

# Administering Jenkins

## JENKINS\_HOME directory

Jenkins needs some disk space to perform builds and keep archives. You can check this location from the configuration screen of Jenkins.

By default, this is set to `~/ .jenkins`, but you can change this in one of the following ways:

- Set "JENKINS\_HOME" environment variable to the new home directory before launching the servlet container.
- Set "JENKINS\_HOME" system property to the servlet container.
- Set JNDI environment entry "JENKINS\_HOME" to the new directory.

See the [container specific documentation collection](#) for more about how to do this for your container.

You can change this location after you've used Jenkins for a while, too. To do this, stop Jenkins completely, move the contents from old JENKINS\_HOME to the new home, set the new JENKINS\_HOME, and restart Jenkins.

JENKINS\_HOME has a fairly obvious directory structure that looks like the following:

```
JENKINS_HOME
+- config.xml      (jenkins root configuration)
+- *.xml          (other site-wide configuration files)
+- userContent    (files in this directory will be served under your http://server/userContent/)
+- fingerprints   (stores fingerprint records)
+- plugins        (stores plugins)
+- workspace      (working directory for the version control system)
  +- [JOBNAME]    (sub directory for each job)
+- jobs
  +- [JOBNAME]    (sub directory for each job)
    +- config.xml (job configuration file)
    +- latest     (symbolic link to the last successful build)
    +- builds
      +- [BUILD_ID] (for each build)
        +- build.xml (build result summary)
        +- log       (log file)
        +- changelog.xml (change log)
```

## Back up and restore

All the settings, build logs, artifact archives are stored under the JENKINS\_HOME directory. Simply archive this directory to make a back up. Similarly, restoring the data is just replacing the contents of the JENKINS\_HOME directory from a back up.

Back ups can be taken without stopping the server, but when you restore, please do stop the server.

## Moving/copying/renaming jobs

You can:

1. Move a job from one installation of Jenkins to another by simply copying the corresponding job directory.
2. Make a copy of an existing job by making a clone of a job directory by a different name.
3. Rename an existing job by renaming a directory. Note that the if you change a job name you will need to change any other job that tries to call the renamed job.

Those operations can be done even when Jenkins is running. For changes like these to take effect, you have to click "reload config" to force Jenkins to reload configuration from the disk.

## Batch renaming jobs

Replacing spaces in job names with underscores

```
$ rename 's/\s/_/g' *
```

## Archive unused jobs

Sometimes you want to remove a job from Jenkins but do so in such a way that you can resurrect it later, if the need arises. You can do this by going to \$JENKINS\_HOME and create an archive of the job directory. The following command illustrates how to archive a job 'xyz' and remove it.

```
$ cd $JENKINS_HOME/jobs
$ tar czf xyz.tgz xyz
// go to Jenkins GUI "Manage Jenkins" page and "Reload Configuration from Disk"
```

As long as you are not building the xyz project while you create an archive, you can do this operation without taking Jenkins offline.

- See also the ["Shelve Project" plugin](#).

## Script Console

Useful for trouble-shooting, diagnostics or batch updates of jobs Jenkins provides a script console which gives you access to all Jenkins internals. These scripts are written in [Groovy](#) and you'll find some samples of them in this [page](#).

## URL Options

```
http://[jenkins-server]/[command]
```

where [command] can be

- `exit` shutdown jenkins
- `restart` restart jenkins
- `reload` to reload the configuration