

¿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)



[Home](#) | [Quienes Somos](#) | [Empleo](#) | [Tutoriales](#) | [Contacte](#)



CoNcept


Lanzado

TNTConcept versión 0.6 (12/07/2007)

Desde [Autentia](#) ponemos a vuestra disposición el software que hemos construido (100% gratuito y sin restricciones funcionales) para nuestra gestión interna, llamado TNTConcept (auTeNTia).

Construida con las últimas tecnologías de desarrollo Java/J2EE (Spring, JSF, Acegi, Hibernate, Maven, Subversion, etc.) y disponible en licencia GPL, seguro que a muchos profesionales independientes y PYMES os ayudará a organizar mejor vuestra operativa.

Las cosas grandes empiezan siendo algo pequeño Saber más en: <http://tntconcept.sourceforge.net/>

<p>Tutorial desarrollado por: Alejandro Perez García 2003-2007 Alejandro es Socio fundador de Autentia y nuestro experto en J2EE, Linux y optimización de aplicaciones empresariales.</p> <p>Si te gusta lo que ves, puedes contratarle para impartir cursos presenciales en tu empresa o para ayudarte en proyectos (Madrid).</p> <p style="text-align: center;">Contacta: alejandropg@autentia.com</p>	<p>NUEVO CATÁLOGO DE SERVICIOS DE AUTENTIA (PDF 6,2MB)</p> <p>www.adictosaltrabajo.com es el Web de difusión de conocimiento de www.autentia.com</p> <p style="text-align: center;">  autentia real business solutions </p> <p style="text-align: center;">Catálogo de cursos</p>
--	---

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

[Firma en nuestro libro de Visitas](#) <-----> [Asociarme al grupo AdictosAlTrabajo en eConozco](#)



Fecha de creación del tutorial: 2007-08-23

Como generar con Maven un histórico de cambios del proyecto

Creación: 20-08-2007

Índice de contenidos

- [1. Introducción](#)
- [2. Entorno](#)
- [3. El fichero changes.xml](#)
- [4. Cambios en nuestro pom.xml](#)
- [5. Modificando el site.xml](#)
- [6. Conclusiones](#)
- [7. Sobre el autor](#)

1. Introducción

Siempre resulta muy interesante tener un informe con los cambios que se van realizando en nuestro proyecto. El típico 'changelog' o 'history' o similar donde se reflejan las nuevas funcionalidades o correcciones que se han realizado en cada versión.

En Maven podemos encontrar el plugin 'maven-changes-plugin' (<http://maven.apache.org/plugins/maven-changes-plugin/>) que nos permite realizar esta tarea.

Este plugin montará un informe en el 'site' del proyecto con la información que tengamos en el fichero src/changes/changes.xml

Para mantener este fichero podemos elegir entre:

- Mantenerlo a mano.
- Integrarlo con el JIRA. El plugin ya viene preparado para esto
- Hacer algún programita o proceso intermedio que saque la información de nuestro repositorio de incidencias y la convierta al formato esperado por maven-changes-plugin.

Evidentemente la primera opción es la peor de todas ya que requiere más trabajo por nuestra parte. Yo os recomiendo hacer el esfuerzo para integrarlo con vuestro sistema de gestión de incidencias, ya que este esfuerzo se vera recompensado.

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).
- Sistema Operativo: GNU / Linux, Debian (unstable), Kernel 2.6.22, KDE 3.5
- Java 1.5
- Maven 2.0.7

3. El fichero changes.xml

Ya hemos dicho que este fichero debe estar en src/changes/ (en las últimas versiones del plugin se puede configurar donde lo queremos tener).

Este fichero tendrá un aspecto como esté (este ejemplo está sacado directamente de la documentación de maven-changes-plugin):

```
<document>
  <properties>
    <title>Changes Tester Project</title>
    <author_email="jrui@exist.com">Johnny R. Ruiz III</author>
  </properties>
  <body>
    <release version="1.1" date="2005-03-01" description="Subsequent release">
      <action dev="jrui" type="add">
        Added additional documentation on how to configure the plugin.
      </action>
      <action dev="aramirez" type="fix" issue="MPJIRA-11">
        Enable retrieving component-specific issues.
      </action>
      <action dev="jrui" type="remove" due-to="Allan Ramirez" due-to-email="aramirez@exist.com">
        The element type " link " must be terminated by the matching end-tag.
        Deleted the erroneous code.
      </action>
    </release>
    <release version="1.0" date="2005-01-01" description="First release">
      <action dev="jrui" type="update">
        Uploaded documentation on how to use the plugin.
      </action>
    </release>
  </body>
</document>
```

Veamos que significa cada elemento:

- <release>
 - version (obligatorio): nombre de la versión en la que han ocurrido los cambios. Para cada cambio habrá un elemento action anidado.
 - date (obligatorio): fecha en la que se ha publicado la versión. Es un texto libre, no requiere ningún formato especial.
 - description: Una descripción opcional para esta versión. Se mostrará en el resumen al principio del informe.
- <action>
 - dev (obligatorio): nombre del desarrollador que ha subido el cambio al repositorio de versiones (por ejemplo al Subversion). Debe coincidir con el id de uno de los desarrolladores descritos en el elemento <developers> del pom.xml.
 - type (obligatorio): tipo del cambio. Puede tomar los siguientes valores:
 - add: algo que se ha añadido nueva en esta versión.
 - fix: algo que se ha arreglado en esta versión.
 - remove: algo que se ha eliminado en esta versión.

- `update`: algo que se ha actualizado en esta versión.
- `issue`: Identificador de nuestro sistema de gestión de incidencias (por ejemplo el id del Bugzilla). Aunque este atributo es opcional, es muy recomendable ya que se generará un enlace en el informe que nos permitirá saltar directamente al bug en nuestro sistema de gestión de incidencias.
- `due-to`: Permite especificar el nombre de la persona que ha hecho el cambio. La diferencia con el atributo `dev` es que `due-to` refleja quien ha hecho el cambio, y `dev` quien lo ha subido al repositorio de versiones. Es decir, los `dev` son los que tienen permiso de escritura en el repositorio de versiones, mientras que los `due-to` no tienen permisos de escritura, así que tienen que hacer un parche y mandárselo a un `dev` para que este lo suba al repositorio.
- `due-to-email`: Correo electrónico del `dev`.

El aspecto del informe generado con este XML será algo como:

Changes Report

Release History		
Version	Date	Description
1.1	2005-03-01	Subsequent release
1.0	2005-01-01	First release

Release 1.1 - 2005-03-01		
Type	Changes	By
	Added additional documentation on how to configure the plugin. Thanks to Dennis Lundberg.	jruiz
	Enable retrieving component-specific issues. Fixes MPJIRA-11 .	aramirez
	The element type " link " must be terminated by the matching end-tag. Deleted the erroneous code. Thanks to Allan Ramirez .	jruiz

Release 1.0 - 2005-01-01		
Type	Changes	By
	Uploaded documentation on how to use the plugin.	jruiz

4. Cambios en nuestro pom.xml

Para que el maven-changes-plugin funcione correctamente vamos a hacer los siguientes cambios en el pom.xml.

Primero vamos a asegurarnos que tenemos declarado el sistema de gestión de incidencias:

```
...
<issueManagement>
  <system>Bugzilla</system>
  <url>http://servidor/bugzilla/</url>
</issueManagement>
...
```

Esta información la va a usar el plugin para crear los enlaces de los cambios en el changes.xml con el sistema de gestión de incidencias. **Mucho ojo al escribir la URL porque esta tiene que terminar en '/'**. Si no ponéis la '/' del final no funcionará correctamente.

Ahora vamos a declarar la lista de desarrolladores:

```
...
<developers>
  <developer>
    <id>alex</id>
    <name>Alejandro Pérez García</name>
    <email>alejandropg@autentia.com</email>
    <organization>Autentia Real Business Solutions</organization>
    <organizationUrl>http://www.autentia.com</organizationUrl>
  </developer>
  ...
</developers>
...
```

Como ya hemos dicho antes, el atributo `dev` del elemento `<action>` en el fichero `changes.xml`, debe coincidir con el elemento `<id>`.

Ahora vamos a indicarle a Maven que queremos que nos genere el informe:

```

...
<reporting>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-changes-plugin</artifactId>
      <configuration>
        <issueLinkTemplate>%URL%/show_bug.cgi?id=%ISSUE%</issueLinkTemplate>
      </configuration>
      <reportSets>
        <reportSet>
          <reports>
            <report>changes-report</report>
          </reports>
        </reportSet>
      </reportSets>
    </plugin>
  </plugins>
</reporting>
...

```

Cabe destacar como con el elemento `<issueLinkTemplate>` le podemos indicar como tiene que componer la URL para apuntar al bug en nuestro sistema de gestión de incidencias.

- `%URL%` representa la URL definida anteriormente con el elemento `<issueManagement>`. Fijaros como después de `%URL%` hemos puesto una `'/'`. Como la barra ya estaba en `issueManagement`, podría parecer que estamos poniéndola 2 veces, pero no, todo lo contrario, si aquí no ponemos la `'/'`, tampoco funcionará.
- `%ISSUE%` es el id del bug que hemos introducido en el fichero `changes.xml` en el atributo `issue` del elemento `<action>`.

Con esto, cada vez que generemos el site del proyecto se nos generará el informe con los cambios.

5. Modificando el site.xml

Simplemente recordar que para que en nuestro site aparezcan correctamente los informes (javadoc, taglist, changes, ...), deberemos añadir la variable de Velocity `${reports}` en el fichero `src/site/site.xml` podría quedar algo como:

```

<?xml version="1.0" encoding="UTF-8"?>
<project name="Proyecto de Autentia">
  <body>
    <menu name="Index">
      <item name="User guide" href="UserGuide.html"/>
      <item name="Developer guide" href="DeveloperGuide.html"/>
      <item name="maven-changes-plugin" href="maven-changes-plugin.html"/>
    </menu>
    ${reports}
  </body>
</project>

```

Con esto nos aparecerá en el menú del site (el menú de la izquierda) la lista con todos los informes que ayamos indicado que queremos generar.

6. Conclusiones

Ya hemos comentado en otros tutoriales la importancia de la correcta gestión de las incidencias. Esto lo debemos hacer siempre a través de una aplicación estilo Bugzilla o JIRA o Trac o Scarab o ... Y de hecho, el fichero `changes.xml` se nos queda muy corto para nuestros propósitos.

No deberíamos usar el fichero `changes.xml` para llevar la gestión de las incidencias; pero si es una muy buena forma para documentar en nuestro proyecto los cambios que se producen en cada versión. Por esto es esencial (como proponíamos al principio del tutorial) integrar nuestro sistema de gestión de incidencia con el el fichero `changes.xml`, de forma que este se genere de forma automática.

7. 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>



This work is licensed under a [Creative Commons Attribution-NonCommercial-No Derivative Works 2.5 License](https://creativecommons.org/licenses/by-nc-nd/2.5/).
[Puedes opinar sobre este tutorial aquí](#)



Recuerda

que el personal de [Autentia](#) te regala la mayoría del conocimiento aquí compartido ([Ver todos los tutoriales](#))

¿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?

info@autentia.com

Somos pocos, somos buenos, estamos motivados y nos gusta lo que hacemos
Autentia = Soporte a Desarrollo & Formación



[Autentia S.L.](#) Somos expertos en:
J2EE, Struts, JSF, C++, OOP, UML, UP, Patrones de diseño ..
 y muchas otras cosas

Nuevo servicio de notificaciones

Si deseas que te enviemos un correo electrónico cuando introduzcamos nuevos tutoriales, inserta tu dirección de correo en el siguiente formulario.

Subscribirse a Novedades	
e-mail	<input type="text"/>
	<input type="button" value="Enviar"/>

Otros Tutoriales Recomendados ([También ver todos](#))

Nombre Corto

[EJB 3.0 y pruebas unitarias con Maven, JUnit y Embedded JBoss](#)
[Proyecto con JSF Java Server Faces Myfaces, Maven y Eclipse: Hibernate \(segunda parte\)](#)
[Crear el sitio web de documentación del proyecto con Maven Site](#)
[Proyecto con JSF Java Server Faces Myfaces, Maven y Eclipse: aplicación multimódulo](#)
[Crear un repositorio remoto y como hacer una 'release' con varios proyectos en Maven y Eclipse](#)
[EJB 3.0, un ejemplo práctico con Maven y JBoss](#)
[Desarrollo de Plugins para Maven](#)
[Pruebas de integración con Maven](#)
[Maven, nunca antes resultó tan fácil compilar, empaquetar, ...](#)
[Proyecto con JSF Myfaces, Maven y Eclipse](#)

Descripción

En este tutorial Alejandro Pérez nos enseña como realizar test unitarios sobre EJB 3.0. Para ello se usará Maven, JUnit y Embedded JBoss

En este artículo se va a continuar con el desarrollo de la aplicación Myfaces JSF con Maven multimódulo que comenzamos en un tutorial anterior. Además también se tratará de la integración de Hibernate con las aplicaciones.

En este tutorial vamos a ver cómo crear los sitios web de documentación del proyecto con Apache Maven.

En este artículo se va a abordar el desarrollo de una aplicación Myfaces JSF con Maven que sea multimódulo.

En este tutorial vamos a explicar como podemos trabajar teniendo varios proyectos relacionados en Maven y en Eclipse

Este tutorial presenta un ejemplo sencillo donde se verá como desarrollar EJBs de sesión y de entidad, inyección de dependencias, llamar a los EJBs desde una aplicación Web, definición de un DataSource, y como configurarlo y hacerlo funcionar en JBoss, y

En este tutorial vamos a hacer un ejemplo práctico de construcción de un plugin para Maven (es la herramienta de gestión de la construcción).

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

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

En este tutorial vamos a aprender a construir una aplicación básica JSF (Java Server Pages) utilizando el Maven 2.0 y las bibliotecas de MyFaces. Lo mejor de todo es que para crear el ejemplo no vamos a programar ni una línea.

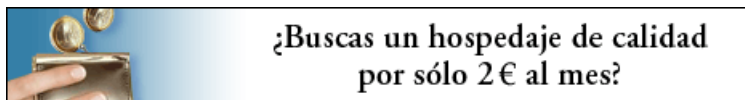
Nota: Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento.

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.

[Patrocinados por enredados.com Hosting en Castellano con soporte Java/J2EE](#)



www.AdictosAlTrabajo.com Optimizado 800X600