Open in app   Get started

Published in Javarevisited

Wei Kang   Follow

Aug 6, 2021 · 3 min read · ▶ Listen

🔖 Save   🐦   f   in   🔗

How To Increment Versions For Maven Build Java Project — Part 1

One of the ways to increment versions for a Maven build Java project.

Photo by NeONBRAND on Unsplash

🏠   🔍   👤

Before we start, let's assume we have the following information in our `pom.xml`.

```
1    ...
2    <groupId>com.github.khubite</groupId>
3    <artifactId>test</artifactId>
4    <version>0.0.0</version>
5    ...
```

increment-version-maven-1-pom.xml hosted with ❤️ by **GitHub**   **view raw**

Reference: https://raw.githubusercontent.com/weikangchia/increment-version-maven/master/pom.xml

The plugins that we will need are:

- org.codehaus.mojo:build-helper-maven-plugin
  A plugin to assist with the Maven build lifecycle

- org.codehaus.mojo:versions-maven-plugin
  A plugin to help manage the versions of artifacts in a project's POM

```
1    <plugins>
2      ...
3      <plugin>
4        <groupId>org.codehaus.mojo</groupId>
5        <artifactId>build-helper-maven-plugin</artifactId>
6        <version>3.2.0</version>
7      </plugin>
8      <plugin>
9        <groupId>org.codehaus.mojo</groupId>
10       <artifactId>versions-maven-plugin</artifactId>
11       <version>2.8.1</version>
12     </plugin>
```

Reference: https://raw.githubusercontent.com/weikangchia/increment-version-maven/steps/1-add-plugins/pom.xml

Build Helper Maven Plugin

The Build Helper Maven plugin provides a goal, called the build-helper:parse-version. This goal parses a version string (default: `${project.version}`) and set properties containing the component parts of the version.

```
[propertyPrefix].majorVersion
[propertyPrefix].minorVersion
[propertyPrefix].incrementalVersion
[propertyPrefix].qualifier
[propertyPrefix].buildNumber
[propertyPrefix].nextMajorVersion
[propertyPrefix].nextMinorVersion
[propertyPrefix].nextIncrementalVersion
[propertyPrefix].nextBuildNumber
```

FYI: The default value for the propertyPrefix is `parsedVersion`

The goal also set the following properties:

```
[formattedPropertyPrefix].majorVersion
[formattedPropertyPrefix].minorVersion
[formattedPropertyPrefix].incrementalVersion
[formattedPropertyPrefix].buildNumber
[formattedPropertyPrefix].nextMajorVersion
[formattedPropertyPrefix].nextMinorVersion
[formattedPropertyPrefix].nextIncrementalVersion
[formattedPropertyPrefix].nextBuildNumber
```
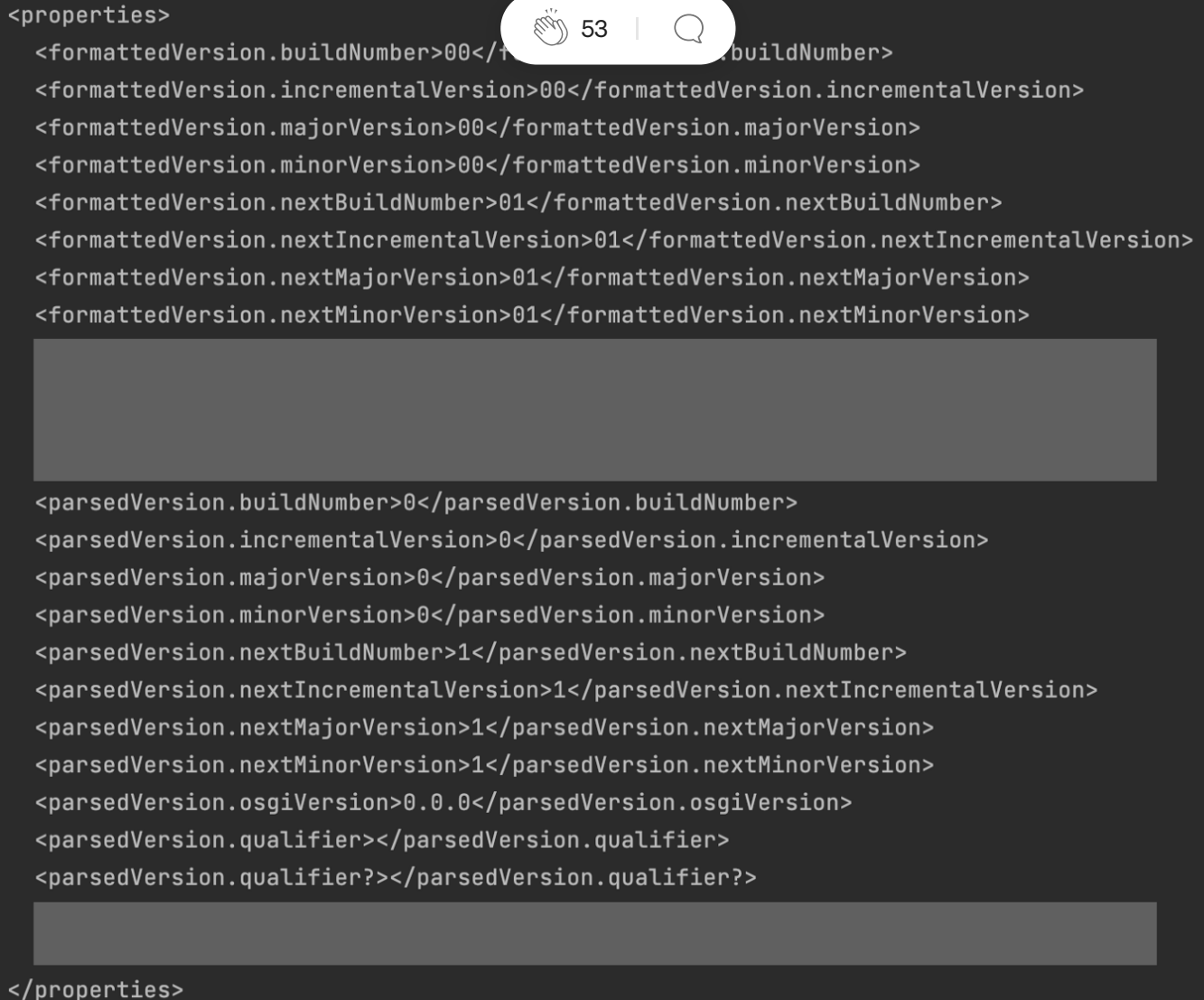
FYI: The default value for the formattedPropertyPrefix is `formattedVersion`

Now let's try with an example:

In the output, we will see the following properties:

```
<properties>
  <formattedVersion.buildNumber>00</f          buildNumber>
  <formattedVersion.incrementalVersion>00</formattedVersion.incrementalVersion>
  <formattedVersion.majorVersion>00</formattedVersion.majorVersion>
  <formattedVersion.minorVersion>00</formattedVersion.minorVersion>
  <formattedVersion.nextBuildNumber>01</formattedVersion.nextBuildNumber>
  <formattedVersion.nextIncrementalVersion>01</formattedVersion.nextIncrementalVersion>
  <formattedVersion.nextMajorVersion>01</formattedVersion.nextMajorVersion>
  <formattedVersion.nextMinorVersion>01</formattedVersion.nextMinorVersion>



  <parsedVersion.buildNumber>0</parsedVersion.buildNumber>
  <parsedVersion.incrementalVersion>0</parsedVersion.incrementalVersion>
  <parsedVersion.majorVersion>0</parsedVersion.majorVersion>
  <parsedVersion.minorVersion>0</parsedVersion.minorVersion>
  <parsedVersion.nextBuildNumber>1</parsedVersion.nextBuildNumber>
  <parsedVersion.nextIncrementalVersion>1</parsedVersion.nextIncrementalVersion>
  <parsedVersion.nextMajorVersion>1</parsedVersion.nextMajorVersion>
  <parsedVersion.nextMinorVersion>1</parsedVersion.nextMinorVersion>
  <parsedVersion.osgiVersion>0.0.0</parsedVersion.osgiVersion>
  <parsedVersion.qualifier></parsedVersion.qualifier>
  <parsedVersion.qualifier?></parsedVersion.qualifier?>



</properties>
```

Screenshot of the properties section of the output from the effective-pom

Bump A Patch Version

Format:

```
${parsedVersion.majorVersion}.${parsedVersion.minorVersion}.${parsedVersion.nextIncre
mentalVersion}
```

Output: 0.0.1

Output: 0.1.0

Bump A Major Version

Format: `${parsedVersion.majorVersion}.0.0`

Output: 1.0.0

Versions Maven Plugin

The Versions Maven plugin provides a goal, called the versions:set. This goal sets the current project's version and based on that change propagates that change onto any child modules as necessary.

Now let's try with an example:

```
mvn versions:set –DnewVersion=0.0.1
```

In the output, we will see the following messages to inform us that it has been updated successfully.

```
[INFO] Processing change of com.github.khubite:test:0.0.0 -> 0.0.1
[INFO] Processing com.github.khubite:test
[INFO]     Updating project com.github.khubite:test
[INFO]         from version 0.0.0 to 0.0.1
[INFO]
[INFO] ------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------
[INFO] Total time:  1.017 s
[INFO] Finished at: 2021-08-06T19:45:52+08:00
[INFO] ------------------------------------------------------------------
```

Screenshot of the output

⌂   🔍   👤

That's it, we have all the ingredients, we can now put them together to increment versions for our Maven build Java project.

First, let's update the Build Helper Maven plugin in the plugin section of our `pom.xml`.

Next, create 3 profiles to do the following:

- bump-patch
  This profile will be activated when given the property, `bumpPatch`

- bump-minor
  This profile will be activated when given the property, `bumpMinor`

- bump-major
  This profile will be activated when given the property, `bumpMajor`

Now let's try out the 3 profiles that we have just created.

bumpPatch

```
mvn validate —DbumpPatch
```

Output:

```
[INFO]      Updating project com.github.khubite:test
[INFO]          from version 0.0.1 to 0.0.2
[INFO]
[INFO] ------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------
[INFO] Total time:  1.518 s
[INFO] Finished at: 2021-08-06T20:06:15+08:00
[INFO] ------------------------------------------------------------------
```

Screenshot of the output

bumpMinor

```
mvn validate —DbumpMinor
```

Output:

```
[INFO] Processing change of com.github.khubite:test:0.0.2 -> 0.1.0
[INFO] Processing com.github.khubite:test
[INFO]      Updating project com.github.khubite:test
[INFO]          from version 0.0.2 to 0.1.0
[INFO]
[INFO] ------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------
[INFO] Total time:  1.695 s
[INFO] Finished at: 2021-08-06T20:09:49+08:00
[INFO] ------------------------------------------------------------------
```

Screenshot of the output

bumpMajor

```
[INFO] Processing change of com.github.khubite:test:0.1.0 -> 1.0.0
[INFO] Processing com.github.khubite:test
[INFO]     Updating project com.github.khubite:test
[INFO]         from version 0.1.0 to 1.0.0
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  1.597 s
[INFO] Finished at: 2021-08-06T20:10:55+08:00
[INFO] ------------------------------------------------------------------------
```

Screenshot of the output

And there we have it. I hope you have found this useful. Thank you for reading. You may find the complete `pom.xml` for this article from <u>here</u>.

Stay tuned for the next article, as I will show you how to increment a version that doesn't follow the <u>Semantic versioning</u> (e.g. 1.0.0_1.0.0)✌ .

How To Increment Versions For Maven Build Java Project — Part 2

One of the ways to increment versions for a Maven build Java project that does not follow Semantic versioning.

weikangchia.medium.com

If you are not a Medium member yet and want to become one, <u>click here</u>.

Join Medium with my referral link - Wei Kang

As a Medium member, a portion of your membership fee goes to writers
~~you read, and you get full access to every story~~

Enjoy the read? Reward the writer. [Beta]

Your tip will go to Wei Kang through a third-party platform of their choice, letting them know you appreciate their story.

Give a tip

Sign up for Sign up for Javarevsiited Newsletter

By Javarevisited

Subscribe to receive top 10 most read stories of Javarevsiited publication + other interesting material — delivered straight into your inbox, monthly. Take a look.

Your email

Get this newsletter

By signing up, you will create a Medium account if you don't already have one. Review our Privacy Policy for more information about our privacy practices.

About   Help   Terms   Privacy

Get the Medium app