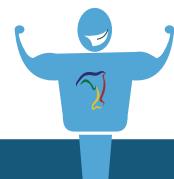


¿Qué ofrece Autentia Real Business Solutions S.L?

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

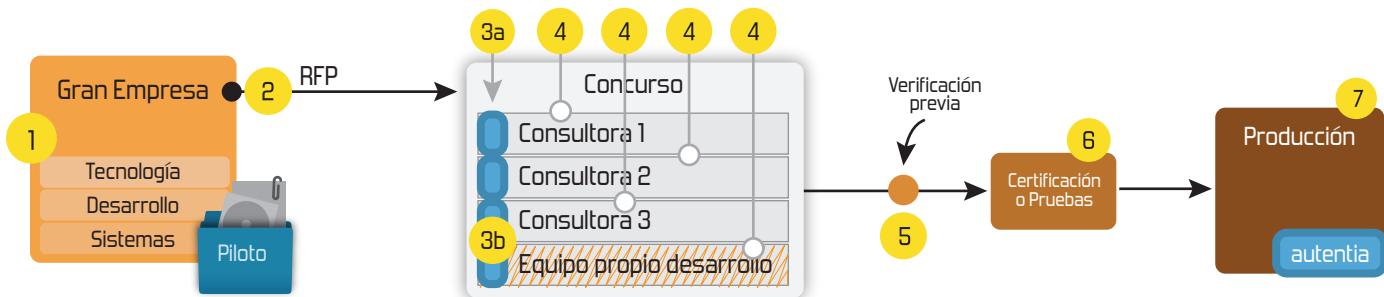
1. Desarrollo de componentes y proyectos a medida



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

Control de autenticación y acceso (Spring Security)
UDDI

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

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

Web Services
Rest Services
Social SSO
SSO (Cas)

Dirección de Proyectos Informáticos.
Metodologías ágiles
Patrones de diseño
TDD

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

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



Entra en Adictos a través de



E-mail

Contraseña

Entrar

Registrarme

Olividé mi contraseña

Inicio Quiénes somos Formación Comparador de salarios Nuestros libros Más



» Estás en: Inicio » Tutoriales » Solución de problemas comunes con la integración de maven en Eclipse.



Jose Manuel Sánchez Suárez

Consultor tecnológico de desarrollo de proyectos informáticos.

Puedes encontrarme en Autentia: Ofrecemos servicios de soporte a desarrollo, factoría y formación

Somos expertos en Java/J2EE

[Ver todos los tutoriales del autor](#)



Fecha de publicación del tutorial: 2014-10-10

Tutorial visitado 1 veces [Descargar en PDF](#)

Solución de problemas comunes con la integración de maven en Eclipse.

0. Índice de contenidos.

- 1. Introducción.
- 2. Entorno.
- 3. Despliegue del artefacto empaquetado en el servidor local.
- 4. Soporte para la compilación con AspectJ.
- 5. Plugin no soportado en el ciclo de vida de maven en Eclipse.
- 6. Referencias.
- 7. Conclusiones.

1. Introducción

En este tutorial vamos a responder a pequeñas, pero importantes, cuestiones relacionadas con la integración de un proyecto que dispone el soporte de maven en Eclipse.

Eclipse proporciona ahora, de forma nativa, el soporte para gestionar las dependencias de las librerías de un proyecto maven; para ello, proporciona una preinstalación de los plugins de m2e y m2e-wtp.

Con versiones anteriores de eclipse los problemas solían ser del soporte del plugin q4e para wtp o tener instalado el plugin de m2e, pero no el de m2e-wtp, con lo que funcionaba la gestión de dependencias de los módulos jars, pero no la de los módulos war; o funcionaba pero no funcionaba el proyecto con lo que no se podían desplegar en un servidor.

Con las últimas versiones de Eclipse esos problemas han desaparecido pero siguen otros y pueden surgir nuevos, y eso es justo lo que vamos a ver en este tutorial.

2. Entorno.

El tutorial está escrito usando el siguiente entorno:

- Hardware: Portátil MacBook Pro 15' (2.3 GHz Intel Core i7, 16GB DDR3).
- Sistema Operativo: Mac OS Mavericks 10.9.4
- Eclipse Luna.
- Apache Maven 3.2.3

3. Despliegue del artefacto empaquetado en el servidor local.

En este punto vamos a hablar tanto del plugin de maven para eclipse como de su integración con el plugin de Apache Tomcat o de cualquier otro plugin que nos permita desplegar nuestro proyecto web en un servidor de aplicaciones.

Según cómo configuremos el servidor local el despliegue se realiza en el directorio de despliegue del servidor o en el directorio del plugin del servidor en el directorio de instalación de Eclipse, véase,
`/WORKSPACE/.metadata/.plugins/org.eclipse.wst.server.core/tmp0/wtpwebapps`

Catálogo de servicios Autentia

www.autentia.com



Síguenos a través de:



Últimas Noticias

- » Curso JBoss de Red Hat
- » Si eres el responsable o líder técnico, considerate desafortunado. No puedes culpar a nadie por ser gris
- » Portales, gestores de contenidos documentales y desarrollos a medida
- » Comentando el libro Start-up Nation, La historia del milagro económico de Israel, de Dan Senor & Salu Singer
- » Screencasts de programación narrados en Español
- [Histórico de noticias](#)

Últimos Tutoriales

- » Integración de SonarQube en Eclipse.
- » Monitorización de Apache Tomcat con psi-probe.
- » Trident, un compañero de viaje para tratar con Storm
- » Cómo se trabaja con un Croma y para qué sirve
- » Creación de un módulo AMP de Alfresco con arquetipo

Tomcat v7.0 Server at localhost

Overview

General Information

Specify the host name and other common settings.

Server name: Tomcat v7.0 Server at localhost

Host name: localhost

Runtime Environment: Apache Tomcat v7.0

Configuration path: /Servers/Tomcat v7.0 Server at loca [Browse...]

[Open launch configuration](#)

Server Locations

Specify the server path (i.e. catalina.base) and deploy path. Server must be published with no modules present to make changes.

Use workspace metadata (does not modify Tomcat installation)

Use Tomcat installation (takes control of Tomcat installation)

Use custom location (does not modify Tomcat installation)

Server path: .metadata/.plugins/org.eclipse.wst.server. [Browse...]

[Set deploy path to the default value \(currently set\)](#)

Deploy path: wtpwebapps [Browse...]

Overview Modules

SonarQu SonarQu Markers Properties Servers Data

Tomcat v7.0 Server at localhost [Stopped]

Maven

Últimos Tutoriales del Autor

» Integración de SonarQube en Eclipse.

» Monitorización de Apache Tomcat con psi-probe.

» Spring Ldap: gestión de transacciones y configuración de un pool de conexiones

» Spring Ldap: operaciones básicas haciendo uso del soporte de plantillas

» Mi primera vista en ZK como desarrollador JSF (II).

Categorías del Tutorial

IDEs

Es frecuente que se produzcan errores de arranque de tu aplicación a causa de la inexistencia de ficheros de propiedades o de dependencias de librerías que aunque sí existen:

- en el workspace,
- dentro del directorio de fuentes,
- como dependencia gestionada por maven, y
- dentro del directorio de build, esto es, en el directorio target

no se trasladan al directorio de despliegue de la aplicación en local, esto es, al directorio wtpwebapps del .metadata o al directorio de despliegue del propio servidor, según la configuración antes vista.

En este punto, en el que sabemos que el fichero o la dependencia sí existe, lo primero es comprobar que efectivamente el fichero se encuentre también en el directorio de despliegue. Si no se encuentra, las causas pueden ser varias:

- que exista un problema en la configuración de algún fichero, por ejemplo, en el pom.xml del proyecto; solemos obviar las marcas de error y solucionarlas, del modo que sea, puede solventar este otro problema colateral,
- idem con problemas producidos en el "Java Build Path", tendremos que solventarlo primero
- aunque el más frecuente es el más simple y recurrente, sobre todo en plataformas windows, atento!

teniendo el servidor arrancado, no se ha parado de forma normal y el proceso de lectura del servidor sobre el directorio de despliegue está aún activo, lo que impide la escritura sobre el mismo

Este último supuesto es el más probable, como comentaba en entornos windows, y se puede solucionar comprobando con una herramienta de tipo processExplorer examinando los hilos de ejecución que apuntan a un directorio de sistema. Incluso desde la propia herramienta se puede terminar con el proceso solucionando el problema.

Este problema es común en todas las versiones y puede reproducirse, o no, básicamente en función de cómo se produzcan las paradas del servidor local.

4. Soporte para la compilación con AspectJ.

Esta segunda tipología de error sí puede ser nueva; imagina que hasta ahora usabas un plugin de maven para la construcción o empaquetación del proyecto y, tras instalar la última versión de Eclipse Luna, comienza a dar problemas de incompatibilidad. Aquello que venía cubriendo el ciclo de vida de maven en Eclipse, a partir de ahora ya no lo cubre.

Vamos a imaginar que el plugin que comienza a darnos problemas es el de aspectJ

```

1 <properties>
2   <aspectj.version>1.8.2</aspectj.version> <!-- specify your version -->
3 </properties>
4
5 <build>
6   <plugins>
7     <plugin>
8       <groupId>org.codehaus.mojo</groupId>
9       <artifactId>aspectj-maven-plugin</artifactId>
10      <version>1.7</version>
11      <dependencies>
12        <dependency>
13          <groupId>org.aspectj</groupId>
14          <artifactId>aspectjtools</artifactId>
15          <version>${aspectj.version}</version>
16        </dependency>

```

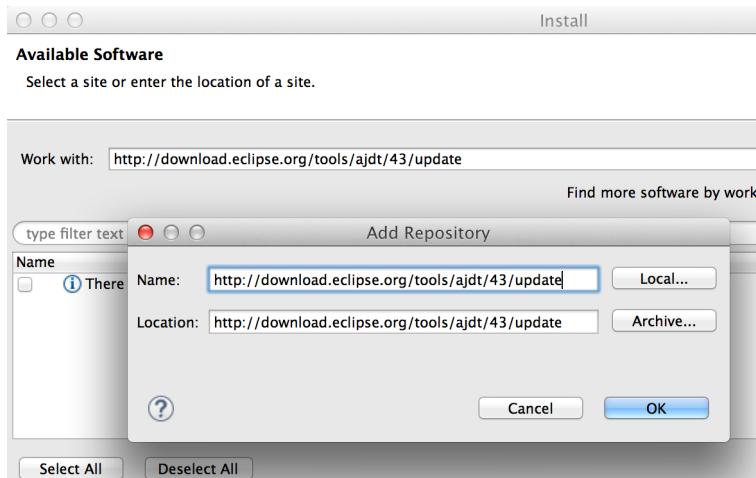
```

17      </dependencies>
18      <executions>
19          <execution>
20              <goals>
21                  <goal>compile</goal>
22                  <goal>test-compile</goal>
23              </goals>
24          </execution>
25      </executions>
26  <configuration>
27      <complianceLevel>1.7</complianceLevel>
28      <encoding>${project.build.sourceEncoding}</encoding>
29      <source>1.7</source>
30      <target>1.7</target>
31  </configuration>
32  </plugin>
33 </plugins>
34 </build>

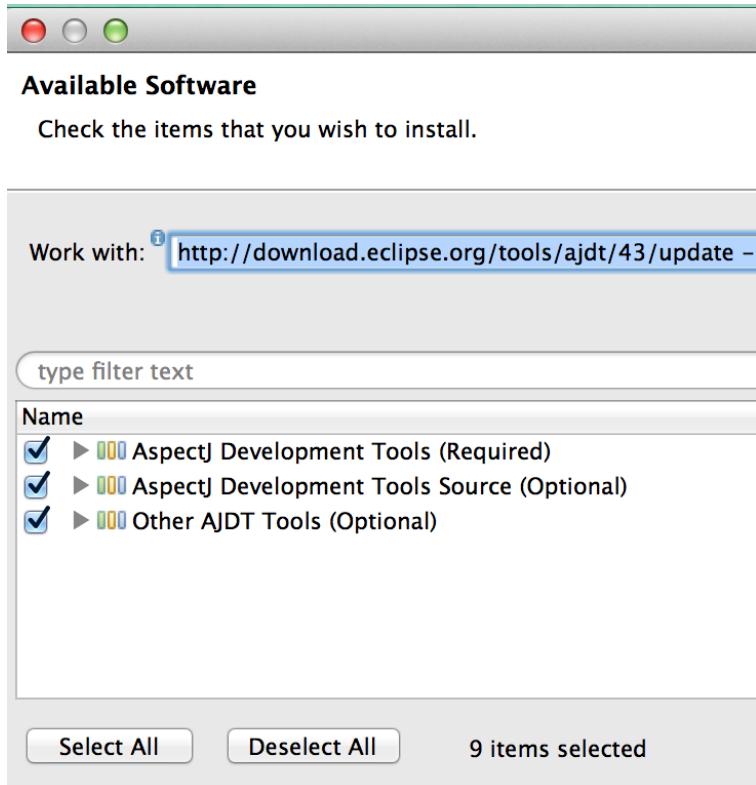
```

En este caso, lo podemos solucionar instalando un plugin de maven que proporciona el soporte necesario para que forme parte del ciclo de vida de maven en Eclipse.

Usando el siguiente sitio <http://download.eclipse.org/tools/ajdt/43/update> y con estos pasos, podemos proceder a su instalación:



Seleccionando estos paquetes:



Y, a continuación con el siguiente <http://dist.springsource.org/release/AJDT/configurator/>



Available Software

Check the items that you wish to install.

Work with: <http://dist.springsource.org/release/AJDT/configurator/>

Name
<input checked="" type="checkbox"/> AjDT M2E configurator
<input checked="" type="checkbox"/> All items are installed

Con ello nuestros problemas comenzarán a desaparecer.

5. Plugin no soportado en el ciclo de vida de maven en Eclipse.

Ahora un segundo plugin, el de jacoco, que venía funcionando hasta ahora y con la nueva instalación ha dejado de funcionar; imaginemos que, en este caso, no disponemos de un plugin de maven para eclipse que instalar y comienza a dar problemas esta simple configuración:

```
1 <build>
2   <plugins>
3     <plugin>
4       <groupId>org.jacoco</groupId>
5       <artifactId>jacoco-maven-plugin</artifactId>
6       <version>0.6.5.201403032054</version>
7       <configuration>
8         <propertyName>jacoco.agent.argLine</propertyName>
9         <destFile>${sonar.jacoco.itReportPath}</destFile>
10        <append>true</append>
11      </configuration>
12      <executions>
13        <execution>
14          <id>agent</id>
15          <goals>
16            <goal>prepare-agent</goal>
17          </goals>
18        </execution>
19      </executions>
20    </plugin>
21    <plugin>
22      <artifactId>maven-failsafe-plugin</artifactId>
23      <version>2.14.1</version>
24      <dependencies>
25        <dependency>
26          <groupId>org.apache.maven.surefire</groupId>
27          <artifactId>surefire-junit47</artifactId>
28          <version>2.14.1</version>
29        </dependency>
30      </dependencies>
31      <configuration>
32        <argLine>${jacoco.agent.argLine}</argLine>
33        <testFailureIgnore>true</testFailureIgnore>
34        <reportsDirectory>${project.build.directory}/surefire-reports</reportsDi
35      </configuration>
36      <executions>
37        <execution>
38          <goals>
39            <goal>integration-test</goal>
40            <goal>verify</goal>
41          </goals>
42        </execution>
43      </executions>
44    </plugin>
45  </plugins>
46 </build>
```

Lo que podemos hacer es habilitar un plugin dentro de la gestión de plugins de maven, en el propio pom.xml, para ignorar la ejecución del plugin de jacoco dentro de la fase correspondiente de ejecución del mismo, en el ciclo de vida de maven, dentro de eclipse.

```
1 <build>
2   <pluginManagement>
3     <plugins>
4       <!--This plugin's configuration is used to store Eclipse m2e settings only.
5           It has no influence on the Maven build itself.-->
6       <plugin>
7         <groupId>org.eclipse.m2e</groupId>
8         <artifactId>lifecycle-mapping</artifactId>
9         <version>1.0.0</version>
10        <configuration>
11          <lifecycleMappingMetadata>
12            <pluginExecutions>
13              <pluginExecution>
14                <pluginExecutionFilter>
15                  <groupId>org.jacoco</groupId>
16                  <artifactId>jacoco-maven-plugin</artifactId>
17                  <versionRange>[0.5,)</versionRange>
```

```

19          <goals>
20              <goal>prepare-agent</goal>
21          </goals>
22      </pluginExecutionFilter>
23      <action>
24          <!-- m2e doesn't know what to do with jacoco,
25              let's ignore it or annoying error markers appear
26              see http://wiki.eclipse.org/M2E_plugin_execution_r
27          -->
28          <ignore></ignore>
29      </action>
30  </pluginExecution>
31  </pluginExecutions>
32  </lifecycleMappingMetadata>
33</configuration>
34</plugin>
35</plugins>
36</pluginManagement>
37</build>

```

Tiene que coincidir el número de versión con el patrón y con ello desaparecerán los errores en el pom.xml y en el proyecto por esta causa.

Claro está que esta configuración no vamos a disponer de la ejecución del plugin de jacoco, puesto que la estamos ignorando; lo normal no es ejecutar el ciclo de tests de integración desde Eclipse sino más bien por línea de comandos con

6. Referencias.

- <http://stackoverflow.com/questions/17613685/aspectj-maven-plugin-not-covered-by-lifecycle-in-kepler>
- http://wiki.eclipse.org/M2E_plugin_execution_not_covered

7. Conclusiones.

Aquí seguimos, intentando eliminar el rojo en los proyectos de nuestro Eclipse ;)

Un saludo.

Jose

jmsanchez@autentia.com

A continuación puedes evaluarlo:

[Regístrate para evaluarlo](#)



Por favor, vota +1 o compártelo si te pareció interesante



Anímate y coméntanos lo que piensas sobre este TUTORIAL:

» [Regístrate](#) y accede a esta y otras ventajas «



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

[PUSH THIS](#)

[Page Pushers](#)

[Community](#)

[Help?](#)

0 people brought clicks to this page

no clicks



powered by [karmocracy](#)

