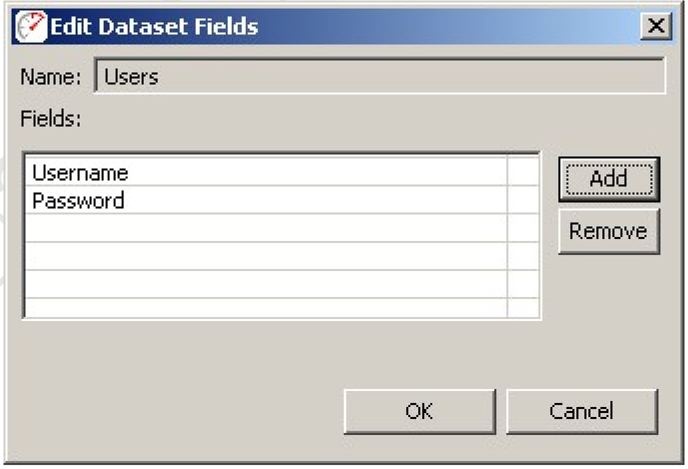
To add new rows, edit the last cell and press <RETURN>. A new row will be created with sample data - continue filling in new rows by typing new entries and pressing the <RETURN> key. Press the <TAB> key to finish editing the last new entry.

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.


Edit Dataset Fields

Name: Users

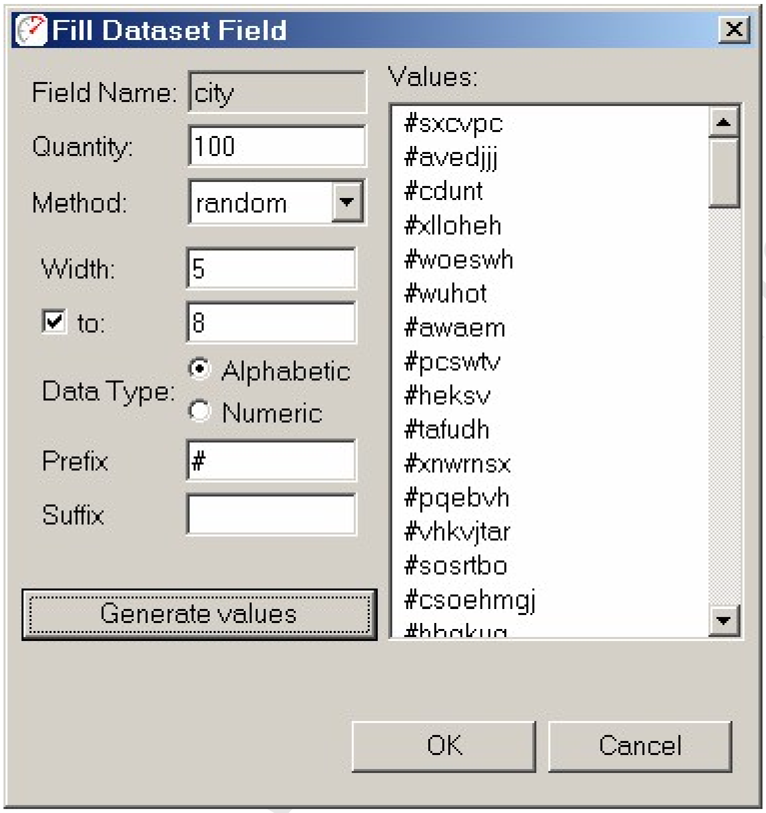
Fields:

Username
Password

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 generated data

The *Fill...* button allows the selected field to be filled with generated values. Select the field to be filled by pressing the field name (column heading) in the table. Then press the *Fill...* button to open the *Fill Dataset Field* dialog:



Field Name: city

Quantity: 100

Method: random

Width: 5

☑ to: 8

Data Type: ● Alphabetic ○ Numeric

Prefix #

Suffix

Values:

```
#sxcvpc
#avedjjj
#cdunt
#xlloheh
#woeswh
#wuhot
#awaem
#pcswtv
#heksv
#tafudh
#xnwrnsx
#pqebvh
#vhkvjtar
#sosrtbo
#csoehmgj
#hhgkug
```

The *Method* field allows selection from three generation types:

1. Random - generate random alphabetic or numeric strings
2. Sequence - generate sequences of numeric strings
3. List - select strings from pre-populated lists

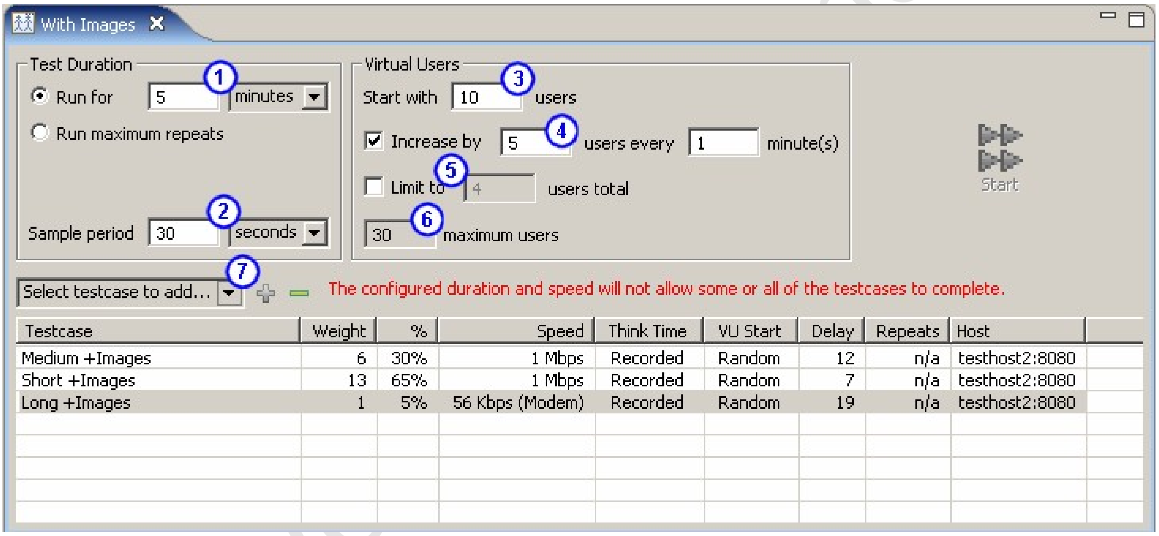
Note that the *Quantity* field will be automatically set to the total number of rows in the dataset and cannot be changed. The *Width* field defines how long each generated value will be. The *Data Type* chooses between alpha and numeric data. A preview of the data is available in the list at the right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

Load 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
● Run for 5 minutes
○ Run maximum repeats
Sample period 30 seconds

Virtual Users
Start with 10 users
☑ Increase by 5 users every 1 minute(s)
☐ Limit to 4 users total
30 maximum users

Select testcase to add...

The configured duration and speed will not allow some or all of the testcases to complete.

Testcase	Weight	%	Speed	Think Time	VU Start	Delay	Repeats	Host
Medium +Images	6	30%	1 Mbps	Recorded	Random	12	n/a	testhost2:8080
Short +Images	13	65%	1 Mbps	Recorded	Random	7	n/a	testhost2:8080
Long +Images	1	5%	56 Kbps (Modem)	Recorded	Random	19	n/a	testhost2:8080

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.

Alternatively, the "Run maximum repeats" option will allow the test to run for as long as necessary for each testcase to be repeated as many times as specified in the testcase's "Repeats" column. This allows for each testcase to run a predetermined number of times, stopping the load test upon completion.

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 ⑥.

Testcases

Testcases are added to the load test using the pull-down menu ⑦ 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.

- Host: All of the URLs contained in the testcase can be modified 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.

- Repeats: When the "Test Duration" section is set to "Run maximum repeats", this column specifies the number of times a testcase should be attempted.

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.

Running the Load Test

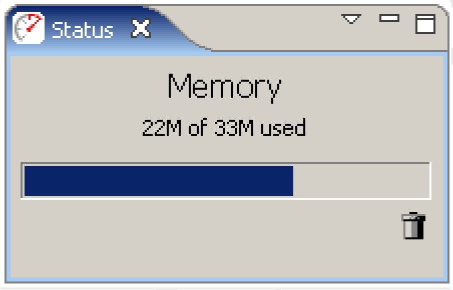
The load test cannot be run until the application detects there are no invalid entries or configuration errors. Once the load test configuration is valid, the *Run* button on the Load Test Configuration Editor is enabled. Selecting this button begins running the load test and opens the Load Test Results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

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.

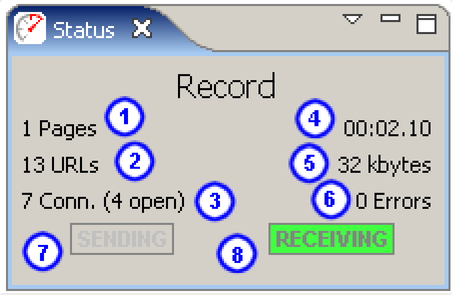


The 🗑 button in the corner runs the garbage collector to recycle memory that is no longer being used. Note that you are never required to push this button manually - the garbage collector automatically runs when needed. But some people really like pushing buttons, so we added one!

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



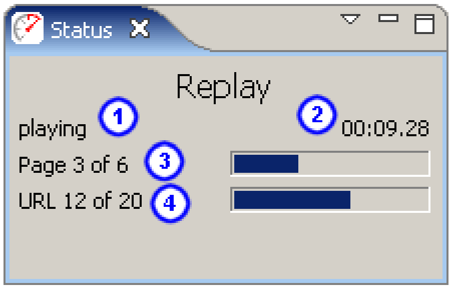
Shortly after a replay ends, the *Status View* will automatically return to displaying the memory status.

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

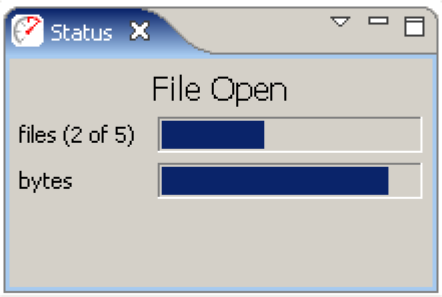
Shortly after a replay ends, the *Status View* will automatically return to displaying the memory status.



File opening status

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

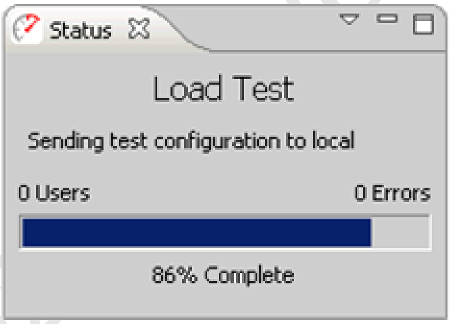
- number of files completed
- bytes read from the current file



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

Load test status

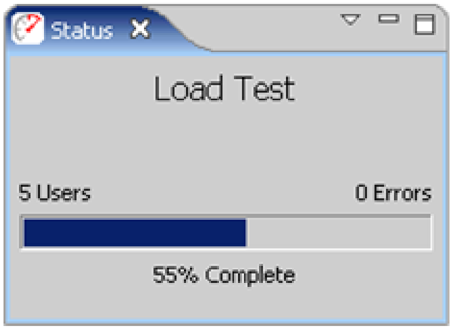
During a load test, the status view will show the progress of the current test stage. The following pictures show examples of the starting, testing and stopping stages.



Sending test configuration to local

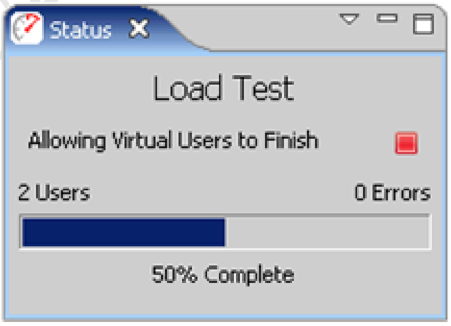
0 Users 0 Errors

86% Complete



5 Users 0 Errors

55% Complete



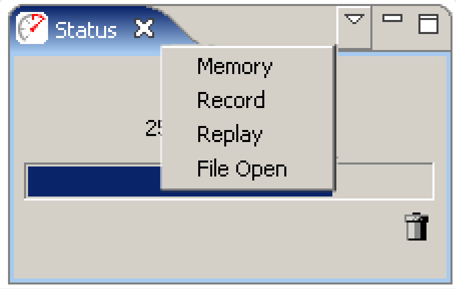
Allowing Virtual Users to Finish

2 Users 0 Errors

50% Complete

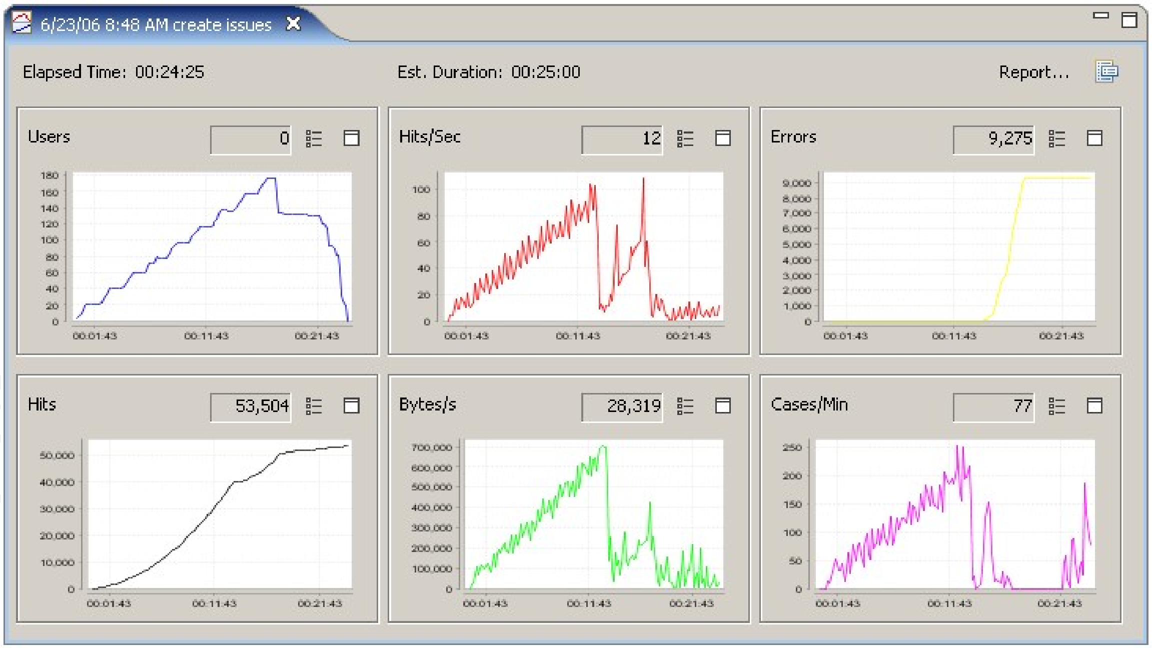
Changing the display

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..



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: 00:24:25

Est. Duration: 00:25:00

Users 0

Hits/Sec 12

Errors 9,275

Hits 53,504

Bytes/s 28,319

Cases/Min 77

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 statistic 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.

Display Properties

Opens the properties editor for configuring the display preferences. Use this to change the number of charts displayed in the display.

Chart Properties

Opens the properties editor for configuring the chart preferences. Use this to change the data displayed in the chart.

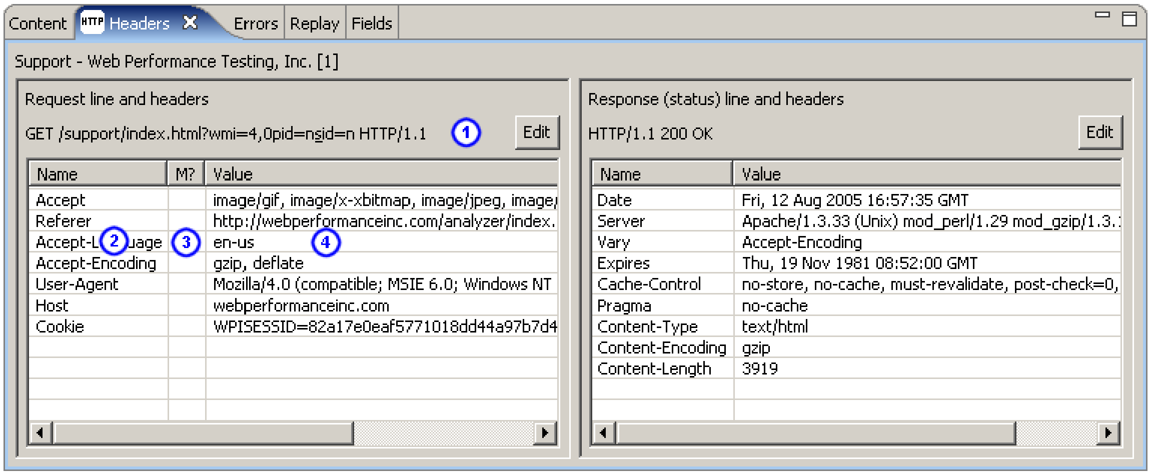
Chart Maximize

Maximize the selected chart.

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.



Support - Web Performance Testing, Inc. [1]

Request line and headers

GET /support/index.html?wmi=4,0pid=nsid=n HTTP/1.1

Name	M?	Value
Accept		image/gif, image/x-xbitmap, image/jpeg, image/
Referer		http://webperformanceinc.com/analyzer/index.
Accept-Language		en-us
Accept-Encoding		gzip, deflate
User-Agent		Mozilla/4.0 (compatible; MSIE 6.0; Windows NT
Host		webperformanceinc.com
Cookie		WPISESSID=82a17e0eaf5771018dd44a97b7d4

Response (status) line and headers

HTTP/1.1 200 OK

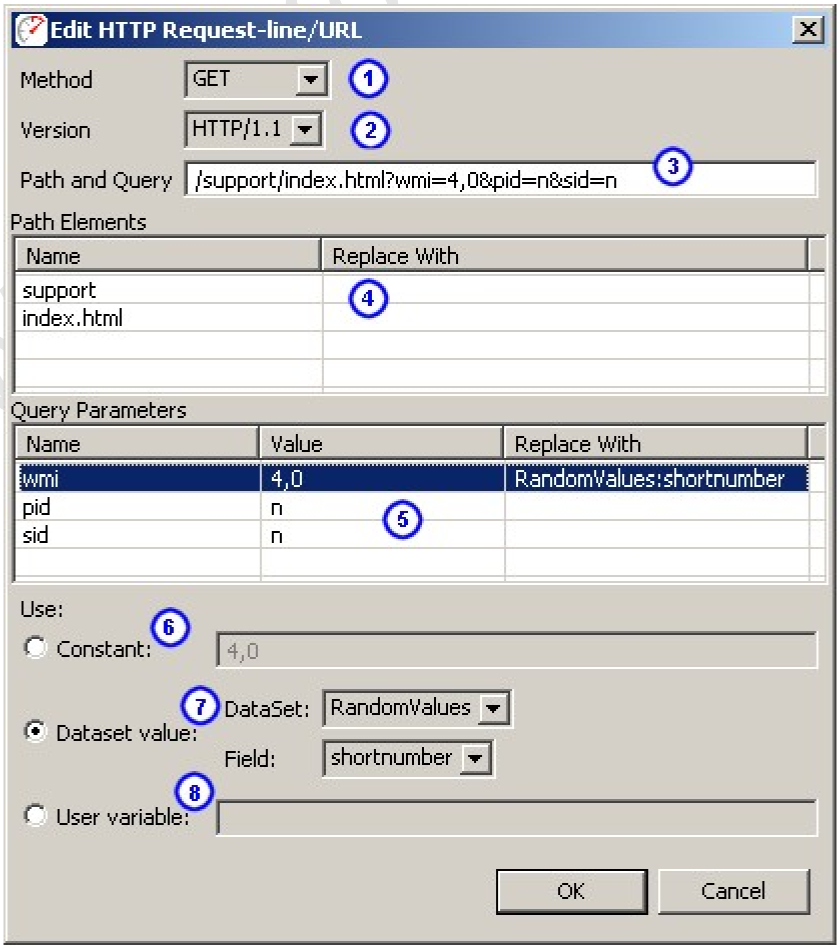
Name	Value
Date	Fri, 12 Aug 2005 16:57:35 GMT
Server	Apache/1.3.33 (Unix) mod_perl/1.29 mod_gzip/1.3.
Vary	Accept-Encoding
Expires	Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control	no-store, no-cache, must-revalidate, post-check=0,
Pragma	no-cache
Content-Type	text/html
Content-Encoding	gzip
Content-Length	3919

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.



Edit HTTP Request-line/URL

Method: GET

Version: HTTP/1.1

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

Path Elements

Name	Replace With
support	
index.html	

Query Parameters

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

Use:

○ Constant: 4,0

● Dataset value: DataSet: RandomValues
Field: shortnumber

○ User variable:

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 immediat

ely.

4. URL path elements - Each path element can be changed or configured with a 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 reply.
8. User variable - Similar to the Dataset value, above. The named value is extracted from the Virtual User's local variable store.

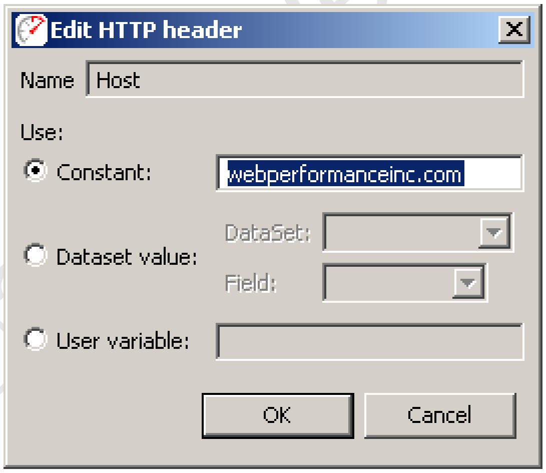
This
feature is
still under
developm
ent.

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.



Edit HTTP header

Name: Host

Use:

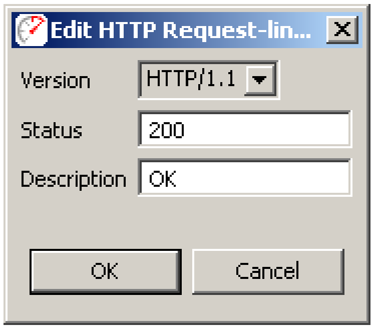
● Constant: webperformanceinc.com

○ Dataset value: DataSet: Field:

○ User variable:

Editing Status-line

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



Edit HTTP Request-lin...

Version: HTTP/1.1

Status: 200

Description: OK

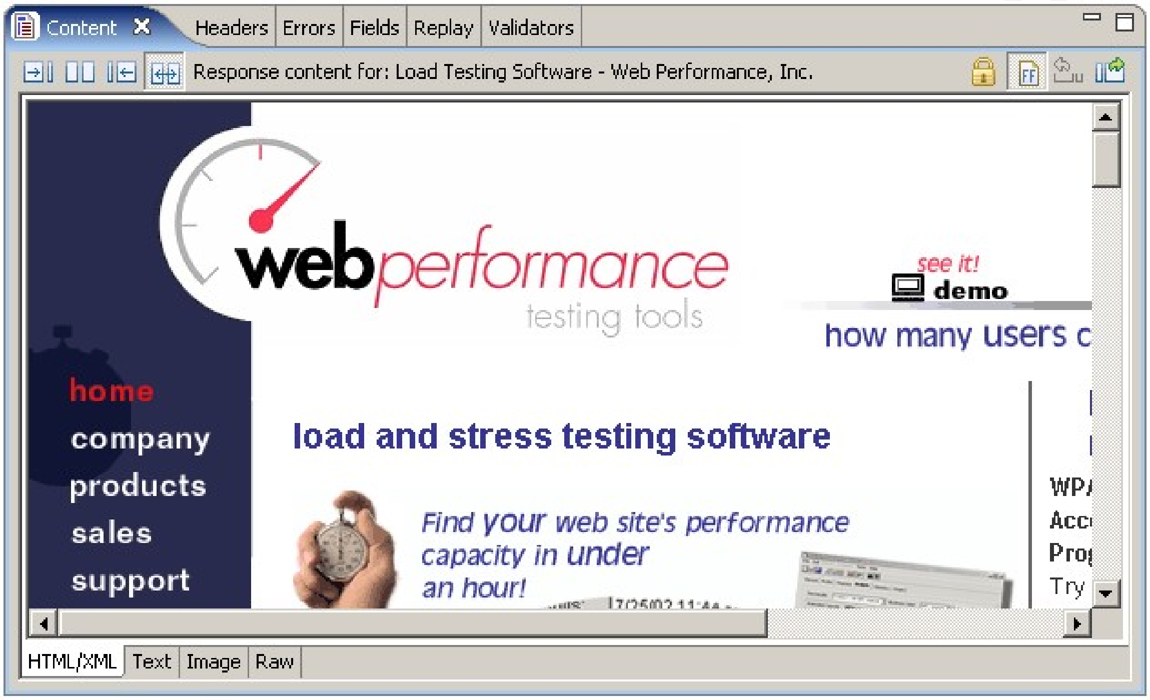
Content View

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

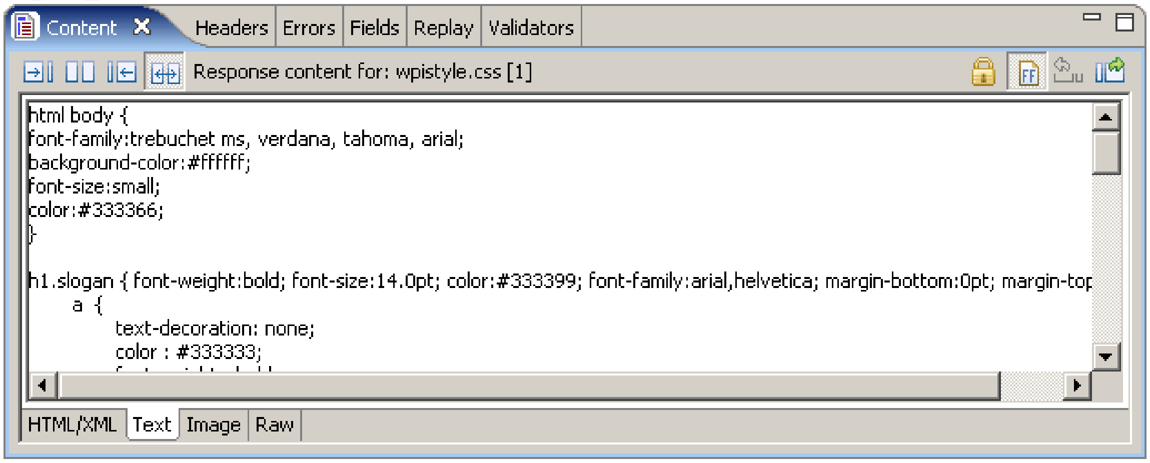
The HTML/XML viewers display the content in an embedded browser that renders the selected page from memory, including all images, style sheets, etc from the recording.



Response content for: Load Testing Software - Web Performance, Inc.

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.

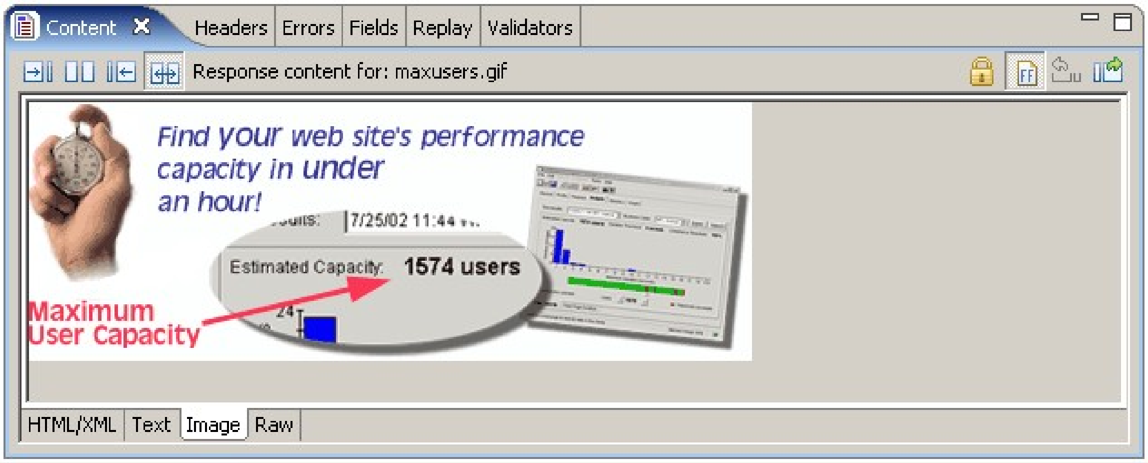


Response content for: wpistyle.css [1]

```
html body {
font-family:trebuchet ms, verdana, tahoma, arial;
background-color:#ffffff;
font-size:small;
color:#333366;
}

h1.slogan { font-weight:bold; font-size:14.0pt; color:#333399; font-family:arial,helvetica; margin-bottom:0pt; margin-top
    a {
        text-decoration: none;
        color : #333333;
```

Image viewer

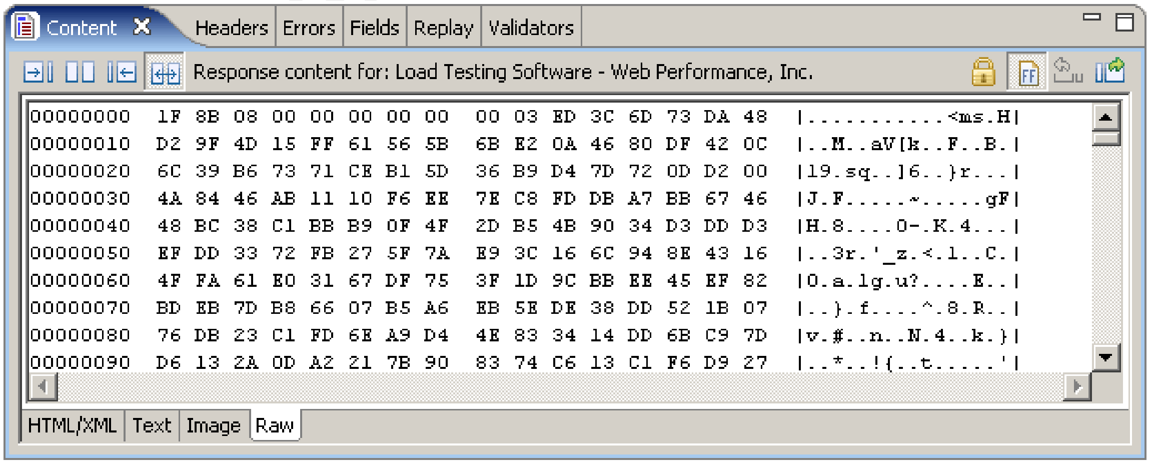
The image viewer displays common image formats, including PNG, GIF, JPEG, and BMP (on Windows).



Response content for: maxusers.gif

Raw viewer

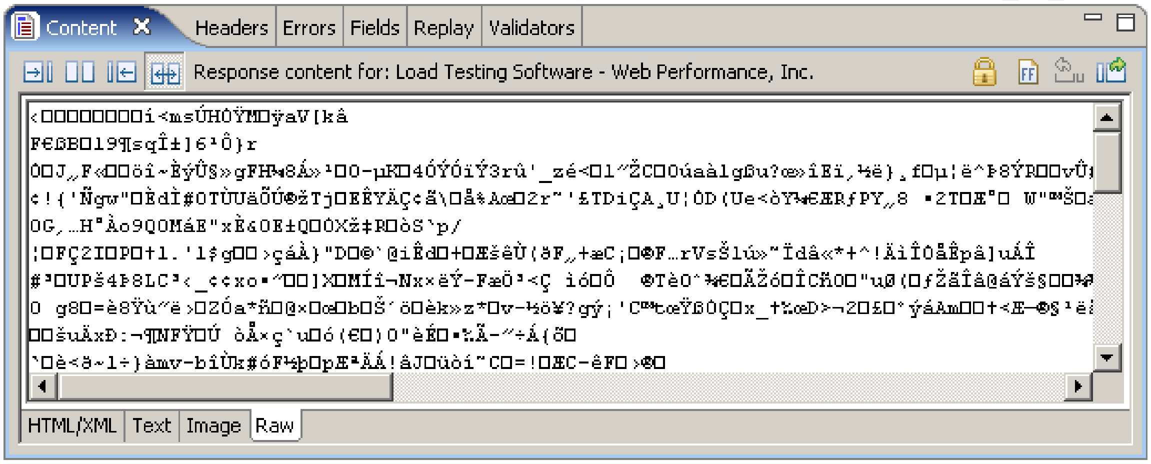
The *Raw* tab displays the content exactly as it was received from the server. By default, the raw content is displayed in hex dump format:



Response content for: Load Testing Software - Web Performance, Inc.

```
00000000  1F 8B 08 00 00 00 00 00  00 03 ED 3C 6D 73 DA 48  |...........<ms.H|
00000010  D2 9F 4D 15 FF 61 56 5B  6B E2 0A 46 80 DF 42 0C  |..M..aV[k..F..B.|
00000020  6C 39 B6 73 71 CE B1 5D  36 B9 D4 7D 72 0D D2 00  |l9.sq..]6..}r...|
00000030  4A 84 46 AB 11 10 F6 EE  7E C8 FD DB A7 BB 67 46  |J.F.....~.....gF|
00000040  48 BC 38 C1 BB B9 0F 4F  2D B5 4B 90 34 D3 DD D3  |H.8....O-.K.4...|
00000050  EF DD 33 72 FB 27 5F 7A  E9 3C 16 6C 94 8E 43 16  |..3r.'_z.<.l..C.|
00000060  4F FA 61 E0 31 67 DF 75  3F 1D 9C BB EE 45 EF 82  |O.a.1g.u?....E..|
00000070  BD EB 7D B8 66 07 B5 A6  EB 5E DE 38 DD 52 1B 07  |..}.f....^.8.R..|
00000080  76 DB 23 C1 FD 6E A9 D4  4E 83 34 14 DD 6B C9 7D  |v.#..n..N.4..k.}|
00000090  D6 13 2A 0D A2 21 7B 90  83 74 C6 13 C1 F6 D9 27  |..*..!{..t.....'|
```

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


Content
Headers
Errors
Fields
Replay
Validators
Response content for: Load Testing Software - Web Performance, Inc.
HTML/XML
Text
Image
Raw

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:

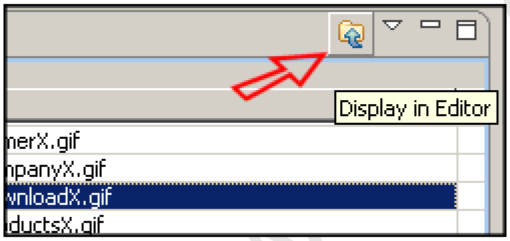
- a Testcase Editor tab, when the replay is displayed
- Web Page or URL in the Testcase Editor when the replay is displayed

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

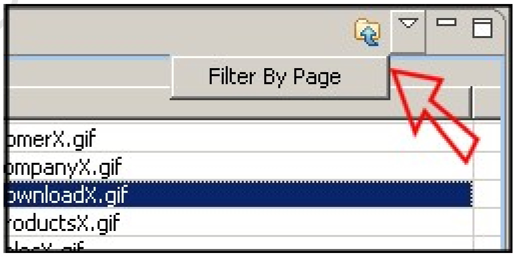
Go to the URL for the error

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).

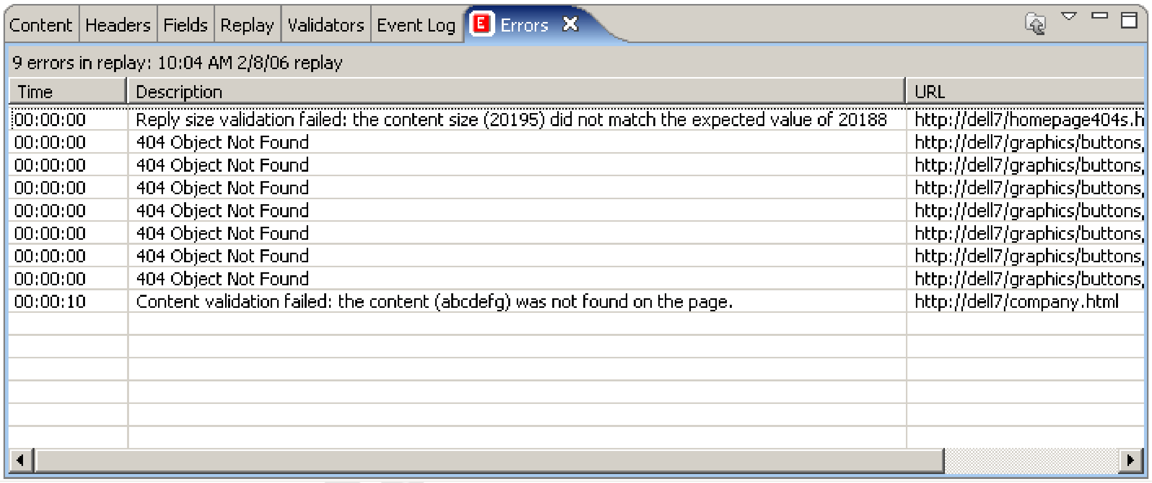


Viewing the Transaction Title and URL

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).

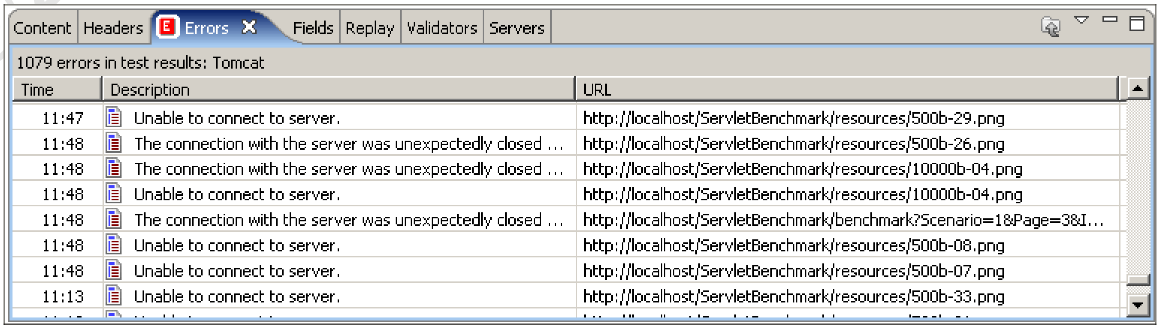


9 errors in replay: 10:04 AM 2/8/06 replay

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.



1079 errors in test results: Tomcat

Time	Description	URL
11:47	Unable to connect to server.	http://localhost/ServletBenchmark/resources/500b-29.png
11:48	The connection with the server was unexpectedly closed ...	http://localhost/ServletBenchmark/resources/500b-26.png
11:48	The connection with the server was unexpectedly closed ...	http://localhost/ServletBenchmark/resources/10000b-04.png
11:48	Unable to connect to server.	http://localhost/ServletBenchmark/resources/10000b-04.png
11:48	The connection with the server was unexpectedly closed ...	http://localhost/ServletBenchmark/benchmark?Scenario=1&Page=3&I...
11:48	Unable to connect to server.	http://localhost/ServletBenchmark/resources/500b-08.png
11:48	Unable to connect to server.	http://localhost/ServletBenchmark/resources/500b-07.png
11:13	Unable to connect to server.	http://localhost/ServletBenchmark/resources/500b-33.png

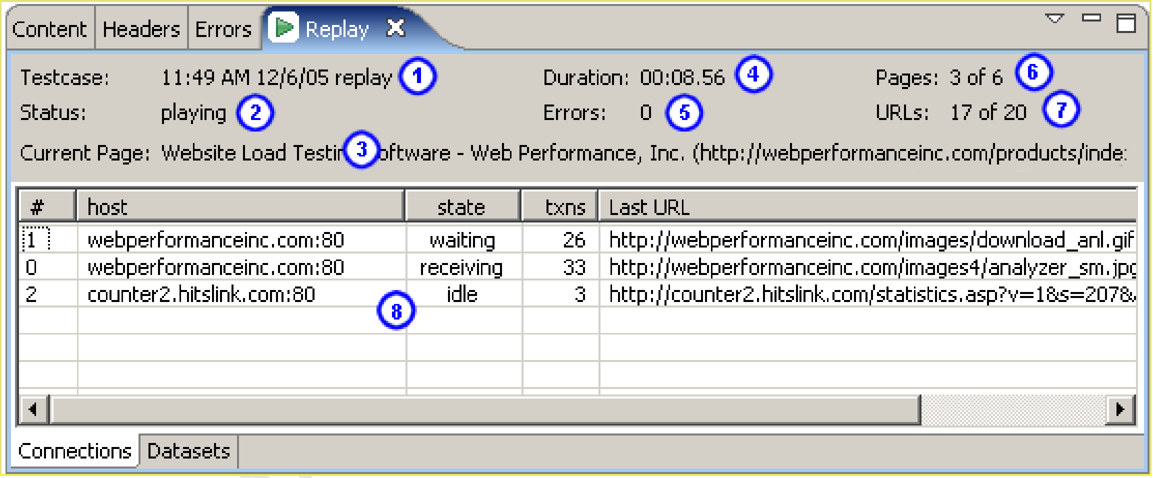
Replay View

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

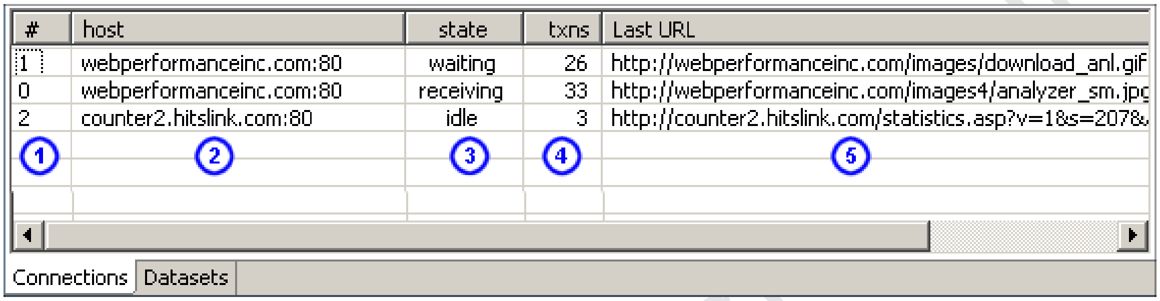


Testcase: 11:49 AM 12/6/05 replay
Status: playing
Duration: 00:08.56
Errors: 0
Pages: 3 of 6
URLs: 17 of 20
Current Page: Website Load Testi oftware - Web Performance, Inc. (http://webperformanceinc.com/products/inde

#	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. 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. additional info, depending on the selected tab

Connections tab



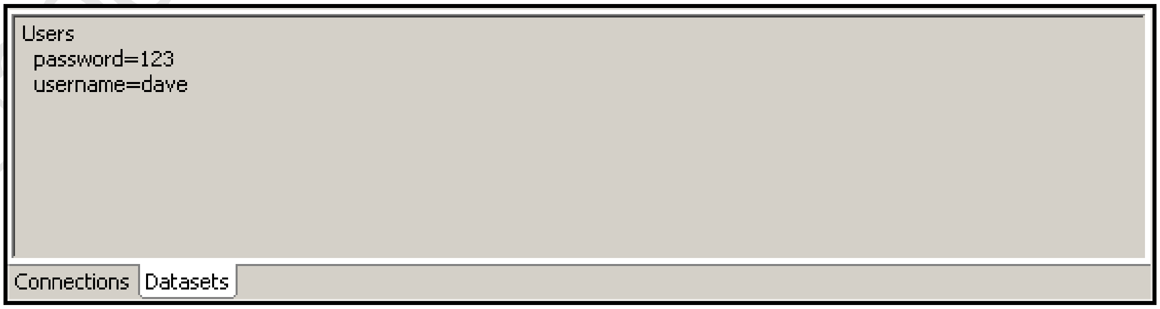
#	host	state	txns	Last URL
1	webperformanceinc.com:80	waiting	26	http://webperformanceinc.com/images/download_anl.gif
0	webperformanceinc.com:80	receiving	33	http://webperformanceinc.com/images4/analyzer_sm.jpg
2	counter2.hitslink.com:80	idle	3	http://counter2.hitslink.com/statistics.asp?v=1&s=207&

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.

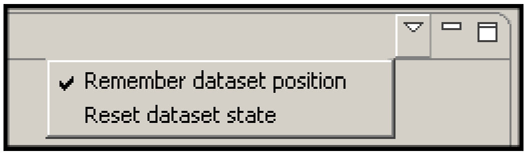


```
Users
  password=123
  username=dave
```

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

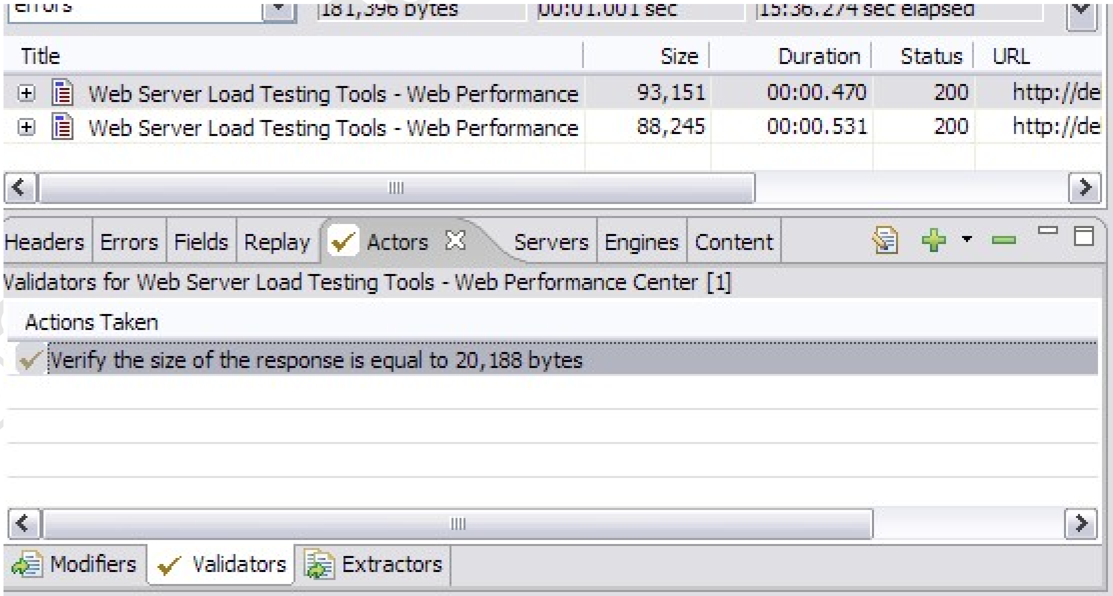
When a VU replays a testcase and it has modifiers configured to pull values from a dataset, the position of the row in the dataset is automatically advanced when the row is returned. This allows a testcase to be replayed multiple times with different data. This setting is on by default. Turning it off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

Actors View

The Actors View displays the list of actors that both control and respond to the content of your testcase during a replay or a load test. The Actors View is opened by selecting Window → Show View → Actors from the main menu.



Title	Size	Duration	Status	URL
Web Server Load Testing Tools - Web Performance	93,151	00:00.470	200	http://de
Web Server Load Testing Tools - Web Performance	88,245	00:00.531	200	http://de

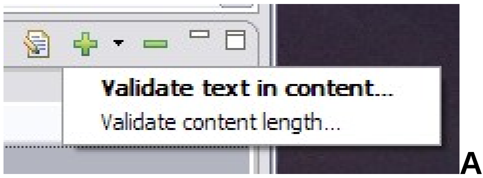
Validators for Web Server Load Testing Tools - Web Performance Center [1]

Actions Taken
✓ Verify the size of the response is equal to 20,188 bytes

At the bottom of the Actors View, there are three tabs that filter out what type of actor the view is currently working with:

- Modifiers
- Validators
- Extractors

The top right of the Actors view additionally displays three buttons that may be used to control various Actors:

1. **Edit**: After selecting an actor, this button will bring up a dialog where the behavior of that actor may be edited. This function is usually only available for actors created within the Actors View via the *Add* button.
2. 

Add: This button allows you to create a new actor. The type of actor created is determined by which tab of the actors view is currently selected. A drop-down arrow next to the button may be pressed to display a list of supported sub-types of actors that may be created.
3. **Remove**: Removes the selected actor(s).

Modifiers

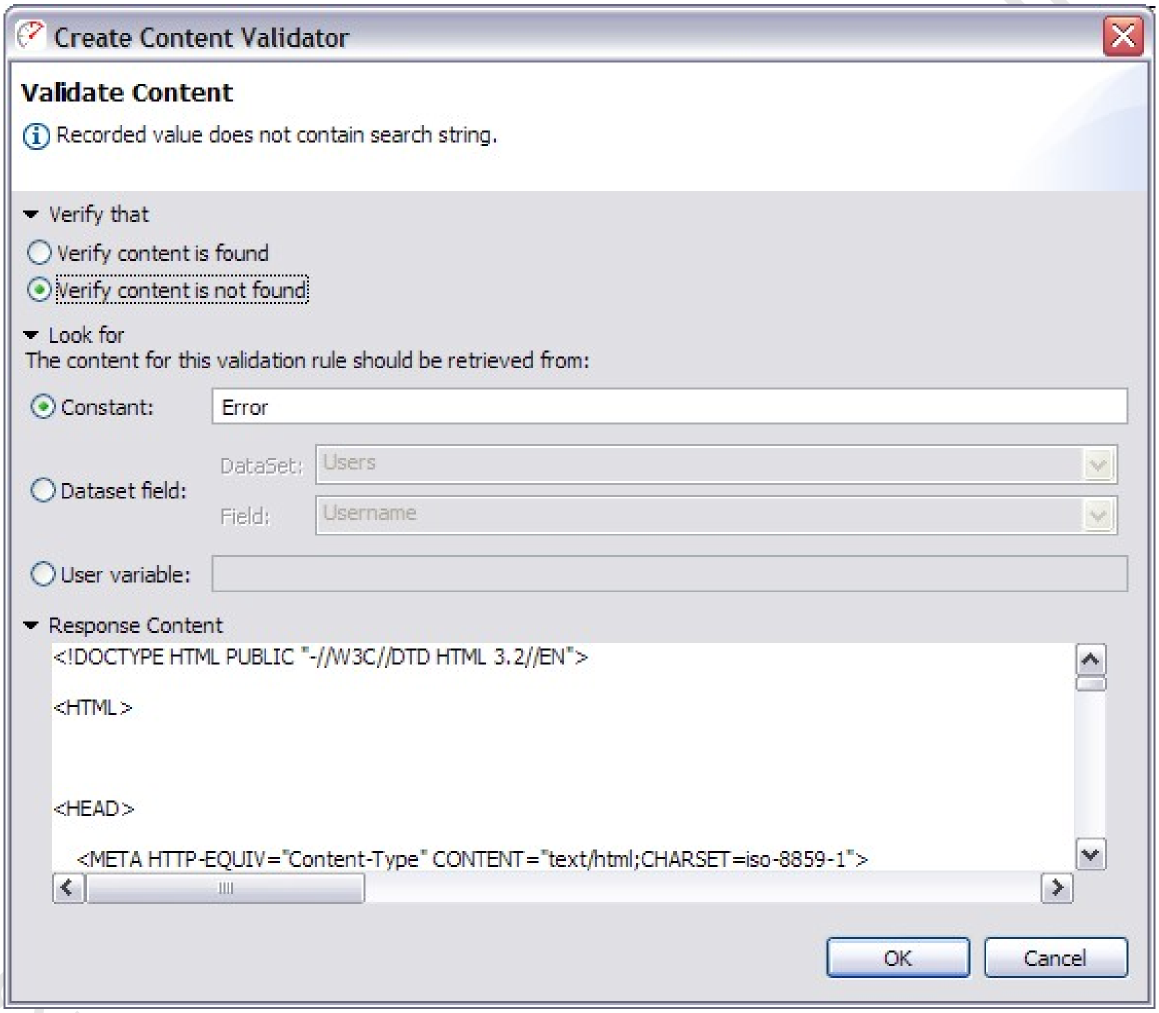
Modifiers may change values submitted to the server from those that were submitted when the testcase was initially recorded.
The Actors View provides a centralized display for reviewing (and if necessary, removing) modifiers that are currently in place in your testcase. To create or revise modifiers, you should use a component of Web Performance Suite™ relevant to the type of value being modified. For more information, please consult the Customizing a Testcase section.

Validators

Validators are used during a replay or load test to examine a received response, and determine if that response was valid or not.
In addition to reviewing validators created automatically by various components of Web Performance Suite™, validators may be created by pressing the *Add* Button of the Actors View. Presently, there are two criteria that a new validator may use (listed from the drop-down menu, located next to the *Add* button): Content, and Size.

Content validation

Using content validation, a search string may be entered. When the response is processed, the page will be deemed valid based on whether or not the response content contained the search string. This style of validation is selected by default when pressing the *Add* button, but may also be accessed through the drop-down menu by selecting "Validate text in content...".



Create Content Validator

Validate Content

ⓘ Recorded value does not contain search string.

▾ Verify that

○ Verify content is found

● Verify content is not found

▾ Look for

The content for this validation rule should be retrieved from:

● Constant: Error

○ Dataset field: DataSet: Users Field: Username

○ User variable:

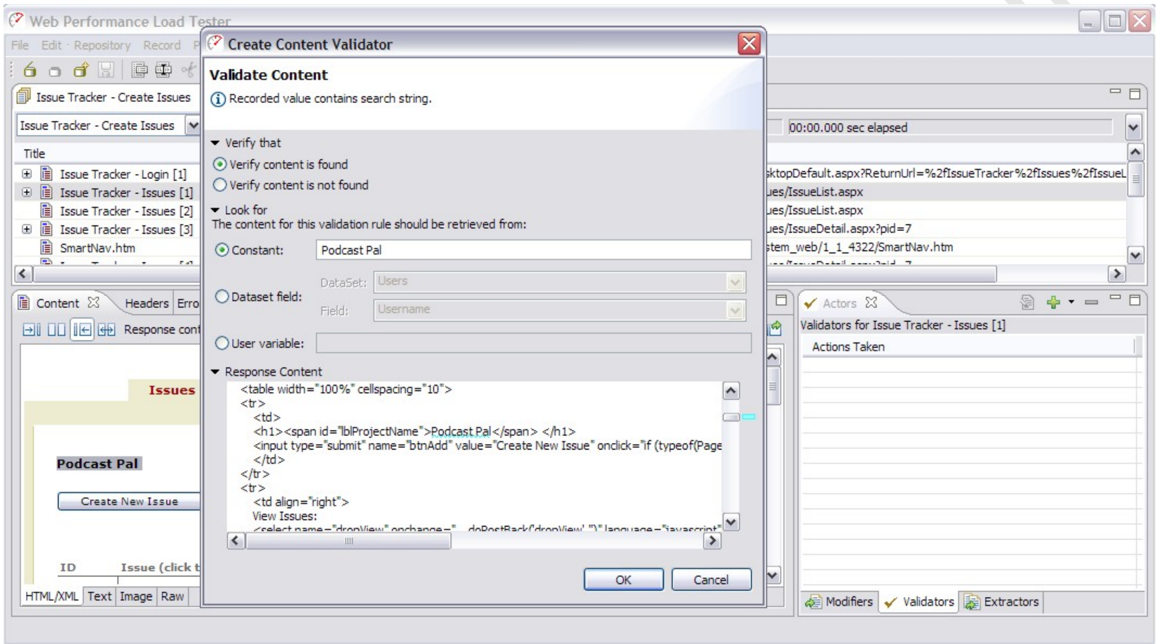
▾ Response Content

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">

<HTML>



<HEAD>

  <META HTTP-EQUIV="Content-Type" CONTENT="text/html;CHARSET=iso-8859-1">
```

The first option on this screen is concerned with whether or not to flag an error when the search string is located. The option "Verify content is not found" is appropriate when entering an error message. When entering a string of text that is unique to this particular page, the option "Verify content is found" is appropriate.

The next section determines what this validator will search the response for. Here, a search string may be entered into the "Constant" field. If it is more appropriate to vary the string being searched for, then it is possible to select the appropriate radio button to obtain this value from the current row of a dataset, or from a user variable.

Finally, the "Response Content" displays the content of the current response for reference. When a "Constant" search string has been entered and located, it is possible to scroll to that point by clicking on the highlighted block that appears next to the content display.


Web Performance Load Tester
File Edit Repository Record
Issue Tracker - Create Issues
Title
Issue Tracker - Login [1]
Issue Tracker - Issues [1]
Issue Tracker - Issues [2]
Issue Tracker - Issues [3]
SmartNav.htm
Content
Headers
Response content
Issues
Podcast Pal
Create New Issue
ID
Issue (click t
HTML/XML
Text
Image
Raw
00:00.000 sec elapsed
Actors
Validators for Issue Tracker - Issues [1]
Actions Taken
Modifiers
Validators
Extractors

Create Content Validator

Validate Content

Recorded value contains search string.

Verify that

● Verify content is found

○ Verify content is not found

Look for

The content for this validation rule should be retrieved from:

● Constant: Podcast Pal

○ Dataset field: DataSet: Users Field: Username

○ User variable:

Response Content

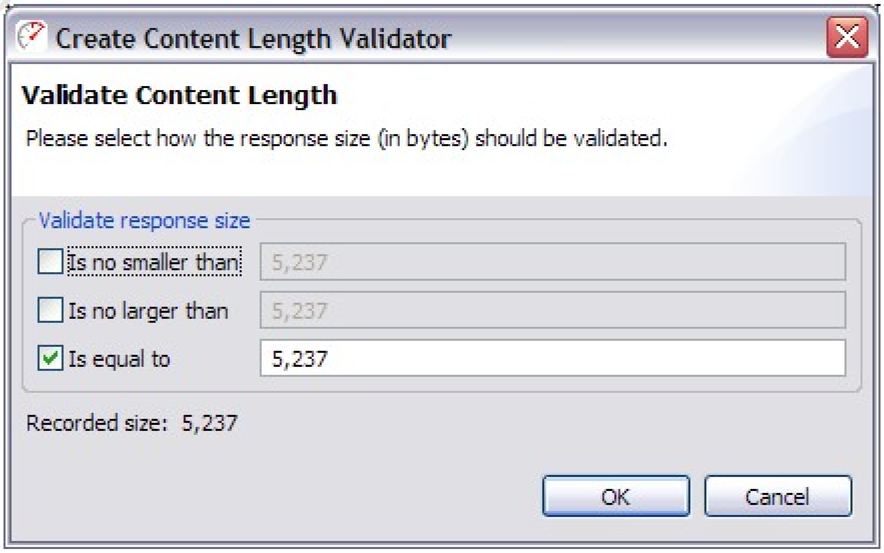
```
<table width="100%" cellspacing="10">
<tr>
   <td>
   <h1><span id="lblProjectName">Podcast Pal</span> </h1>
   <input type="submit" name="btnAdd" value="Create New Issue" onclick="if (typeof(Page
   </td>
</tr>
<tr>
   <td align="right">
   View Issues:
```

OK Cancel

Tip: It is not necessary to tab back and forth between the Content View and the Actors View. Either view may be displayed side-by-side with the other by dragging the top tab of the view to the edge of the other view. This makes it easy to Copy text from the Content View, press the "Add" button of the Actors View, and Paste the text into the "Constant" field to validate on.

Size validation

It is possible to also validate that the size of a response remains within reasonable bounds by using size validation. This style of validation may be used by selecting "Validate content length..." from the drop-down menu next to the *Add* button.



Create Content Length Validator

Validate Content Length

Please select how the response size (in bytes) should be validated.

Validate response size

☐ Is no smaller than 5,237

☐ Is no larger than 5,237

☑ Is equal to 5,237

Recorded size: 5,237

OK Cancel

Using this validator, it is possible to verify the size of the response. Simply check the appropriate options, and enter the corresponding size constraints (measured

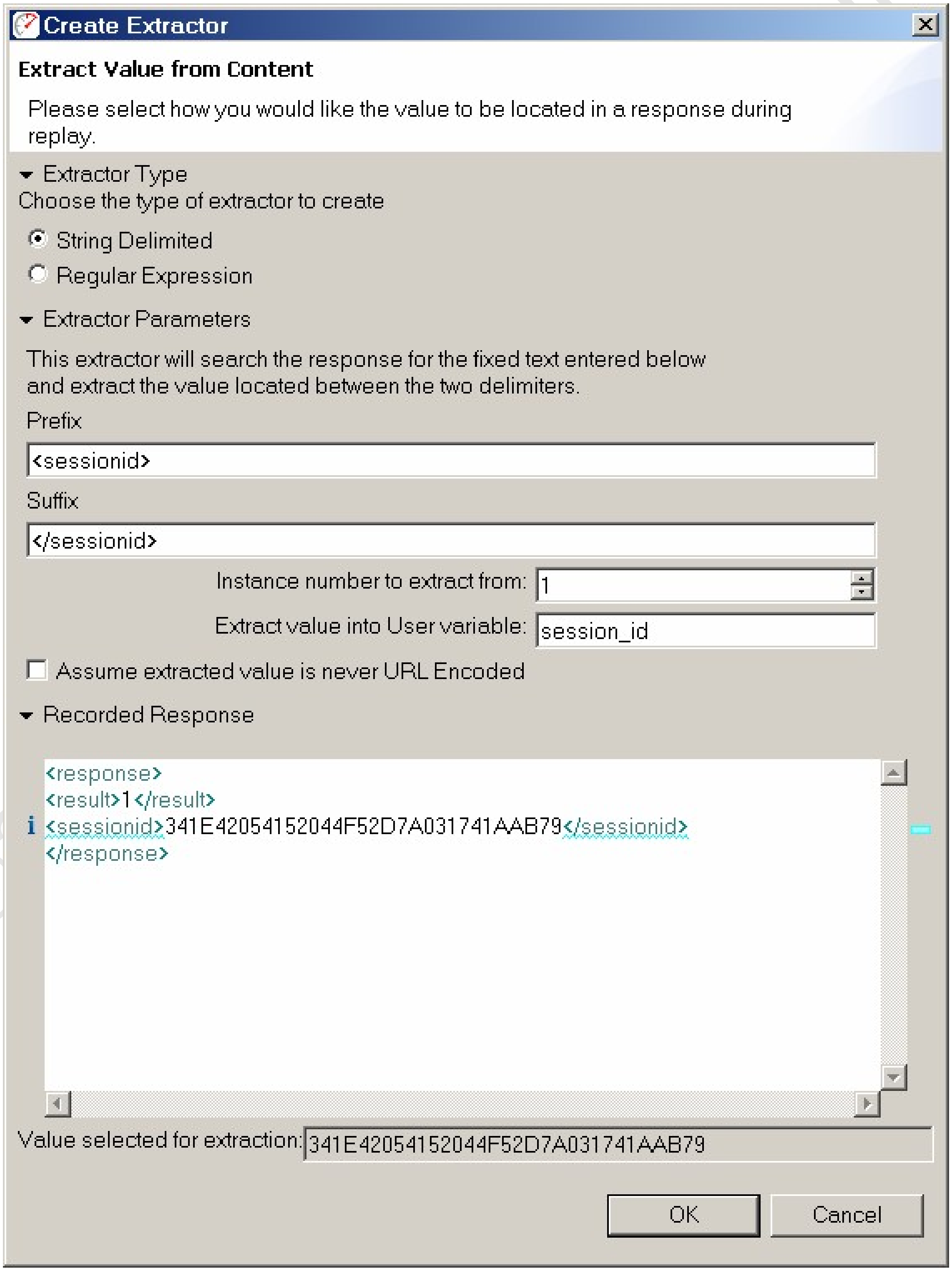
in bytes). Use "Is no smaller than" to specify a minimum allowable size, and/or "Is no larger than" for a maximum allowable size; or select "Is equal to" to specify a single exact value.
Note: Due to a server's ability to vary the transmitted size of a response (for example: by altering compression scheme or transfer encoding), this option may not be available for some responses.

Extractors

Extractors are able to examine a response and extract information that can be re-used later by a Modifier.

String-delimited Extractors

The Actors View allows you create and edit simple extractors capable of extracting a value into a User Variable. Since the value will likely be changing, an extractor may be specified by using a pair of delimiting anchors to denote the beginning and end of the value to be extracted.



Create Extractor

Extract Value from Content

Please select how you would like the value to be located in a response during replay.

Extractor Type

Choose the type of extractor to create

● String Delimited

○ Regular Expression

Extractor Parameters

This extractor will search the response for the fixed text entered below and extract the value located between the two delimiters.

Prefix

<sessionid>

Suffix

</sessionid>

Instance number to extract from: 1

Extract value into User variable: session_id

☐ Assume extracted value is never URL Encoded

Recorded Response

```
<response>
<result>1</result>
<sessionid>341E42054152044F52D7A031741AAB79</sessionid>
</response>
```

Value selected for extraction: 341E42054152044F52D7A031741AAB79

The first section of this dialog allows the prefix and suffix anchors to be entered. The extractor will search for these anchors in the response received during playback, and extract the value located between them. The "Instance number to extract from" can be increased if the extractor should skip and ignore initial occurrences of the prefix anchor.

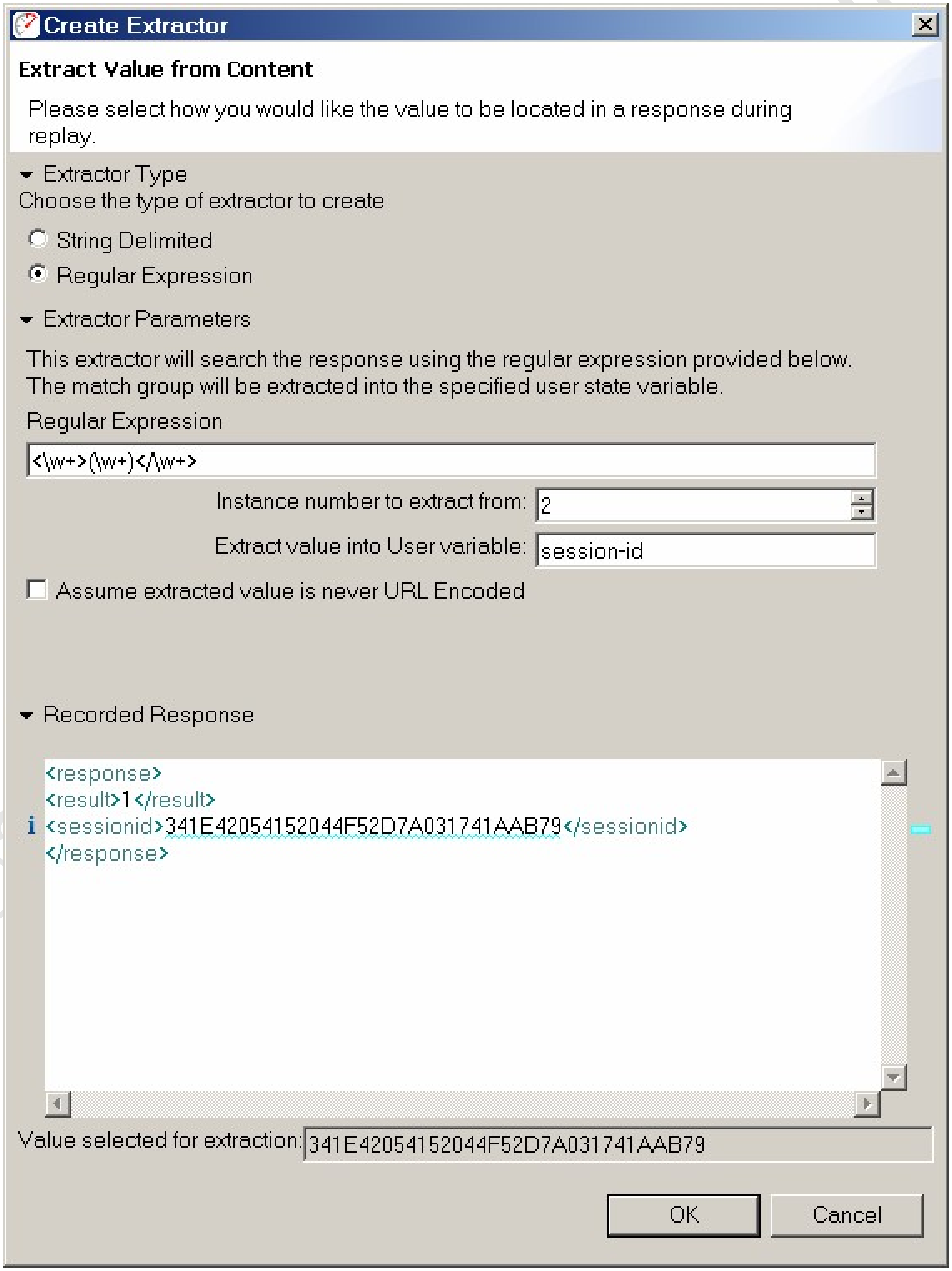
Next, the extractor needs to know the name of a user variable to extract the value into. The name of the user variable is used to identify the value in the user state - such as in a modifier that needs the extracted value later in the testcase. Please note that variable names starting with a non-alphanumeric character (e.g. '#') are reserved for use by Web Performance Suite™ and may be overwritten by the software as needed.

The field "Assume extracted value is never URL Encoded" controls the context of the extracted value, and how encoding is performed when a modifier re-submits this value. This capability is available for advanced users, the default value (unselected) will suffice for most normal cases. If the extracted value appears URL Encoded, and can potentially contain the characters "+" and/or "%", but no other characters that would be encoded by a URL Encode process, then this field may be checked to indicate that those characters **must** be encoded ("%2B" and "%25", respectively) when the extracted value is re-transferred back up to the HTTP server.

Finally, the bottom section of this dialog shows the response that was received when this testcase was recorded. Additionally, a sample value is displayed of what would be extracted, if this extractor processed a response from the server identical to the response being displayed.

Regular Expression Extractors

For more flexible searches, choose the *Regular Expression* option in the *Extractor* type section. Configuration of this extractor is very similar to the String-delimited extractor. The primary difference is that instead of supplying prefix and suffix strings, a single regular expression is supplied.



Extract Value from Content

Please select how you would like the value to be located in a response during replay.

▾ Extractor Type

Choose the type of extractor to create

○ String Delimited

● Regular Expression

▾ Extractor Parameters

This extractor will search the response using the regular expression provided below. The match group will be extracted into the specified user state variable.

Regular Expression

```
<\w+>(\w+)</\w+>
```

Instance number to extract from: 2

Extract value into User variable: session-id

□ Assume extracted value is never URL Encoded

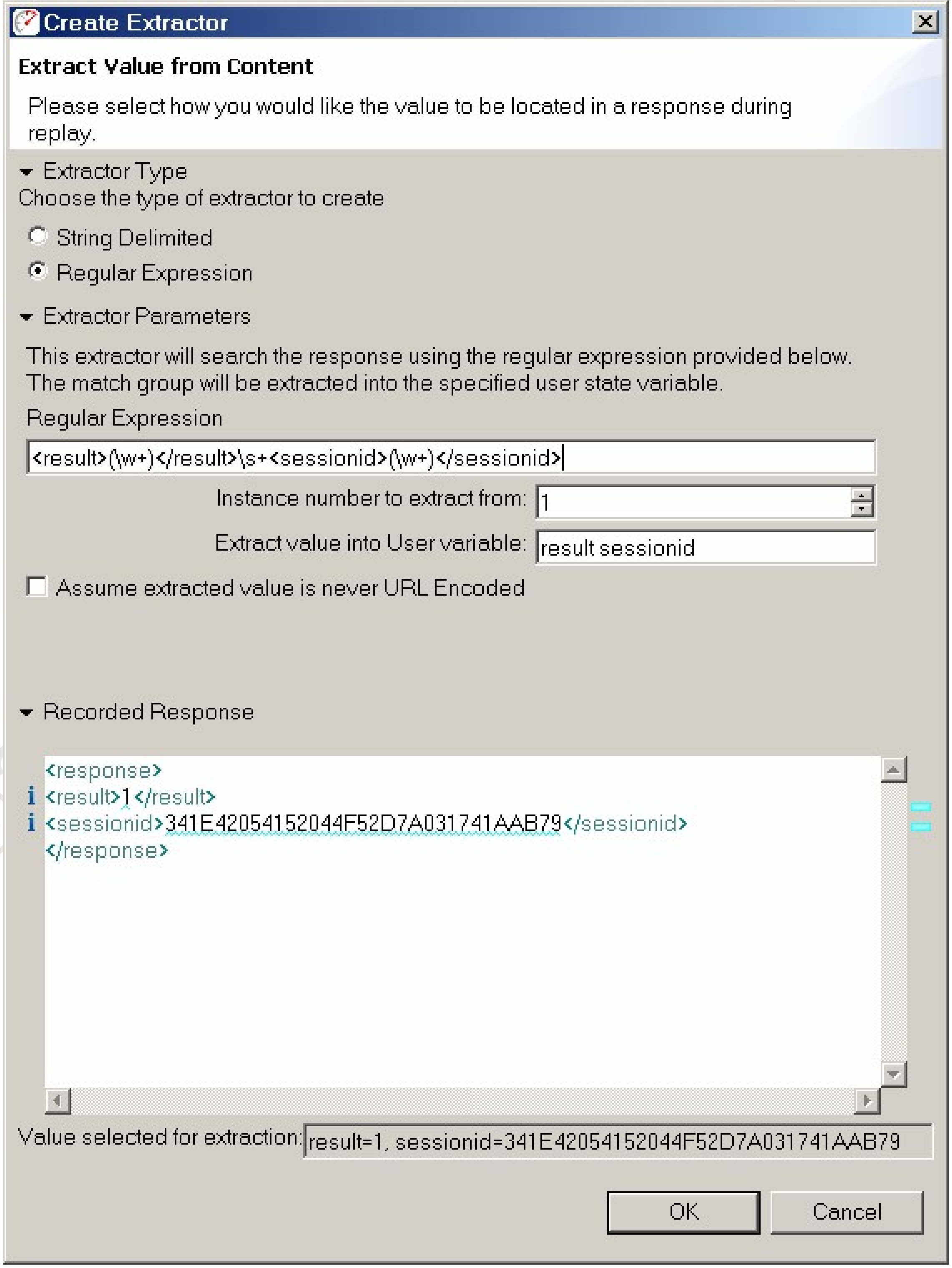
▾ Recorded Response

```
<response>
<result>1</result>
<sessionid>341E42054152044F52D7A031741AAB79</sessionid>
</response>
```

Value selected for extraction: 341E42054152044F52D7A031741AAB79

Using Multiple match groups

More advanced regular expressions that contain multiple match groups may also be used. In the example below, two match groups are specified in the regular expression and two variable names are provided in the user variable field. Note that when multiple match groups are used, the user variable names may not contain spaces.



Create Extractor

Extract Value from Content

Please select how you would like the value to be located in a response during replay.

Extractor Type

Choose the type of extractor to create

○ String Delimited

● Regular Expression

Extractor Parameters

This extractor will search the response using the regular expression provided below. The match group will be extracted into the specified user state variable.

Regular Expression: `<result>(\w+)</result>\s+<sessionid>(\w+)</sessionid>`

Instance number to extract from: 1

Extract value into User variable: result sessionid

□ Assume extracted value is never URL Encoded

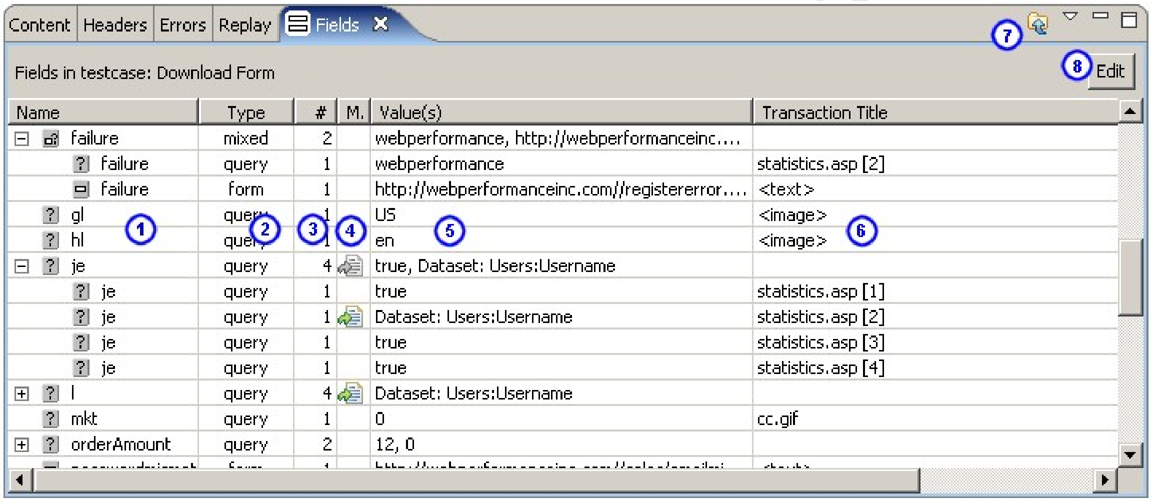
Recorded Response

```
<response>
<result>1</result>
<sessionid>341E42054152044F52D7A031741AAB79</sessionid>
</response>
```

Value selected for extraction: result=1, sessionid=341E42054152044F52D7A031741AAB79

Fields View

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



Fields in testcase: Download Form

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)*, form field (*field*), POST content (post), multipart (*part*), or form field file upload (*file*). 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

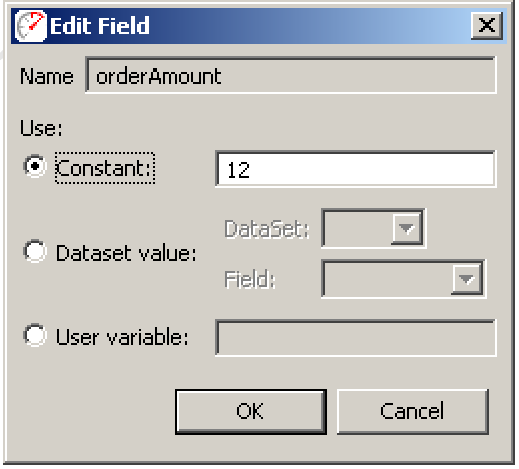
Go to Transaction

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



Name orderAmount

Use:

● Constant: 12

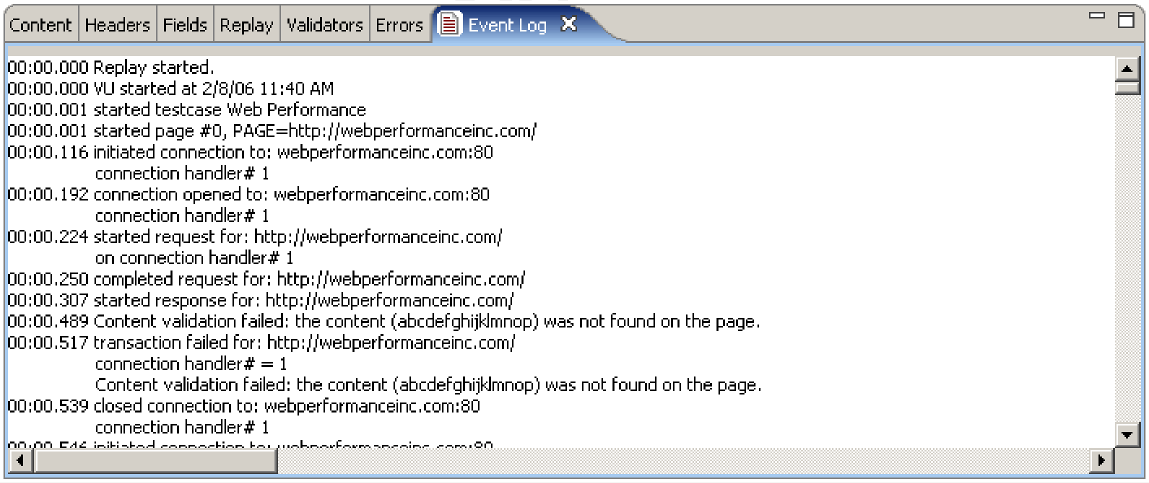
If multiple fields are selected, they will all be changed by this dialog. If they have different settings when the dialog is opened, the *Constant/Dataset/User* buttons will be initially de-selected - one must be chosen before editing can take place. For more information on using modifiers to customize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



```
00:00.000 Replay started.
00:00.000 VU started at 2/8/06 11:40 AM
00:00.001 started testcase Web Performance
00:00.001 started page #0, PAGE=http://webperformanceinc.com/
00:00.116 initiated connection to: webperformanceinc.com:80
          connection handler# 1
00:00.192 connection opened to: webperformanceinc.com:80
          connection handler# 1
00:00.224 started request for: http://webperformanceinc.com/
          on connection handler# 1
00:00.250 completed request for: http://webperformanceinc.com/
00:00.307 started response for: http://webperformanceinc.com/
00:00.489 Content validation failed: the content (abcdefghijklmnop) was not found on the page.
00:00.517 transaction failed for: http://webperformanceinc.com/
          connection handler# = 1
          Content validation failed: the content (abcdefghijklmnop) was not found on the page.
00:00.539 closed connection to: webperformanceinc.com:80
          connection handler# 1
00:00.546 initiated connection to: webperformanceinc.com:80
```

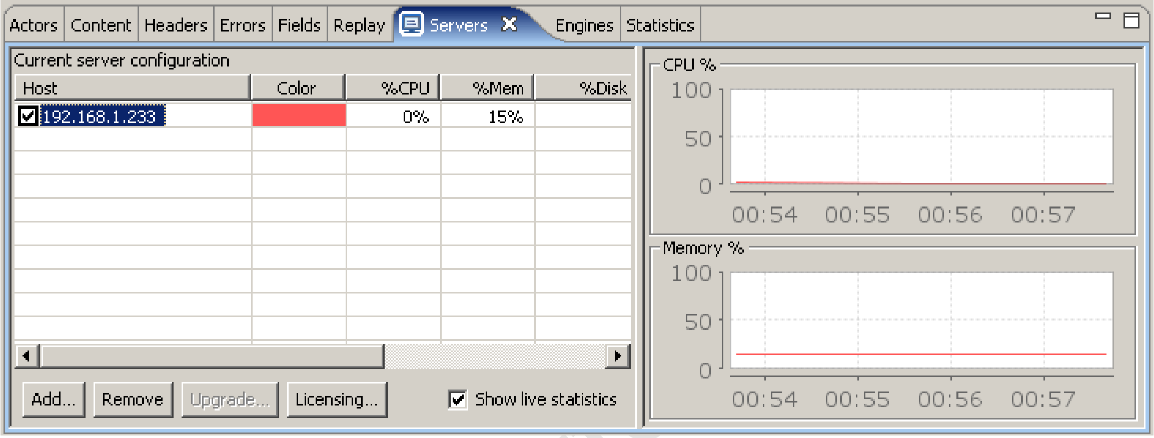
In order to activate the replay logging feature, the view must be activated before starting the replay. This means that it must have been opened at some point prior to starting the replay (it does not need to be visible at that time the replay starts). After the replay is complete, the log may be viewed 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

For more information about server monitoring, see the Introduction to Server Monitoring page.

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.



Current server configuration

Host	Color	%CPU	%Mem	%Disk
☑ 192.168.1.233	🟥	0%	15%	

☑ Show live statistics

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, you may need to first add the server to the view, depending on the type of server monitoring you are using. See the Basic Server Monitoring and Advanced Server Monitoring pages for more information.

To monitor a server during a load test, be sure the server is selected (via the checkboxes in the view) before starting the load test.

The following options may be used to manage the Server Monitoring configuration.

- Add - Start monitoring a new server, as described on the Basic Server Monitoring page.
- Remove - Stop and remove the highlighted monitor(s)

Additionally, the following controls may be used to configure Server Monitoring Agents.

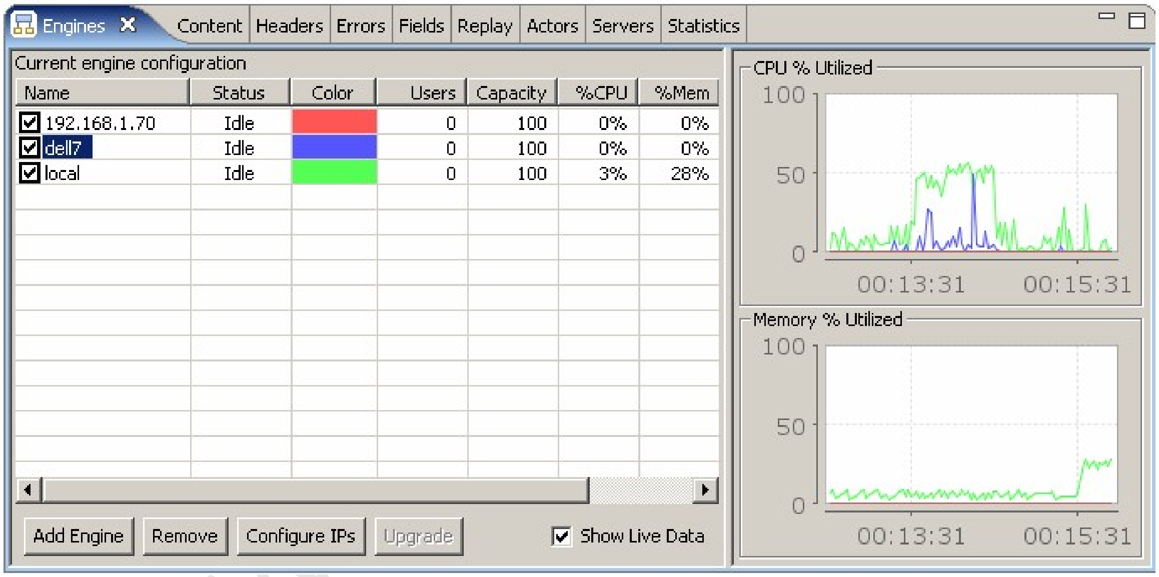
- Upgrade - Upgrades the remote Server Monitor Agent to the same version being used by the controller. This operation will require that the agent application be quit and re-opened on the controller before the upgrade can take full effect.
- Licensing - Manage the licenses installed on the remote Server Monitor Agent.

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.

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.



Current engine configuration

Name	Status	Color	Users	Capacity	%CPU	%Mem
☑ 192.168.1.70	Idle	🟥	0	100	0%	0%
☑ dell7	Idle	🟦	0	100	0%	0%
☑ local	Idle	🟩	0	100	3%	28%

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", "Initializing", and "Stopping."

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 to 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.

If you receive an error attempting to add an engine that is known to be running, then it may be necessary to configure any firewalls and the engine to permit connections with the controller. For more information, please see Configuring a Load Engine.

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.

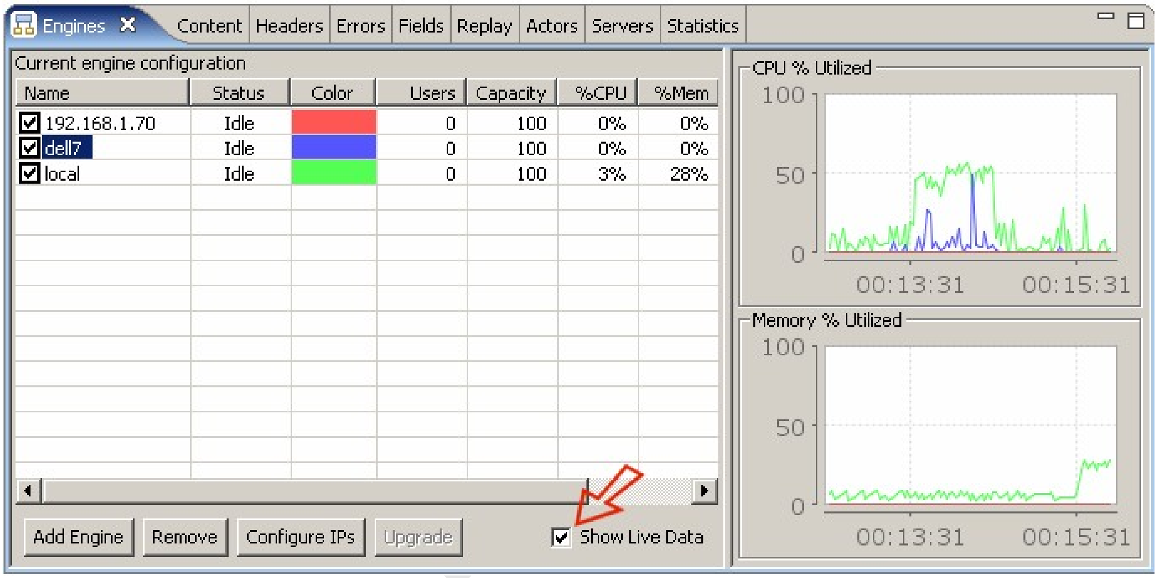
Configuring Engines

For engines which support using multiple IP addresses, it is possible to configure which IP addresses the engine will use during a load test. To do this, select an enabled engine, and press the *Configure IPs* button. A dialog will be displayed allowing you to configure the IP Aliasing settings for the engine.

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:



Current engine configuration

Name	Status	Color	Users	Capacity	%CPU	%Mem
☑ 192.168.1.70	Idle	🟥	0	100	0%	0%
☑ dell7	Idle	🟦	0	100	0%	0%
☑ local	Idle	🟩	0	100	3%	28%

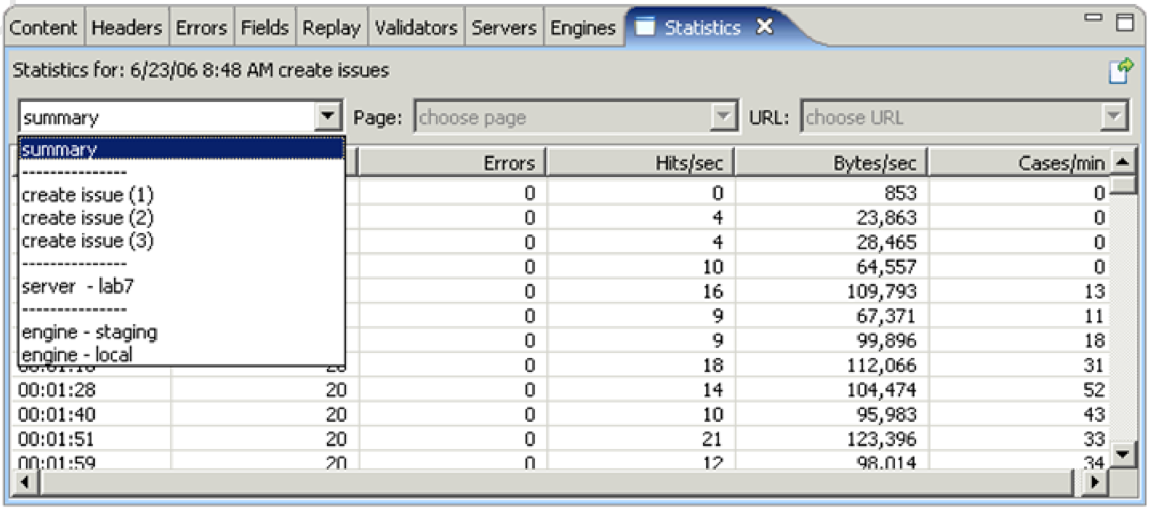
☑ Show Live Data

Statistics View

This view provides access to the detailed statistics gathered during a load test. The drop-down choice fields at the top provide navigation to the various categories and levels of statistics.

The first field allows selection from the 4 main categories:

1. summary - statistics relating to the entire test
2. testcases - statistics for each testcase entry in the load configuration
3. servers - statistics for the servers monitored during the test (if any)
4. engines - statistics for the load-generating engines used during the test



Statistics for: 6/23/06 8:48 AM create issues

summary Page: choose page URL: choose URL

		Errors	Hits/sec	Bytes/sec	Cases/min
		0	0	853	0
		0	4	23,863	0
		0	4	28,465	0
		0	10	64,557	0
		0	16	109,793	13
		0	9	67,371	11
		0	9	99,896	18
[illegible]	[illegible]	0	18	112,066	31
00:01:28	20	0	14	104,474	52
00:01:40	20	0	10	95,983	43
00:01:51	20	0	21	123,396	33
00:01:59	20	0	12	98,014	34

When a testcase is selected in the first choice field, the remaining fields allow selection of the page and URL within the testcase.

Tooltips for each column describe the meaning of each column.

Exporting data

The data displayed in the view can be exported at any time to CSV format. To export, press the *export* button and then provide a file location.

Settings

General Settings

Dialogs

This section controls the default behavior some of the most commonly used dialogs.

Testcase Editor

These settings affect the default behavior of the Testcase Editor.

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.

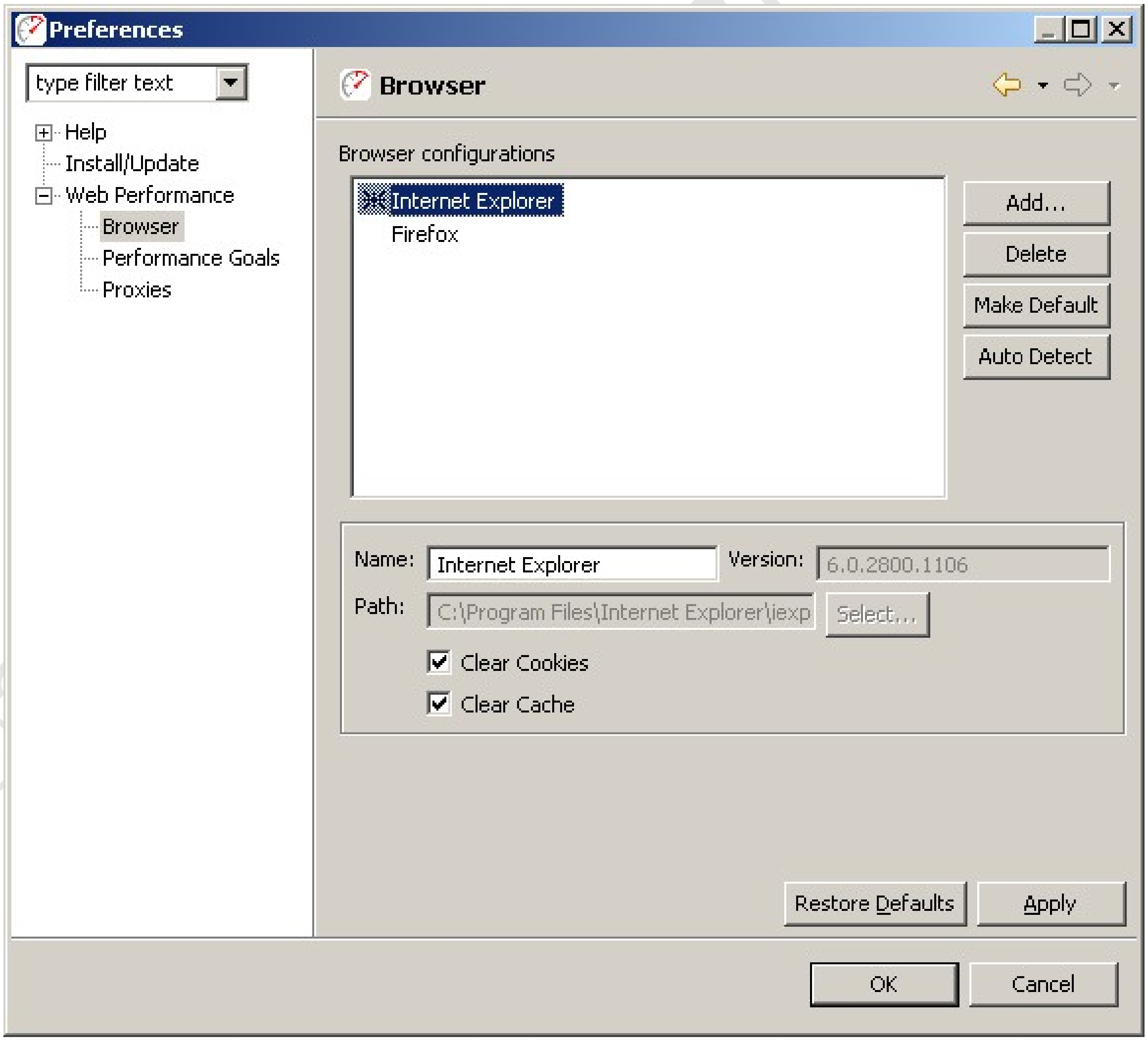
- Replays containing NTLM or Basic authentication should have the user identitiy configured before attempting to replay the testcase. In some scenarios, this behavior may need to be overridden. Changing the *Allow replay without authentication configured* item allows a replay to proceed even if the authentication 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.

- By default the Virtual User will wait for a page to complete prior to loading the child resources on the page. Turning off this setting will cause the Virtual User to start loading child resources as soon as the start of the response is received.

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.



Browser configurations

- *Internet Explorer
- Firefox

Name: Internet Explorer

Version: 6.0.2800.1106

Path: C:\Program Files\Internet Explorer\iexp

☑ Clear Cookies

☑ Clear Cache

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 a 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

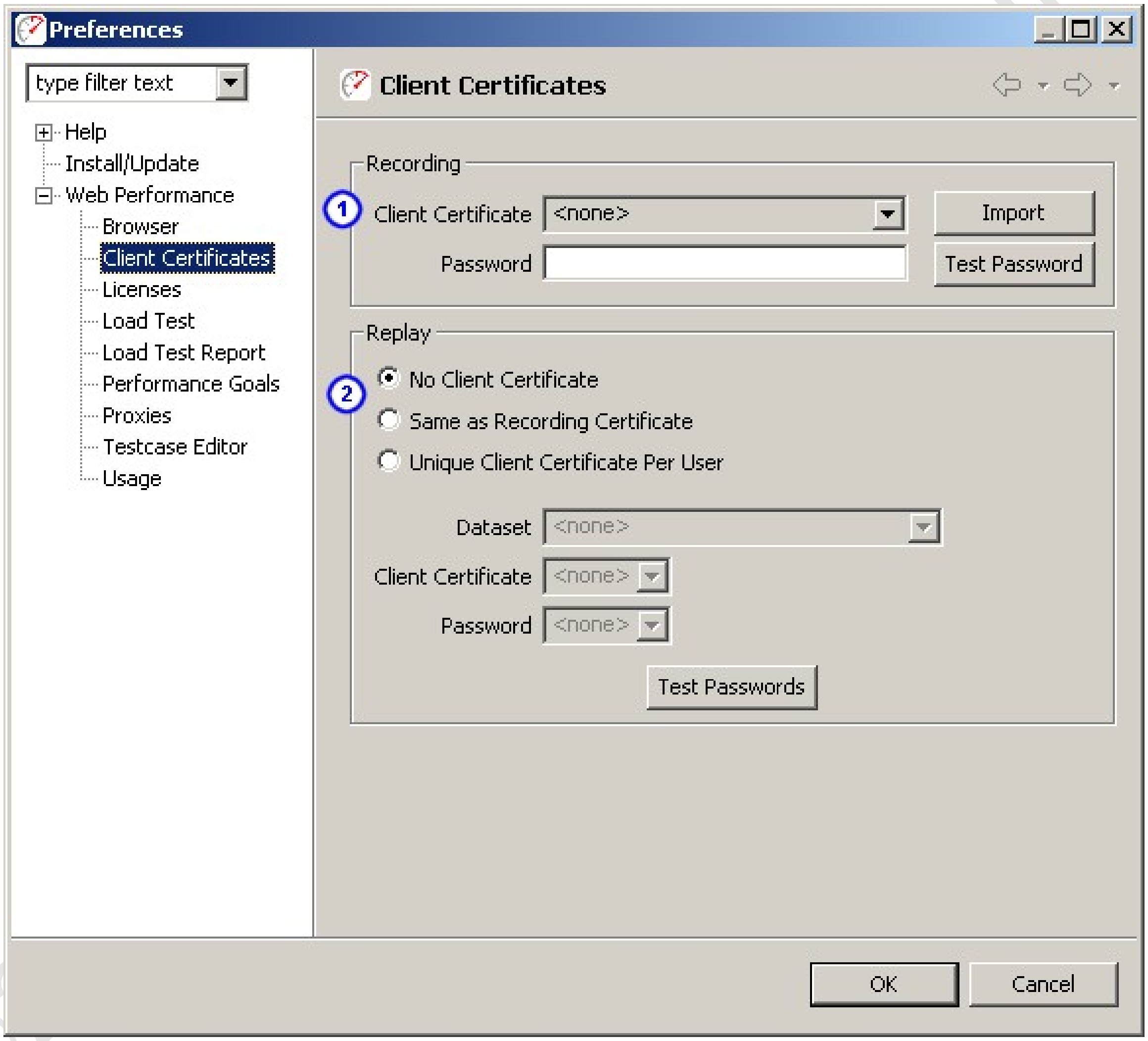
By default, the browser's cookies and cache are cleared prior to initiating recording and restored shortly afterwards. If this is not done, the recording would reflect the state of the browser's cache. For example, cached pages or images might never be requested from the server - which would result in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

Client Certificates

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



Recording

Client Certificate <none>

Password

Replay

● No Client Certificate

○ Same as Recording Certificate

○ Unique Client Certificate Per User

Dataset <none>

Client Certificate <none>

Password <none>

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.

License Key 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 license keys

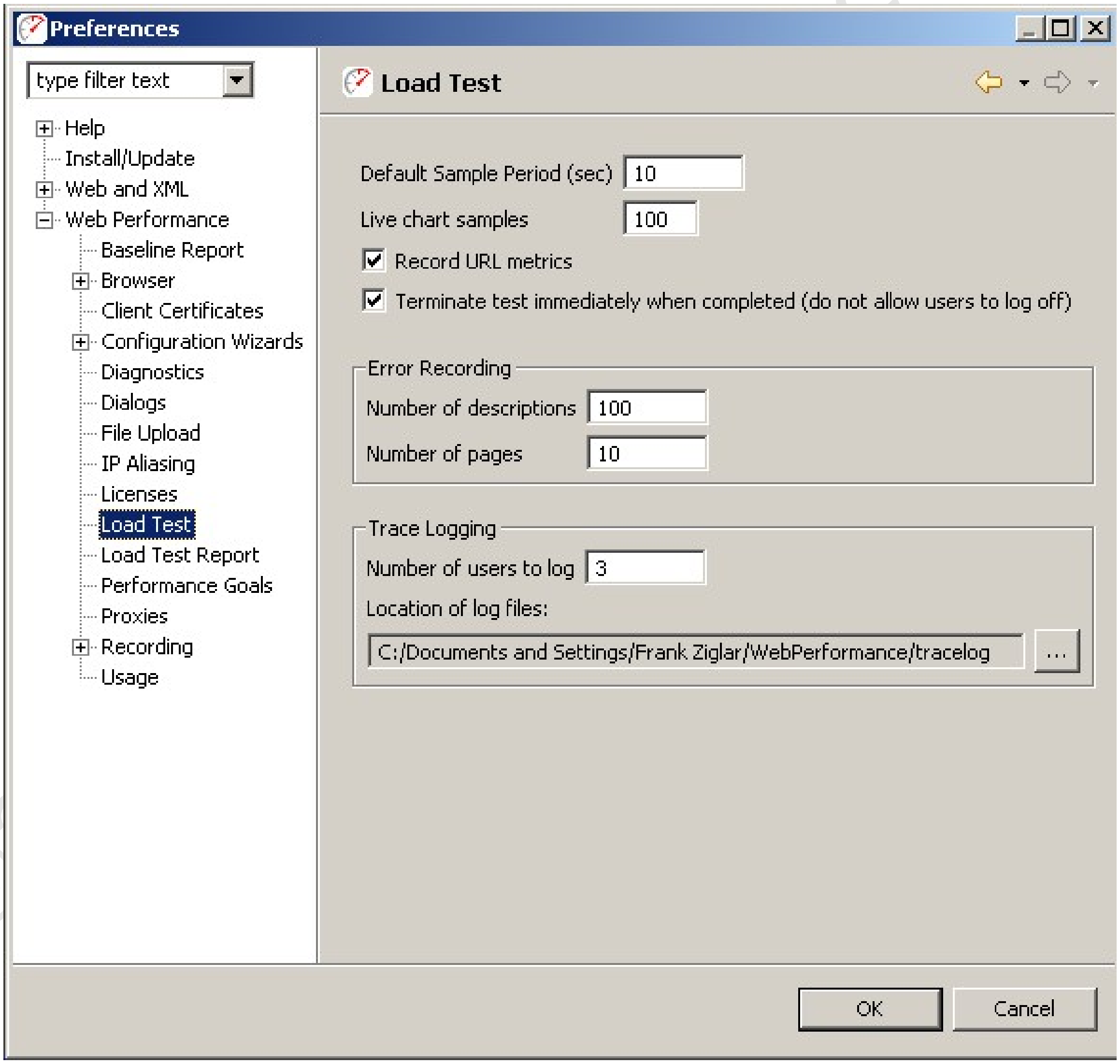
Expired license keys are indicated with *(expired)* in the license key descriptions. You may safely delete these license keys with no effect on the program.

Disabled

When a license key 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.

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.



Load Test

Default Sample Period (sec) 10

Live chart samples 100

☑ Record URL metrics

☑ Terminate test immediately when completed (do not allow users to log off)

Error Recording

Number of descriptions 100

Number of pages 10

Trace Logging

Number of users to log 3

Location of log files:

C:/Documents and Settings/Frank Ziglar/WebPerformance/tracelog ...

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.

Live Chart Samples

Specifies the number of points to plot in each chart while a test is under way. Increasing this value will allow each chart to show data for a greater length of time, but may increase the amount CPU load required by each chart as it is animated. This setting has no effect when reviewing data from a previous test.

Record URL metrics

This setting determines whether or not the load test will collect metrics for each URL in your testcase. Disabling this value will greatly reduce the amount of data collected by the load test, while enabling it will provide more detailed information about each individual component within a web page. Page and test level data are always recorded.

Terminate test immediately when completed

Causes the load test to terminate when the allotted time for the test has finished. Normally, the test will give each virtual user a chance to complete their testcase, logging them off from the test site (if applicable). Enabling this option will cause all virtual users to cease running regardless of how far along they are in the page. Note that an active test may be halted in a similar fashion by pressing the red stop button in the Status View.

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.

VU Trace Log formats

A VU Trace Log is a detailed log of all the transactions performed by a virtual user. Trace logging is enabled from the *Play* menu. Please note that this feature will consume a large amount of disk space very quickly and has a significant effect on the performance of the product. As a result the feature will be reset (deactivated) each time Load Tester is restarted, to prevent accidental use during normal testing.

Trace logs provide detailed information about the activity of virtual users that is not needed for most load-testing tasks. It is provided only for troubleshooting and diagnostics. As such it is considered a "rough" feature that does not have a convenient graphical user interface. A text editor (e.g. notepad) must be used to view the log files.
The trace logs will be generated for the first 5 virtual users that run in the test. The resulting logs are stored in the *tracelog* folder under the Web Performance Trainer™ installation folder. The logs are stored in the following folder structure:

- T0
- T1
- Tn
    - BC0
    - BC1
    - BCn
        - VU0
        - VU1
        - VUn
            - R0
            - R1
            - Rn

T0...Tn: For each test run, a new folder will be created under the *tracelog* folder.
BC0...BCn: For each Business Case in the test, a corresponding folder will be created under the test folder.
VU0...VUn: For each virtual user running the Business Case, a corresponding folder will be created under the Business Case folder.
R0...Rn: For each repeat of the Business Case that the virtual user performs, a corresponding folder will be created under the virtual user folder.
Within each repeat folder (R0...Rn), a log of the user activities will be contained in a file called *log.txt*. An example of this file is shown here:

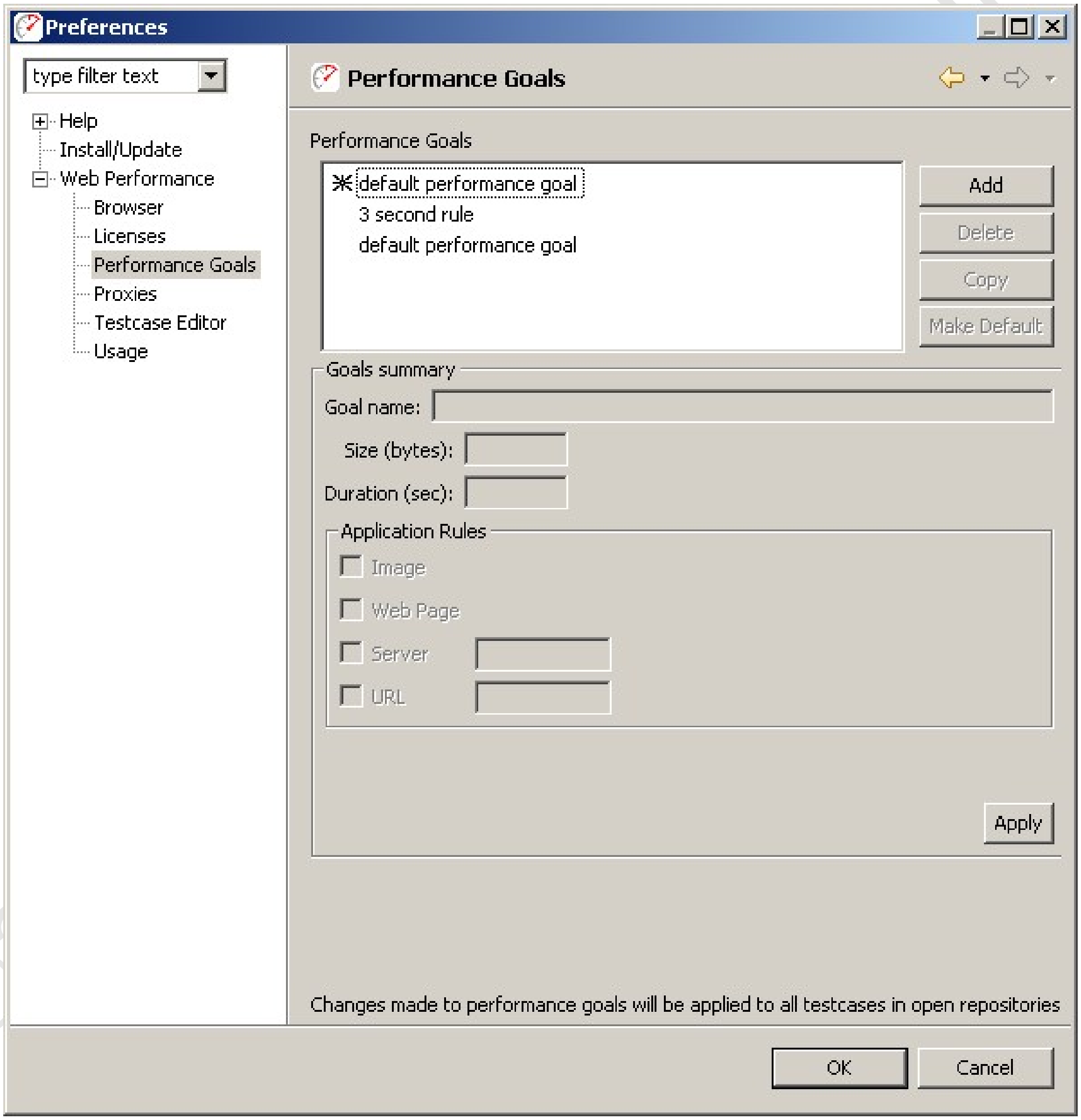
```
00:01.248 BCS: started testcase, BC=Case1
00:01.248 WPS: started page #0, PAGE=http://dell2:81/
00:01.249 CNI: initiated connection to: dell2:81
connection# = 20102672
00:01.251 CNO: connection opened to: dell2:81
connection# = 20102672
00:01.251 TQS: started request for: /
connection# = 20102672
00:01.253 TQE: completed request for: /
connection# = 20102672
00:01.257 TSS: started response for: /
connection# = 20102672
00:01.265 TSE: completed response for: /
connection# = 20102672
HTTP/1.1 200 OK
00:01.266 CNI: initiated connection to: dell2:81
connection# = 17099451
00:01.267 TQS: started request for: /tomcat.gif
```

```
connection# = 20102672
00:01.268 TQE: completed request for: /tomcat.gif
connection# = 20102672
00:01.269 CNO: connection opened to: dell2:81
connection# = 17099451
00:01.269 TQS: started request for: /jakarta-banner.gif
connection# = 17099451
00:01.281 TQE: completed request for: /jakarta-banner.gif
connection# = 17099451
00:01.282 TSS: started response for: /tomcat.gif
connection# = 20102672
00:01.286 TSE: completed response for: /tomcat.gif
connection# = 20102672
HTTP/1.1 200 OK
00:01.314 WPE: completed page #0, PAGE=http://dell2:81/
00:01.314 CNC: closed connection to: dell2:81
connection# = 20102672
00:01.314 CNC: closed connection to: dell2:81
connection# = 17099451
00:01.314 BCE: completed testcase, BC=Case1
```

Additionally, the full contents of each transaction (both request and response) are saved in files called *T0.txt*...*Tn.txt*. These files contain the full request and response, including the start-line, headers and content. If any parts of the request are modified from the original before sending to the server (e.g. data replacement), the file will reflect these changes exactly as they were sent to the server. For SSL transactions, the content will reflect the unencrypted content.

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).



default performance goal
3 second rule
default performance goal

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

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

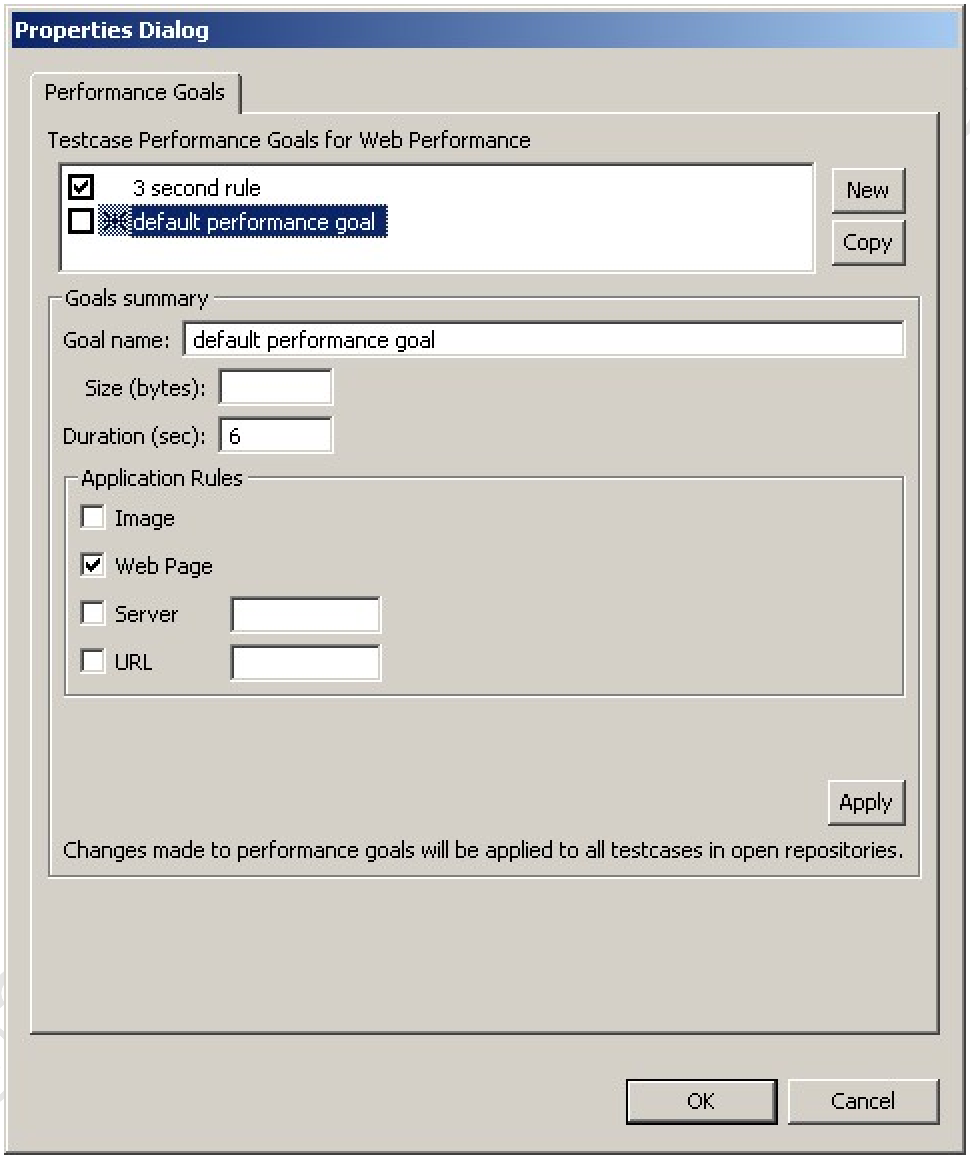
Notes

- 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.



Testcase Performance Goals for Web Performance

☑ 3 second rule
□ default performance goal

Goals summary

Goal name: default performance goal

Size (bytes):

Duration (sec): 6

Application Rules

□ Image
☑ Web Page
□ Server
□ URL

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

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, 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

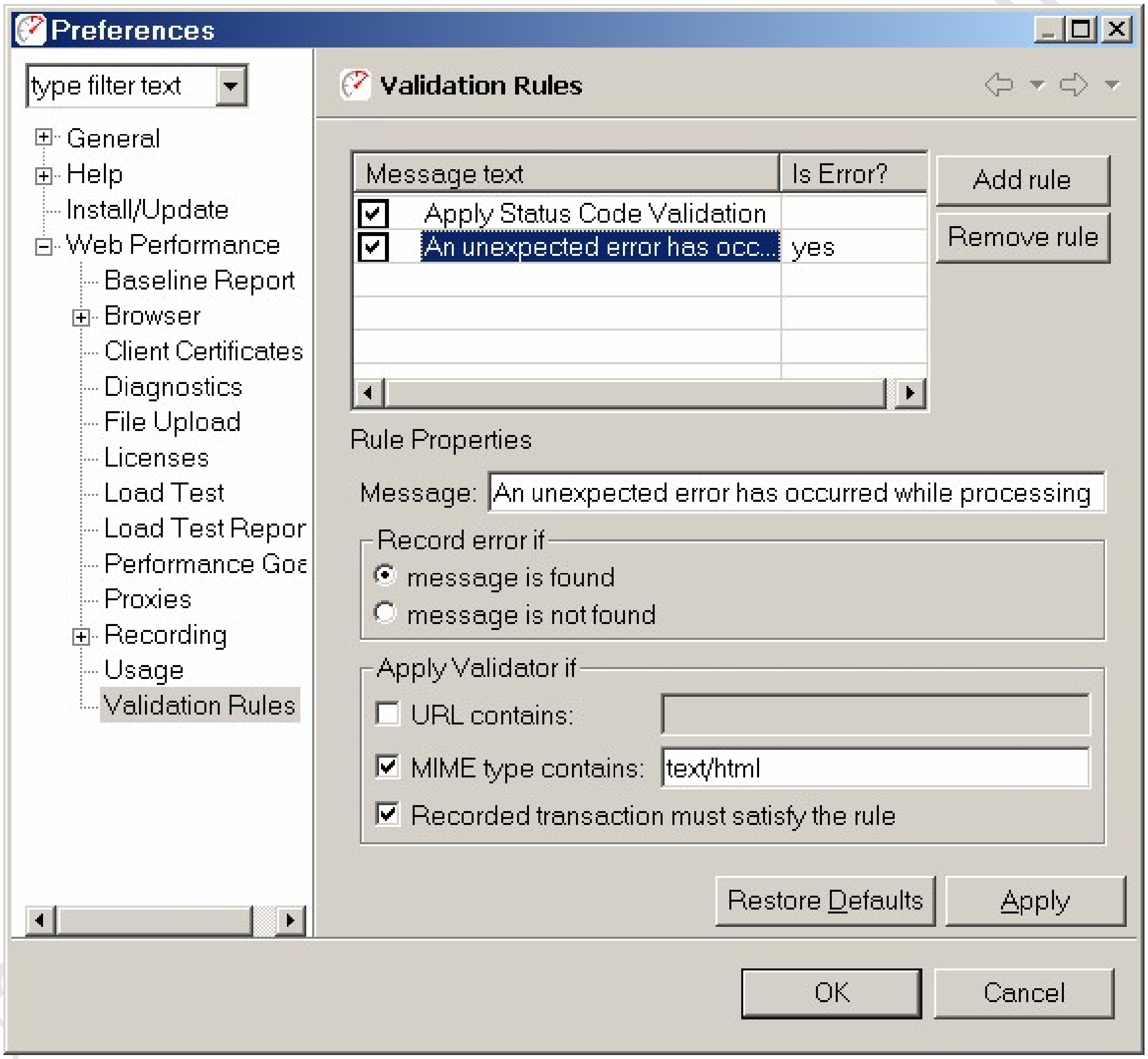
Notes

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.

Validation Preferences

The Validation Rules preferences page allows general rules to be entered that affect how the Validation Wizard will configure default validation rules for a testcase. Since pages may come back with different content as a test is being executed, these rules will help determine when a page may have encountered an error that should be flagged. The preferences page may be accessed by selecting Window->Preferences, and selecting the "Web Performance->Validation Rules" section. Changes on this page will only take affect once the Validation Wizard has been used to reconfigure a testcase.



Validation Rules

	Message text	Is Error?
☑	Apply Status Code Validation	
☑	An unexpected error has occ...	yes

Rule Properties

Message: An unexpected error has occurred while processing

Record error if
- ● message is found
- ○ message is not found

Apply Validator if
- □ URL contains:
- ☑ MIME type contains: text/html
- ☑ Recorded transaction must satisfy the rule

Creating a Validation Rule

To create a new validation rule, select the *Add Rule* button. Then, a message may be entered, which will be used as a search string in the content of resulting pages. Next, designate if the message entered indicates an error if and when it is found by selecting the appropriate option in the "Record error if" section. If the message is an error message, use the "Record error if: message is found" option. If the message indicates a string that is always expected to be found, the "Record error if: message is not found" option should be selected.

Apply Validator if

The rule may be refined to only apply to certain pages by using the "Apply Validator if" section. By default, each rule will only be applied to every URL in the testcase, except where the rule would have flagged an error with the initial recording. The following options are available:

URL contains

When this option is selected, a search string may be entered into the text block to constrain the transactions by URL. Only those transactions that have the search string present anywhere in their URL will be validated against using this rule. For example, in cases where a test might contain multiple host, and the error message was specific to the host "192.168.10.10", then this field might be set to "192.168.10.10".

MIME type contains

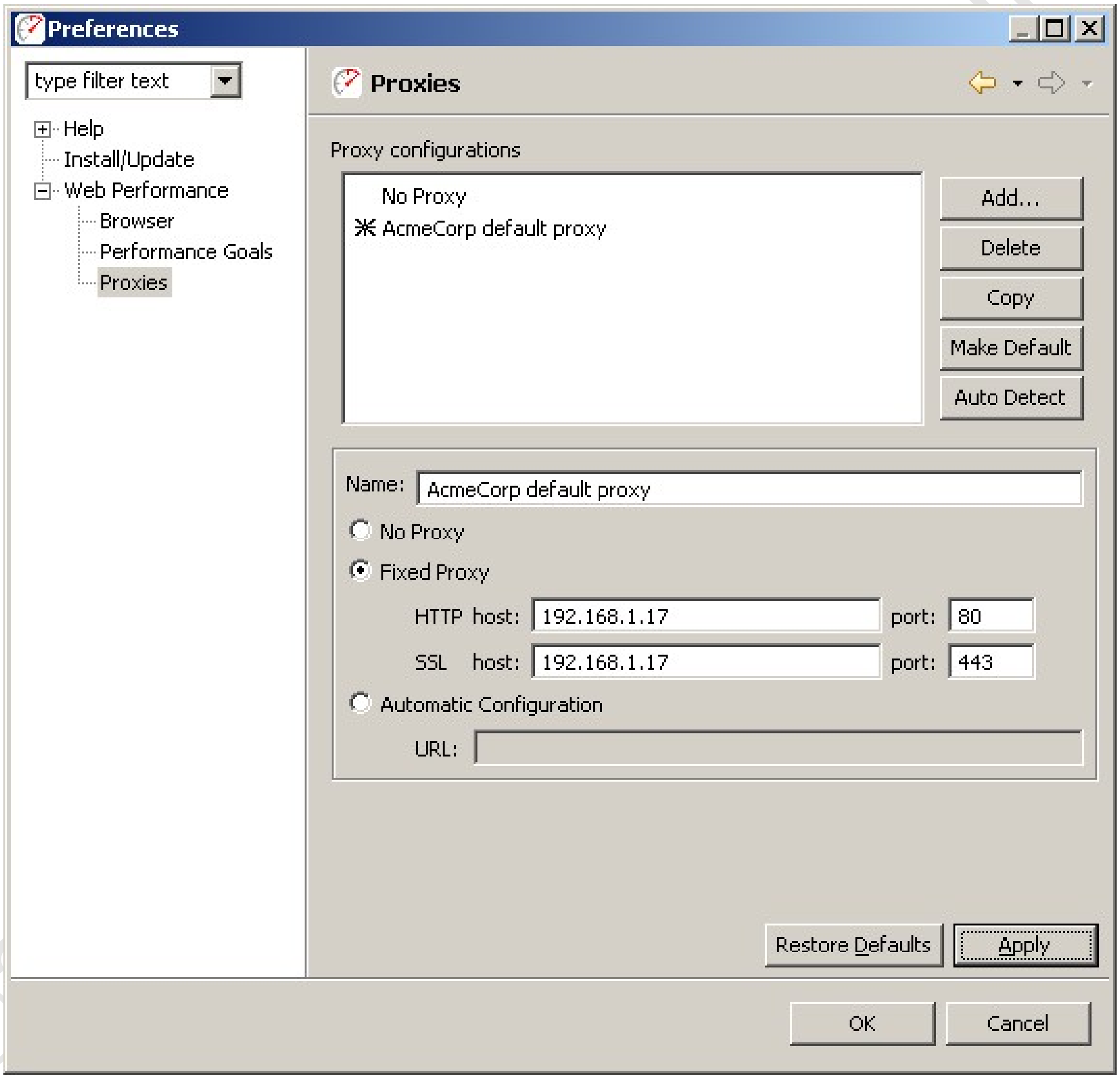
Selecting this option allows the validation to be constrained to only a certain type of resource, determined by it's MIME type. Only those resources with specified MIME types that contain the search string will be eligible for this form of validation. For example, to validate only HTML documents, this field might be set to "text/html" in order to skip validation on images. Likewise, setting this field to "text/xml" will apply validation only to XML results.

Recorded transaction must satisfy rule

When this option is enabled, this rule is quietly ignored for transactions whose recorded content would have caused this rule to raise an error. By default, this setting is enabled.

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.



Proxies

Proxy configurations

No Proxy

* AcmeCorp default proxy

Name: AcmeCorp default proxy

○ No Proxy

● Fixed Proxy

HTTP host: 192.168.1.17 port: 80

SSL host: 192.168.1.17 port: 443

○ Automatic Configuration

URL:

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.

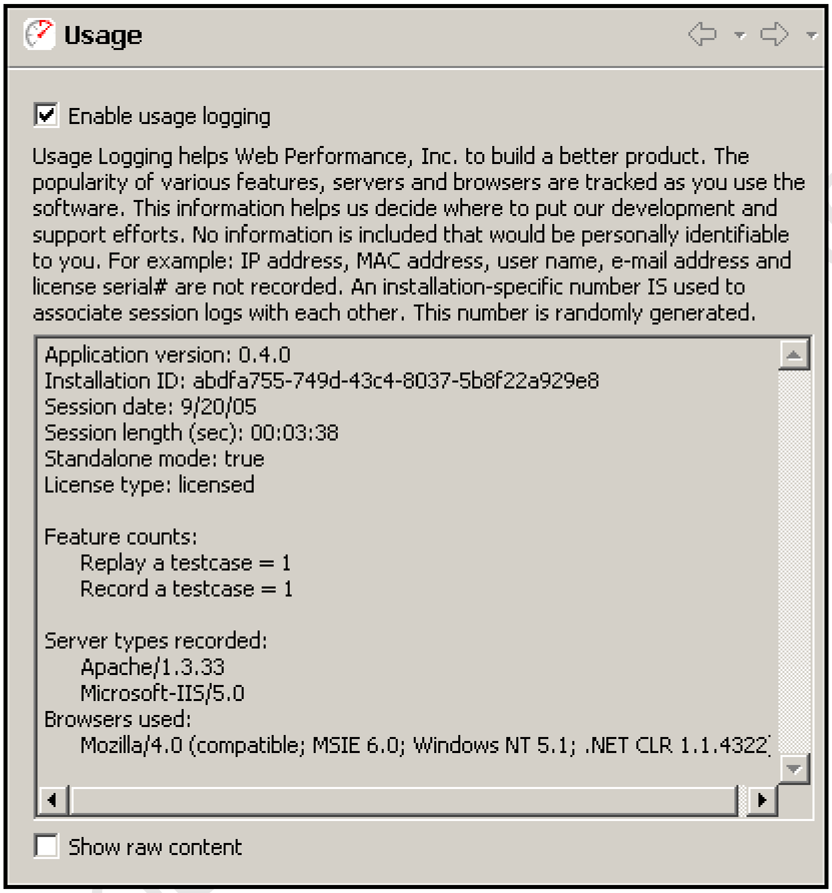
Deleting a proxy

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.

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.



☑ Enable usage logging

Usage Logging helps Web Performance, Inc. to build a better product. The popularity of various features, servers and browsers are tracked as you use the software. This information helps us decide where to put our development and support efforts. No information is included that would be personally identifiable to you. For example: IP address, MAC address, user name, e-mail address and license serial# are not recorded. An installation-specific number IS used to associate session logs with each other. This number is randomly generated.

```
Application version: 0.4.0
Installation ID: abdfa755-749d-43c4-8037-5b8f22a929e8
Session date: 9/20/05
Session length (sec): 00:03:38
Standalone mode: true
License type: licensed

Feature counts:
    Replay a testcase = 1
    Record a testcase = 1

Server types recorded:
    Apache/1.3.33
    Microsoft-IIS/5.0
Browsers used:
    Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; .NET CLR 1.1.4322
```

☐ Show raw content

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.

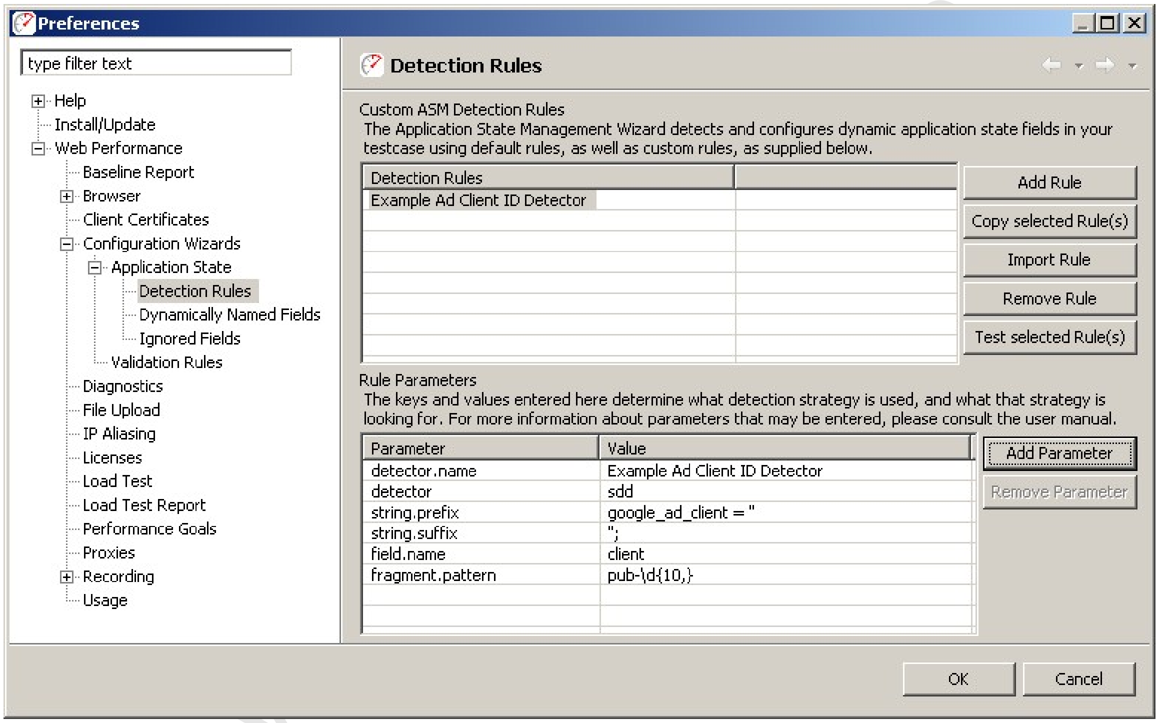
Advanced

Custom ASM Configuration

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.

Detectors may be defined by using the "Detection Rules" preferences page. This page may be accessed by selecting Window → Preferences… and then selecting Web Performance → Configuration Wizards → Application State → Detection Rules.



Custom ASM Detection Rules

The Application State Management Wizard detects and configures dynamic application state fields in your testcase using default rules, as well as custom rules, as supplied below.

Detection Rules	
Example Ad Client ID Detector	

Rule Parameters

The keys and values entered here determine what detection strategy is used, and what that strategy is looking for. For more information about parameters that may be entered, please consult the user manual.

Parameter	Value
detector.name	Example Ad Client ID Detector
detector	sdd
string.prefix	google_ad_client = "
string.suffix	";
field.name	client
fragment.pattern	pub-\d{10,}

To create a new rule, simply press the "Add Rule" button, and then enter the parameters for the detector. The parameters for a detector will vary based on the type of detection strategy desired. There are presently three basic types of detectors:

- Single Field String Delimited Detectors
- Variable Field String Delimited Detectors
- Dynamic Path Segment Detectors

Single Field String Delimited Detectors

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

Parameter	Value
detector	sdd
detector.name	Example Javascript Detector
string.prefix	setField('ID','
string.suffix	');
field.name	ID

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.

fragment.pattern

A regular expression which allows this detector to detect only a dynamic fragment of the value of the field. For more information on the behavior of this field, please see the Fragmented Value Detectors section.

Fragmented Value Detectors

Both of the String Delimited Detectors can be made to search for fragmented values (instead of complete values) by adding the Parameter "fragment.pattern". The value of this field should be a regular expression, which must match the isolated fragment of the field.

To understand how this works, consider an example field "client" with the value "ca-pub-1971486205579989". Now, let us suppose that the HTML document contains a Javascript fragment:

```
google_ad_client = "pub-1971486205579989";
```

In this case, only part of the value of the field has been declared in the source of the script. The full value is determined at a later point in time, by concatenating the prefix "ca-" with the variable value declared. In order to play back this case, the detector should only detect the dynamic fragment. This may be accomplished in our example using the following detector configuration:

Parameter	Value

detector	sdd
detector.name	Example Ad Client ID Detector
string.prefix	google_ad_client = "
string.suffix	";
field.name	client
fragment.pattern	pub-\d{10,}

In this case, the additional field "fragment.pattern" allows this detector to use a dynamic value defined by the HTML to replace "pub-1971486205579989" within the value "ca-pub-1971486205579989".

Variable Field String Delimited Detectors

Like the String Delimited Detector, 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 Variable Delimited Detector could be configured to detect both of these assignments (1234 and 56789, respectively) with the following configuration:

Parameter	Value
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.

fragment.pattern

A regular expression which allows this detector to detect only a dynamic fragment of the value of the field. For more information on the behavior of this field, please see the Fragmented Value Detectors section.

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

Parameter	Value
detector	dpsd
segment.pattern	(\d{6,})

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)?`

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*.

Configuring your computer for Multiple IP Addresses

This section covers how to configure your computer to generate virtual users from more than one IP address. This is only needed if your web application makes use of the client's IP addresses, which is quite rare, or if a piece of hardware such as load balancer uses client IP addresses. The concepts behind networks and which IP addresses are valid for your network are beyond the scope of the manual. **Please consult with your network administrator before going any further.** The following modifications have a high probability of rendering your computer inoperable if done incorrectly.

Do not use this configuration unless you are sure it is required!

An IP address is intended to identify the source of a computer's network traffic, and is used to route network packets on a network. By default virtual users will originate from the IP address of the computer running Web Performance Suite, but there are reasons why you may want virtual users to each have their own IP address. For example, some hardware load balancing devices use the IP address to route packets to different computers.

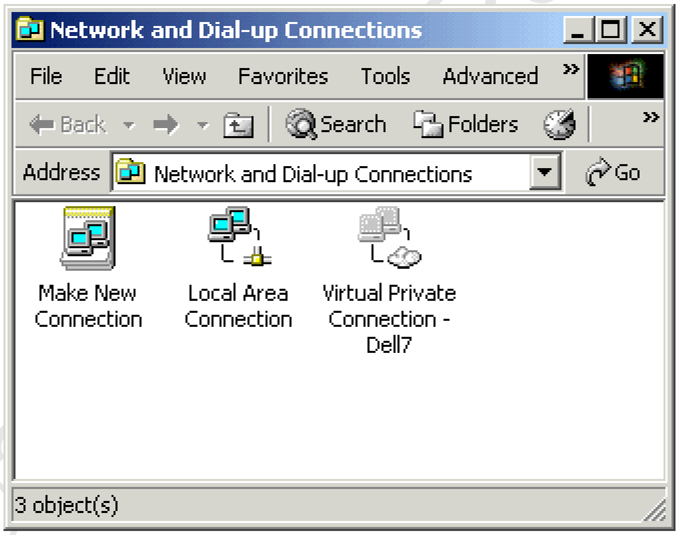
By default Web Performance Suite will use the Operating System to select an available network address, but at most you might have four network cards, which is not nearly enough to give every virtual user its own IP address. To get past this limitation the multiple IP address feature uses the ability of your operating system to configure virtual network devices. When it starts, Web Performance Suite will create a list of all real and virtual network devices. During a performance test as each virtual user is created it will be assigned a new IP address; if there are more users than IP addresses, the virtual users will grab an IP address from the front of the list.

The use of multiple IP addresses will also work if you have multiple playback engines, but you must configure virtual network devices on each computer separately.

The following sections describe how to configure virtual network devices on the different operating systems. Note that this feature of Web Performance Suite makes use of the built-in feature of your operating system to configure virtual network devices, and the complicated setup procedure is required by the operating system.

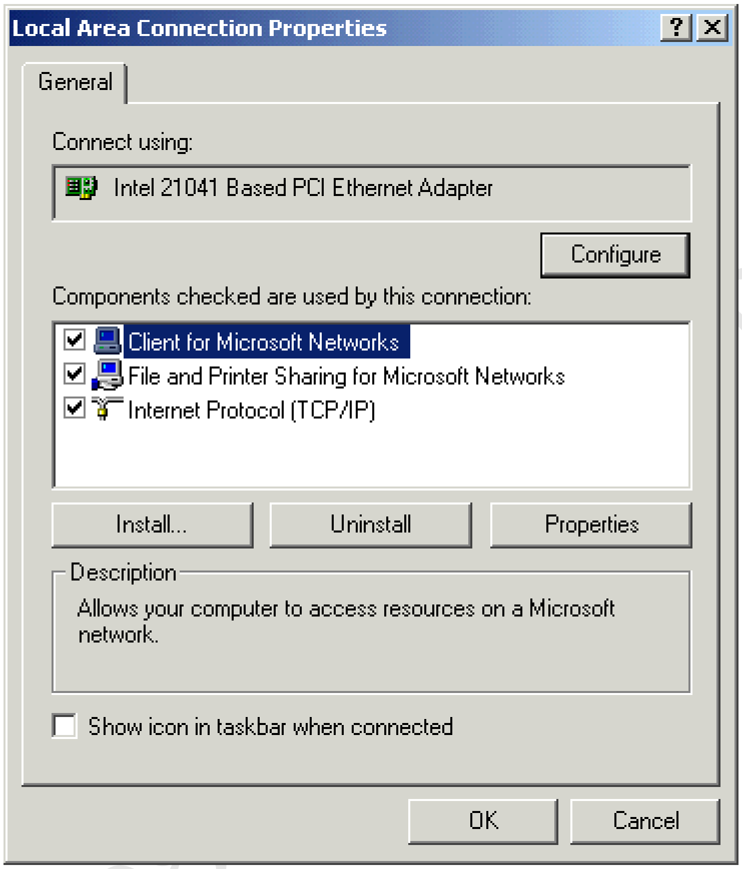
Windows

To configure a Windows machine to use multiple virtual IP addresses for right-click on *My Network Places* (on the *Desktop*) or execute *Start->Control Panel*, and double click on *Network and Dial-up Connections*:



Note that a limitation of Windows is that you can only configure virtual network devices using a Local Area Connection; VPNs and ISDN or other modem connections do not have this ability.

The next step is to edit the properties of your network connection, bringing up the following dialog:



Local Area Connection Properties

Connect using:

Intel 21041 Based PCI Ethernet Adapter

Components checked are used by this connection:

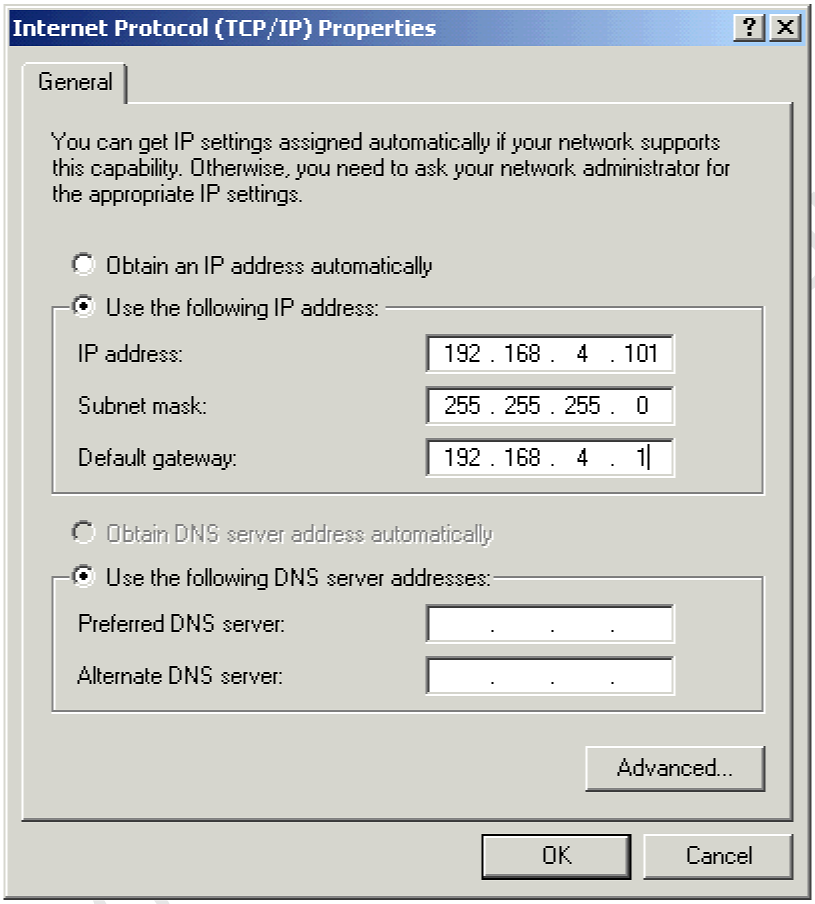
☑ Client for Microsoft Networks
☑ File and Printer Sharing for Microsoft Networks
☑ Internet Protocol (TCP/IP)

Description

Allows your computer to access resources on a Microsoft network.

☐ Show icon in taskbar when connected

Select *Internet Protocol (TCP/IP)* and click on the *Properties* button, which brings up the following dialog:



Internet Protocol (TCP/IP) Properties

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address: 192 . 168 . 4 . 101

Subnet mask: 255 . 255 . 255 . 0

Default gateway: 192 . 168 . 4 . 1

○ Obtain DNS server address automatically

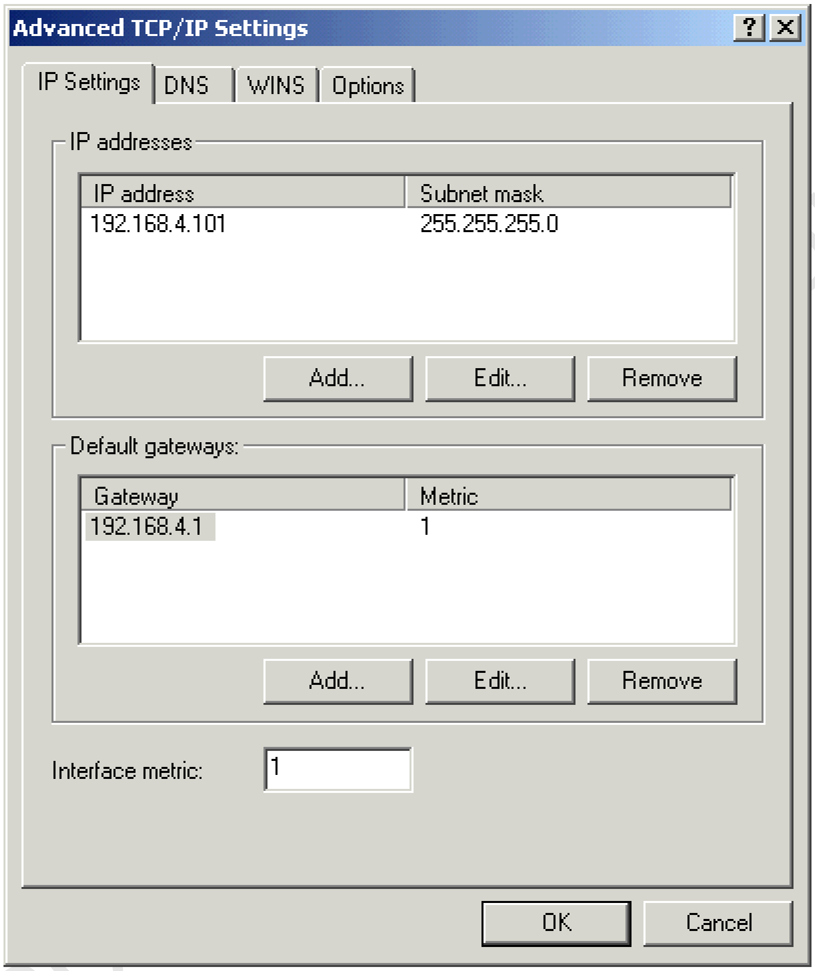
● Use the following DNS server addresses:

Preferred DNS server: . . .

Alternate DNS server: . . .

Note that in order to configure virtual IP addresses your computer must be configured to use fixed IP addresses; DHCP is not supported. If you are not on control of the IP addresses on your local network you should work with your network administrator to reserve a block of IP addresses.

The next step is to click on the *Advanced* button, bringing up this dialog:



IP addresses

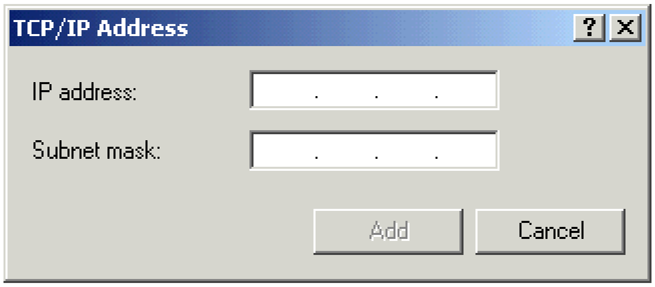
IP address	Subnet mask
192.168.4.101	255.255.255.0

Default gateways:

Gateway	Metric
192.168.4.1	1

Interface metric: 1

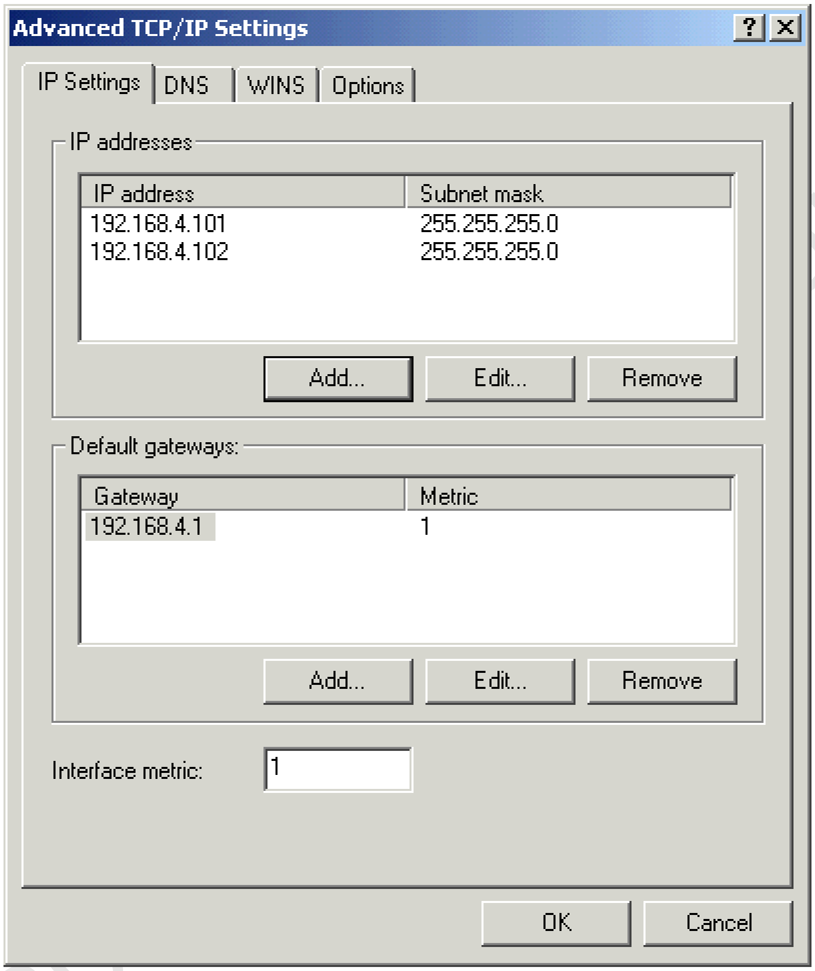
The above dialog (*Advanced TCP/IP Settings)* shows the list of IP addresses for your local computer. To add virtual network devices click on the *Add* button, which brings up the *TCP/IP Address Dialog*:



IP address: . . .

Subnet mask: . . .

Enter the IP address and subnet mask of the virtual network device you wish to configure, and this will be added to the list shown in the *Advanced TCP/IP Settings Dialog*:



IP addresses

IP address	Subnet mask
192.168.4.101	255.255.255.0
192.168.4.102	255.255.255.0

Default gateways:

Gateway	Metric
192.168.4.1	1

Interface metric: 1

The procedure should be repeated for each virtual IP address/network device that you wish to add.

Linux/UNIX

As with the Windows configuration, choosing valid IP addresses is beyond the scope of this manual, but typically you would want to perform this modification on a computer using private IP addresses in a test lab.

The `ifconfig` command can be used to dynamically create virtual network device. The following example shows the creating of a single virtual network device:

```
[root@bigtoe root]# ifconfig eth0:0 10.1.1.1
[root@bigtoe root]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:A0:C9:5A:DF:F7
   inet addr:10.0.0.100  Bcast:10.0.0.255
 Mask:255.255.255.0
   UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
   RX packets:454545 errors:0 dropped:0 overruns:0 frame:0
   TX packets:311037 errors:0 dropped:0 overruns:0
carrier:0
   collisions:0 txqueuelen:100
   RX bytes:94017376 (89.6 Mb)  TX bytes:31798276 (30.3 Mb)
   Interrupt:10 Base address:0xdc00 Memory:ef201000-
ef201038

eth0:0    Link encap:Ethernet  HWaddr 00:A0:C9:5A:DF:F7
   inet addr:10.1.1.1  Bcast:10.1.1.255  Mask:255.255.255.0
   UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
   RX packets:0 errors:0 dropped:0 overruns:0 frame:0
   TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
   collisions:0 txqueuelen:100
   RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
   Interrupt:10 Base address:0xdc00 Memory:ef201000-
ef201038
```

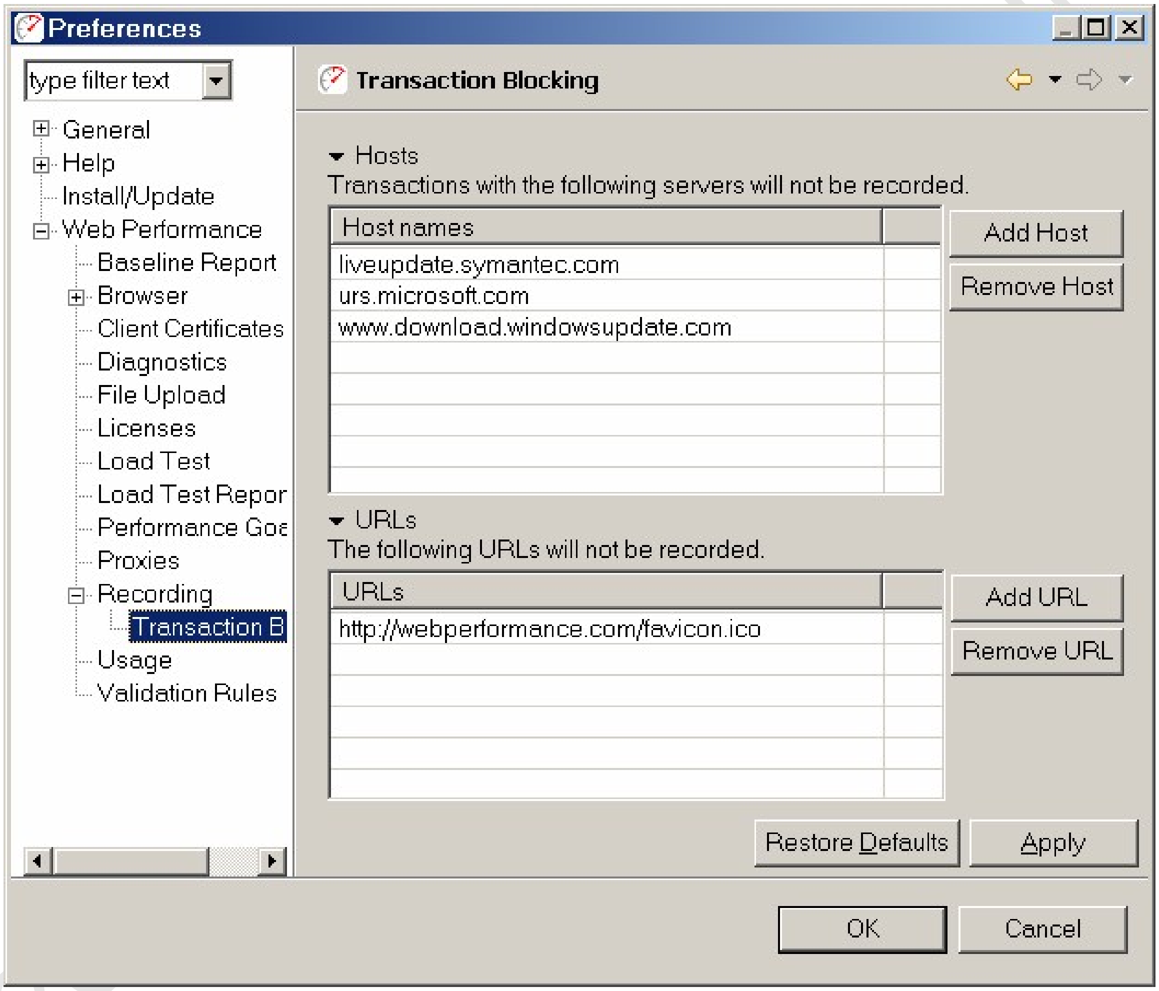
This command would then have to be repeated with different parameters to add more than one virtual device. To make this permanent on a BSD or SysV style system like RedHat you can modify the `/etc/rc.d/rc.local` startup script. For more information please consult the Linux IP Alias mini-HOWTO .

Customizing IP Selections During a Load Test

Once your workstation has been configured with the IP addresses desired, you may also wish to configure the IP aliasing options of Web Performance Suite.

Blocking Undesired Transactions

During recording, users may occasionally notice URLs to sites that are unaffiliated with the Test Case being recorded. These URLs can be triggered by automatic update checks within the web browser, or third party plugins. Alternatively, they might also include hit counters or advertisements that do not affect the actual behavior of the application being tested. All of these URLs can be manually removed in the Test Case Editor, but it may be easier to configure Analyzer to simply ignore all transactions occurring with a given host while testing.
The Transaction Blocking Preferences screen may be accessed by selecting Window → Preferences… and then selecting Web Performance → Recording → Transaction Blocking. To block all transactions with a selected server, simply press "Add Host" (next to the "Hosts" section) and enter the name of the server as it appears in the URL. In order to specify a subdomain, just include a "." in front of the domain name, and all hosts ending with that domain will be blocked. If only blocking for a specific URL is desired, press the "Add URL" option (next to the "URLs" section), and enter the full URL of the resource to be blocked. When finished editing the blocked resources, press "OK" to save the configuration. The rules entered will take affect in future recordings, and transactions matching the rules provided will be omitted from those recordings.



Transaction Blocking

Hosts

Transactions with the following servers will not be recorded.

Host names
liveupdate.symantec.com
urs.microsoft.com
www.download.windowsupdate.com

URLs

The following URLs will not be recorded.

URLs
http://webperformance.com/favicon.ico

Advanced Cookie Handling

In most cases, the default cookie handling used by Web Performance products will operate with your server with practically no manual configuration. If your server sends normal cookies to be relayed back by the user's web browser, then no further configuration is required. Some applications, however, may depend on the end user or their web browser to alter cookies before being sent back to the application. For these scenarios some configuration may be required.

In order to over-ride the default cookie handling, a configuration file must be created. To do this, locate the directory for custom configuration files and create a new file named "cookies.cfg".

Each configuration in this file must start a new line specifying the name of the cookie in the form:

```
cookie1.name=BROWSERSUPPORT
```

Here the name of the cookie being sent back to the server is "BROWSERSUPPORT".

The next line of the file should appear as

```
cookie1.instance=1
```

The instance field here indicates the first usage of this cookie in your recording where the rule should be applied. For example, if default handling should be applied for the first usage, and custom handling from there on, this value could be changed to

```
cookie1.instance=2
```

Lastly, we will want to specify what the value of this cookie should be changed to. There are two possible options:

- To use a fixed value, you may enter the line:

  ```
  cookie1.setFromConstant=supported
  ```

  Here, the text "supported" could be any fixed string that should be used whenever this cookie is being sent back to your server during playback.
- To use a dynamic value loaded from a dataset, use the lines:

  ```
  cookie1.setFromDataset=browser-support-cookies
  cookie1.setFromField=field1
  ```

  Here, the value "browser-support-cookies" is the name of a dataset saved in your repository that will be used by the virtual users, and "field1" is the name of the field within the dataset containing the corresponding values for each cookie.

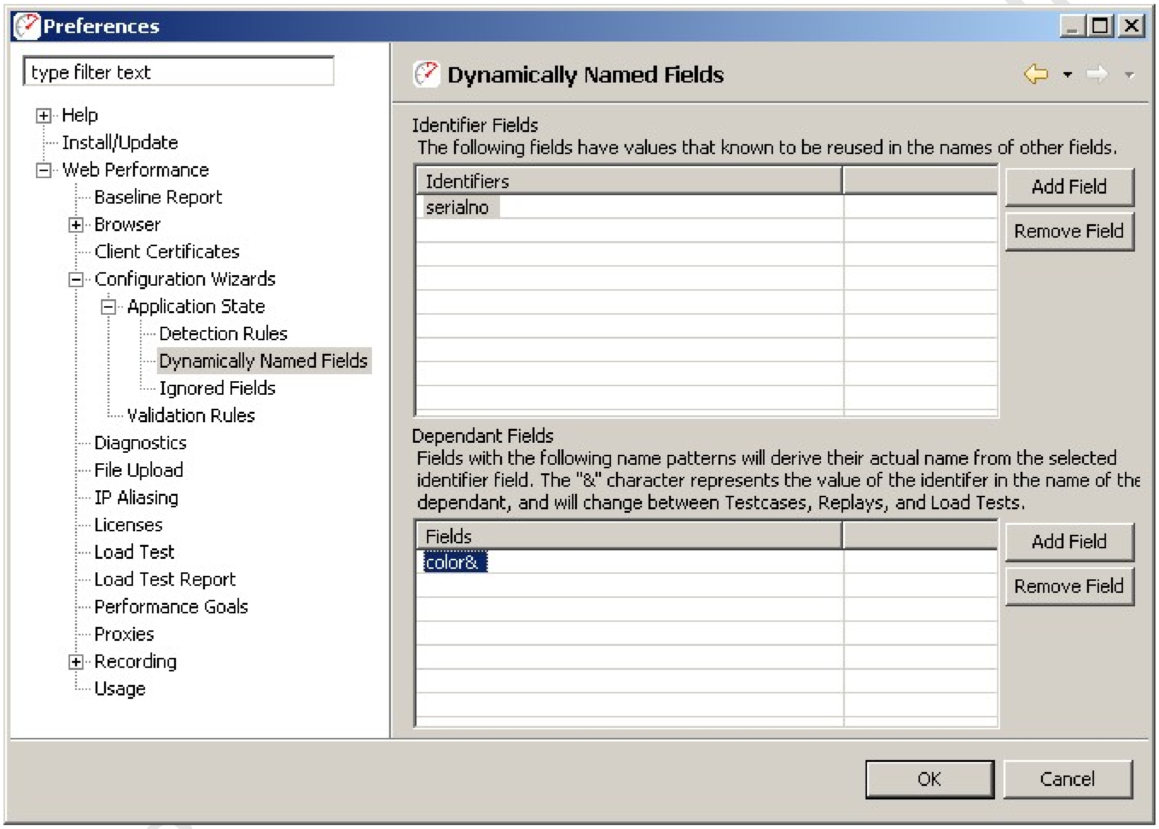
As many cookies as desired may be added into following lines of this configuration file. To add further cookies, rule number must be incremented sequentially for each additional cookie in the file. For example, further cookies rules would start with "cookie2", "cookie3", etc. It is recommended that once this file has been changed, that Web Performance Suite be closed and re-started in order to ensure the changes take full effect.

Dynamically Named Fields

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 configure the "Dynamically Named Fields" preferences how to detect this behavior in your application. This option may be configured through a preference page, accessed by selecting Window→Preferences… and then selecting Web Performance→Configuration Wizards→Application State→Dynamically Named Fields.



Identifier Fields

The following fields have values that known to be reused in the names of other fields.

Identifiers	
serialno	

Dependant Fields

Fields with the following name patterns will derive their actual name from the selected identifier field. The "&" character represents the value of the identifer in the name of the dependant, and will change between Testcases, Replays, and Load Tests.

Fields	
color&	

Configuring these fields is done in two phases. The first is to select the "Add Field" next to the "Identifiers" table, and enter the name of the field that identifies a value. In our example, the identifier is "serialno", whose value will be used later to identify the name of the next variable.

Next, select the field in the Identifiers table to display the dependant fields associated with it, and press the "Add Field" button next to the bottom "Fields" table to create a new dependant field. The name of the variable may be entered here, replacing the dynamic part of the name with an ampersand (&). In our example, the color field would be entered as "color&".

The next time the Application State Management Wizard is run on a testcase, fields starting with the name "color", and ending their name with a value from the field "serialno" will be dynamically renamed when the testcase is replayed or run in a load test.

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 by adding their names to the "Fields" table:

- color&
- weight&_in_lbs
- &_assembly_date

This configuration will allow the Application State Management Wizard to correctly assume name dependencies for all three dependent variables.
It is also permitted for a dynamically named field to be associated with multiple identifiers. For example, consider another case:

```
itemid=123456789
checkbox123456789=checked
legacyid=123
checkbox123=unchecked
```

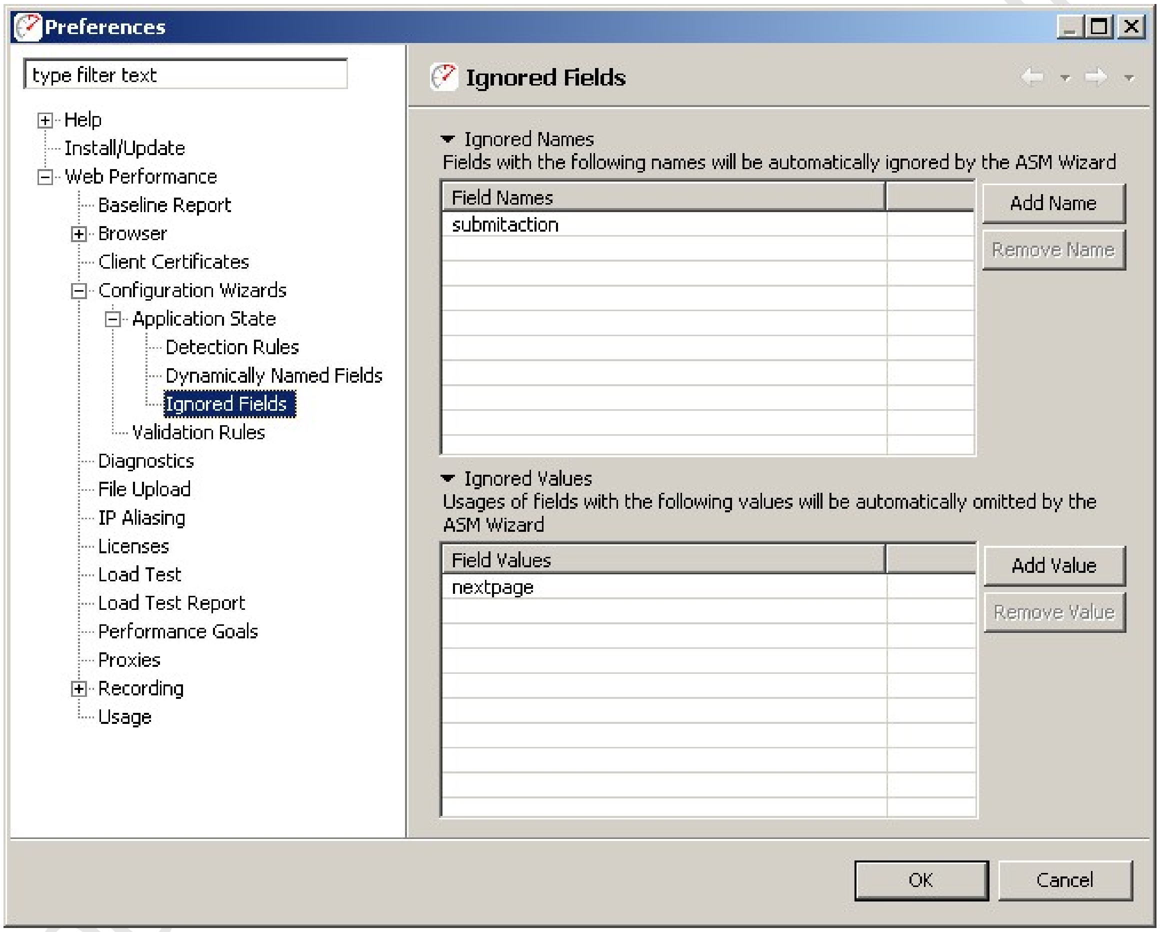
To configure this case, simply create to identifier fields:

- itemid
- legacyid

Next, add the dependant field "checkbox&" to both identifier fields. The Application State Management Wizard will examine both uses of the "checkbox" fields, and only associate dependency when the name of the field corresponds to the value of the identifier. In this example, the wizard will associate the first "checkbox" field as being dependant on "itemid", and associate the second "checkbox" field as dependant on the field "legacyid".

Ignoring Fields in the Application State Management Wizard

The Application State Management 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 Application State Management Wizard offers ignore preferences in order to automatically ignore those fields which are not intended to be treated as dynamic. These preferences may be accessed by selecting Window → Preferences… and then selecting Web Performance → Configuration Wizards → Ignored Fields.



Preferences

type filter text

- Help
- Install/Update
- Web Performance
  - Baseline Report
  - Browser
  - Client Certificates
  - Configuration Wizards
    - Application State
      - Detection Rules
      - Dynamically Named Fields
      - Ignored Fields
    - Validation Rules
  - Diagnostics
  - File Upload
  - IP Aliasing
  - Licenses
  - Load Test
  - Load Test Report
  - Performance Goals
  - Proxies
  - Recording
  - Usage

Ignored Fields

Ignored Names

Fields with the following names will be automatically ignored by the ASM Wizard

Field Names
submitaction

Ignored Values

Usages of fields with the following values will be automatically omitted by the ASM Wizard

Field Values
nextpage

This page contains two lists, one for omitting fields by name, and one for omitting specific uses of a field by their value. 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 Application State Management Wizard manually, or you could specify that this field always be automatically removed with either ignore list

Ignored Names		Ignored Values
btnSubmit	OR	submit

Be very careful not to include a blank entry unless you intend for the Wizard to treat blank values as fixed values as well. The next time you run the Application State Management Wizard, any usage with their name or value specified in one of the ignore lists will be automatically ignored or hidden by the wizard.

Charts & Reports

Viewing Reports

There are three types of reports that can be viewed within the application or opened within the application and sent to an external browser. These reports are:

1. Testcase Report

   This report can be viewed by selecting the Testcase in the Navigator View, then right-clicking to view the menu and selecting the *Open Testcase Report* menu item. It can also be opened from the *Report* item in the Edit menu when a testcase is selected in the Navigator or a testcase editor is active. There is also a toolbar button for opening a report from the active testcase editor and an item in the pull-down menu in the upper-right corner of the editor.
2. Baseline Performance Analysis
   This report can be viewed for a load configuration or for a single testcase (which creates a matching load configuration). From a testcase, it can be opened by selecting the Testcase in the Navigator View, then selecting the *Open Baseline Report* item from the pop-up menu. This will prompt for information required to create a Load Configuration. From a pre-existing load configuration, the report can be viewed by selecting the Load configuration in the Navigator View and selecting the *Open Baseline Report* item from the pop-up menu.

3. Load Test Results reports

   This report can be viewed by either:

   - Selecting the *Report* button from the Load Test Results Editor

   - Selecting the Load Test Result in the Navigator View, then right-clicking to view the menu and selecting the *Open Test Report* menu item.

Printing the Reports

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 such as that provided with Adobe Acrobat. There other programs available - some, such as PDF Creator are available free. Search the Internet for "pdf writer".

Saving charts

A chart image may be saved to a file by launching the report in a browser (using the *Launch* button) and using the browser's *Save Image As...* feature.

Exporting Data

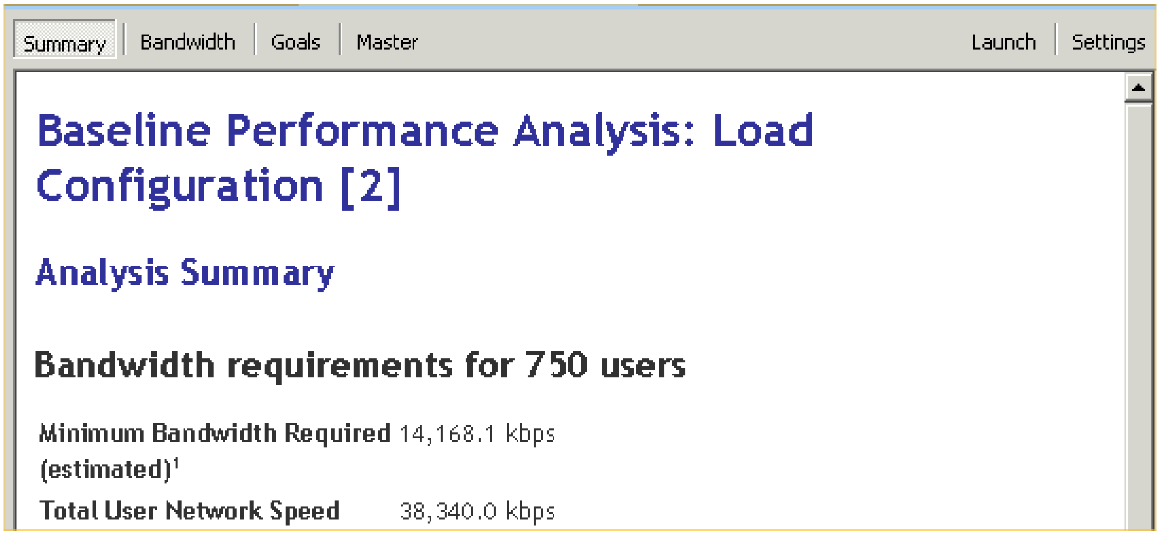
Selecting the *CSV format* link below each table will provide CSV-formatted representation of the data which is importable into most spreadsheet programs, such as Microsoft Office and Open Office.

Testcase Report

The Testcase Report contains 5 sections:

1. Summary - high-level statistics relevant to the entire testcase
2. Pages - detailed statistics about each page, similar to the information displayed in the testcase editor
3. URLs - detailed statistics about each URL in each page, similar to the information displayed in the testcase editor
4. Trend - performance trend plots and detailed statistics for web page durations and sizes (only applicable if one or more replays have been performed)
5. Errors - a list of errors similar to that displayed in the Errors view

Baseline Performance Report



Baseline Performance Analysis: Load Configuration [2]

Analysis Summary

Bandwidth requirements for 750 users

Minimum Bandwidth Required (estimated)[1]	14,168.1 kbps
Total User Network Speed	38,340.0 kbps

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.

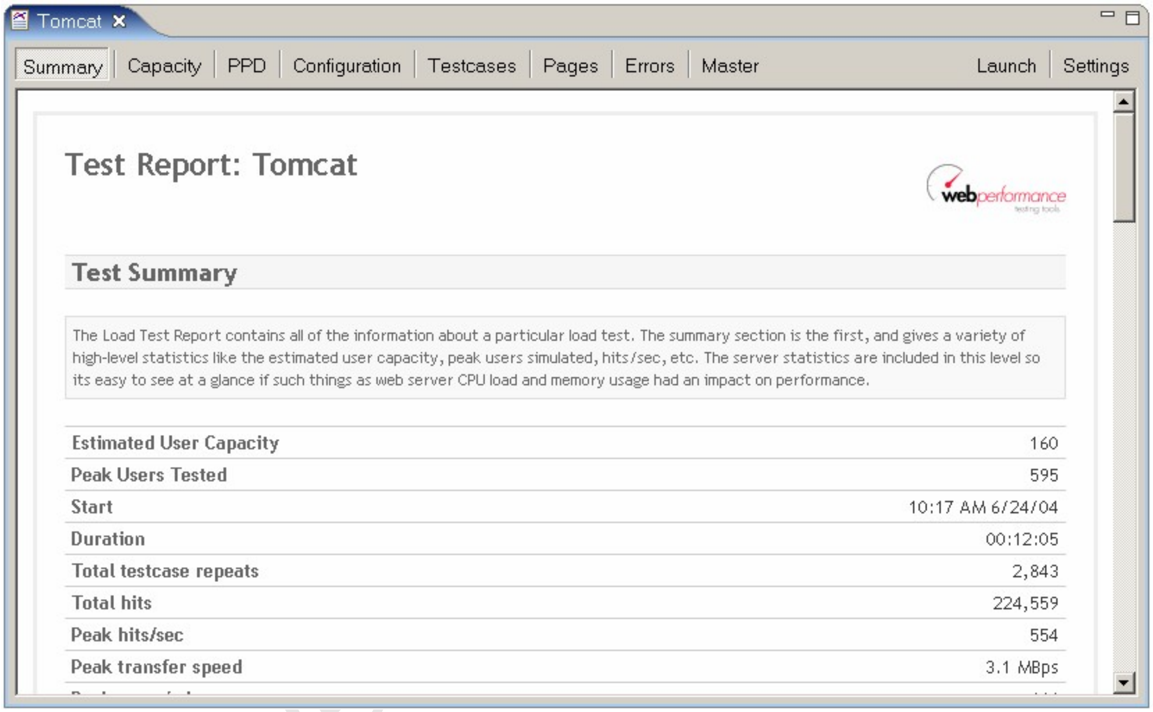
The Analysis Summary gives an overall summary of the report's findings from the two other major sections which look at Performance Goals and Bandwidth Estimates.

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 perform 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.



Test Report: Tomcat

Test Summary

The Load Test Report contains all of the information about a particular load test. The summary section is the first, and gives a variety of high-level statistics like the estimated user capacity, peak users simulated, hits/sec, etc. The server statistics are included in this level so its easy to see at a glance if such things as web server CPU load and memory usage had an impact on performance.

Estimated User Capacity	160
Peak Users Tested	595
Start	10:17 AM 6/24/04
Duration	00:12:05
Total testcase repeats	2,843
Total hits	224,559
Peak hits/sec	554
Peak transfer speed	3.1 MBps

Server Monitoring

Server Monitoring Introduction

Monitoring the server is a critical part of load testing. The Load Tester™ software automatically measures the change in user experience in response to load. However, additional information about the activity on the server is crucial to any diagnosis or tuning of the performance.

The software provides several different options for monitoring the server during a load test. Those methods fall into two categories:

- Basic monitoring
- Advanced monitoring

Basic Server Monitoring

Basic monitoring provides information about the CPU and memory utilization of the server during the test. This can be accomplished in four ways:

1. Server Monitoring Agent - an application which may be downloaded to each server to provide basic monitoring support. Additionally, the agent may be licensed to enable Advanced Server Monitoring.

2. Direct windows monitoring - using the built-in performance measurement APIs, this method can be used to monitor a Windows server as long as the controller (the main GUI of the Load Tester™ software) is also running on Windows and has the necessary authentication credentials
3. Unix/Java monitoring - on any Java-based application server, our WAR file can be deployed to provide the monitoring function
4. HTTP and custom script - using a simple format, any script can return the CPU and memory measurements in a plain text that is understood by the Load Tester™ software.

This monitoring option is included with all Load Tester™ licenses.

More information is available on the Basic Server Monitoring page.

Advanced Server Monitoring

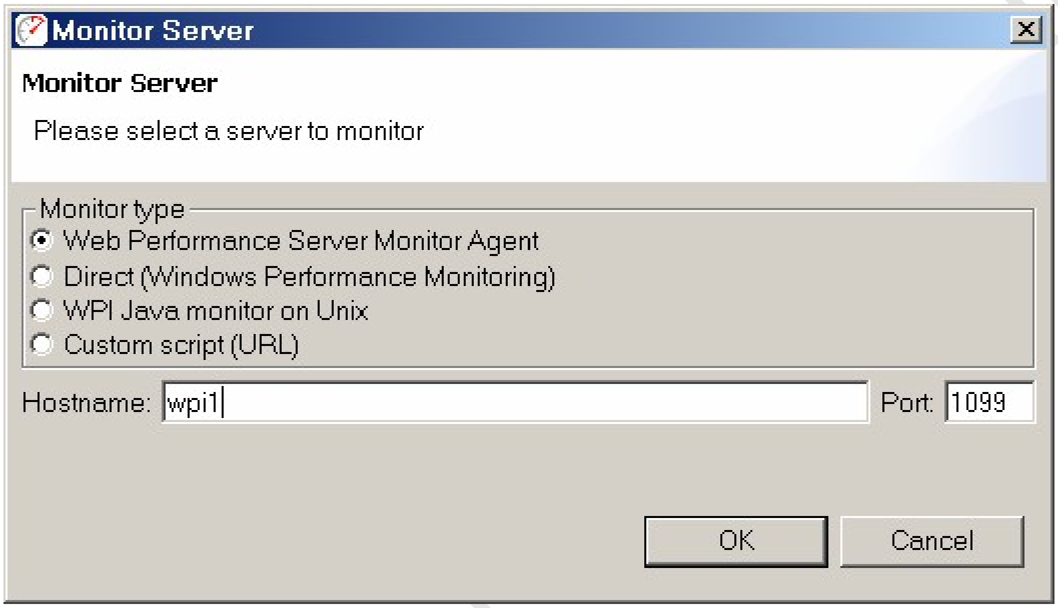
This module, purchased separately, allows Load Tester™ to take more detailed measurements of the server performance. These metrics are listed in the Server Metrics section.

This feature requires installation of a Server Monitoring Agent on each server. Like our remote load engines, this also features auto-detection on the local network, which eliminates the configuration steps required for basic monitoring.

More information is available on the Advanced Server Monitoring page.

Basic Server Monitoring

To start monitoring a new server, simply press the *Add...* button in the Servers View. This dialog will appear:



Monitor Server

Monitor Server

Please select a server to monitor

Monitor type

● Web Performance Server Monitor Agent
○ Direct (Windows Performance Monitoring)
○ WPI Java monitor on Unix
○ Custom script (URL)

Hostname: wpi1 Port: 1099

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

1. Web Performance Server Monitor Agent - The downloadable server monitor agent (available for Windows or Linux) provides basic monitoring of CPU and Memory, with an option for Advanced monitoring of vital server performance data. Please see the Server Monitoring Agent for further information.
2. 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).
3. 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.
4. 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).

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.

Server Configuration

Web Performance Server Monitor Agent

This option may be used to connect to an installed and running copy of the Web Performance Server Monitoring Agent. Typically, the agents are automatically detected and enabled. However, depending on network conditions, it may be necessary to add the configuration manually by entering either the IP Address or hostname of the server on which the agent is running.

If the agent has been configured to use a specific port, then the specified value of the RmiRegistryPort should be entered in the Port option. Otherwise, the default port is 1099.

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.

> note: the WPIMonitor.war file can be found in the product installation folder.

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.

Advanced Server Monitoring And Analysis

With no license installed, the Server Monitoring Agent will collect CPU% and Memory% metrics -- similar to Basic Server Monitoring. To unlock the full potential of the agent, you must purchase and install a license for the Advanced Server Analysis™ module.

Installing the License

To install a license follow these steps:

1. Open Web Performance Load Tester
2. Select the Servers view
3. Select the server where the license should be installed
4. Select the "Licensing..." button
5. Follow the wizard to completed activation and installation of the license key

Note that the procedures for activating, deactivating and moving licenses for the server monitoring agent is similar to the procedures for Load Tester. Please consult the License Key Activation page for more details.

Viewing Collected Data

The data collected by the monitoring agent is automatically integrated into the test results.

In the Load Test Report, select the *Servers* section for charts and tables detailing the server performance as well as a performance checklist.

In the Statistics View, use the drop-down controls to select the server and the metrics will be presented in the table.

Server Metrics

Note that not all metrics are available on all operating systems.

CPU group

CPU % (Processor Time for all processors) - the percentage of elapsed time that the processor spends executing non-Idle threads. This metric is the primary indicator of processor activity, and displays the average percentage of busy time observed during the sample interval. This metric is expected to increase proportionally to the load applied to the system.

Context switches/sec - the rate of switches from one thread to another. Thread switches can occur either inside of a single process or across processes. A thread switch can be caused either by one thread asking another for information, or by a thread being preempted by another, higher priority thread becoming ready to run. This metric is expected to increase proportionally to the load applied to the system.

Process Queue Length - the number of processes waiting to be scheduled to run. On windows, a sustained processor queue of less than 10 threads per processor is normally acceptable, dependent on the workload.

Memory group

% Memory - The percentage of available memory that is in use. Large values of this metric suggest that the frequency of page swaps to disk will be high. For Windows servers, this is the percentage of virtual memory currently committed. For Linux servers, this is the percentage of physical memory in use (non free, cached or buffered).

Page reads/sec - the rate at which the disk was read to resolve hard page faults. Hard page faults occur when a process references a page in virtual memory that is not in working set or elsewhere in physical memory, and must be retrieved from disk. This counter is a primary indicator of the kinds of faults that cause system-wide delays. It includes read operations to satisfy faults in the file system cache (usually requested by applications) and in non-cached mapped memory files. Large increases in this metric can degrade system performance. Increasing physical memory can alleviate the problem).

Page Writes/sec - the rate at which pages are written to disk to free up space in physical memory. Pages are written to disk only if they are changed while in physical memory, so they are likely to hold data, not code. Large increases in this metric can degrade system performance. Increasing physical memory can alleviate the problem.

Disk group

% I/O Time Utilized - The percentage of time during the sample interval that the disk was executing I/O.

Service time: The average amount of time required for each I/O transfer.

Reads/sec - the rate of read operations on the disk. A plateau in this metric could indicate a performance bottleneck.

Writes/sec - the rate of write operations on the disk. A plateau in this metric could indicate a performance bottleneck.

Queue Length - The average number of write requests that were queued for the disk during the sample interval.

Network group

Packets Received/sec - the rate at which packets are received on the network interfaces. This metric is expected to increase proportionally to the applied load. A greater-than-linear increase could indicate less efficient operation of the network. A less-than-linear increase indicates a limitation of network and/or server capacity.

Packets Sent/sec - the rate at which packets are sent on the network interfaces. This metric is expected to increase proportionally to the applied load. A greater-than-linear increase could indicate less efficient operation of the network. A less-than-linear increase indicates a limitation of network and/or server capacity.

Bytes received/sec - the rate at which data is received on the network interfaces. This metric is expected to increase proportionally to the applied load. A greater-than-linear increase could indicate less efficient operation of the network. A less-than-linear increase indicates a limitation of network and/or server capacity.

Bytes sent/sec - the rate at which data is sent on the network interfaces. This metric is expected to increase proportionally to the applied load. A greater-than-linear increase could indicate less efficient operation of the network. A less-than-linear increase indicates a limitation of network and/or server capacity.

Packets Received Errors - the number of inbound packets that contained errors preventing them from being deliverable to a higher-layer protocol. These errors are considered a serious network degradation.

Packets Sent Errors - the number of outbound packets that contained errors preventing them from being deliverable to a higher-layer protocol. These errors are considered a serious network degradation.

Collisions/sec - the rate at which outgoing ethernet packets must be re-transmitted. When this metric exceeds 5% of packets sent/sec, this indicates a network problem or network capacity limit.

Connections Established - the number of TCP connections for which the current state is either ESTABLISHED or CLOSE-WAIT. This metric is expected to increase proportionally to the load applied to the system.

Connection Failures - the number of times TCP connections have gone from SYN-SENT or SYN-RCVD to CLOSED.

TCP Segments Retransmitted - the number of TCP segments which were previously transmitted, but had to be retransmitted again.

External Evaluation

Developers or advanced users who need to see sources that these metrics are measured from should refer to the Locating Server Metric Counters section.

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.

note: the CER and PEM certificate files may be copied directly from the following folder (where <user> is your windows username) if the download links do not work:

C:\Documents and Settings\<user>\.webperformance

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.

Manual Browser Configuration

Do you have a VPN active? If yes, be sure to perform the VPN configuration steps at the end.

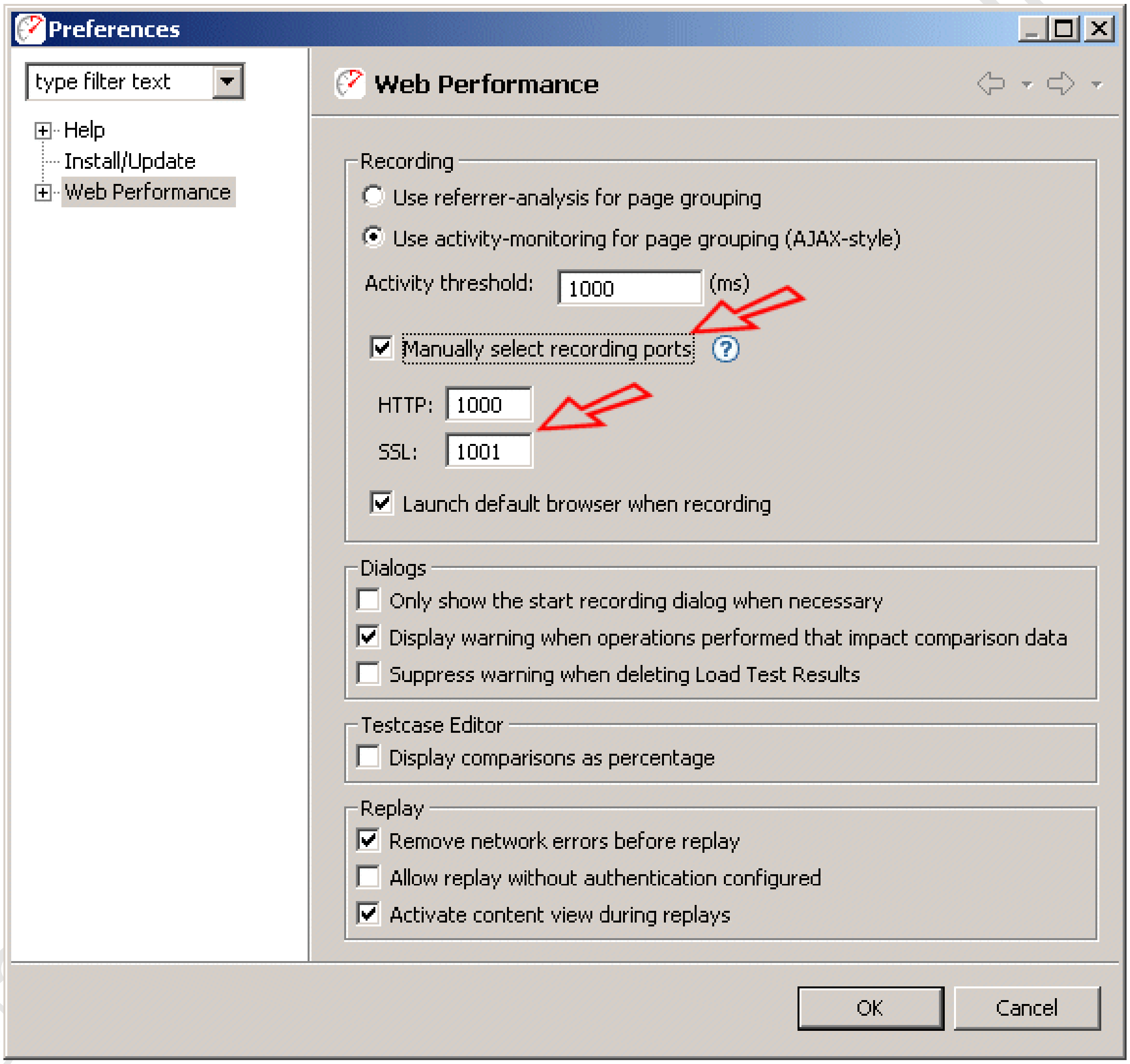
Under the most common configurations, Web Performance software will automatically detect and configure the default browser for your platform (IE on Windows, Firefox on Linux/Unix, Safari on Mac OSX). When it does not, or the configuration is unusual, it may be necessary to configure the browser manually.

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



Preferences

type filter text

- Help
- Install/Update
- Web Performance

Web Performance

Recording

○ Use referrer-analysis for page grouping

● Use activity-monitoring for page grouping (AJAX-style)

Activity threshold: 1000 (ms)

☑ Manually select recording ports

HTTP: 1000

SSL: 1001

☑ Launch default browser when recording

Dialogs

□ Only show the start recording dialog when necessary

☑ Display warning when operations performed that impact comparison data

□ Suppress warning when deleting Load Test Results

Testcase Editor

□ Display comparisons as percentage

Replay

☑ Remove network errors before replay

□ Allow replay without authentication configured

☑ Activate content view during replays

Step 2 - Configure the browser

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

Manual configuration instructions are available for these browsers:

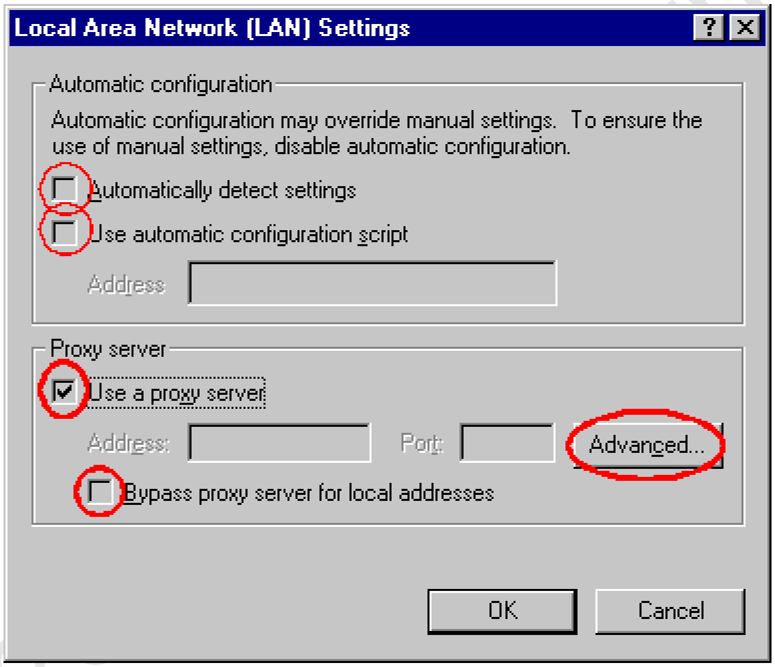
- Internet Explorer (6)
- Firefox (1.5) - also applies to Mozilla (1.x) and Netscape (6 and up)

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



Local Area Network (LAN) Settings

Automatic configuration

Automatic configuration may override manual settings. To ensure the use of manual settings, disable automatic configuration.

□ Automatically detect settings

□ Use automatic configuration script

Address

Proxy server

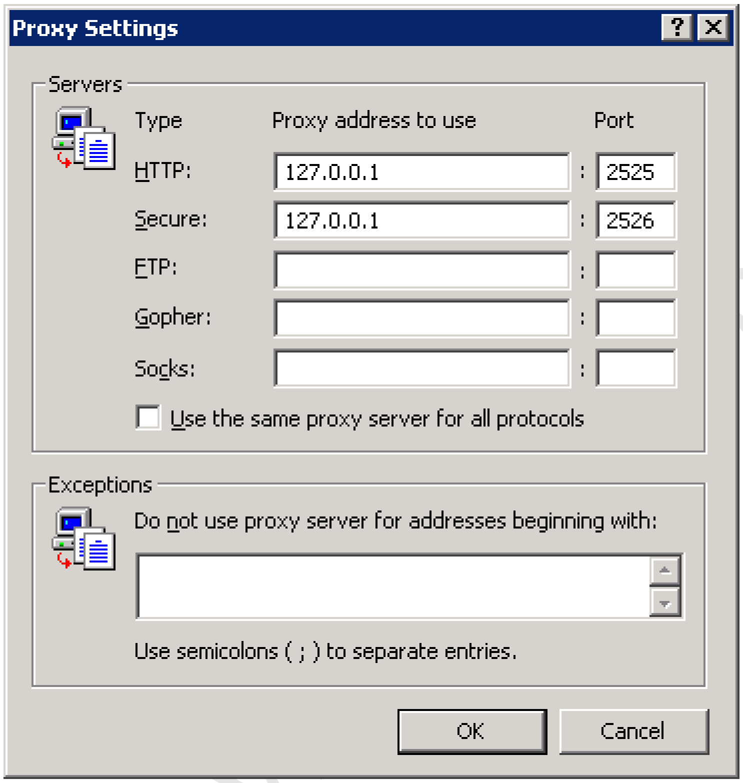
☑ Use a proxy server

Address: Port: Advanced...

□ Bypass proxy server for local addresses

OK Cancel

- 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



Proxy Settings

Servers

Type	Proxy address to use	Port
HTTP:	127.0.0.1	2525
Secure:	127.0.0.1	2526
FTP:		
Gopher:		
Socks:		

□ Use the same proxy server for all protocols

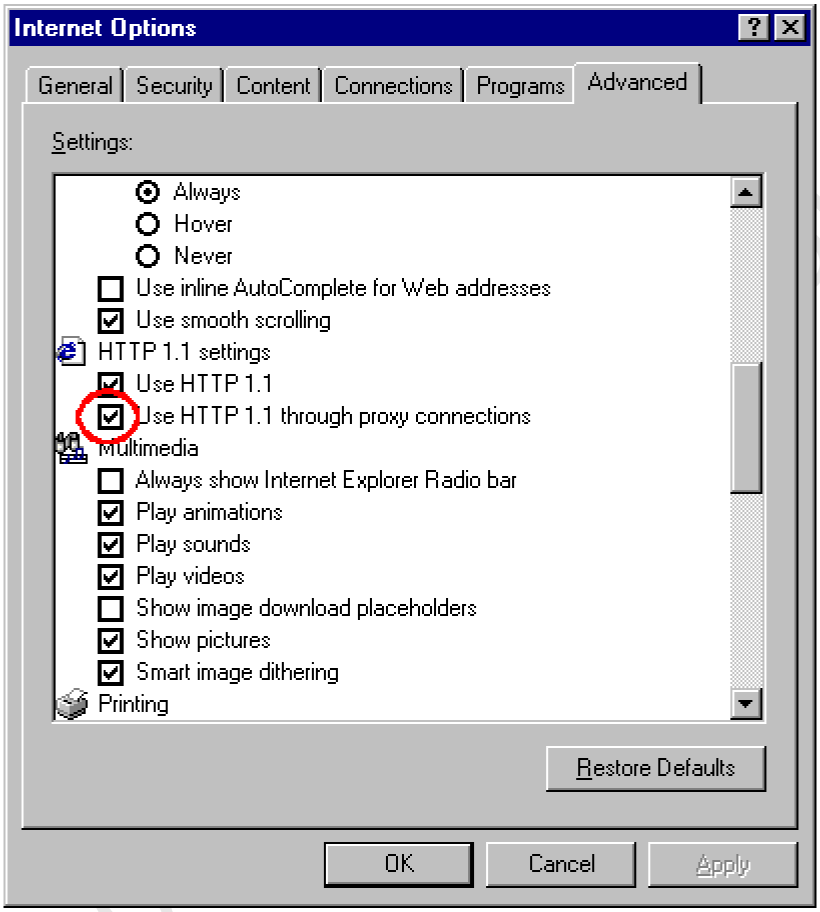
Exceptions

Do not use proxy server for addresses beginning with:

Use semicolons (;) to separate entries.

- On the *Proxy Settings* dialog, IE must be provided with the address the analyzer is configured to listen to, which 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.



● Always
○ Hover
○ Never
□ Use inline AutoComplete for Web addresses
☑ Use smooth scrolling

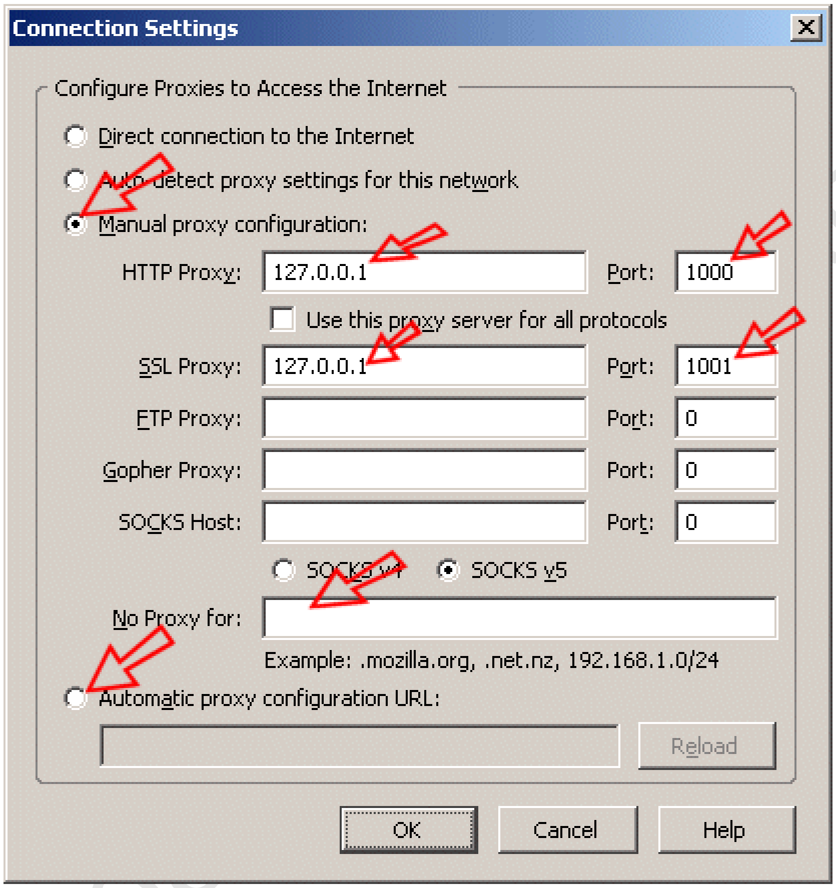
HTTP 1.1 settings
☑ Use HTTP 1.1
☑ Use HTTP 1.1 through proxy connections

Multimedia
□ Always show Internet Explorer Radio bar
☑ Play animations
☑ Play sounds
☑ Play videos
□ Show image download placeholders
☑ Show pictures
☑ Smart image dithering

- 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)



Connection Settings

Configure Proxies to Access the Internet

○ Direct connection to the Internet

○ Auto-detect proxy settings for this network

● Manual proxy configuration:

HTTP Proxy: 127.0.0.1 Port: 1000

□ Use this proxy server for all protocols

SSL Proxy: 127.0.0.1 Port: 1001

FTP Proxy: Port: 0

Gopher Proxy: Port: 0

SOCKS Host: Port: 0

○ SOCKS v4 ● SOCKS v5

No Proxy for:

Example: .mozilla.org, .net.nz, 192.168.1.0/24

○ Automatic proxy configuration URL:

- 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

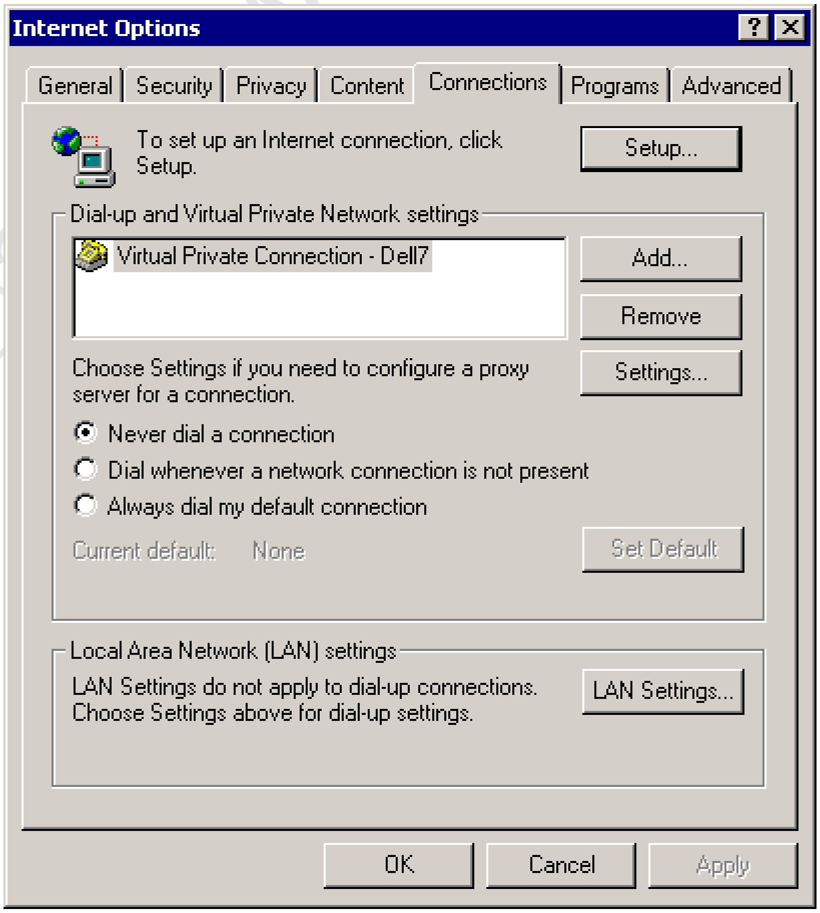
After these configuration steps are finished, press the refresh button in your browser to retry the diagnostic page. If the URL of the diagnostic page is no longer in the URL field, you may enter this:

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

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

VPN and modem configuration

If your Windows computer is connected to the internet via an ISDN modem, phone-based modem or VPN there is an extra configuration step that must be completed. Unfortunately the normal Windows network settings are ignored when the internet connection is made via these methods and there is a simple change that must be made before and after using Web Performance Analyzer™.
To tell if your computer requires this extra step bring up Internet Explorer and bring up the Internet Options Dialog. (select the *Tools->Internet Options* menu item). Click on the *Connections* tab to examine the network configurations:



To set up an Internet connection, click Setup.

Dial-up and Virtual Private Network settings

Virtual Private Connection - Dell7

Choose Settings if you need to configure a proxy server for a connection.

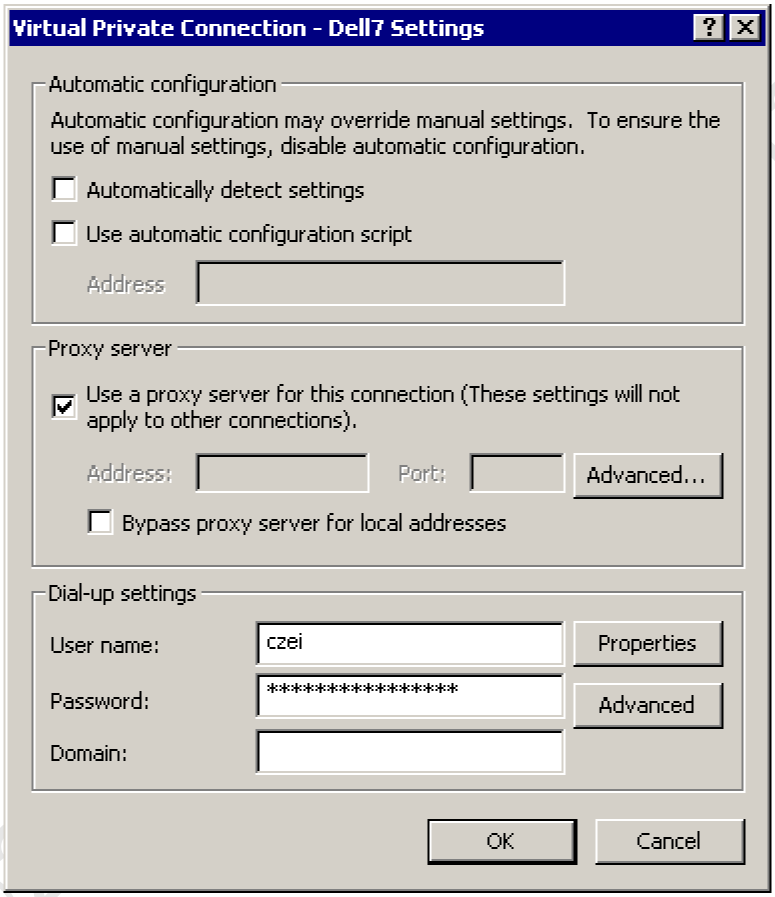
● Never dial a connection
○ Dial whenever a network connection is not present
○ Always dial my default connection

Current default: None

Local Area Network (LAN) settings

LAN Settings do not apply to dial-up connections. Choose Settings above for dial-up settings.

If extra configuration is needed you will see an entry in the Dial-up and Virtual Private Network settings. Select the dial-up or VPN connection you are using and push the *Settings* button:



Virtual Private Connection - Dell7 Settings

Automatic configuration

Automatic configuration may override manual settings. To ensure the use of manual settings, disable automatic configuration.

☐ Automatically detect settings

☐ Use automatic configuration script

Address

Proxy server

☑ Use a proxy server for this connection (These settings will not apply to other connections).

Address: Port:

☐ Bypass proxy server for local addresses

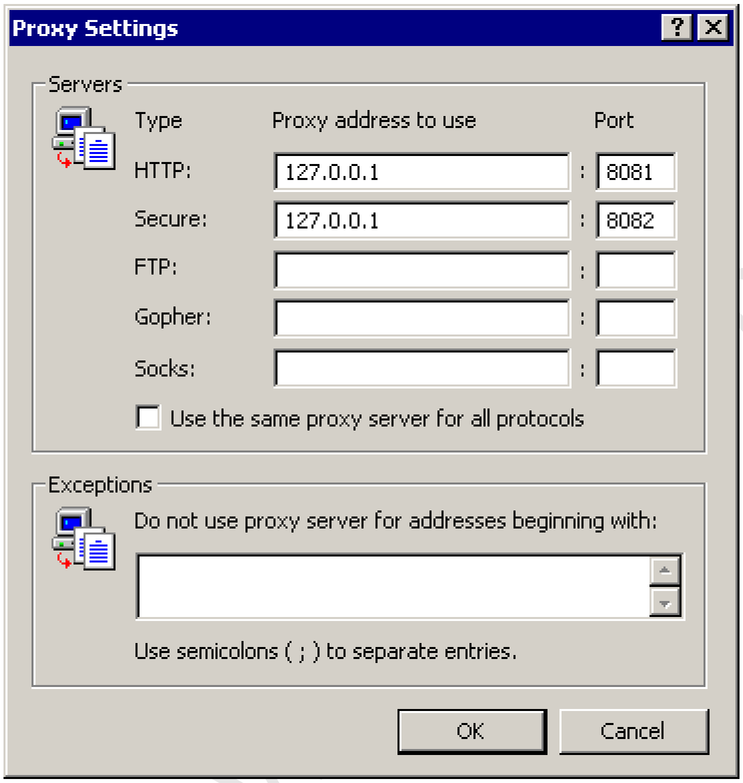
Dial-up settings

User name: czei

Password: ****************

Domain:

Make sure the "Use a proxy server for this connection" is checked, and then click on the *Advanced* button:



Proxy Settings

Servers

Type	Proxy address to use	Port
HTTP:	127.0.0.1	8081
Secure:	127.0.0.1	8082
FTP:		
Gopher:		
Socks:		

☐ Use the same proxy server for all protocols

Exceptions

Do not use proxy server for addresses beginning with:

Use semicolons (;) to separate entries.

If you are using the default settings use the loopback address 127.0.0.1 and the port numbers as shown in the Web Performance Preferences, in Step 1.

Server Agents

The Web Performance Load Engine is an optional component of the Web Performance Suite software that monitors key performance metrics on a server during a load test. These measurements are recorded, cached (in case of server crash) and sent to the controller during and after a load test for inclusion in the Load Test Report.

Installing a Server Agent

Installers for Windows and Linux 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:

```
ServerAgent_Linux_3.0.bin –i console
```

Starting a Server Agent

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

The server agent is started on Linux and UNIX platforms using the installed shortcut or the startup script:

/usr/local/bin/WebPerformanceSuite_N.N/ServerAgent

Configuring a Load Engine

Network and Firewall Considerations

In many cases, the engine and controller will be able to immediately connect to one another, and will even automatically detect one another if they are both connected to the same local LAN. This section outlines the configuration options available for the engine, should an error be received while attempting to connect to it.

In order to connect to a Load Engine, the engine will need to be able to accept connections from the controller on at least one network interface. The IP address and port number used to accept connections may be controlled by using a plain text editor to edit the file "system.properties" (creating it if it does not exist), located in the directory where the engine was installed. The following lines may be added:

```
EngineRMIAddress=192.168.1.62
RmiRegistryPort=1099
RmiEnginePort=1100
```

These values have the following effect:

EngineRMIAddress

Controls which IP address the engine will accept incoming connections from. If set, the engine will accept connections from the controller only through the specified IP address. By default, the engine will accept connections through any of it's available IP addresses. However, setting this field may eliminate connection problems; particularly if the engine has IP addresses from different protocols such as IPv4 and IPv6.

RmiRegistryPort

Controls which port the engine will accept incoming connections from. If this field is omitted, it will default to using port 1099.

RmiEnginePort

Controls which port the engine will use to communicate information with the controller once connected. If this field is omitted, it will default to using any available port.

Additionally, it may be necessary to specify the public IP address from which the engine will be accessed, especially if that IP address is assigned to a NAT or firewall device other than the machine running the engine application. This may be specified by editing the file "Load Engine.lax", and adding the line

```
lax.nl.java.option.additional=-
Djava.rmi.server.hostname=site.mycompany.com
```

Once the settings have been entered and saved into the engine's configuration file, the engine may be restarted in order to pick up the correct settings.

Accessing an Engine behind a Firewall

In order to access an engine behind a firewall, it may be necessary to configure the port settings used by the engine for accessibility. The RmiRegistryPort and RmiEnginePort should be set, and the firewall should be configured to allow connections through these ports. For more information on configuring your firewall, please contact your network administrator.

After configuring the ports and starting the engine, the engine is ready to be added to the controller. The Engines View may be used to add a remote engine. When prompted, the IP address should be an address the controller can connect directly to. If your firewall uses a NAT, then this is the IP address of the firewall; otherwise it is that of the engine itself. The port option should reflect the value of the RmiRegistryPort configured on the engine.

Further Configuration

Once the controller has been able to successfully connect to a server agent, the agent may be managed through the Servers View.

Advanced Configuration

Setting user levels

In some cases, it may be desirable to manually determine how many VUs will run on each engine, rather than allowing automatic distribution. However, since overloading an engine can result in collection of load test data that may be invalid, it would be unwise to override the distribution completely.

Each remote load engine has a configuration file (system.properties) in the installation folder. in this file, add this line:
  EngineUserCapacity=NN
(for the local load engine, see the Configuration Files section for the location of the system.properties file)

Change the value of NN to reflect the number of total users that is desired for this engine. Repeat this for each engine and restart the engines. This setting will place a maximum limit on the number of users the engine will report that it can support. Since the controller will not allocate more than the reported capacity to each engine, manipulating these settings will allow manual selection of the distribution.

Note that the capacity of the engines will always be reported as "100" when a test is not running. Also, the reported capacity may be lower than configured, based on the CPU and memory utilization as analyzed by the overload prevention feature.

Command Line Tools

Replay a testcase

Testcases in a repository may be replayed from the command line. A command-line interface launcher (cli) is included in the installation folder (*cli.exe* on Windows).

Replay command-line format:

```
usage: cli replay [options] repository
-t testcase to replay
-w workspace location (default is <home>/WebPerformance)
```

If a testcase is provided (-t option), only that testcase will be replayed. If omitted, all testcases in the repository will be replayed. After completion of the last replay, the repository will be saved.

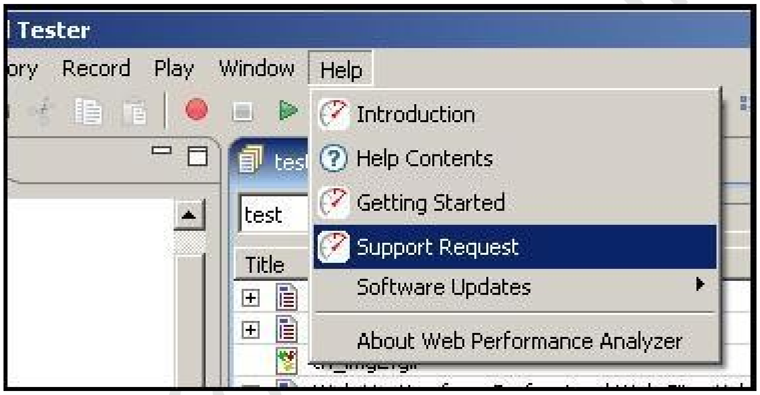
Return codes:

```
0-N - The number of errors encountered during the replays.
-1 - Illegal arguments provided
-2 - Software error occurred (send cli.log to Web Performance Support)
```

Support Request

The Support Request is used to send a question or issue to the Web Performance Support team, or attach files to an existing issue.

The Support Request Form is available from the *Support Request* item on the *Help* menu.



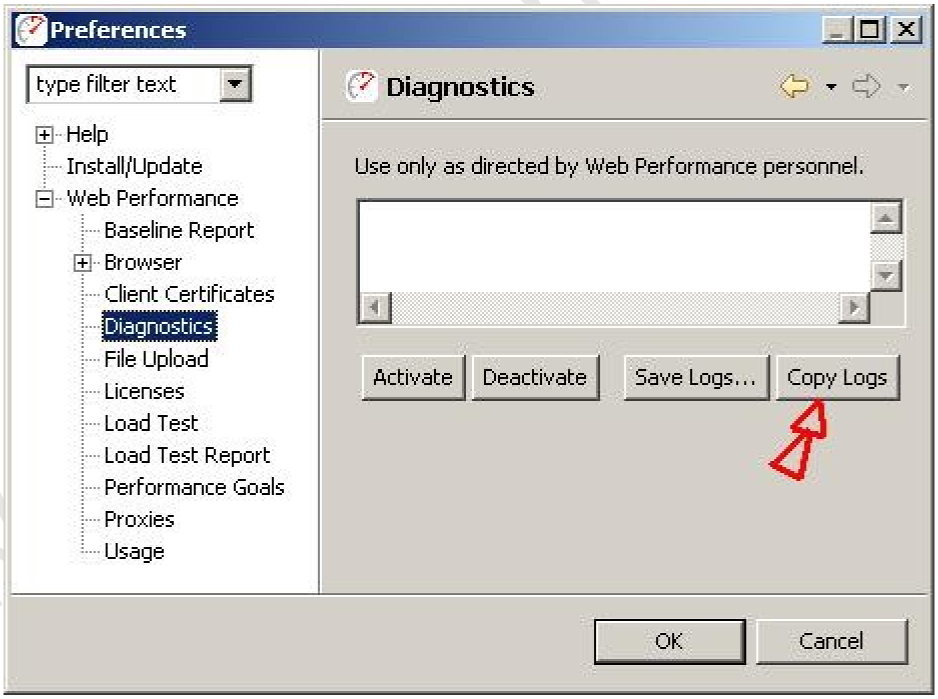
On the first page of the support wizard, you choose to create a new issue or attach files to an existing request. If attaching files to an existing issue, use the exact number received in the e-mail when the issue was created (e.g. WPA-111 or WPL-222). Once an e-mail address is entered, you may proceed to describing the issue and attaching files to the support request.
If an error occurs while sending the request, please visit our website and manually submit the form as described in the next section

Manual support request submission

Support request can be submitted on the Support Section of our website. You must create an account in order to submit a request. Once you have an account, login and select the *Create New Issue* item at the top of the screen. Fill in the required fields and select the *Create* to finish the request. When submitting the request, please update the *Description* to include either:

- The information from the support wizard if the tool failed to send the support request.
- The information from the Diagnostics Preference page:

Go to Window->Preferences and select the *Diagnostics* tab. Click the *Copy Logs* item to put the requested information into the system clipboard, and paste it into the support request. (Or use the *Save Logs* item to save the logs to a file and attach them to the support request once it has been created.)



Use only as directed by Web Performance personnel.

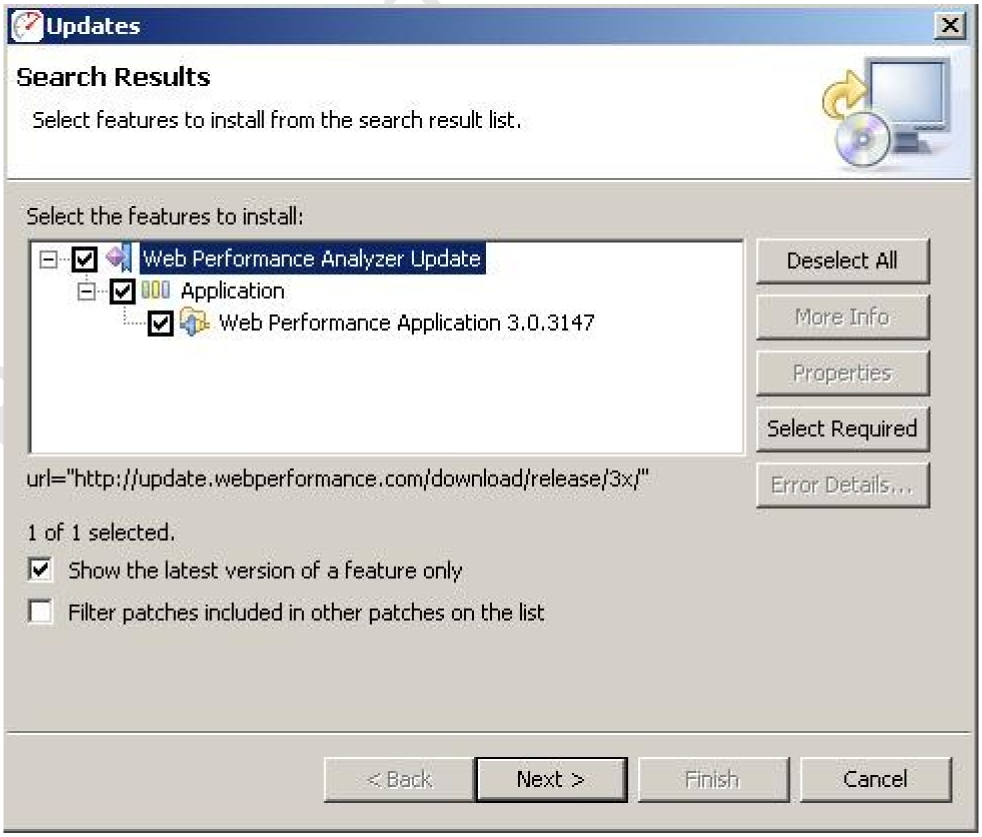
Managing software updates

Retrieving patch updates

For service (patch) updates, the update manager is used to find and install the new software. To view and install patches available, follow these steps:

1. From the *Help* menu, select *Software Updates...*

2. Select the *Search for updates...* item and then press *Finish*



Updates

Search Results

Select features to install from the search result list.

Select the features to install:

- ☑ Web Performance Analyzer Update
  - ☑ Application
    - ☑ Web Performance Application 3.0.3147

url="http://update.webperformance.com/download/release/3x/"

1 of 1 selected.

☑ Show the latest version of a feature only

☐ Filter patches included in other patches on the list

3. If there are service updates available for your software, the results are displayed (as shown). To install the update, make sure the version is checked then follow the wizard instructions.

Note that all file downloads are performed in the background and there is no status displayed. You may continue to work in the product until the download has completed and the update wizard re-appears.

4. Follow the instructions on the next few wizard pages to complete the installation.

Configuring Updates to upgrade to new minor versions

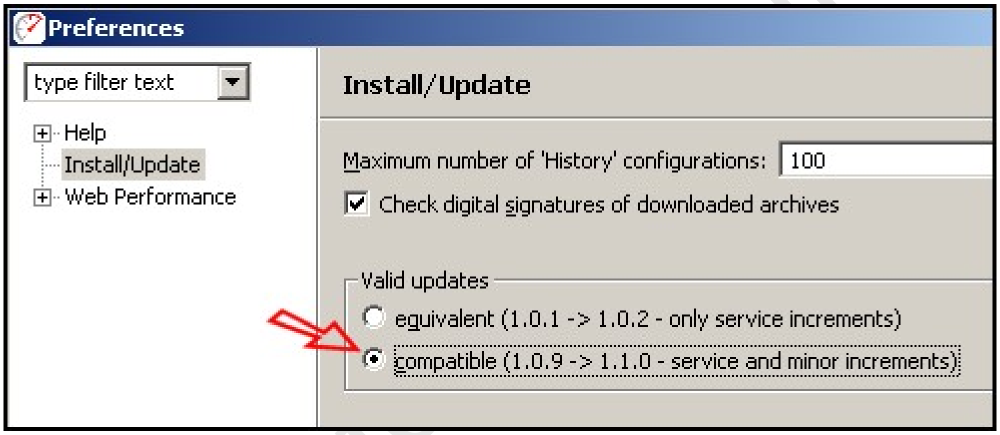
By default, the software only downloads patches. This prevents accidental upgrades to a version of the software for which the installed license is not valid. Once this happens, a re-install is required to get back to the previous version.

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

A new license is required with minor version upgrades. If you have not requested a new license and wish to upgrade your software, see our licensing information web page before continuing.

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*.



Install/Update

Maximum number of 'History' configurations: 100

☑ Check digital signatures of downloaded archives

Valid updates

○ equivalent (1.0.1 -> 1.0.2 - only service increments)

● compatible (1.0.9 -> 1.1.0 - service and minor increments)

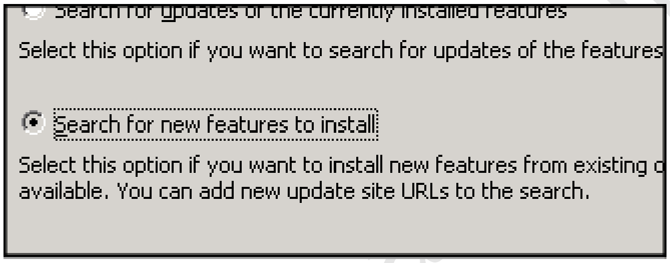
Once the update manager has been configured, follow the procedure described under *Retrieving patch updates* to view and install minor version updates. After installation of the new software, install the new license (see License Management for detailed instructions).

Updating from a new Update Site

If you have been instructed to download a new version of the software from a new update site, the update manager is used to configure the application to use the new site in the search for software changes.

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

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



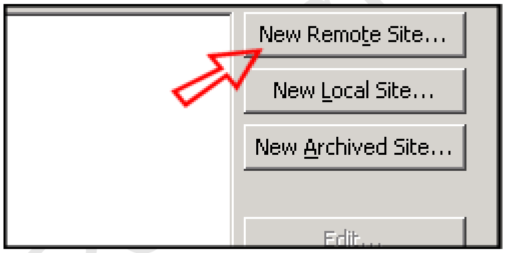
○ Search for updates of the currently installed features

Select this option if you want to search for updates of the features

● Search for new features to install

Select this option if you want to install new features from existing o
available. You can add new update site URLs to the search.

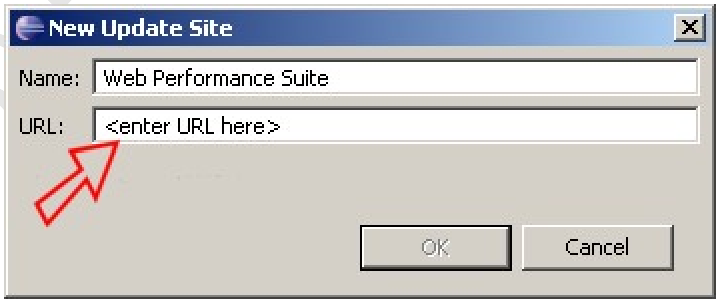
3. Select *New Remote Site...*



4. Enter "Web Performance Suite" for the *Name* and the new update site for the *URL*

Select *OK*

Select *Finish*



Name: Web Performance Suite

URL: <enter URL here>

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\<username>".

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\Web Performance Suite <version>") 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\<username>\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.

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.
- In the Status View, select the *Garbage Collection* icon.
- Run the Repository Cleanup Wizard
- Change the Load Test Settings to disable collection of URL Statistics when running a Load Test
- Check your operating system to see if there are any background processes taking up memory.
- Try increasing the memory available to Web Performance Suite (See below).

Stand-alone program

The default setting for the maximum amount of memory allocated to Web Performance Suite is 200MB (the default). The program will encounter errors if this limit is reached. To increase this value, use the following steps:

1. Locate the file "*webperformance.ini"* in the directory where you installed the program.

2. Create a backup copy of the file.
3. Edit the file with a plain-text editor and locate the lines:

```
-Xms200m
-Xmx200m
```

4. Change the "200" in each line to the desired amount (in MB) of memory to use, and save the file.

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

Configuring your computer for Multiple IP Addresses

This section covers how to configure your computer to generate virtual users from more than one IP address. This is only needed if your web application makes use of the client's IP addresses, which is quite rare, or if a piece of hardware such as load balancer uses client IP addresses. The concepts behind networks and which IP addresses are valid for your network are beyond the scope of the manual. **Please consult with your network administrator before going any further.** The following modifications have a high probability of rendering your computer inoperable if done incorrectly.

Do not use this configuration unless you are sure it is required!

An IP address is intended to identify the source of a computer's network traffic, and is used to route network packets on a network. By default virtual users will originate from the IP address of the computer running Web Performance Suite, but there are reasons why you may want virtual users to each have their own IP address. For example, some hardware load balancing devices use the IP address to route packets to different computers.

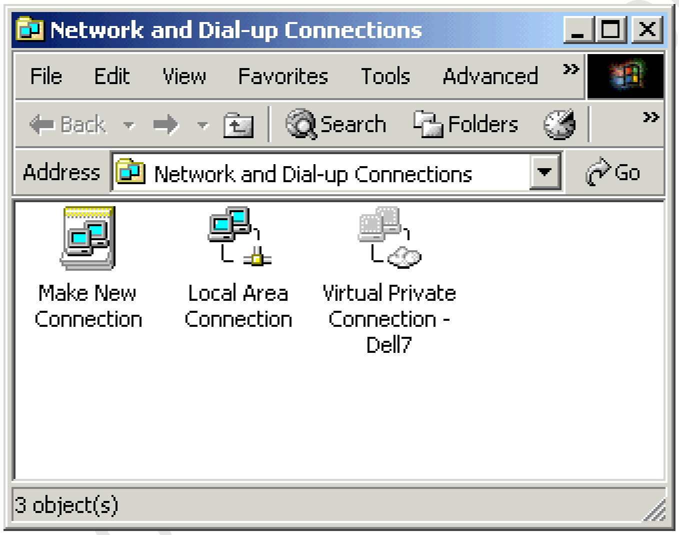
By default Web Performance Suite will use the Operating System to select an available network address, but at most you might have four network cards, which is not nearly enough to give every virtual user its own IP address. To get past this limitation the multiple IP address feature uses the ability of your operating system to configure virtual network devices. When it starts, Web Performance Suite will create a list of all real and virtual network devices. During a performance test as each virtual user is created it will be assigned a new IP address; if there are more users than IP addresses, the virtual users will grab an IP address from the front of the list.

The use of multiple IP addresses will also work if you have multiple playback engines, but you must configure virtual network devices on each computer separately.

The following sections describe how to configure virtual network devices on the different operating systems. Note that this feature of Web Performance Suite makes use of the built-in feature of your operating system to configure virtual network devices, and the complicated setup procedure is required by the operating system.

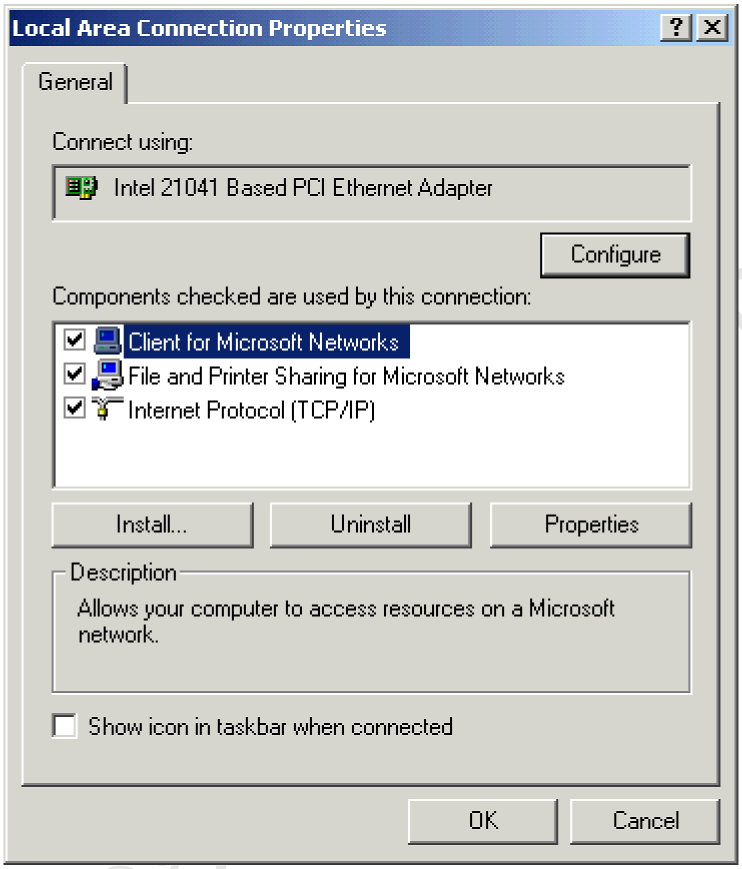
Windows

To configure a Windows machine to use multiple virtual IP addresses for right-click on *My Network Places* (on the *Desktop*) or execute *Start->Control Panel*, and double click on *Network and Dial-up Connections*:



Note that a limitation of Windows is that you can only configure virtual network devices using a Local Area Connection; VPNs and ISDN or other modem connections do not have this ability.

The next step is to edit the properties of your network connection, bringing up the following dialog:



Connect using:

Intel 21041 Based PCI Ethernet Adapter

Components checked are used by this connection:

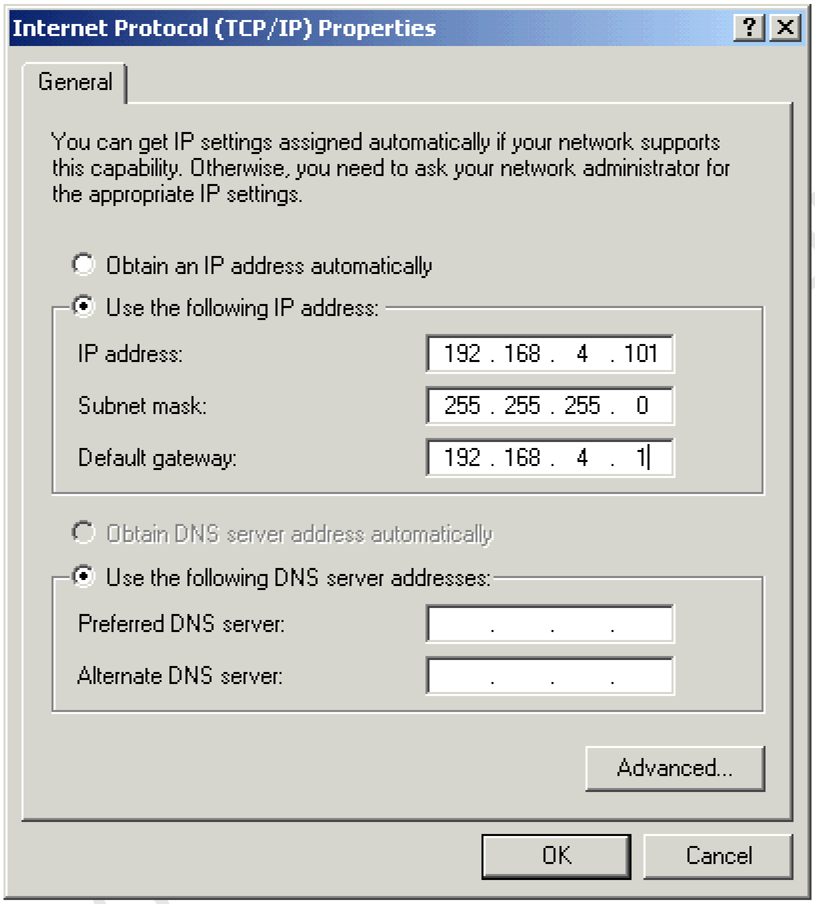
☑ Client for Microsoft Networks
☑ File and Printer Sharing for Microsoft Networks
☑ Internet Protocol (TCP/IP)

Description

Allows your computer to access resources on a Microsoft network.

□ Show icon in taskbar when connected

Select *Internet Protocol (TCP/IP)* and click on the *Properties* button, which brings up the following dialog:



Internet Protocol (TCP/IP) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address: 192 . 168 . 4 . 101

Subnet mask: 255 . 255 . 255 . 0

Default gateway: 192 . 168 . 4 . 1

○ Obtain DNS server address automatically

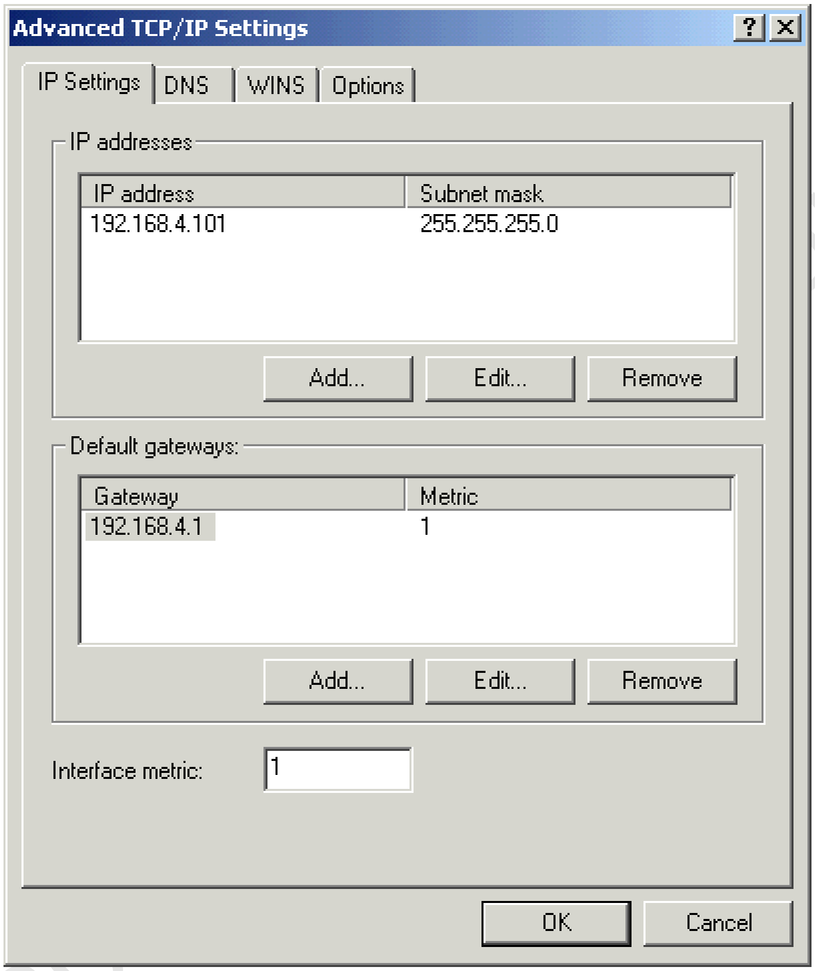
● Use the following DNS server addresses:

Preferred DNS server: . . .

Alternate DNS server: . . .

Note that in order to configure virtual IP addresses your computer must be configured to use fixed IP addresses; DHCP is not supported. If you are not on control of the IP addresses on your local network you should work with your network administrator to reserve a block of IP addresses.

The next step is to click on the *Advanced* button, bringing up this dialog:



IP addresses

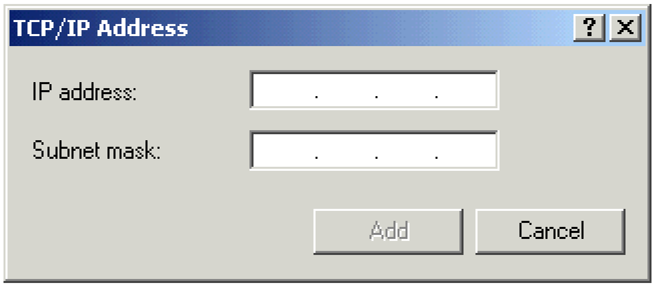
IP address	Subnet mask
192.168.4.101	255.255.255.0

Default gateways:

Gateway	Metric
192.168.4.1	1

Interface metric: 1

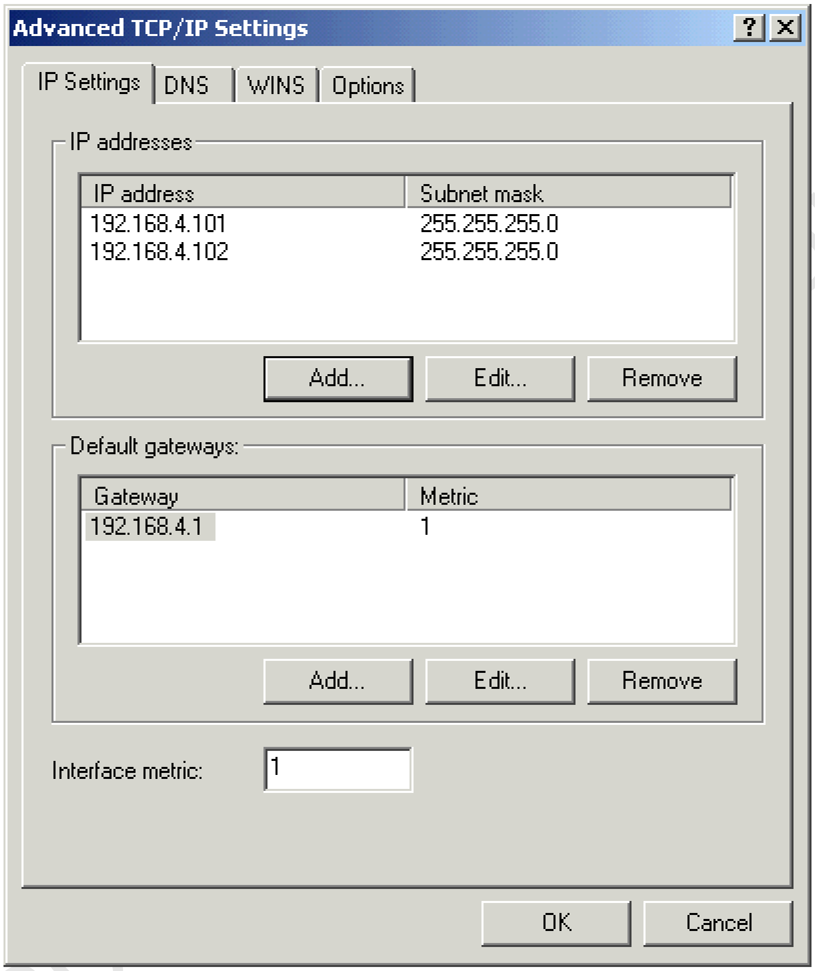
The above dialog (*Advanced TCP/IP Settings)* shows the list of IP addresses for your local computer. To add virtual network devices click on the *Add* button, which brings up the *TCP/IP Address Dialog*:



IP address: . . .

Subnet mask: . . .

Enter the IP address and subnet mask of the virtual network device you wish to configure, and this will be added to the list shown in the *Advanced TCP/IP Settings Dialog*:



IP address	Subnet mask
192.168.4.101	255.255.255.0
192.168.4.102	255.255.255.0

Gateway	Metric
192.168.4.1	1

Interface metric: 1

The procedure should be repeated for each virtual IP address/network device that you wish to add.

Linux/UNIX

As with the Windows configuration, choosing valid IP addresses is beyond the scope of this manual, but typically you would want to perform this modification on a computer using private IP addresses in a test lab.

The `ifconfig` command can be used to dynamically create virtual network device. The following example shows the creating of a single virtual network device:

```
[root@bigtoe root]# ifconfig eth0:0 10.1.1.1
[root@bigtoe root]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:A0:C9:5A:DF:F7
   inet addr:10.0.0.100  Bcast:10.0.0.255
 Mask:255.255.255.0
   UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
   RX packets:454545 errors:0 dropped:0 overruns:0 frame:0
   TX packets:311037 errors:0 dropped:0 overruns:0
carrier:0
```

```
   collisions:0 txqueuelen:100
   RX bytes:94017376 (89.6 Mb)  TX bytes:31798276 (30.3 Mb)
   Interrupt:10 Base address:0xdc00 Memory:ef201000-
ef201038

eth0:0    Link encap:Ethernet  HWaddr 00:A0:C9:5A:DF:F7
   inet addr:10.1.1.1  Bcast:10.1.1.255  Mask:255.255.255.0
   UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
   RX packets:0 errors:0 dropped:0 overruns:0 frame:0
   TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
   collisions:0 txqueuelen:100
   RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
   Interrupt:10 Base address:0xdc00 Memory:ef201000-
ef201038
```

This command would then have to be repeated with different parameters to add more than one virtual device. To make this permanent on a BSD or SysV style system like RedHat you can modify the `/etc/rc.d/rc.local` startup script. For more information please consult the Linux IP Alias mini-HOWTO .

Customizing IP Selections During a Load Test

Once your workstation has been configured with the IP addresses desired, you may also wish to configure the IP aliasing options of Web Performance Suite.

License Key Activation

Single-user license keys require activation when they are imported into the Web Performance software. This locks the license key for use by that user on that computer.

We realize that some license key activation schemes can be more cumbersome for licensed users than they are for software pirates. Therefore, we have attempted to make license key activation as painless as possible:

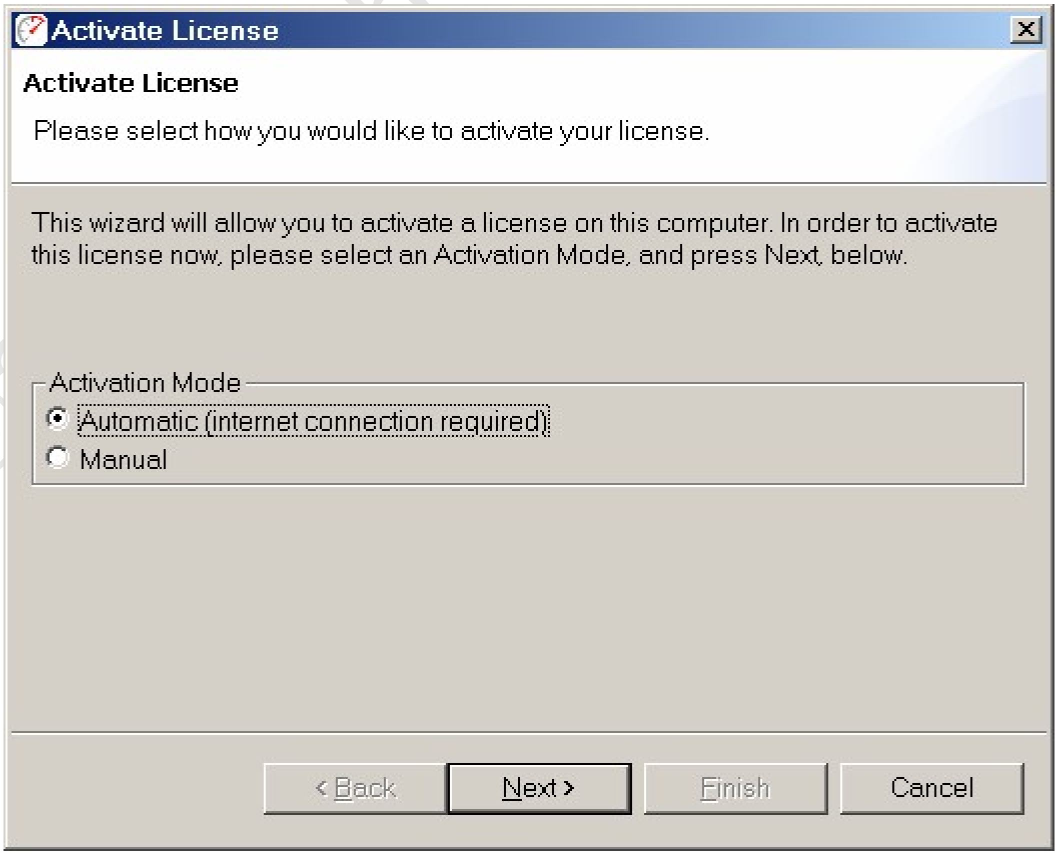
- First-time users on an Internet-connected computer will be activated automatically without any additional steps.
- Moving a license key from one computer to another requires only one additional step (de-activation) before installing the license key on the new computer.
- Users on machines that do not have an Internet connection can use a form on our website to activate the license key - which requires a total of 3 additional steps. We estimate this will take 2 minutes to complete. Alternatively, you can email the activation request file provided by our software to us and we'll return an activated license key to you.

- Once the license key is activated, the software does not require access to our server for future operation. Among other things, this means that our server could disappear forever and the software will still function normally.

First-time Activation

When a license key that requires activation is installed for the first time, the wizard will ask you to choose the activation mode: *automatic* or *manual*. If the computer does not have internet access, you will need to choose the manual mode and follow the instructions, described later, for activating without Internet access.

When automatic mode is selected, the software will contact the Web Performance server and activate the license key. The software is then ready to use.



Activate License

Please select how you would like to activate your license.

This wizard will allow you to activate a license on this computer. In order to activate this license now, please select an Activation Mode, and press Next, below.

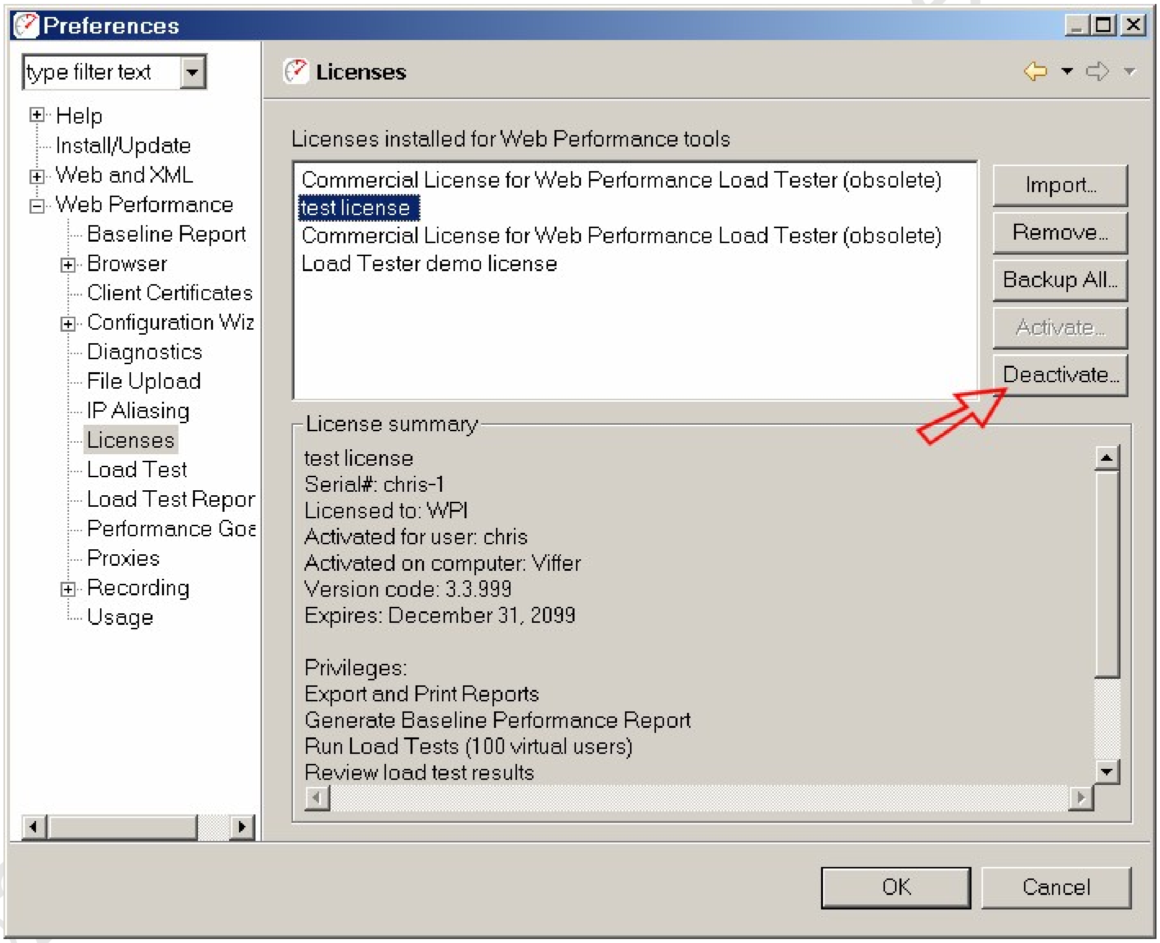
Activation Mode

● Automatic (internet connection required)

○ Manual

Moving the License Key

When a user upgrades to a new computer, the license key will need to be moved as well. Prior to the move, you should deactivate the license key on the currently-activated computer via the *Deactivate...* button on the License Key Management preference page in our software. Then you can install the deactivated license key on the new computer and repeat the activation process.



Licenses installed for Web Performance tools

Commercial License for Web Performance Load Tester (obsolete)
test license
Commercial License for Web Performance Load Tester (obsolete)
Load Tester demo license

License summary

test license
Serial#: chris-1
Licensed to: WPI
Activated for user: chris
Activated on computer: Viffer
Version code: 3.3.999
Expires: December 31, 2099

Privileges:
Export and Print Reports
Generate Baseline Performance Report
Run Load Tests (100 virtual users)
Review load test results

Note: When you deactivate a license key, you will be prompted to save the deactivated license key so that it may be moved and reactivated on another computer. If you have misplaced that key, you may use the original deactivated license key (the key mailed to you after purchase) for activation on the new computer. However, you can NOT skip the deactivation step. If you have already lost all data from the original computer and cannot retrieve it from a backup, you will need to contact the sales team at Web Performance for a replacement.

Activation and De-activation without Internet Access

If the computer running the software does not have access to the Internet, the activation and deactivation procedures may be performed manually via our website from another computer or via email.

To activate a license key manually, select the *manual* activation mode in the activation wizard. If a license key requires activation, the wizard will appear automatically when importing the license key on the License Key Management preference page. The wizard will give you an activation request file (or deactivation file) that can be submitted to our website (http://webperformance.com/activate). In return, you will receive a pre-activated license key that can be installed on the License Management preference page.

FAQ

Q: What happens if a second user needs to use the software on the same computer?
A: If the second user needs to use the software *in addition* to the first user, you should purchase an additional license. If the first user no longer needs to user the software, you can *move* that license key from one user to another - by following the same procedure as moving the license key from one computer to another.

Q: What happens if I copy the activated license key to another computer?
A: At best, the software will not run. At worst, it may trigger a license revocation - preventing the license key from working on any computer. You would need to contact support to issue a new license key.

Q: I backed up my entire OS on one computer and restored it to another. Now the software will not run. What can I do?
A: You need to deactivate/reactivate the license key, as explained above.

Q: Does the activation process contact your server or use the Internet?
A: Yes, it contacts our server to validate that the license key has not already been activated.

Q: Does the software ever contact the server again?
A: Yes, the software will periodically verify that the license key is still valid. It is also contacted when you deactivate the license.

Q: How does a license key get revoked?
A: When we detect that the license enforcement mechanism has been circumvented, the license key will be revoked. We will make every attempt to contact you and/or your company to resolve the problem before this step is taken. We hope we never have to take this step.

Q: I lost my activated license key file. Is there an easy way to reset it so that I can activate it again?
A: No. You will need to contact the sales team for a new license key. The original serial number will be revoked and a new key must be issued under a new serial number.

Glossary of Terms

Page Duration - The total time to complete all the transactions for a web page. This time starts when a connection for the first transaction is initiated and ends when the response for the final transaction is completed. The page duration does not include the think time after the page is completed.

Repeat delay - the delay between the time when a VU finishes a testcase and begins the next.

Sample - A set of data collected during a specific time period.

Sample period - the length of time during which a sample was measured.

Testcase - A series of web (HTTP) transactions between a client and one or more servers. A testcase is usually created by recording the interactions between a browser and the servers. The testcase is represented as a series of pages; each page can contain one or more transactions.

Testcase Elapsed Time - the total time to perform all the pages in the test case and the intervening periods of think time. This will usually be much larger than the sum of all the Page Durations in the testcase, due to the addition of the think times.

Think Time - The time delay between the end of one page and the start of the next. The name reflects that this time delay represents the time the user spends reading, entering data or thinking about the last page received.

Transaction - A pair of HTTP messages (request and response) between a client (e.g. browse) and a server.

Virtual User - A simulation of a real user using a browser (or other web application) to interact with a server

Troubleshooting Guide

Recording Configuration Troubleshooting Guide

Getting Started

If you are reading this page, then you have attempted to record your website but were unsuccessful. Rest assured, Analyzer™ is compatible with every network configuration we have encountered. However, Analyzer™ is not always able to automatically configure the workstation correctly and must occasionally be configured manually. The remainder of this section will guide you through the process of determining what settings are correct for your particular network configuration.
Which of the following best describes the problem you are experiencing?

- The Recording Configuration Wizard appears. After entering in the URL of the site to be recorded, the wizard indicates that it was unsuccessful connecting to the site and requires a manual configuration. CONTINUE ▶
- A Web Browser is launched, but indicates "Proxy configuration failed" or displays a similar error page. CONTINUE ▶
- A Web Browser is launched, and displays a message stating that

  The Analyzer™ module of the Web Performance Suite™ is now successfully recording your test case.

  However, when attempting to connect to the desired site, an error message is displayed, or nothing is displayed. CONTINUE ▶

Index