



Automation User Guide

Version 12.0

RADVIEW

The software supplied with this document is the property of RadView Software and is furnished under a licensing agreement. Neither the software nor this document may be copied or transferred by any means, electronic or mechanical, except as provided in the licensing agreement. The information in this document is subject to change without prior notice and does not represent a commitment by RadView Software or its representatives.

WebLOAD Automation User Guide

© Copyright 2018 by RadView Software. All rights reserved.

November, 2018, RadView Publication Number WLPRO-1013-AUG84

WebLOAD, TestTalk, Authoring Tools, ADL, AppletLoad, and WebExam, are trademarks or registered trademarks of RadView Software IBM, and OS/2 are trademarks of International Business Machines Corporation. Microsoft Windows, Microsoft Windows 95, Microsoft Windows NT, Microsoft Word for Windows, Microsoft Internet Explorer, Microsoft Excel for Windows, Microsoft Access for Windows and Microsoft Access Runtime are trademarks or registered trademarks of Microsoft Corporation. SPIDERSESSION is a trademark of NetDynamics. UNIX is a registered trademark of AT&T Bell Laboratories. Solaris, Java and Java-based marks are registered trademarks of Sun Microsystems, Inc. HP-UX is a registered trademark of Hewlett-Packard. SPARC is a registered trademark of SPARC International, Inc. Netscape Navigator and LiveConnect are registered trademarks of Netscape Communications Corporation. iReport is a registered trademark of JasperSoft Corporation. Any other trademark name appearing in this book is used for editorial purposes only and to the benefit of the trademark owner with no intention of infringing upon that trademark.

For product assistance or information, contact:	
Toll free in the US:	1-888-RadView
Fax:	+1-908-864-8099
World Wide Web:	www.RadView.com
North American Headquarters:	International Headquarters:
RadView Software Inc. 991 Highway 22 West, Suite 200 Bridgewater, NJ 08807 Email: info@RadView.com Phone: 908-526-7756 Fax: 908-864-8099 Toll Free: 1-888-RadView	RadView Software Ltd. 13 Haamal Street, Park Afek Rosh Haayin, 4809249 Israel Email: info@RadView.com Phone: +972-3-915-7060 Fax: +972-3-915-7011

Table of Contents

Chapter 1. Introduction	1
WebLOAD Documentation.....	1
Typographical Conventions.....	2
Where to Get More Information.....	3
Online Help	3
Technical Support Website.....	3
Technical Support.....	4
Chapter 2. WebLOAD Automation Overview	5
Chapter 3. Running WebLOAD from the Command Line	6
Registering and Updating your WebLOAD License through the CLI	6
Syntax.....	7
Parameters.....	7
Examples.....	7
Running a WebLOAD Console Test through the CLI.....	8
Syntax.....	8
Parameters.....	8
Examples.....	9
Generating an Analytics Report through the CLI.....	10
Running WLANalyticsCMD.exe.....	10
Running WLANalytics.exe.....	12
Running WebLOAD Recorder Testing through the CLI	14
Syntax.....	14
Parameters.....	14
Examples.....	15
Chapter 4. Running WebLOAD in Jenkins	16
Installing the WebLOAD Plugin for Jenkins	17
Creating a WebLOAD Job in Jenkins.....	20
Defining the Execute Load Session Build Step.....	23
Defining the Generate Analytics Report Build Step	25
Common Usage Scenarios of an Analytics Build Action.....	27

Chapter 5. Running WebLOAD using the Open API.....	29
Accessing the REST Open API	29
Understanding how to use the WebLOAD REST Open API.....	30

Introduction

Welcome to WebLOAD Professional, the premier performance, scalability, reliability testing solution for internet applications.

WebLOAD is easy to use and delivers maximum testing performance and value. WebLOAD verifies the scalability and integrity of internet applications by generating a load composed of Virtual Clients that simulate real-world traffic. Probing Clients let you refine the testing process by acting as a single user that measures the performance of targeted activities, and provides individual performance statistics of the internet application under load.

This section provides a brief introduction to WebLOAD technical support, including both documentation and online support.

IMPORTANT NOTE: In previous WebLOAD versions, a WebLOAD script was called an “Agenda”. From version 12.0, it is referred to simply as a script. Wherever “Agenda” is still displayed, we are referring to the WebLOAD script.

WebLOAD Recorder was formerly referred to as WebLOAD IDE.

WebLOAD Documentation

WebLOAD is supplied with the following documentation:

WebLOAD™ Installation Guide

Instructions for installing WebLOAD and its add-ons.

WebLOAD™ Recorder User Guide

Instructions for recording, editing, and debugging load test scripts to be executed by WebLOAD to test your Web-based applications.

WebLOAD™ Console User Guide

A guide to using WebLOAD console, RadView’s load/scalability testing tool to easily and efficiently test your Web-based applications. This guide also includes a quick start section containing instructions for getting started quickly with WebLOAD using the RadView Software test site.

WebLOAD™ Analytics User Guide

Instructions on how to use WebLOAD Analytics to analyze data and create custom, informative reports after running a WebLOAD test session.

WebRM™ User Guide

Instructions for managing testing resources with the WebLOAD Resource Manager.

WebLOAD™ Scripting Guide

Complete information on scripting and editing JavaScript scripts for use in WebLOAD and WebLOAD Recorder.

WebLOAD™ JavaScript Reference Guide

Complete reference information on all JavaScript objects, variables, and functions used in WebLOAD and WebLOAD Recorder test scripts.

WebLOAD™ Extensibility SDK

Instructions on how to develop extensions to tailor WebLOAD to specific working environments.

WebLOAD™ Automation Guide

Instructions for automatically running WebLOAD tests and reports from the command line, or by using the WebLOAD plugin for Jenkins.

WebLOAD™ Cloud User Guide

Instructions for using RadView's WebLOAD Cloud to view, analyze and compare load sessions in a web browser, with full control and customization of the display.

The guides are distributed with the WebLOAD software in online help format. The guides are also supplied as Adobe Acrobat files. View and print these files using the Adobe Acrobat Reader. Install the Reader from the Adobe Web site <http://www.adobe.com>.

Typographical Conventions

Before you start using this guide, it is important to understand the terms, icons, and typographical conventions used in the documentation.

The following icons appear next to the text to identify special information.

Table 1: Icon Conventions

Icon	Type of Information
	Indicates a note.
	Indicates a feature that is available only as part of a WebLOAD Add-on.

The following kinds of formatting in the text identify special information.

Table 2: Typographical Conventions

Formatting convention	Type of Information
Special Bold	Items you must select, such as menu options, command buttons, or items in a list.
<i>Emphasis</i>	Use to emphasize the importance of a point or for variable expressions such as parameters.
CAPITALS	Names of keys on the keyboard. for example, SHIFT, CTRL, or ALT.
KEY+KEY	Key combinations for which the user must press and hold down one key and then press another, for example, CTRL+P or ALT+F4.

Where to Get More Information

This section contains information on how to obtain technical support from RadView worldwide, should you encounter any problems.

Online Help

WebLOAD provides a comprehensive online help system with step-by-step instructions for common tasks.

You can press the **F1** key on any open window for an explanation of the options or select **Help > Contents** to open the online help contents and index.

Technical Support Website

The technical support pages on our website contain:

- The option of opening a ticket
- Links to WebLOAD documentation

Technical Support

For technical support in your use of this product, contact:

North American Headquarters	International Headquarters
e-mail: support@RadView.com Phone: 1-888-RadView (1-888-723-8439) (Toll Free) 908-526-7756 Fax: 908-864-8099	e-mail: support@RadView.com Phone: +972-3-915-7060 Fax: +972-3-915-7011



Note: We encourage you to use e-mail for faster and better service.

When contacting technical support please include in your message the full name of the product, as well as the version and build number.

WebLOAD Automation Overview

This guide describes the two methods of automating WebLOAD tasks:

- Running WebLOAD components from a command line interface.
- Running a WebLOAD job in Jenkins.

Running WebLOAD from the Command Line

You can run the following WebLOAD components from the command line:

- WebLOAD License – You can register and update your WebLOAD license.
- WebLOAD Console – You can launch a WebLOAD test that invokes the WebLOAD Console and runs a specified test according to the specified parameters.
- WebLOAD Analytics – You can launch WebLOAD Analytics and generate a report for a specified session, then publish or print it.
- WebLOAD Recorder – You can launch a WebLOAD Recorder test that invokes the WebLOAD Recorder and runs a specified test according to the specified parameters.

Registering and Updating your WebLOAD License through the CLI

You can register or upload a WebLOAD license through a command line interface. You can enter the WebLOAD Update License command into a batch file or into an external script and it will run directly, without user intervention, using the parameters specified.

To register or update you WebLOAD license through the command line interface:

Enter the `wlUpdateLicenseApplicationCmd` command together with the optional parameters (described below) into your external script, to automatically launch a WebLOAD License action.

Syntax

Use the following syntax to define the parameters for registering or updating a WebLOAD license through a command line interface:

```
wlUpdateLicenseApplicationCmd {-help}|{-info}|{-free}|{-file
license_file}|{-server server_address}|{-hostid}
```

Parameters

Parameter	Description	Comments
-help	Display all available command parameters and their syntax.	
-info	Display the full license details of the installed license.	
-free	Create and install a license for a free WebLOAD version. This version provides 50 virtual clients, and is available for an unlimited period of time. You can use the free WebLOAD edition immediately after running this command.	
-file license_file	Install the specified license file, which you received from RadView (refer to the <i>WebLOAD Installation Guide</i> for further explanations). If license registration is successful, a confirmation appears in the command-line output. You can then begin working with WebLOAD.	
- server server_address	Connect to the WebRM server, whose address is the specified IP address. This option is relevant if you installed a WebRM server. For further explanations, refer to the <i>WebLOAD Installation Guide</i> .	
-hostid	Display the host ID of the computer. This information is needed if you wish to receive a WebLOAD license file from RadView. For further explanations, refer to the <i>WebLOAD Installation Guide</i> .	

Examples

Example 1:

```
wlUpdateLicenseApplicationCmd -info
```

This command provides as output the license information.

Example 2:

```
wlUpdateLicenseApplicationCmd -file weblod.lic
```

This command uploads the specified license file, and installs it. Following successful installation, you can begin working with WebLOAD.

Running a WebLOAD Console Test through the CLI

You can perform load testing through a command line interface. You can enter the WebLOAD Console launch command into a batch file or into an external script and WebLOAD Console will run directly, without user intervention, using the parameters specified.

To run WebLOAD Console through the command line interface:

Enter the `WebLOAD.exe` command together with a series of optional parameters (described below) into your external script to automatically launch a WebLOAD test. When your script runs, the executable file will invoke WebLOAD Console and run the specified test according to the specified parameters.



Note: Verify that the script used with the template specified, and any included files, are accessible to the Load Template or Load Session file that will be run.

Syntax

Use the following syntax to define the parameters for running a WebLOAD test through a command line interface:

```
WebLOAD.exe [<file name to open>][<file name to save>]
[<flags>]
```

Parameters

Parameter	Description	Comments
File name to open	The name of the *.tpl or *.ls file (Load Template or Load Session file) to run.	Optional parameter
File name to save	The name of the *.ls file containing the test data. This file will be saved in the current directory unless otherwise specified.	Optional parameter

Parameter	Description	Comments
Flags	<ul style="list-style-type: none"> • /ar – Automatically run the WebLOAD test without waiting for user input. If this flag is not specified, the Console is called up with the specified Load Template/Load Session but the test is not automatically run. The system waits for user input. • /ar <time> – Automatically run the test for the length of time specified in <time>/ • /vc <num>– The number of Virtual Client licenses to allocate when using WebRM License Server. • /pc <num>– The number of Probing Client licenses to allocate when using WebRM License Server. • /rc <results_file_name> – Place the results in the specified file (an XML file). • /ag <script name> - The name of an existing script (*.wlp) to open. 	Optional parameter

The parameters are all optional. If no parameters are entered, the executable launches the Console and does not run a test. If the autorun flag `</ar>` flag is not set, the `<file name to save>` and the `<time>` parameters are ignored.



Note: If there is a conflict between the time defined in this command and the time defined in the WebLOAD Scheduler, the load test runs for the shorter of the two periods defined.

Examples

Example 1:

```
WebLOAD.exe test1.tpl
```

This command opens the Console and the `test1.tpl` template. The Console waits for user input.

Example 2:

```
WebLOAD.exe test1.tpl march9.ls /ar 30
```

This command opens the Console and automatically runs a test using the `test1.tpl` template file. The test results are saved in the Load Session file `march9.ls`, which includes all of the test data and results. This file is saved in the current directory, unless otherwise specified. The autorun flag is set, meaning that the test runs without user intervention. The test will run for 30 seconds.

Example 3:

```
WebLOAD.exe /ag c:\agendas\MyAgenda.wlp
```

This command opens the Console and the WebLOAD Wizard to the script/Mix Selection dialog box. The `MyAgenda.wlp` script is automatically selected and the WebLOAD Wizard waits for user input.

Example 4:

```
WebLOAD.exe test1.tpl march9.ls /ar 30 /rc result1.xml
```

This command performs all the actions described in *Example 2* above, and in addition the execution return code is saved in `result1.xml`.

Example 5:

```
WebLOAD.exe test1.tpl march9.ls /ar 30 /vc 100 /pc 3
```

This command performs all the actions described in *Example 2* above, and in addition it allocates 100 virtual clients and 3 probing clients from the WebRM server.

Generating an Analytics Report through the CLI

WebLOAD Analytics can be executed in command line mode. This enables incorporating WebLOAD Analytics in scripts. Two executables are available:

- **WLANalyticsCMD.exe** – Automatically generates a report for a specified session, and publishes or prints it. WebLOAD Analytics then closes.
- **WLANalytics.exe** – Launches the WebLOAD Analytics UI, and generates a report for a specified session.

The executables are located in `<Installation dir>\bin`. For example:
`C:\Program Files\RadView\WebLOAD\bin`.

Running WLANalyticsCMD.exe

Use this executable to generate a report for a specified session, and publish or print it.

Syntax

```
WLANalyticsCmd.exe -m U|P {-t template_path} | {-p portfolio_path}
{-s session_name} | {-ls session_path} [-f
DOC|ODT|HTML|XLS|RTF|PDF] [-l report_location] [-n
output_report_name] [-h]
```

Parameters

Parameter	Description	Comments
-m	Indicates the action. Specify one of the following U – Publish. P – Print.	Mandatory parameter.
-t <i>template_path</i>	Generates a chart from a specified template. You must specify the path to the template directory (either absolute or relative to the gallery).	You must specify one of the two options: -t or -p .
-p <i>portfolio_path</i>	Generates a report from a portfolio. You must specify the path to the portfolio directory (either absolute or relative to the Portfolio category).	
-s <i>session_name</i>	Specifies a session already loaded into WebLOAD. You must specify the session name. Note: You can use this parameter multiple times to specify multiple sessions. This is necessary if you are generating a regression chart.	You must specify one of the two options: -s or -ls .
-ls <i>session_path</i>	Specifies a load session file to import into WebLOAD. You must specify the full path. Note: You can use this parameter multiple times to load multiple sessions. This is necessary if you are generating a regression chart.	
-f	Specifies the output format for a published report. Select one of the following: DOC , ODT , HTML , XLS , RTF , or PDF . If you do not specify an output format, the default format, specified in Analytics Preferences, is used.	Optional parameter.
-l <i>report_location</i>	Specifies the location of the published report. If you do not specify a location, the default location, specified in Analytics Preferences, is used.	Optional parameter.
-n <i>output_report_name</i>	Specifies a name for the newly created report. If you do not specify a name, the application provides a default name.	Optional parameter.
-h	Displays the help.	Optional parameter.



Note: Note that you must specify:

- Publish or print.
- A template or portfolio.
- A session, either previously loaded or to be imported.

Examples:**Example 1:**

To load the `mysession.ls` Load Session, generate a 'General/Load Size Summary' chart, and publish it in the default file format, in the default location, under the name `test-report`:

```
WLANalyticsCmd.exe -m U -t "General\Load Size Summary" -ls
"C:\mysession.ls" -n "test-report"
```

Example 2:

To use the loaded `first-session` Load Session, generate a 'Summary Portfolio' portfolio, and print it:

```
WLANalyticsCmd.exe -m P -p "Summary Portfolio" -s "first-
session"
```

Example 3:

To use the loaded `first-session` and `second-session` Load Sessions, generate a 'Regression/Load Size Summary' regression chart, and publish it as a PDF file in `C:\myreports`, using a default name:

```
WLANalyticsCmd.exe -m U -t "Regression\Load Size Summary" -s
"first-session" -s "second-session" -f PDF -l "c:\myreports"
```

Running WLANalytics.exe

Use this executable to open the WebLOAD Analytics UI, and open a report or generate a report for a specified session.

Syntax

```
WLANalytics.exe {-t template_path}|{- p portfolio_path}  
{-s session_name}|{-ls session_path} [-h] [-noSplash]
```

Parameters

Parameter	Description	Comments
<code>-t</code> <i>template_path</i>	Generates a chart from a specified template. You must specify the path to the template directory (either absolute or relative to the gallery).	You must specify one of the two options: -t or -p .
<code>-p</code> <i>portfolio_path</i>	Generates a report from a portfolio. You must specify the path to the portfolio directory (either absolute or relative to the Portfolio category).	
<code>-s</code> <i>session_name</i>	Specifies a session already loaded into WebLOAD. You must specify the session name. Note: You can use this parameter multiple times to specify multiple sessions. This is necessary if you are generating a regression chart.	You must specify one of the two options: -s or -ls .
<code>-ls</code> <i>session_path</i>	Specifies a load session file to import into WebLOAD. You must specify the full path. Note: You can use this parameter multiple times to load multiple sessions. This is necessary if you are generating a regression chart.	
<code>-h</code>	Displays the help.	Optional parameter.
<code>-noSplash</code>	Launches without a Splash screen.	Optional parameter.



Note: Note that you must specify:

- A template, report, or portfolio.
- A session, either previously loaded or to be imported.

Examples

Example 1:

To open the WebLOAD Analytics UI, load the `mysession.ls` Load Session, and generate a 'General/Load Size Summary' chart:

```
WLANalytics.exe -t "General\Load Size Summary" -ls "C:\mysession.ls"
```

Example 2:

To open the WebLOAD Analytics UI, use the loaded `first-session` Load Session, and generate a 'Summary Portfolio' portfolio:

```
WLANalytics.exe -p "Summary Portfolio" -s "first-session"
```

Running WebLOAD Recorder Testing through the CLI

You can initiate WebLOAD Recorder testing directly through the CLI. You can enter the WebLOAD Recorder launch command into a batch file or into an external script and WebLOAD Recorder will run directly, without user intervention, using the parameters specified.

To run WebLOAD Recorder testing through the CLI:

Enter the `weblloadIDE.exe` command together with a series of optional parameters (described below) into your external script to automatically launch a WebLOAD Recorder test. When your script runs, the executable file will invoke WebLOAD Recorder and run the specified test according to the specified parameters.

Syntax

Use the following syntax to define the parameters for running a WebLOAD Recorder test **through a Command Line Interface**:

```
weblloadide.exe [<flags>][<project or session name to open>]
[<session name to save to>][<Number of rounds to run>]
```

To run more than one session, append all relevant parameters at the end of the syntax. See examples 2 and 3 in *Examples* (on page 15).

Parameters

When running a test invoked by the executable, you can specify the following parameters:

Parameter	Description	Comments
Flags	/a - auto run Automatically run the WebLOAD Recorder test without waiting for user input. If this flag is not specified, WebLOAD Recorder is opened with the specified project / session but the test is not automatically run. The system waits for user input.	Optional parameter
Project or session name to open	The name of the <code>.wlp</code> file or <code>.wls</code> file (Project file or Session file) to run.	Optional parameter
Session name to save to	The name of the <code>.wls</code> file containing the test data. This file will be saved in the current directory unless otherwise specified.	Optional parameter
Number of rounds to run	The number of iterations to run during runtime. The default value is 1.	Optional parameter

The parameters are all optional. If no parameters are entered, the executable launches WebLOAD Recorder and does not run a test. If the autorun flag `/a` flag is not set, the `< Session name to save to >`, and the `< Number of rounds to run >` parameters are ignored.

Examples

Example 1:

```
webloadide.exe test1.wlp
```

This command opens WebLOAD Recorder with the `test1` project file and waits for user input.

Example 2:

```
webloadide.exe /a test1.wlp test2.wlp 3
```

This command:

- Opens WebLOAD Recorder and automatically runs a test using the `test1.wlp` project file.
- Runs the project for three iterations.
- Saves the test results in the WebLOAD Recorder session file `test1.wls`, which includes all of the test data and results.

Example 3:

```
webloadide.exe /a test1.wlp test1.wls 3 /a test2.wlp test2.wls 2
```

This command:

- Opens WebLOAD Recorder and automatically runs a test using the `test1.wlp` project file.
- Runs the project `test1.wlp` for three iterations.
- Saves the test results in the WebLOAD Recorder session file `test1.wls`, which includes all of the test data and results.
- Opens the WebLOAD Recorder project file `test2.wlp`.
- Runs the project `test2.wlp` for two iterations.
- Saves the test results in the WebLOAD Recorder session file `test2.wls`, which includes all of the test data and results.

Running WebLOAD in Jenkins

Jenkins CI is a Continuous Integration server that is growing in popularity. Agile development teams use Jenkins to run their builds, tests and deployments as well as their performance tests if possible. Jenkins is extendable by means of plugins. As of WebLOAD 10.1, a WebLOAD plugin is available for Jenkins.

To run WebLOAD jobs in Jenkins, you must:

1. Install a Jenkins CI server.

Jenkins needs to be able to control a full WebLOAD installation (in which both the WebLOAD Console and WebLOAD Analytics are installed, and which has an active license). This can be done either by installing the Jenkins Master on the WebLOAD machine, or by setting up a Jenkins Slave (recommended). For instructions, refer to the Jenkins documentation at <http://jenkins-ci.org/>.



Note: If Jenkins is running as a Service, make sure that you are using a logged in user, and not the Local System Account.

2. Install the WebLOAD plugin in Jenkins. This is a one-time task. Refer to *Installing the WebLOAD Plugin for Jenkins* below.
3. Define a WebLOAD job in Jenkins whenever desired. Refer to *Creating a WebLOAD Job in Jenkins* on page 20.

Installing the WebLOAD Plugin for Jenkins

To install the WebLOAD Plugin for Jenkins:

1. In the Jenkins dashboard (homepage), click **Manage Jenkins**.

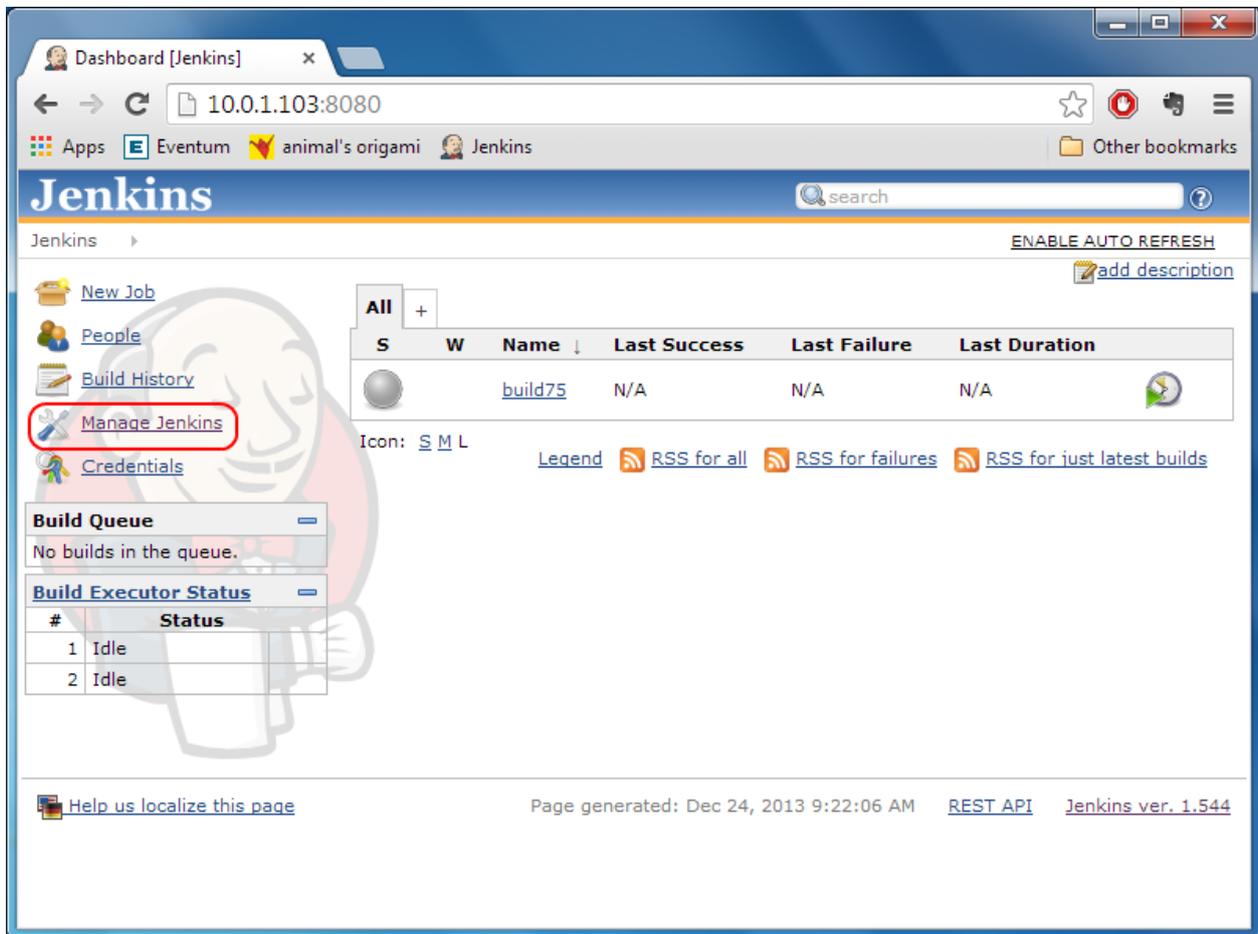


Figure 1: Selecting Manage Jenkins

2. In the Manage Jenkins page, click **Manage Plugins**.



Figure 2: Selecting Manage Plugins

3. Select the **Available** tab.

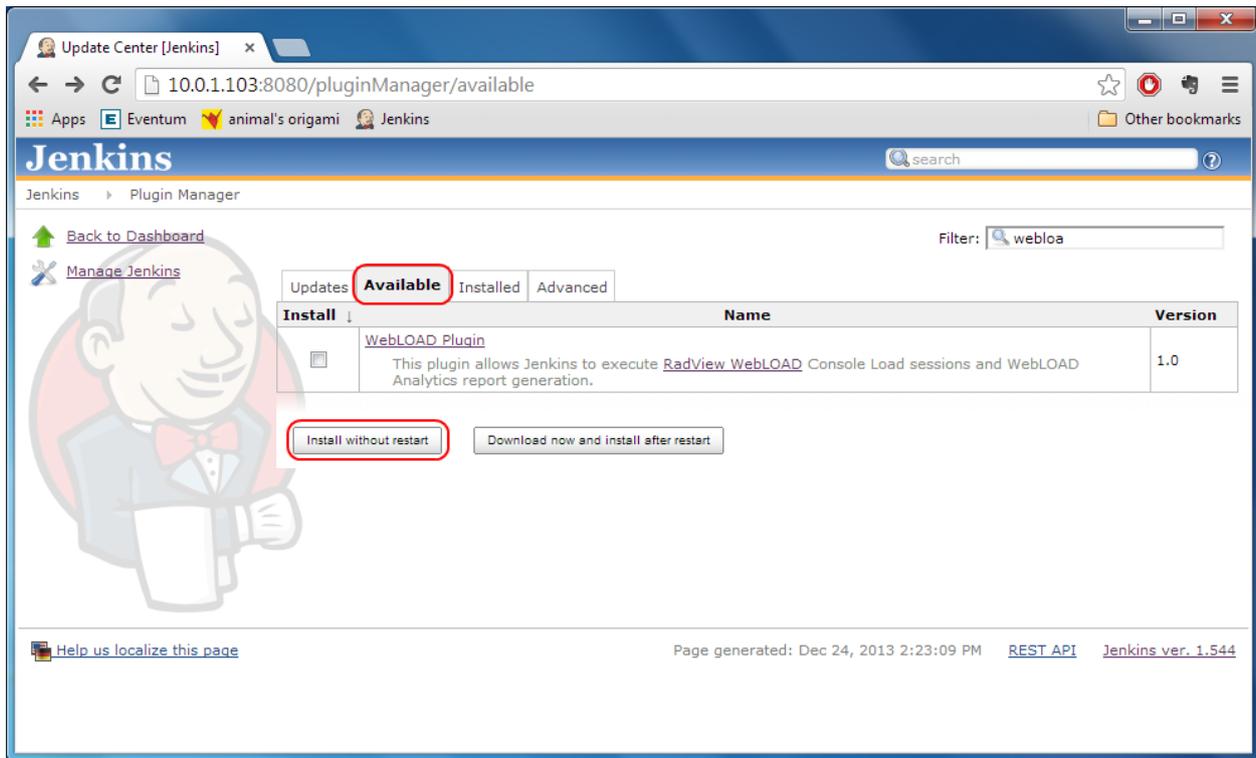


Figure 3: Available Plugins

4. In the Available page:
 - a. Select the **WebLOAD Plugin** in the list.
 - b. Click **Install without restart**.

You are returned to the Jenkins dashboard.

Creating a WebLOAD Job in Jenkins

1. In the Jenkins dashboard (homepage), click **New Job**.

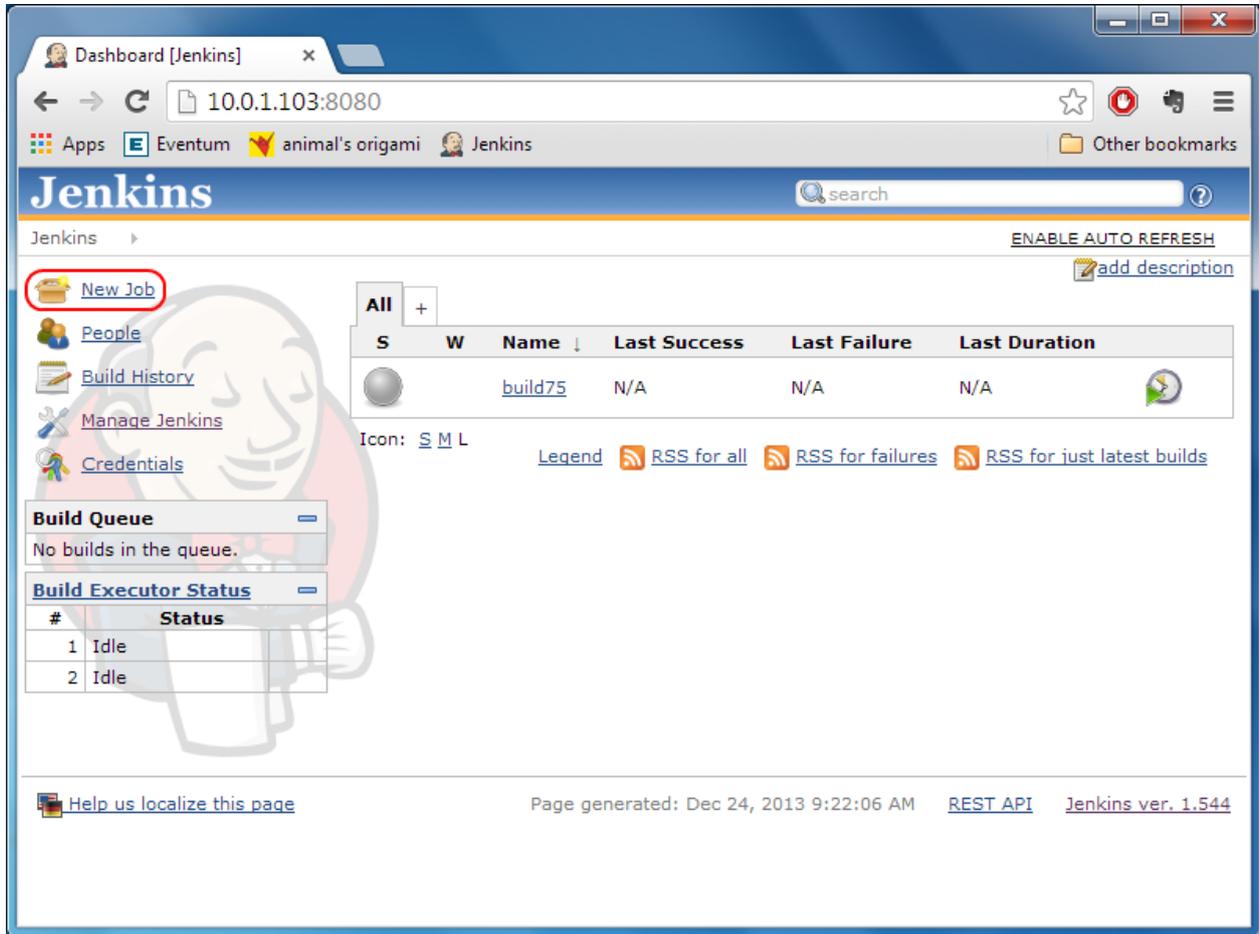


Figure 4: Selecting New Job

2. Enter a name for the new job, and define the type of job by selecting from the options displayed under the job name. Typically you might select **Build a free-style software project** or **Copy existing Job**.

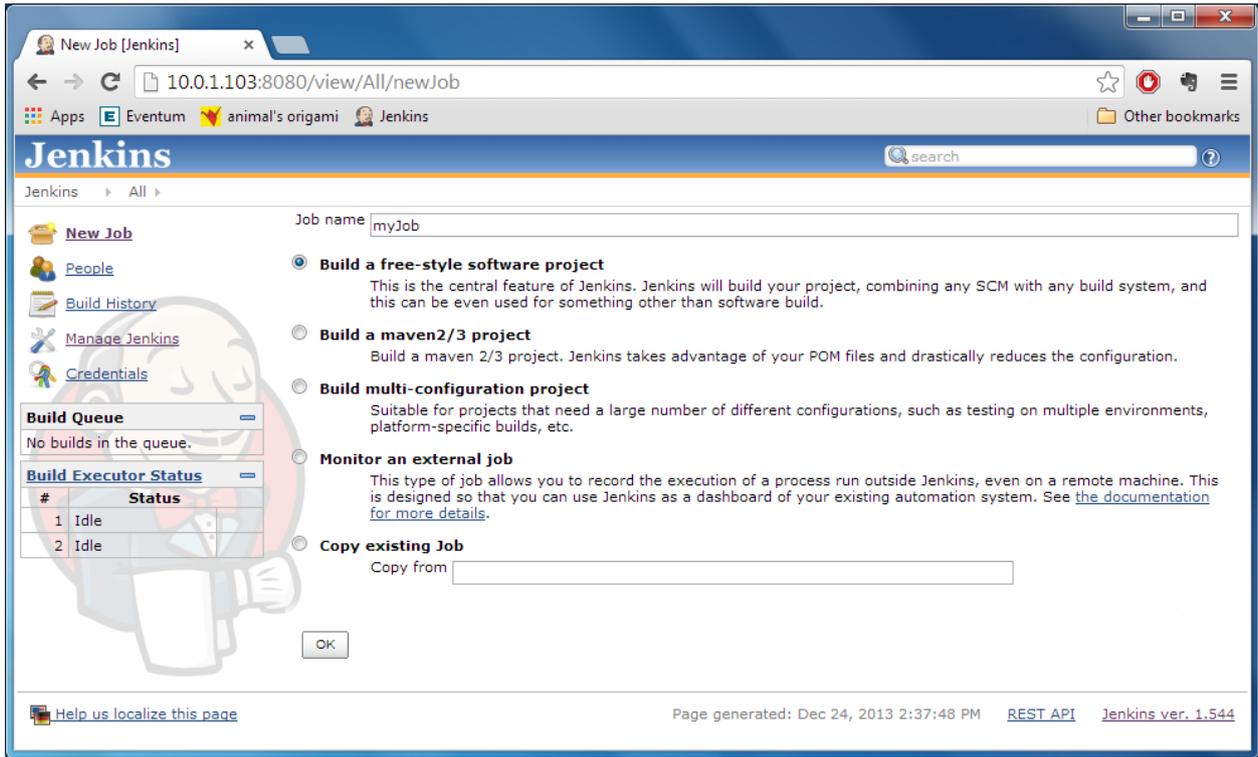


Figure 5: Defining a Job Name and Type

3. Click OK.

A standard Jenkins New Job page appears.

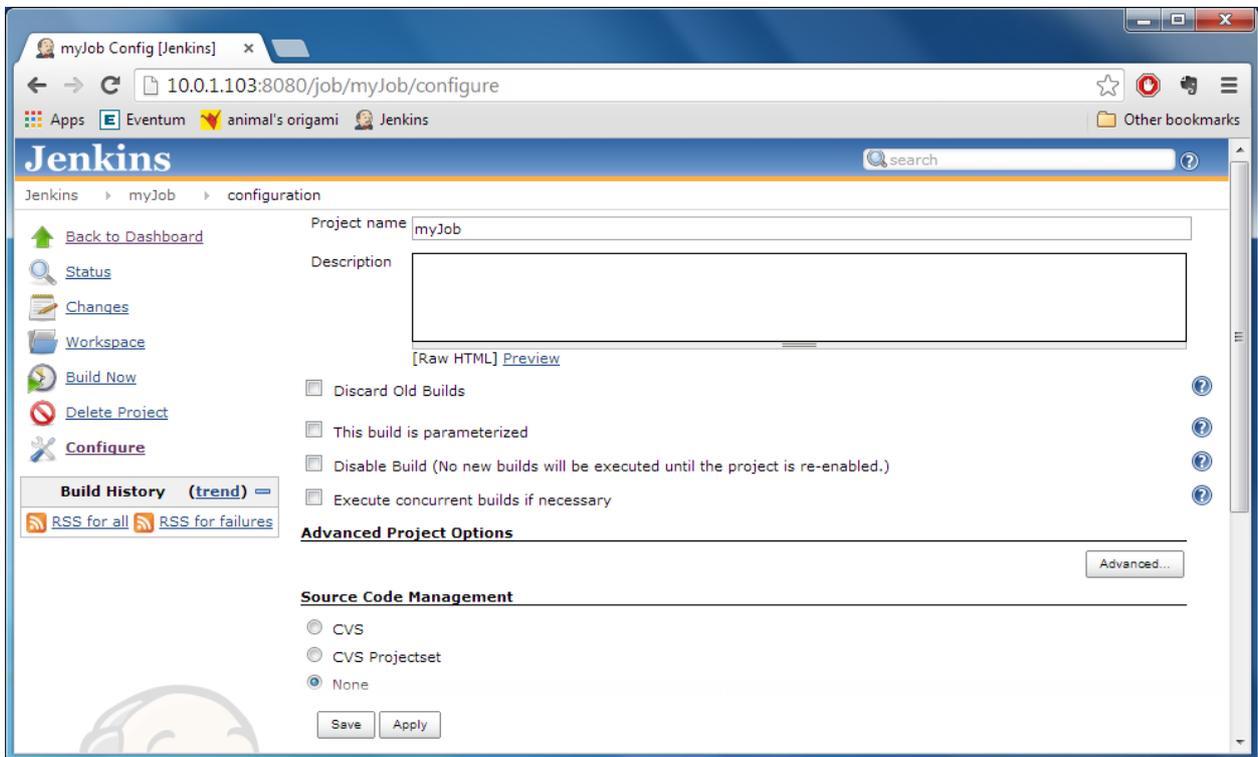


Figure 6: New Job Page

- Define the project (for help, refer to Jenkins documentation). For example, you might click **Build periodically** to define a project that runs automatically every Thursday at 2 AM.
- In the **Build** section, click **Add build step** and select a step. The two steps offered by the WebLOAD plugin are **Execute WebLOAD load session** and **Generate WebLOAD Analytics Report**.

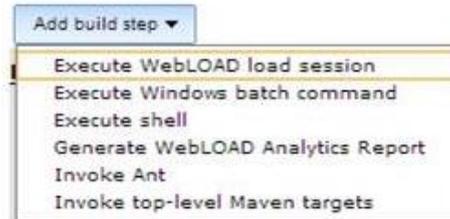
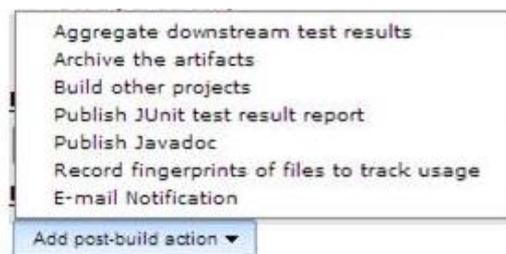


Figure 7: Plugins List

The screen refreshes to enable defining the settings of the build step. Refer to:

- *Defining the Execute Load Session Build Step* on page 23
 - *Defining the Generate Analytics Report Build Step* on page 25
- Repeat the previous step as many times as desired to add as many build actions as desired.
 - Click **Add post-build action** and specify a post-build action or actions. For example, if you specified the JUNIT output format for an Analytics report, you can select **Publish JUnit test result report**.



- Click **Save** or **Apply**.

Defining the Execute Load Session Build Step

If you select **Execute WebLOAD load session** as a build step, the following appears:

The screenshot shows a 'Build' step configuration window. At the top, it says 'Build'. Below that, there is a section titled 'Execute WebLOAD load session'. There are two input fields: 'Template File' with the value 'c:\temp\demo.tpl' and 'Load Session File' with the value 'demo_\${BUILD_NUMBER}.ls'. To the right of each field is a help icon (a question mark in a circle). Below the input fields, there is a yellow pencil icon, a button labeled 'Advanced...' (which is highlighted with a red circle), and a 'Delete' button.

Figure 8: Execute WebLOAD load session – Defining

1. Optionally click **Advanced** to further refine your definitions. The following appears:

The screenshot shows the 'Advanced Definition' configuration window for the 'Execute WebLOAD load session' build step. It has a 'Build' header. The section is titled 'Execute WebLOAD load session'. There are five input fields: 'Template File' (c:\temp\demo.tpl), 'Load Session File' (demo_\${BUILD_NUMBER}.ls), 'Execution duration' (30), 'Allocate floating Virtual Clients' (0), and 'Allocate floating Probing Clients' (0). Each field has a help icon (a question mark in a circle) to its right. At the bottom right, there is a 'Delete' button.

Figure 9: Execute WebLOAD load session – Advanced Definition

2. Enter information in the fields, as described in Table 3.

Table 3: WebLOAD Load Session Parameters

Parameter	Description
Template File	The name and path of the *.tpl (Template file) or *.ls file (Load Session file) to run. The template defines which agenda to run, on which load generator, and for how long.
Load Session File	The name of the Load Session results file (*.ls) that will store the load session results. If you do not enter a name, the template file name will be used (and the .tpl suffix will be replaced with .ls) This file will be saved in the workspace unless you enter a different path.
Execution duration	The length of time (in seconds) that the test is to run. This setting overrides the template definitions.
Allocate floating Virtual Clients	The number of Virtual Client licenses to allocate when using WebRM License Server.
Allocate floating Probing Clients	The number of Probing Client licenses to allocate when using WebRM License Server.

Defining the Generate Analytics Report Build Step

If you select **Generate WebLOAD Analytics Report** as a build step, the following appears:

Generate WebLOAD Analytics Report

Input Load Session File: demo_\${BUILD_NUMBER}.ls

Output format: JUNIT

Portfolio Name:

Report name: demo

Compare to previous builds: 0

Compare to sessions:

Advanced... (highlighted)

Delete

Figure 10: Generate WebLOAD Analytics Report – Defining

1. Optionally click **Advanced** to further refine your definitions. The following appears:

Generate WebLOAD Analytics Report

Input Load Session File: demo_\${BUILD_NUMBER}.ls

Output format: JUNIT

Portfolio Name:

Report name: demo

Compare to previous builds: 0

Compare to sessions:

Location:

Delete

Figure 11: Generate WebLOAD Analytics Report – Advanced Definition

2. Enter information in the fields, as described in Table 4.

Table 4: WebLOAD Analytics Report Parameters

Parameter	Description
Input Load Session File	The Load Session file name (*.ls) for which to generate a report. The Load Session file was created by executing a WebLOAD Console load session.
Output format	The output format of the report. Select one of the following: <ul style="list-style-type: none"> • JUNIT - JUnit XML Report. Can be used in conjunction with the Post Build step Publish JUnit test result report. Available from WebLOAD 10.1. • HTML - HTML Report • DOC - Microsoft Word document • ODT - OpenDocument format • XLS - Microsoft Excel spreadsheet • XLSX - Microsoft Excel spreadsheet (2007 and newer) • RTF - Rich Text Format • PDF - Portable Document Format • CSV - Comma Separated Values (text) • RAW - Export the report raw data in CSV (text) format
Portfolio Name	The templates portfolio to use in order to generate the report. You can specify any portfolio created in the WebLOAD Analytics tool. Note that the predefined portfolios are: <ul style="list-style-type: none"> • Summary Portfolio – used by default • Regression Portfolio – use when comparing several sessions. Available from WebLOAD 10.1 • Session Comparison Portfolio – use when comparing to a single baseline session • Extended Summary Portfolio
Report name	The name of the generated report. The name is used both as the report title and as the output filename. If you do not specify a name, the application provides a default name.
Compare to previous builds	Compare to the session results of the defined number of previous builds. For example, if you enter 8, the report will compare the current build with the past eight builds. The previous result files (.ls) are expected to have the same name, in the previous build artifacts.
Compare to sessions	Compare to these specified Load Session files. The sessions should be entered as a comma separated list of *.ls files. Use absolute paths, or paths that are relative to the workspace.
Location	The location of the generated report, relative to the workspace.

Common Usage Scenarios of an Analytics Build Action

Generate a summary report of the last build

Used to present the results of the recently-run load session.

- Input Load Session File: The Load Session from the Execute Console command.
- Output format: PDF, DOC or HTML
- Portfolio Name: Summary Portfolio

Compare last build to X number of previous builds

Used for regression testing against several previous runs of the Jenkins job (builds).

- Input Load Session File: The Load Session from the Execute Console command.
- Output format: PDF, DOC or HTML
- Portfolio Name: Regression Portfolio
- Compare to previous builds: Number of previous builds to compare with, for example 8.

Compare last build to a known session

Used to compare the results to a base / benchmark load session.

- Input Load Session File: The Load Session from the Execute Console command.
- Output format: PDF, DOC or HTML
- Portfolio Name: Session Comparison Portfolio
- Compare to sessions: Path to the base session file, for example C:\base_results.ls

Generate a report in JUnit format for the last build

Used to export the validation results in JUnit format.

- Input Load Session File: The Load Session from the Execute Console command.

- Report Name: Enter a name for the report. For example, entering 'result' will create a 'result.xml' file.
- Output format: JUNIT
- Portfolio Name: Summary Portfolio



Note: Only templates with Validation Rules are exported.



Note: You can optionally instruct Jenkins to display the generated report. To do so, click **Add post-build action**, select **Publish JUnit test result report** and specify the report file name (result.xml in our example).

Running WebLOAD using the Open API

You can run WebLOAD using a new REST Open API. With this API, you can remotely activate a test, remotely create Analytics Reports, and access any statistics that WebLOAD saves during a test.

Accessing the REST Open API

To access the REST Open API:

1. Install the WebLOAD Dashboard, which is part of the regular WebLOAD installation, as follows:
In the Select Components window of the WebLOAD installation wizard, select either of the following options:
 - Select **Full Installation** and check the **WebLOAD Dashboard** checkbox to have the dashboard installed locally as part of a full installation.
 - Select **Dashboard Server** to only install a central dashboard server.

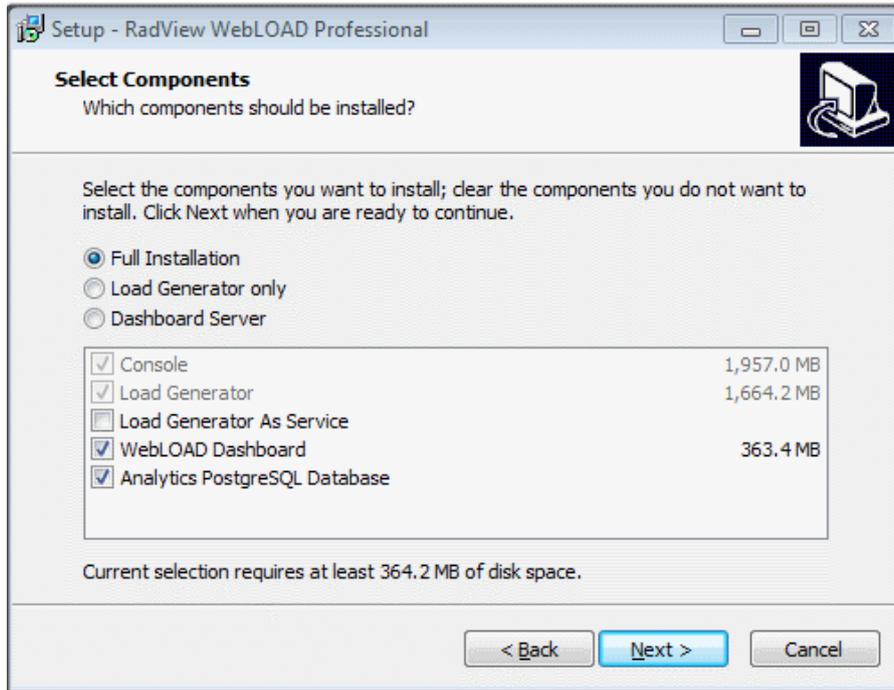


Figure 12: Web Dashboard Installation Options

2. Go to <http://localhost:8080/api-docs/docs.html>.

Understanding how to use the WebLOAD REST Open API

A description of the method and its parameters is embedded within each method in the Open API.

- The result of the Get Analytics method is an Analytics report, in the specified format (PDF, by default).
- The result of the Post Session method is that a specified load session starts running in the Console.
- The result of the other Open API methods is retrieved data in JSON format.