

Millora de l'entorn de desenvolupament

A qui va dirigit

Aquest how-to va dirigit a tots aquells que desenvolupen aplicacions canigó.

Versió de Canigó

Els passos descrits en aquest document són d'aplicació a les versions de Canigó a partir de la 2.X.

Modificació paràmetres de configuració eclipse.ini

Cal realitzar una còpia de seguretat del fitxer eclipse.ini abans de realitzar els canvis:

```
-splash  
-vmargs  
-XX:+AggressiveOpts  
-XX:+UseParallelOldGC  
-XX:ParallelGCThreads=2  
-XX:ThreadPriorityPolicy=1  
-Xverify:none  
-Xmx900m  
-Xms128m  
-XX:MaxPermSize=128m
```

AggressiveOpts → Habilita les opcions de rendiment per defecte.

UseParallelOldGC → Utilitza una estratègia de màxima paral·lelització a l'hora d'executar el Garbage Collector.

ParallelGCThreads → Nombre de processos per al garbage collector

ThreadPriorityPolicy → S'indica una prioritat dels threads agressiva.

Xverify:none → No verifica les classes contingudes en el bootclasspath i la resta de jars a l'iniciar. **Alerta al deshabilitar aquesta opció si s'instal·len plugins estranys.**

Instal·lació fileSync

FileSync és un plugin per a l'eclipse que permet mapejar directoris d'un path a un altre. Podeu trobar més informació a la pàgina web:

<http://andrei.gmxhome.de/filesync/index.html>

Per a instal·lar-lo cal anar a l'eclipse "Help → Install New Software" i afegir la següent url:
<http://andrei.gmxhome.de/eclipse/>

Configuració de l'aplicació

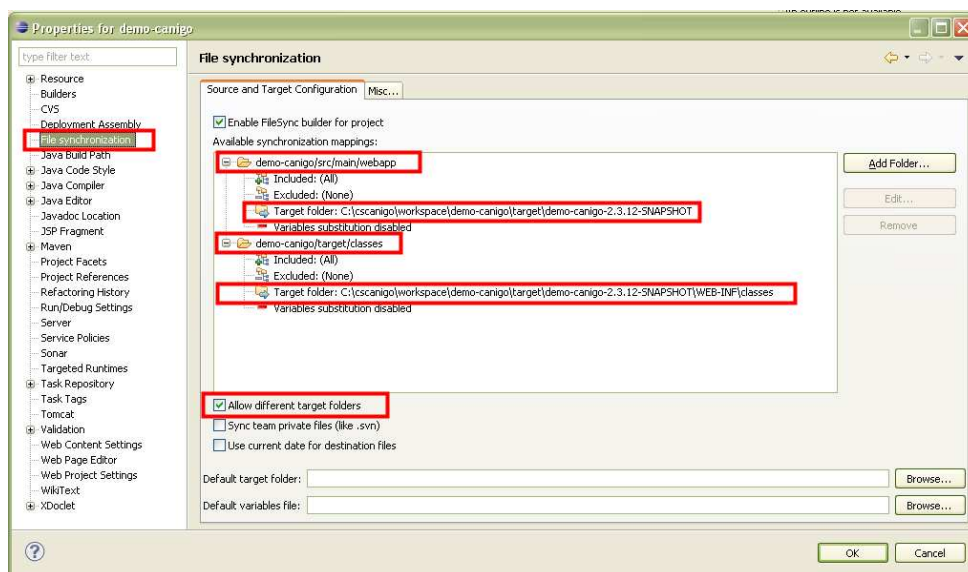
S'haurà de modificar el pom.xml per a que la compilació només la faci eclipse, no la torni a fer maven. Per això afegirem el següent codi:

```
<plugin>  
  <artifactId>maven-compiler-plugin</artifactId>  
  <configuration>  
    <source>1.4</source> <!-- versió de JDK -->  
    <target>1.4</target>  
    <!-- Exclude la compilació de les classes -->  
    <excludes>  
      <exclude>**/*.java</exclude>
```

Millora de l'entorn de desenvolupament

```
</excludes>
</configuration>
</plugin>
```

Utilitzant el plugin **fileSync** redirigirem les classes que compila l'eclipse **/target/classes** a **/src/target/nom-aplicacio/WEB-INF/classes** i el contingut web **/src/main/webapp/** a **/target/nom-aplicacio/WEB-INF/**. D'aquesta manera tindrem actualitzat el sempre les classes i el contingut web i no serà necessari realitzar cada cop que es realitzi un canvi un mvn install.



Per a configurar l'aplicació al tomcat haureu de tenir dintre del directori `$TOMCAT_HOME/conf/Catalina/localhost/` un fitxer `.xml` amb el nom de la vostra aplicació i el següent contingut:

```
<Context path="/nomaplicacio" reloadable="true" docBase="path\target\nomaplicacio"
workDir="path\nomaplicacio\work" >
  <Logger className="org.apache.catalina.logger.SystemOutLogger" verbosity="4"
timestamp="true"/>
  <Resource
    name="formacioDS"
    type="javax.sql.DataSource"
    driverClassName="org.h2.Driver"
    password="cscanigo"
    maxIdle="2"
    maxWait="5000"
    username="cscanigo"
    url="jdbc:h2:~/exemple"
    maxActive="4"/>
</Context>
```

D'aquesta manera, no serà necessari realitzar el mvn clean i mvn package cada cop que es realitzi un canvi, sinó que l'eclipse serà l'encarregat de compilar les classes i deixar-les en el lloc corresponent per a que només sigui necessari reiniciar el servidor d'aplicacions si s'ha modificat un java o xml.