

Upgrade Weblogic 10g

A qui va dirigit

Desenvolupadors d'aplicacions basades en Canigó.

Versió de Canigó

Utilitzable en qualsevol versió de Canigó.

Introducció

Aquest document pretén guiar a les aplicacions per a que pugin desplegar en un Weblogic 10g.

Migració Weblogic 8.1 a 9.2

Una aplicació que estigui funcionant en Weblogic 8.1 sense fer cap tipus de modificació hauria de funcionar amb Weblogic 9.2. Tot i així s'han de tenir en compte alguns punts:

- Si s'estan fent servir classes marcades com a "deprecated" de la JDK 1.4 és possible que no estiguin en la JDK 1.5.
- També caldrà revisar el nom de les variables, ja que podria ser que alguns noms estiguin reservats a partir de la JDK 1.5 com per exemple una variable anomenada "enum" amb JDK 1.4 no dona problemes en canvi en JDK 1.5 sí.

El gran avantatge de migrar a Weblogic 9.2 és la possibilitat d'utilitzar la versió 2.4 del Servlet i JSP 2.0. La migració a JSP 2 si que te una afectació i hi ha un HOWTO que el podeu trobar a:

<http://canigo.ctti.gencat.net/confluence/pages/viewpage.action?pageId=229>

amb el nom de:

[Canigo - HowTo - Com fer upgrade a Servlet 2.4 i JSP 2.0 \(J2EE 1.4\).pdf](#)

Des del Departament de Treball ja varen migrar les seves aplicacions de Weblogic 8.1 a 9.2 amb èxit.

Encara que no s'hagin de realitzar canvis en el codi de l'aplicació, sempre es necessari que la pròpia aplicació realitzi una bateria de proves per validar que tot funciona correctament.

Upgrade Weblogic 10g

Migració Weblogic 9.2 a 10.3

Weblogic 10.3 obliga a utilitzar una JDK 1.6 o superior. Per tal de poder migrar a aquesta versió de Weblogic cal realitzar una sèrie de modificacions en els següents fitxers de configuració de l'aplicació pel correcte funcionament dels serveis de Persistència i de Web Services:

hibernate.cfg.xml → Cal afegir la següent propietat:

```
<property name="hibernate.query.factory_class">  
    org.hibernate.hql.classic.ClassicQueryTranslatorFactory  
</property>
```

weblogic.xml → Cal indicar la propietat prefer-web-inf-classes a true.

```
<?xml version="1.0" encoding="UTF-8"?>  
<weblogic-web-app xmlns="http://www.bea.com/ns/weblogic/weblogic-web-app"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xsi:schemaLocation="http://www.bea.com/ns/weblogic/weblogic-web-app  
http://www.bea.com/ns/weblogic/weblogic-web-app/1.0/weblogic-web-app.xsd">  
<container-descriptor>  
<prefer-web-inf-classes>true</prefer-web-inf-classes>  
</container-descriptor>  
</weblogic-web-app>
```

També caldrà tenir en compte que si s'estan fent servir classes marcades com a "deprecated" de la JDK 1.5 és possible que no estiguin en la JDK 1.6.

Si l'aplicació utilitza xFire, cal desplegar l'aplicació en un ear, afegint el fitxer weblogic-application.xml amb el següent contingut.

```
<?xml version="1.0" encoding="UTF-8"?>  
<weblogic-application xmlns="http://www.bea.com/ns/weblogic/90">  
<application-param>  
<param-name>webapp.encoding.default</param-name>  
<param-value>UTF-8</param-value>  
</application-param>  
<prefer-application-packages>  
<package-name>javax.jws.*</package-name>  
<package-name>antlr.*</package-name>  
<package-name>org.codehaus.xfire.*</package-name>  
<package-name>org.apache.axis2.*</package-name>  
<package-name>javax.xml.*</package-name>  
</prefer-application-packages>  
</weblogic-application>
```