

Installation avec IzPack

Cet outil permet de créer le programme d'installation et de livrer les différentes versions téléchargeables disponibles sur Sourcesup.

Préparer l'installation

- Changer le numéro de version dans /resources/properties/misc/application.xml
- Changer les valeurs des variables dans le pom.xml :

```
<version>X.X.X</version>
```

```
<your.standalone-compiler.version>4.3.2</your.standalone-compiler.version>
```

```
<zip.location>${basedir}/src/main/util/utls/7zip/7za.exe</zip.location>
```

```
<izpack.application.name>ChaineditXXX</izpack.application.name>
```

<data.dir>datarelease</data.dir> pour indiquer le répertoire contenant les chartes et datas à intégrer dans la livraison

- your.standalone-compiler.version permet de spécifier la version du plugin izpack

- zip.location permet de préciser le chemin d'accès à l'outil 7zip. Le chemin est à changer en fonction de votre installation (chemin et système Linux ou Windows).

- izpack.application.name permet de préciser le nom qui sera utiliser par izpack pour l'appel au service Tomcat sous Windows. Le nom ne doit pas contenir de caractères spéciaux ni point ".", ni tiret "-". Remplacer **XXX** par le numéro de version sans les points.

- Lancer : mvn install

Pour plus de détails voir la documentation esup-commons :

<http://www.esup-portail.org/display/PROJCOMMONS/3.7.1+Installeur+IzPack>

Debug de l'installation

1. Ouvrir une ligne de commande : Exécuter → cmd
2. Aller dans le répertoire du jar d'install
3. Exécuter l'install sous la forme : java -jar -DTRACE=TRUE chainedit2-2.0-quickstart.jar

Les modifications

Dans le apache

Dézipper le apache-tomcat-x.x.x.zip Modifier les fichiers comme indiqué ci-dessous en ajoutant les lignes précisées au début de chaque fichier. Attention à la sauvegarde des .sh à faire sous Linux. Rezipper le apache-tomcat-x.x.x.zip et le mettre dans src/main/util/utls/tomcat

La procédure de construction de l'exé d'install est répartie entre le pom et le fichier install.xml de /src/main/installer/config/jar/ Dans le répertoire bin du Tomcat zippé

Le service.bat

1. A début : set CATALINA_HOME=%INSTALL_PATH%\apache-tomcat-6.0.29\
1. A la fin remplacer : "%EXECUTABLE%" US%SERVICE_NAME% ++JvmOptions "-
Djava.io.tmpdir=%CATALINA_BASE%\temp;-
Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager;-
Djava.util.logging.config.file=%CATALINA_BASE%\conf\logging.properties;-
Dapplication.config.location=%INSTALL_PATH\data\config.properties;-
Dapplication.security.location=%INSTALL_PATH\data\security" -Jvms 256 -JmMx 512

Le catalina.bat

set CATALINA_OPTS = "-Dapplication.config.location=%INSTALL_PATH\data\config.properties -
Dapplication.security.location=%INSTALL_PATH\data\security"

Le catalina.sh

export CATALINA_OPTS = "-Dapplication.config.location=%INSTALL_PATH/data/config.properties -
Dapplication.security.location=%INSTALL_PATH/data/security"

Explication

Au lancement Chainedit regarde si application.config.location existe. Si oui, il prend ce fichier de config. Sinon il prend celui qui est dans properties. Cette information est définie dans le applicationContext.xml :

```
<bean id="propertyConfigurer"  
class="org.springframework.beans.factory.config.PropertyPlaceholderConfigure  
r">  
  <property  
    name = "locations">  
    <list>  
      <value>classpath:/properties/config.properties</value>  
      <value>file:${application.config.location}</value>  
    </list>  
  </property>  
  <property name = "ignoreResourceNotFound" value="true" />  
</bean>
```

```
</bean>
```

De plus, le paramètre `application.security.location` permet de pointer vers les fichiers de sécurité dans `/data`. Lors de la procédure de création du fichier d'install, le fichier `web.xml` est remplacé par le fichier `web-war.xml` qui contient ce paramètre. Il doit pointer sur le répertoire.

From:

<https://wiki.univ-rennes1.fr/chainedit/> - **chainedit wiki**

Permanent link:

https://wiki.univ-rennes1.fr/chainedit/doku.php?id=developpeurs:installeur_izpack

Last update: **2017/03/30 12:53**

