

---

# **jgitver Documentation**

**Matthieu Brouillard**

**Aug 08, 2018**



<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Clean git history</b>	<b>7</b>
<b>3</b>	<b>Maven usage</b>	<b>9</b>
3.1	All installation methods . . . . .	9
<b>4</b>	<b>Gradle usage</b>	<b>11</b>
<b>5</b>	<b>Maven configuration</b>	<b>13</b>
<b>6</b>	<b>Gradle configuration</b>	<b>15</b>
<b>7</b>	<b>Contribution</b>	<b>17</b>
<b>8</b>	<b>jgitver library</b>	<b>19</b>
<b>9</b>	<b>Indices and tables</b>	<b>21</b>



The main documentation for the site is organized into a couple sections:

- *[Quickstart](#)*
- *[User documentation](#)*
- *[Developer documentation](#)*



# CHAPTER 1

---

## Introduction

---

`jgitver` in essence is a set of tool providing auto-computation of projects versioning and comes with plugins for maven-usage and gradle-usage.

For those who can wait go directly to maven-usage or gradle-usage pages.

Main features include:

- project version calculation (`semver` compatible)
- standardized but configurable computations
- **0** file modification and thus **0** additional commit
- plugins for maven-usage and gradle-usage

When activated, `jgitver` is able for example to compute version number *à la maven* as following:

but depending on your needs, you could also configure `jgitver` to produce:

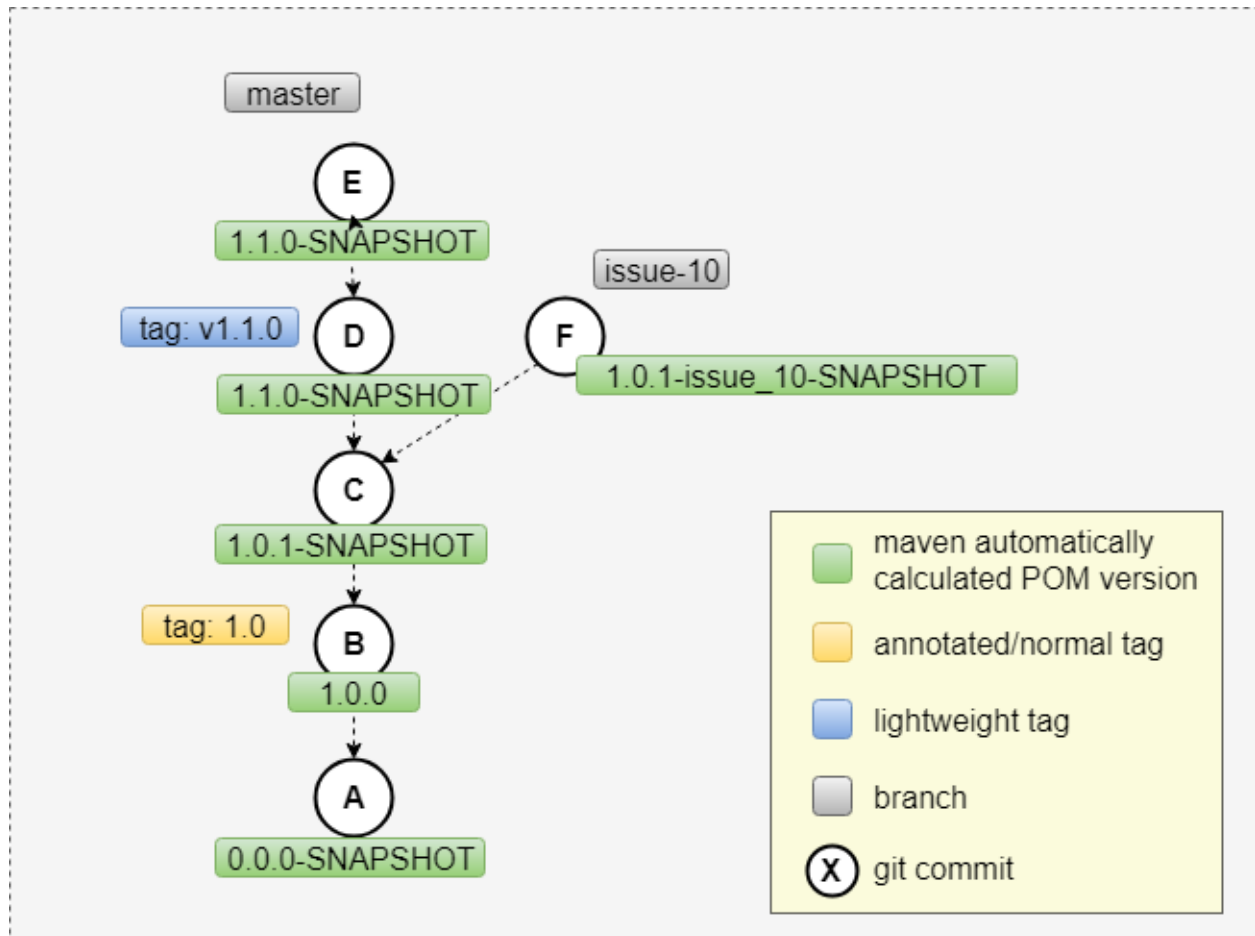


Fig. 1: jgitver in 'maven' mode

Using a maven like configuration, it is possible for your project to automatically:

- use SNAPSHOTs
- have dedicated versioning for branch
- have released versions for release tags



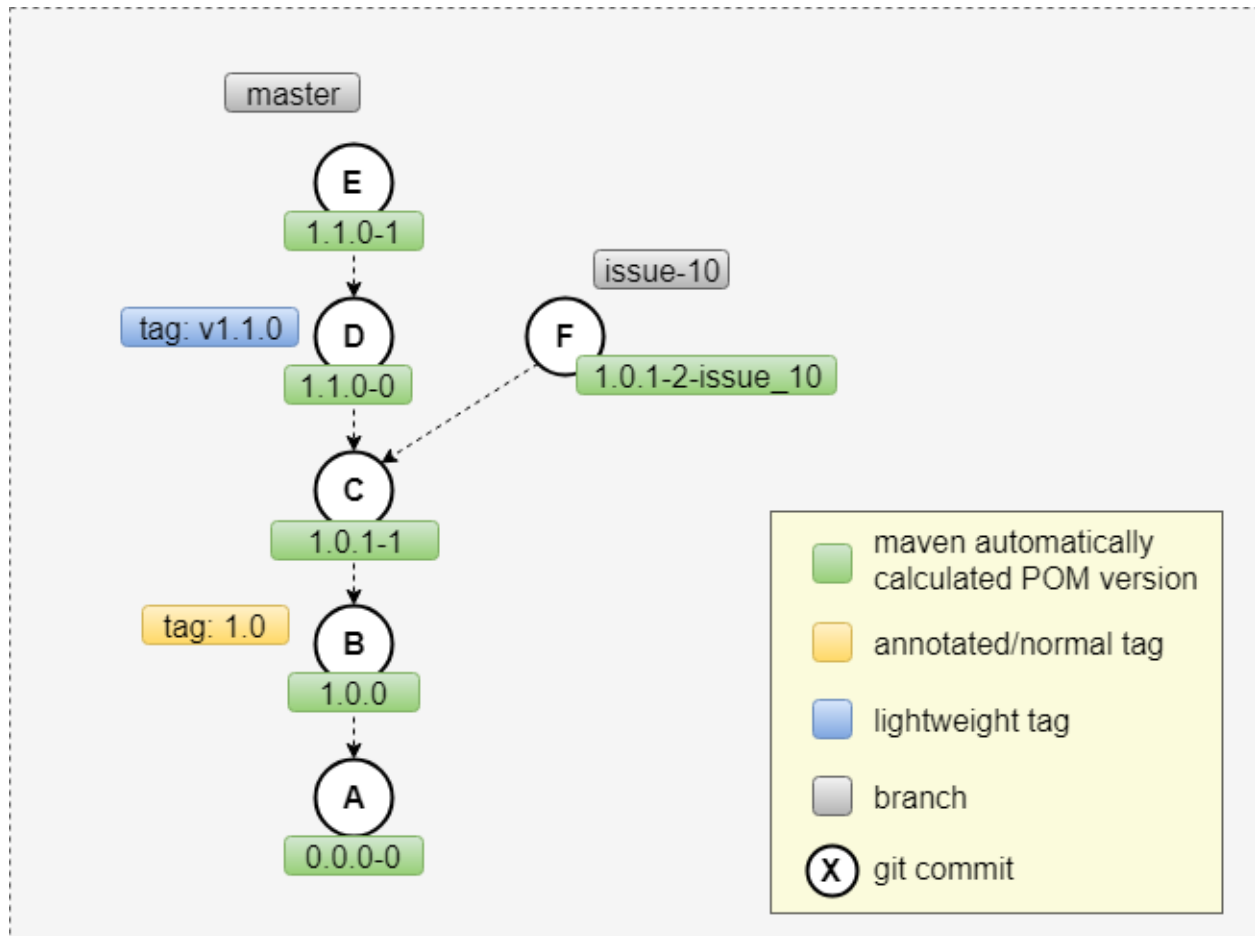


Fig. 2: jgitver producing unique versions for each commit

**In this mode:**

- versions are suffixed with the distance to the base tag
- have dedicated versioning for branch
- have released versions for release tags



## CHAPTER 2

---

### Clean git history

---

jgitver has been created with **DRY** & **KISS** principles in mind especially to keep a clean git history.

Having this in mind, jgitver has been built to allow to **NOT** modify any project descriptor (pom.xml or build.gradle).

The benefits from the above:

- no pollution of git history for unnecessary commits. Your git log contains **ONLY** business changes ; *tag* & *deploy* actions are enough for your project
- project version follows defined (*but configurable*) guidelines
- version can be automatically differentiated when working in branch

To summarize if you are familiar with the following unnecessary commits (*in red*) from *maven release* plugin

```

MINGW64:/d:/dev/projects/oss/maven/maven-war-plugin
$ git lg --oneline --no-color | grep -E --color=auto '^.*maven-release-plugin.*$'
* 344b55d [INFRA-16467] move components documentation out of CMS space
* a81cbdf [maven-release-plugin] prepare for next development iteration
* 28e74a4 [maven-release-plugin] prepare release maven-war-plugin-3.2.2
* 589cac9 [MWAR-303] - filtering of ${project.developers[0].id} does not work
* f1c6132 [MWAR-417] - Upgrade to plexus-interpolation to version 1.25
* c62549b Revert "[MWAR-317] - Upgrade to plexus-interpolation to version 1.25"
* bfb0a59 [MWAR-317] - Upgrade to plexus-interpolation to version 1.25
* ca152a6 Added Github Documentation.
* 62b3e37 [maven-release-plugin] prepare for next development iteration
* 3b292a3 [maven-release-plugin] prepare release maven-war-plugin-3.2.1
* a65162d [MWAR-416] - Upgrade plexus-archiver to 3.6.0
* 5e8fa13 [MNGSITE-332] - Changed download templates of plugins not to reference .md5 anymore
* 434c917 [MWAR-413] - Upgrade xstream to 1.4.10
* 8a3a3d4 [MWAR-414] - Upgrade mave-surefire/failsafe-plugin 2.21.0
* 2a16d37 [MWAR-401] - Upgrade the WAR lifecycle to use the maven-compiler-plugin 3.7.0
* da97036 [MWAR-412] - Upgrade parent to 31
* 17e5976 moved to git
* 9a4763f [maven-release-plugin] prepare for next development iteration
* 43a95d1 [maven-release-plugin] prepare release maven-war-plugin-3.2.0
* b530fad [MWAR-407] Binary files are modified during web.xml filtering; revert MWAR-404
* 9edaedd [MWAR-410] Upgrade plexus-utils to version 3.1.0
* 87ab36b [MWAR-409] Upgrade maven-archiver to 3.2.0 / plexus-archiver 3.5 o Upgraded maven-archiver to
s in consequence to life the JDK minimum to JDK 7.
* 8b1b564 [maven-release-plugin] prepare for next development iteration
* 8a2cc82 [maven-release-plugin] prepare release maven-war-plugin-3.1.0
* 6e32e02 Undoing [MWAR-401] Upgrade the WAR lifecycle to use the maven-compiler-plugin 3.6.0. based on i
* 22d262d CANCEL VOTE.
* c26a483 [maven-release-plugin] prepare for next development iteration
* 8a13abd [maven-release-plugin] prepare release maven-war-plugin-3.1.0
* 1d4dc9d [MWAR-404] <filteringDeploymentDescriptors>true</> is not honored o Added IT to prove wrong be
* fa71109 [MWAR-405] workaround XStream incompatibility with Java9 Contributed by Enrico Olivelli, review

```

Then be happy to hear that *git tag* and *mvn deploy* || *gradle publish* are just enough when using jgitver.

## CHAPTER 3

---

### Maven usage

---

jgitver operates as a maven core extension and needs a per project installation.

Installing is as simple as running the following command from the root directory of your project (*see below for other methods*):

```
sh -c "$(curl -fsSL https://raw.githubusercontent.com/jgitver/jgitver-maven-plugin/  
↪master/src/doc/scripts/install.sh)"
```

Congratulations, your project now uses jgitver, run *mvn validate* to be convinced that jgitver works

```
$ mvn validate  
[INFO] no suitable configuration file found, using defaults  
[INFO] Scanning for projects...  
[INFO] Using jgitver-maven-plugin [1.3.0] (sha1:↪  
↪ef8eec9f820d662e63a84f1210c377183e450cbd)  
[INFO] jgitver-maven-plugin is about to change project(s) version(s)  
[INFO]      fr.brouillard.oss::jgitver::0 -> 0.7.0-SNAPSHOT
```

---

**Note:** As jgitver uses the maven core extension mechanism it requires a maven version  $\geq 3.3.2$

---

### 3.1 All installation methods

All the installation scripts below will use the latest version available ; if you are updating find the [latest version here](#) or [there](#).

#### 3.1.1 curl

```
sh -c "$(curl -fsSL https://raw.githubusercontent.com/jgitver/jgitver-maven-plugin/  
↪master/src/doc/scripts/install.sh)"
```

### 3.1.2 wget

```
sh -c "$(wget https://raw.githubusercontent.com/jgitver/jgitver-maven-plugin/master/  
↪src/doc/scripts/install.sh -O -)"
```

### 3.1.3 manual installation

- Create a directory `.mvn` under the root directory of your project.
- Create file `.mvn/extensions.xml`
- Put the following content to `.mvn/extensions.xml` (adapt to [latest version](#)).

```
<extensions xmlns="http://maven.apache.org/EXTENSIONS/1.0.0" xmlns:xsi="http://www.w3.  
↪org/2001/XMLSchema-instance"  
  xsi:schemaLocation="http://maven.apache.org/EXTENSIONS/1.0.0 http://maven.apache.  
↪org/xsd/core-extensions-1.0.0.xsd">  
  <extension>  
    <groupId>fr.brouillard.oss</groupId>  
    <artifactId>jgitver-maven-plugin</artifactId>  
    <version>1.3.0</version>  
  </extension>  
</extensions>
```

## CHAPTER 4

---

### Gradle usage

---





## CHAPTER 5

---

### Maven configuration

---



## CHAPTER 6

---

### Gradle configuration

---



## CHAPTER 7

---

Contribution

---



## CHAPTER 8

---

jgitver library

---





---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`

design Theme <<https://sphinx-rtd-theme.readthedocs.io/en/latest/>>