

WebLOAD Plugin

Plugin Information

View WebLOAD Load Testing [on the plugin site](#) for more information.

This plugin allows Jenkins to execute [RadView WebLOAD](#) Console Load sessions and WebLOAD Analytics report generation.

About this Plugin

This plugin allows you to trigger a WebLOAD Console load-test, and generate reports using WebLOAD Analytics. The plugin is intended to be used with WebLOAD version 10.1 and up.

Installation Notes

1. Install a full installation of WebLOAD on the machine that will run the WebLOAD jobs (this is either the master, or better, a slave).
2. The Load-Generator (if used) do not require any special set-up
3. If running Jenkins as a Service, make sure it running using a normal user, and not the Local System Account

User Guide

Set up a job - execute a load session

1. Install a full installation of WebLOAD on the machine running the job (this is either the master, or better, a slave)
2. Go to the Jenkins Server home page.
3. Click the **New Job** link or select an existing job.
4. Enter a Job name (for a new job).
5. Select **Build a free-style software project** and click **OK**.
6. In the Project Configuration section scroll down to the **Build** section.
7. Expand the **Add build step** drop-down and select **Execute WebLOAD Load Session**
8. Set the **Template File** - create this file in WebLOAD Console - it defines which scripts to run, on which load-generators and what is the load session schedule
9. Set the **Load Session File** - this is where the results will be saved. It is recommended you use the build-number parameter, to differentiate between the runs, for example, `my_result_${BUILD_NUMBER}.ls`

Set up a job - generate Analytics report

Follow the steps above to create a new Job, or more commonly, add another build step to the previous Job.*

1. Expand the **Add build step** drop-down and select **Generate WebLOAD Analytics Report**
2. Set **Input Load Session File** to the .ls file (normally, this will be the name set in step 9 above)
3. Set **Output format** to the desired report format, normally HTML or PDF. You can also use JUNIT to create an xml report that can be used in conjunction with Jenkins's **Publish JUnit Test Result report** Post Build action.
4. Set **Portfolio Name** to define which templates will be included in the report. By default the **Summary Portfolio** is used. The **Regression Portfolio** is recommended when using **Compare to previous builds** is used.
5. Set **Report name**. Used as the report title and also the output filename
6. Set **Compare to previous builds** to compare the current run to the specified number of previous builds.
7. Set **Compare to sessions** to a path of a baseline .ls session file to compare to (or a comma-separated list of such files)