

¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
Ese apoyo que siempre quiso tener...

1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces,
HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
Gestor de contenidos (Alfresco)
Aplicaciones híbridas

Tareas programadas (Quartz)
Gestor documental (Alfresco)
Inversión de control (Spring)

Control de autenticación y
acceso (Spring Security)
UDDI
Web Services
Rest Services
Social SSO
SSO (Cas)

JPA-Hibernate, MyBatis
Motor de búsqueda empresarial (Solr)
ETL (Talend)

Dirección de Proyectos Informáticos.
Metodologías ágiles
Patrones de diseño
TDD

BPM (jBPM o Bonita)
Generación de informes (JasperReport)
ESB (Open ESB)



powered by

autentia

Hosting patrocinado por

enREDados

[Inicio](#)[Quienes somos](#)[Tutoriales](#)[Formación](#)[Empleo](#)[Colabora](#)[Comunidad](#)[Libro de Visitas](#)[Comic](#)

NUEVO ¿Quieres saber cuánto ganas en relación al mercado? pincha aquí...

[Ver cursos que ofrece Autentia](#)[Descargar comics en PDF y alta resolución](#)

[¡NUEVO!] 2008-09-01



2008-07-31



2008-07-08



2008-06-22

Estamos escribiendo un libro sobre la profesión informática y estas viñetas formarán parte de él. Puedes opinar en la sección [comic](#).

Tutorial desarrollado por



Alejandro Pérez García

Alejandro es socio fundador de Autentia y nuestro experto en J2EE, Linux y optimización de aplicaciones empresariales.

Ingeniero en Informática

Si te gusta lo que ves, puedes contratarle para impartir **cursos presenciales** en tu empresa o para **ayudarte en proyectos** (Madrid). Puedes encontrarme en [Autentia](#)

Catálogo de servicios de Autentia

[Descargar \(6,2 MB\)](#)[Descargar en versión comic \(17 MB\)](#)

[AdictosALTrabajo.com](#) es el Web de difusión de conocimiento de [Autentia](#).

[Catálogo de cursos](#)

Descargar este documento en formato PDF: [mavenLicensePlugin.pdf](#)

Fecha de creación del tutorial: 2008-10-17

maven-license-plugin Como gestionar la licencia de nuestros ficheros fuentes

Creación: 17-10-2008

Índice de contenidos

1. Introducción
2. Entorno
3. El fichero con la cabecera de la licencia
4. Configuración del proyecto
5. Comprobando si tenemos correctamente la licencia en la cabecera de los ficheros fuentes
6. Actualizando la cabecera de licencia en todos los ficheros fuente
7. Como se excluyen ficheros para que no se tengan en cuenta ni al comprobar ni al actualizar
8. Como poner la cabecera de la licencia en ficheros no reconocidos por el plugin
9. La configuración completa después de todos los cambios
10. Conclusiones
11. Sobre el autor

1. Introducción

Cuando construimos aplicaciones, lo normal es que estas estén bajo alguna tipo de licencia, ya sea esta free software (GPL, LGPL, ...), open source (MPL, Apache, ...) o una licencia comercial propietaria.

El caso es que lo habitual es poner en la cabecera de todos nuestros ficheros fuentes (.java, .xml, .jsp, .properties, ...) un texto indicando la propiedad (el copyright) y la licencia que "protege" ese código fuente.

Esto no es tarea fácil teniendo en cuenta que este texto debe ser un comentario y cada fichero fuente tiene su propio tipo de comentario, y sobre todo teniendo en cuenta, no el hecho de añadir el mensaje, sino su mantenimiento, comprobación y actualización.

Para solucionar todos estos problemas y simplificarlos casi hasta el absurdo viene a rescatarnos **maven-license-plugin** (<http://code.google.com/p/maven-license-plugin/>).

maven-license-plugin es un plugin de [Maven](#) (como su propio nombre indica ;) que se puede enganchar al [ciclo de vida](#) de [Maven](#), normalmente a la fase **verify**, para comprobar y/o actualizar el texto de la licencia en todos nuestros ficheros fuente (con independencia del tipo de estos ficheros).

En este tutorial vamos a ver como configurar y usar este plugin (este tutorial está basado en la estupenda documentación que proporciona su creador [mathieu.carbou](#) en la home del plugin).

Catálogo de servicios Autentia (PDF 6,2MB)

[En formato comic...](#)☐ Web☒ [www.adictosaltrabajo.com](#)

Últimos tutoriales

2008-10-17
[maven-license-plugin: cómo gestionar la licencia de nuestros ficheros fuentes](#)

2008-10-10
[Cypal Studio: plugin de GWT para Eclipse](#)

2008-10-10
[Planificación de tareas con Spring](#)

2008-10-09
[Tutorial de Google Calendar Sync](#)

2008-10-06
[Instalación de GWT 1.5](#)

2008-10-03
[Crear un portal web con OpenCms 7](#)

2008-09-30
[Introducción a UMLet](#)

2008-09-25
[Introducción a la gestión de contenidos en Java con Liferay](#)

2008-09-23
[Archiva: gestión de repositorios maven \(IV\)](#)

2008-09-19
[Archiva: gestión de repositorios maven \(III\)](#)

Últimas ofertas de empleo

2008-10-03
[Marketing - Experto en Marketing - MADRID.](#)

2008-10-01
[Atención a cliente - Call Center - MADRID.](#)

2008-09-11
[Otras Sin catalogar - BARCELONA.](#)

2. Entorno

El tutorial está escrito usando el siguiente entorno:

- Hardware: Portátil Asus G1 (Core 2 Duo a 2.1 GHz, 2048 MB RAM, 120 GB HD).
- Nvidia GEFORCE GO 7700
- Sistema Operativo: GNU / Linux, Debian (unstable), Kernel 2.6.26, KDE 3.5
- Java Sun 1.6.0_07
- Maven 2.0.9
- maven-license-plugin 1.3.2

2008-08-11
Atención a cliente - Call Center - MADRID.

2008-07-28
Comercial - Ventas - MALAGA.

Anuncios Google

3. El fichero con la cabecera de la licencia

El texto que se pondrá como cabecera en todos los ficheros fuentes, debe ser un fichero de texto plano sin ninguna marca de comentario en especial.

Lo podríamos guardar, por ejemplo, en `src/main/config/copyrightNotice.txt`

Aquí os pongo un ejemplo, suponiendo que usamos la licencia LGPL en nuestra aplicación:

```
Copyright (C) 2008 Autentia Real Business Solutions S.L.

This file is part of Autentia Framework.

Autentia Framework is free software: you can redistribute it and/or modify
it under the terms of the GNU Lesser General Public License as published by
the Free Software Foundation, version 3 of the License.

Autentia Framework is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License
along with Autentia Framework. If not, see <http://www.gnu.org/licenses/>.
```

Lo interesante de este fichero es que donde hemos puesto: "Copyright (C) 2008 Autentia Real Business Solutions S.L.", podríamos haber usado: "Copyright (C) \${year} \${author} <\${email}>", y el propio plugin se encargará de sustituir las variables con los valores que nosotros le demos.

4. Configuración del proyecto

Usar este plugin es muy sencillo, basta con añadir a nuestro `pom.xml` las siguientes líneas:

```
view plain print ?
01. ...
02. <pluginRepositories>
03. ...
04.   <pluginRepository>
05.     <id>mc-release</id>
06.     <name>Local Maven repository of releases </name>
07.     <url>http://mc-repo.googlecode.com/svn/maven2/releases </url>
08.     <snapshots>
09.       <enabled>>false</enabled>
10.     </snapshots>
11.     <releases>
12.       <enabled>>true</enabled>
13.     </releases>
14.   </pluginRepository>
15. ...
16. </pluginRepositories>
17. ...
```

De esta forma el plugin se descargará automáticamente.

También podemos no dar de alta este repositorio, y el plugin se descargará de los repositorios oficiales de Maven. Pero como su propio creador nos indica, desde que sale una versión hasta que se actualiza en los repositorios oficiales de Maven pasa un cierto tiempo, así que si queremos estar siempre a la última es mejor declarar el repositorio (por ejemplo, a la hora de escribir estas líneas en el repositorio oficial se encuentra la versión 1.3.1, mientras que si damos de alta el repositorio de su creador tendremos acceso a la versión 1.3.2).

5. Comprobando si tenemos correctamente la licencia en la cabecera de los ficheros fuentes

En el `pom.xml` añadiremos las siguientes líneas:

```
view plain print ?
01. ...
02. <plugins>
03. ...
04.   <plugin>
05.     <groupId>com.google.code.maven-license-plugin </groupId>
06.     <artifactId>maven-license-plugin</artifactId>
07.     <configuration>
08.       <header>src/main/config/copyrightNotice.txt </header>
09.     </configuration>
10.   </plugin>
11. ...
12. </plugins>
13. ...
```

Ahora lo lanzaremos desde línea de comandos con:

```
$ mvn license:check
```

Si estamos usando variables en la plantilla de la nota de licencia, podemos ejecutarlo con:

```
$ mvn license:check -Dyear=2008 -Dauthor=Autentia -Demail=proyecto@autentia.com
```

Si lo que queremos es que la comprobación se haga siempre en la fase **verify** (cosa muy recomendable), pondríamos las siguientes líneas en nuestro pom.xml:

```
view plain print ?
01. ...
02. <plugins>
03. ...
04. <plugin>
05. <groupId>com.google.code.maven-license-plugin </groupId>
06. <artifactId>maven-license-plugin</artifactId>
07. <configuration>
08. <header>src/main/config/copyrightNotice.txt </header>
09. </configuration>
10. <executions>
11. <execution>
12. <goals>
13. <goal>check</goal>
14. </goals>
15. </execution>
16. </executions>
17. </plugin>
18. ...
19. </plugins>
20. ...
```

Para realizar la verificación explícitamente, ejecutaríamos el comando:

```
$ mvn verify
```

y, teniendo en cuenta que ahora está "enganchado" al ciclo de vida de Maven, también se realizara la comprobación si ejecutamos:

```
$ mvn install
$ mvn deploy
...
```

6. Actualizando la cabecera de licencia en todos los ficheros fuente

Para actualizar la cabecera de licencia en todos los ficheros fuentes, basta con ejecutar:

```
$ mvn license:format
```

Se han hecho varias pruebas, y "actualizar" significa:

- Si el fichero no tiene licencia, se lo añade.
- Si la plantilla de la licencia (la que creamos en el punto 3) ha cambiado, se actualiza el texto de la licencia de cada fichero fuente. Es decir, no se vuelve a añadir la licencia, sino que se modifica el texto actual del fichero fuente.
- Si se ha modificado el texto de la licencia en algún fichero fuente, se actualiza el fichero fuente, restaurando el texto del fichero fuente para que coincida con la plantilla (la que definimos en el punto 3).

7. Como se excluyen ficheros para que no se tengan en cuenta ni al comprobar ni al actualizar

Por defecto ya hay un conjunto de ficheros y directorios que son excluidos (visitar "Default excludes" en <http://code.google.com/p/maven-license-plugin/wiki/Configuration> para ver la lista completa). Pero también podemos excluir todos los ficheros que queramos. Por ejemplo, en la configuración que ya tenemos en nuestro pom.xml, añadiríamos:

```
view plain print ?
01. ...
02. <configuration>
03. ...
04. <excludes>
05. <exclude>**/LICENSE.txt</exclude>
06. <exclude>**/COPYING.LESSER</exclude>
07. <exclude>**/COPYING</exclude>
08. </excludes>
09. ...
10. </configuration>
11. ...
```

Como se puede ver, estamos excluyendo los ficheros LICENSE.txt, COPYING.LESSER y COPYING, con independencia del directorio en el que se encuentren (*/*). Es decir, se admiten expresiones regulares, así que el sistema es bastante potente.

8. Como poner la cabecera de la licencia en ficheros no reconocidos por el plugin

El plugin reconoce una lista de extensiones bastante amplia (ver lista completa en "Supported file extensions" en <http://code.google.com/p/maven-license-plugin/>), pero no soporta todas las extensiones posibles. Aún así añadir nuevas extensiones es muy sencillo, por ejemplo, a nuestra configuración añadiríamos las siguientes líneas:

```
view plain print ?
01. ...
02. <configuration>
03. ...
04. <mapping>
05. <jsp>xml</jsp>
06. </mapping>
07. ...
08. </configuration>
09. ...
```

Con esto lo que le estamos diciendo es que los ficheros con extensión **jspx** (el nombre de la etiqueta) se interpreten como si fueran **xml** (el contenido de la etiqueta).

9. La configuración completa después de todos los cambios

Aquí os dejo la configuración final después de ir aplicando los diferentes cambios que hemos ido viendo:

```
view plain print ?
01. ...
02. <plugins>
03. ...
04. <plugin>
05. <groupId>com.google.code.maven-license-plugin</groupId>
06. <artifactId>maven-license-plugin</artifactId>
07. <configuration>
08. <header>src/main/config/copyrightNotice.txt</header>
09. <excludes>
10. <exclude>**/LICENSE.txt</exclude>
11. <exclude>**/COPYING.LESSER</exclude>
12. <exclude>**/COPYING</exclude>
13. </excludes>
14. <mapping>
15. <jsp>xml</jsp>
16. </mapping>
17. </configuration>
18. <executions>
19. <execution>
20. <goals>
21. <goal>check</goal>
22. </goals>
23. </execution>
24. </executions>
25. </plugin>
26. ...
27. </plugins>
28. ...
```

10. Conclusiones

Como podéis ver, a día de hoy prácticamente todo ya está hecho, así que siempre merece la pena buscar un buen rato en Internet, porque o encontramos nuestro problema resuelto o veremos tantas cosas que ya tendremos una buena idea de como resolver nuestro problema.

11. Sobre el autor

Alejandro Pérez García, Ingeniero en Informática (especialidad de Ingeniería del Software)

Socio fundador de Autentia (Formación, Consultoría, Desarrollo de sistemas transaccionales)

<mailto:alejandropg@autentia.com>

Autentia Real Business Solutions S.L. - "Soporte a Desarrollo"

<http://www.autentia.com>

- Puedes opinar sobre este tutorial [haciendo clic aquí](#).
- Puedes firmar en nuestro libro de visitas [haciendo clic aquí](#).
- Puedes asociarte al grupo AdictosAlTrabajo en XING [haciendo clic aquí](#).
- [Añadir a favoritos Technorati.](#)



Esta obra está licenciada bajo [licencia Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5](#)

Recuerda

[Autentia](#) te regala la mayoría del conocimiento aquí compartido ([Ver todos los tutoriales](#)). Somos expertos en: J2EE, Struts, JSF, C++, OOP, UML, UP, Patrones de diseño ... y muchas otras cosas.

¿Nos vas a tener en cuenta cuando necesites consultoría o formación en tu empresa?, ¿Vas a ser tan generoso con nosotros como lo tratamos de ser con vosotros?

Somos pocos, somos buenos, estamos motivados y nos gusta lo que hacemos ...

Autentia = Soporte a Desarrollo & Formación.

info@autentia.com

Formación en nuevas tecnologías

Servicio de notificaciones:

Si deseas que te enviemos un correo electrónico cuando introduzcamos nuevos tutoriales.

Formulario de subcripción a novedades:

E-mail

Tutoriales recomendados

Nombre	Resumen	Fecha	Visitas	pdf
Archiva: gestión de repositorios maven (I)	Archiva es un gestor de repositorios maven desarrollado por el equipo de Jakarta Maven. Aprenderemos a instalarlo, configurarlo y administrarlo en este primer tutorial.	2008-09-19	681	pdf
Búsqueda de dependencias para maven	Nuestro compañero nos da piestas muy útiles para encontrar librerías para maven según las necesidades de nuestro desarrollo.	2008-01-18	1438	pdf
Como generar con Maven un histórico de cambios del proyecto	Como generar con Maven un histórico de cambios del proyecto.	2007-08-23	1954	pdf
Creación de un repositorio maven interno accesible por SSH	En este tutorial os enseñamos a hacer un repositorio maven privado para vuestro equipo de desarrollo	2008-01-03	2972	pdf
Crear el sitio web de documentación del proyecto con Maven Site	En este tutorial vamos a ver cómo crear los sitios web de documentación del proyecto con Apache Maven.	2007-06-28	2423	pdf
Crear un repositorio remoto y como hacer una 'release' con varios proyectos en Maven y Eclipse	En este tutorial vamos a explicar como podemos trabajar teniendo varios proyectos relacionados en Maven y en Eclipse	2006-11-29	6224	pdf
Pruebas de integración con Maven	Este tutorial nos muestra un ejemplo para lanzar las pruebas de integración "engañando" a Maven para que no se lancen en la fase de test teniendo únicamente un módulo para ambas	2007-02-08	3766	pdf
Desarrollo de Plugins para Maven	En este tutorial vamos a hacer un ejemplo práctico de construcción de un plugin para Maven (es la herramienta de gestión de la construcción).	2007-05-04	3060	pdf
Maven, nunca antes resultó tan fácil compilar, empaquetar, ...	En este tutorial aprenderemos el uso de esta herramienta que nos permite compilar, empaquetar, generar documentación, pasar los test, preparar las builds de nuestros proyectos	2006-09-19	12031	pdf
Q, un plugin de Eclipse para gestionar Maven 2	en este tutorial aprendemos a instalar y usar Q, un plugin de Eclipse para gestionar Maven 2	2007-10-24	3495	pdf

Nota:

Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento. Los contenidos y comentarios de los tutoriales son responsabilidad de sus respectivos autores. En algún caso se puede hacer referencia a marcas o nombres cuya propiedad y derechos es de sus respectivos dueños. Si algún afectado desea que incorporemos alguna reseña específica, no tiene más que solicitarlo. Si alguien encuentra algún problema con la información publicada en este Web, rogamos que informe al administrador rcanales@adictosaltrabajo.com para su resolución.