**Jenkins Script Console and Jenkins CLI**

Jenkins features a nice Groovy script console which allows to run arbitrary scripts on the Jenkins server or on slave nodes. This feature can be accessed from the "manage Jenkins" link, typically at your <http://server/jenkins/script>.

http://<PATHTOJENKINS>:8080/script

Some useful scripts available on

https://wiki.jenkins-ci.org/display/JENKINS/Jenkins+Script+Console

# Remote access with CLI

User can execute groovy scripts remotely sending post request to /script/ url or /scriptText/ to have response returned without the html wrapping.

Jenkins has a built-in command line interface that allows you to access Jenkins from a script or from your shell. This is convenient for automation of routine tasks, bulk updates, trouble diagnosis, and so on.

## Obtaining the CLI client

You can download the JAR file for the client from the URL "/[jnlpJars/jenkins-cli.jar](https://jenkins.example.com/jnlpJars/jenkins-cli.jar)" on your Jenkins server, e.g. <https://jenkins.example.com/jnlpJars/jenkins-cli.jar>  
In theory, the CLI JAR is dependent on the version of Jenkins, but in practice, we expect to be able to retain compatibility between different versions of Jenkins. In case of problems, just re-download the latest JAR from your Jenkins server.

This interface is accessed via the Jenkins CLI client, which is a Java JAR file distributed with Jenkins.

**Jenkins CLI** offers the possibility to execute groovy scripts remotely using groovy command or execute groovy interactivelly via groovysh. However, once again curl can be used to execute groovy scripts by making use of bash Command Substitution. In the following examplesomescript.groovy is a groovy script in the current working directory.

## Running a CLI command

The general syntax is as follows (the design is similar to tools like svn/git):

java -jar jenkins-cli.jar [-s JENKINS\_URL] command [options...] [arguments...]

JENKINS\_URL can be specified via the environment variable $JENKINS\_URL. This environment variable is automatically set when Jenkins fork a process during builds, which allows you to use Jenkins CLI from inside the build without explicitly configuring the URL.

## Getting help

>java -jar jenkins-cli.jar -s http://localhost:8080 help

**List the available jobs**

>java -jar jenkins-cli.jar -s http://localhost:8080 list-jobs

**List the current added plugins**

>java -jar jenkins-cli.jar -s http://localhost:8080 list-plugins

**Enable job MyJob1**

java -jar jenkins-cli.jar -s http://localhost:8080 enable-job MyJob1

**Disable job MyJob1**

java -jar jenkins-cli.jar -s http://localhost:8080 disable-job MyJob1

**Restart the Jenkins**

java -jar jenkins-cli.jar -s http://localhost:8080 restart

**Safely Restart the Jenkins**

java -jar jenkins-cli.jar -s http://localhost:8080 safe- restart

**safely shutdown the Jenkins**

java -jar jenkins-cli.jar -s http://localhost:8080 safe- shutdown

The list of the available commands depends on the server you are talking to. Visit <https://jenkins.example.com/cli> or use 'help' command to list them all:

java -jar jenkins-cli.jar -s https://jenkins.example.com help [command]

More detailed help for individiual commands can be found by adding the command name after help (e.g. help build).  
The same information is available via the web UI, by clicking on a command name on the Jenkins CLI page.

## <http://javadoc.jenkins-ci.org/>