

General | Source Code Management | Build Triggers | Build Environment | Build | Post-build Actions

Description: `Simple job to release a staging repository to maven central`
[Safe HTML] [Preview](#)

- Enable project-based security
- Discard old builds [?](#)
- GitHub project
- Promote builds when... [?](#)

Rebuild options:

- Rebuild Without Asking For Parameters [?](#)
- Disable Rebuilding for this job
- This build requires lockable resources
- This project is parameterized [?](#)

String Parameter X [?](#)

Name: [?](#)

Default Value:

Description: `staging repository id to release` [?](#)
[Safe HTML] [Preview](#)

Trim the string [?](#)

Add Parameter ▼

- Throttle builds [?](#)
- Disable this project [?](#)
- Execute concurrent builds if necessary [?](#)

JDK: [?](#)

JDK to be used for this project

- Restrict where this project can be run [?](#)

[Save](#) [Apply](#)

Advanced...

Jenkins > test >

Source Code Management

None

Git

Build Triggers

- Trigger builds remotely (e.g., from scripts) ?
- Build after other projects are built ?
- Build periodically ?
- Build when another project is promoted
- Gerrit event ?
- GitHub Pull Request Builder
- GitHub hook trigger for GITScm polling ?
- Poll SCM ?

Build Environment

- Delete workspace before build starts
- Use secret text(s) or file(s) ?
- Provide Configuration files ?
- Abort the build if it's stuck
- Add timestamps to the Console Output
- Prepare SonarQube Scanner environment ?
- Run Xvnc during build ?
- SSH Agent

Credentials

X

Specific credentials Parameter expression

genie.cdi (ssh://genie.cdi@projects-storage.eclipse.org) Add

Add

Ignore missing credentials

- Set GitHub commit status with custom context and message (Must configure upstream job using GHPRB trigger) ?
- Setup Kubernetes CLI (kubect!) ?
- With Ant ?

Save
Apply

Build

Jenkins ▶ test ▶

Execute shell

Command

```
TOOLS_PREFIX='/opt/tools'
JAVA_PREFIX="${TOOLS_PREFIX}/java/oracle"
MVN_HOME="${TOOLS_PREFIX}/apache-maven/latest"
JAVA_HOME="${JAVA_PREFIX}/jdk-8/latest"
PATH="${MVN_HOME}/bin:${JAVA_HOME}/bin:/usr/local/sbin:/usr/local/bin:

# Maven plugins
HELP_PLUGIN='org.apache.maven.plugins:maven-help-plugin:3.1.0'
NEXUS_PLUGIN='org.sonatype.plugins:nexus-staging-maven-plugin:1.6.7'
NEXUS_PLUGIN_PARAMS='-DnexusUrl=https://oss.sonatype.org/ -DserverId=c

mvnq() {
    # filter out progress reports (-B) and download details
    mvn -B "$@" | grep -v '^\[INFO\] Download'
}

mvnq ${NEXUS_PLUGIN_PARAMS} ${NEXUS_PLUGIN}:rc-list
echo "Releasing repositoryID=$ID"
mvnq -DstagingRepositoryId="$ID" ${NEXUS_PLUGIN_PARAMS} ${NEXUS_PLUGIN
```

See the list of available environment variables

[Advanced...](#)[Add build step ▼](#)

Post-build Actions

[Add post-build action ▼](#)

Jenkins ver. 2.176.2

[Save](#)[Apply](#)