

[Inicio](#) » [Java](#) » [Enterprise Java](#) » Prueba del complemento de lanzamiento de Maven Número de versión de incremento automático

## SOBRE MICHAEL REMIJAN



Michael Remijan es arquitecto de sistemas en el Banco de la Reserva Federal de St. Louis. Es coautor de 'EJB 3 In Action Second', un blogger activo en la comunidad Java EE, un guardián de Java EE y presentador de JavaOne. Ha desarrollado sistemas empresariales para comercio B2C y B2B, manufactura, astronomía, agricultura, telecomunicaciones, defensa nacional, salud y áreas financieras.



# Prueba del número de versión de aumento automático del complemento de lanzamiento de Maven

Publicado por: Michael Remijan en Enterprise Java 25 de febrero de 2020 0 334 Vistas

## Resumen

El complemento de lanzamiento de Maven es una herramienta muy poderosa y confío mucho en él para coordinar los lanzamientos de software. Por lo general, los números de versión de software siguen un

1.0.0.0-SNAPSHOT

formato simple. Pero recientemente tuve que agregar un calificador al número de versión, algo así como

1.0-beta-SNAPSHOT

o

1.0.0-fix-bug-description-SNAPSHOT

. Después de mi primer intento de agregar el calificador, el complemento de lanzamiento de Maven incrementó automáticamente el número de versión de una manera inesperada. Por lo tanto, necesitaba investigar el comportamiento de incremento automático de la versión del complemento de lanzamiento de Maven, pero hacerlo sin hacer un montón de etiquetas sin sentido en el repositorio de producción de Git (no tenía un repositorio de desarrollo para trabajar). El propósito de este blog es mostrar cómo se puede configurar el complemento de lanzamiento de Maven para que pueda ejecutarse localmente y no realizar cambios en el control de la fuente de producción.

## Descargo de responsabilidad

Esta publicación es únicamente informativa. Piensa críticamente antes de usar cualquier información presentada. Aprenda de él, pero finalmente tome sus propias decisiones bajo su propio riesgo.

## Requisitos

Hice todo el trabajo para esta publicación usando las siguientes tecnologías principales. Es posible que pueda hacer lo mismo con diferentes tecnologías o versiones, pero sin garantías.

- NetBeans 11.2
- Maven 3.3.9 (incluido con NetBeans)
- Maven-Release-Plugin 2.5.1

## BOLETIN INFORMATIVO

¡Los iniciados ya disfrutan de ag semanales y libros blancos grat

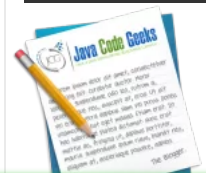
Únase a ellos ahora para obtene a las últimas noticias en el munc como información sobre Android otras tecnologías relacionadas.

Introduce tu correo electrónico

☐ Acepto los Términos y la P  
privacidad

Regístrate

## ÚNETE A NOSOTROS



Con 1.1. únicos n 500 al ubicado: principa relacion. Estar co buca d

no se encuentra el comando, debe asegurarse de que Git esté instalado independientemente de su IDE y en el

PATH

## Descargar

Visite mi página de GitHub <https://github.com/mjremijan> para ver todos mis proyectos de código abierto. El código para esta publicación se encuentra en: <https://github.com/mjremijan/thoth-maven-release-plugin>

## Instalar Git

Lo primero que deberá hacer es instalar Git. Incluso si Git está integrado en su IDE (como con NetBeans), el complemento de lanzamiento de Maven asume que los ejecutables que necesita están disponibles en la RUTA. Descargue, instale y configure PATH para su entorno.

## Etiqueta POM <scm>

La segunda cosa que deberá hacer es configurar la etiqueta POM <scm>. Cuando se ejecuta el complemento de lanzamiento de Maven, utiliza los valores de etiqueta <scm> para identificar la ubicación del sistema de control de origen.

La configuración relevante está debajo, vea el pom.xml completo en GitHub.

Como quiero ejecutar localmente y no conectarme a un sistema de control de fuente de producción, mi ejemplo usa Git. Git funciona muy bien sin necesidad de un servidor. Echemos un vistazo a la etiqueta <scm>.

## Listado 1 - Etiqueta <scm>

```
1 <scm>
2 <url>scm:git:file://D:/Projects/thoth-maven-release-plugin/.git</url>
3 <connection>
4   scm:git:file://D:/Projects/thoth-maven-release-plugin/.git</connection>
5 <developerConnection>
6   scm:git:file://D:/Projects/thoth-maven-release-plugin/.git
7 </developerConnection>
8 </scm>
```

El código en el Listado 1 muestra mi configuración para la etiqueta <scm>. Obviamente, la referencia a la

.git

carpeta del proyecto en el sistema de archivos de mi computadora debe destacarse. **Debe cambiar este valor para su computadora.** ¡Recuerde, el objetivo es trabajar localmente! Esta configuración te lleva allí.

Después de configurar la etiqueta <scm>, lo siguiente que debe hacer es configurar el complemento maven-release. Veamos eso a continuación.

## POM maven-release-plugin

Lo tercero que debe hacer es configurar el artefacto maven-release-plugin. Esta configuración es para el repositorio de preparación de Maven.

La configuración relevante está debajo, vea el pom.xml completo en GitHub. Echemos un vistazo a esta configuración.

## Listado 2 - etiqueta maven-release-plugin

```
01 <build>
02   <plugins>
03     <plugin>
04       <groupId>org.apache.maven.plugins</groupId>
05       <artifactId>maven-release-plugin</artifactId>
06       <version>2.5.1</version>
07       <dependencies>
08         <dependency>
09           <groupId>org.apache.maven.shared</groupId>
10           <artifactId>maven-invoker</artifactId>
11           <version>2.2</version>
12         </dependency>
13       </dependencies>
14     </plugin>
15     <configuration>
16       <stagingRepository>
17         localForm::default::file://C:/Users/Michael/.m2/repository
18       </stagingRepository>
19     </configuration>
20   </plugins>
```



La línea 8 establece la dependencia de maven-invocador a la versión

```
2.2
```

. Maven 3.3 cambió el nombre del script (en Windows) de

```
mvn.bat
```

a

```
mvn.cmd
```

. El maven-release-plugin debajo de la versión

```
2.5.2
```

asume

```
mvn.bat
```

. Cambiar maven-invocador a la versión

```
2.2
```

cambia la suposición para

```
mvn.cmd
```

que se pueda usar Maven 3.3+ (Lucas, 2015). Si usa maven-release-plugin mayor o igual que la versión

```
2.5.2
```

, no necesita esta configuración de maven-invoker.

La línea 15 establece la ubicación del repositorio de preparación de Maven. Por lo general, tendría esto configurado

```
settings.xml
```

. Si no está configurado, obtendrá errores de compilación. Muestro cómo configurarlo aquí. Si lo tiene,

```
settings.xml
```

puede eliminar esta configuración.

Ahora que

```
pom.xml
```

está configurado, veamos a continuación cómo ejecutar esto.

## Comando MVN

Con la configuración ahora establecida, es hora de ejecutar el proyecto Maven y probar que todo está funcionando. El Listado 3 muestra las propiedades y modificadores para

```
mvn
```

.

## Listado 3 - comando mvn

```
1 mvn
2 -Dmaven.test.skip=true
3 -Dmaven.javadoc.failOnError=false
4 4 --batch-mode release:clean release:prepare release:stage
```

No hay mucho en esta línea de comando. La línea 4 es la más importante. Especifica los objetivos de Maven-Release-Plugin para ejecutar.

Ahora estás listo para la prueba. Ejecute el comando y vea qué sucede.

## Resumen

Si está utilizando un IDE como NetBeans que mira y recarga automáticamente los archivos cuando se detectan cambios en el sistema de archivos, puede abrir el

```
pom.xml
```

, luego ejecutar el




Recuperado de <https://stackoverflow.com/questions/29755620/failed-to-execute-goal-maven-releaseprepare> .

Publicado en Java Code Geeks con permiso de Michael Remijan, socio de nuestro programa JCG . Vea el artículo original aquí: Prueba del complemento de lanzamiento de Maven Número de versión de incremento automático

Las opiniones expresadas por los contribuyentes de Java Code Geeks son propias.

Etiquetado con: MAVEN

 ( 0 calificación, 0 votos )  
Debe ser un miembro registrado para calificar esto.

🗨 Inicie el debate 👁 334 reproducciones i 🐦 Tweet!

## ¿Quieres saber cómo desarrollar tus habilidades para convertirte en un Rockstar de Java?



¡Suscríbete a nuestro boletín para comenzar a rockear  
Para que comiences, ¡te ofrecemos nuestros eBooks más vendidos  
ahora mismo.  
**GRATIS!**

1. JPA Mini libro
2. Guía de resolución de problemas de JVM
3. Tutorial JUnit para pruebas unitarias
4. Tutorial de anotaciones de Java
5. Preguntas de la entrevista Java
6. Preguntas de la entrevista de primavera
7. Diseño de la interfaz de usuario de Android

y muchos más ....

☐ Acepto los Términos y la Política de privacidad

[Regístrate](#)

¿TE GUSTA ESTE ARTÍCULO? LEER MÁS DE JAVA CODE GEEKS

Deja una respuesta



Este sitio usa Akismet para reducir el spam. Aprende cómo se procesan los datos de tus comentarios.

Suscribir ▼

BASE DE CONOCIMIENTOS

- Cursos
- Ejemplos
- Minibooks
- Recursos
- Tutoriales

SOCIOS

Mkyong

LA RED CODE GEEKS

- Geeks de código .NET
- Java Code Geeks
- Geeks del código del sistema
- Geeks de código web

SALÓN DE LA FAMA

- Serie "Tutorial de aplicación completa de Android"
- 11 sitios web de aprendizaje en línea que debe consultar
- Ventajas y desventajas de la computación en la nube: ventajas y desventajas de la computación en la nube
- Tutorial de Android Google Maps
- Tutorial de Android JSON con Gson
- Tutorial
- Aplicación de servicios basados en la ubicación de Android: ubicación GPS
- Tutorial de preferencias rápidas de Android
- Diferencia entre Comparador y Comparable en Java
- GWT 2 Spring 3 JPA 2 Hibernate 3.5 Tutorial
- Mejores prácticas de Java - Vector vs ArrayList vs HashSet

ACERCA DE JAVA CODE GEEKS

JCGs (Java Code Geeks) es una comunidad en línea independiente enfocada en ser el mejor centro de recursos para desarrolladores Java a Java; dirigido al líder del equipo técnico (desarrollador senior), gerente de proyecto y desarrollador junior por igual. Los JCG sirven a las comunidades de Java, SOA, Agile y DevOps. Noticias diarias escritas por expertos en dominios, artículos, tutoriales, entrevistas, anuncios, fragmentos de código y proyectos de código abierto.

DESCARGO DE RESPONSABILIDAD

Todas las marcas comerciales y marcas comerciales registradas que aparecen en Java Code Geeks son propiedad de sus respectivos dueños. Java es una marca registrada de Oracle Corporation en los Estados Unidos y en otros países. Ejemplos Java Code Geeks no está conectado a Oracle Corporation patrocinado por Oracle Corporation.

