Install Howto

Table of contents

Ce document décris la procédure d'installation complète depuis un snapshot. Vous pouvez récupérer cette archive à la page de téléchargement et la décompresser.

1. Prérequis

Pour installer le système vous devez avoir préalablement installé :

- un serveur web apache pour :
  - l'affichage des pièces jointes
  - l'hébergement du site d'update de l'application RCP
- un serveur LDAP openldap pour :
  - l'authentification des utilisateurs
  - la gestion des droits

  Le serveur doit être configuré avec les schémas suivants :

  - core.schema
  - cosine.schema
  - nis.schema
  - inetorgperson.schema
- un serveur de base de données postgresql
  - pour héberger la base de donnée principale
- un serveur d'application jBoss
  - pour faire tourner le coeur de l'application

Pour la compilation et l'installation, vous devez disposer d'un JDK (1.5) et de ANT.

Ci-dessous les versions testées avec succès :

Apache	1.3.33 - 2.0.54	http://httpd.apache.org
PostgreSQL	7.4.7 - 8.0.3	http://www.postgresql.org
OpenLDAP	2.2.23 - 2.2.28	http://www.openldap.org
JBoss	4.0.0 (uniquement cette version)	http://www.jboss.com
JDK	1.5	http://java.sun.com/j2se/1.5.0/download.jsp
Apache ANT	1.6.2	http://ant.apache.org

Note:

2 choses importantes à retenir : **jBoss 4.0.0 et jdk1.5**

2. Architecture Matérielle

Vous pouvez installer postgresql, apache, jboss et openldap sur des machines différentes. La seule condition est que le serveur jboss ait accès au système de fichier du serveur apache (partage NFS, samba ...).

Bien entendu, une seule machine peut héberger l'ensemble du système.

Bien souvent, le server LDAP est un serveur existant sur lequel nous allons juste ajouter quelques unités organisationnelles et groupes pour le bon fonctionnement de l'application.

Nous considérons ici 3 machines différentes :

- MAINHOST : jboss et apache, jdk et ant. La machine héberge le serveur d'application, le serveur web et fait les builds.
- DBHOST : postgesql
- DIRHOST : openldap

Toutes ces serveurs sont en LAN et les clients sont dans ce même LAN. (ce n'est pas un prérequis, mais c'est plus simple).

3. Etapes préparatoires

Le but du jeu est de renseigner le fichier build.properties présent à la racine du répertoire de snapshot. Ne faites aucun build tant que ce fichier n'est pas correctement renseigné. Bien entendu les exemple donnés ne sont pas à utiliser tels quels, mais sont à adapter à votre propre configuration.

3.1. Utilisateurs

Le soin vous est laissé de gérer vos utilisateur et la sécurité de votre système. Veuillez vous assurer des choses suivantes :

- l'utilisateur qui execute ant doit avoir les droits suivant :
  - ecriture sur le répertoire contenant le site d'update
  - ecriture sur le répertoire 'deploy' de l'instance jboss
  - ecriture sur le répertoire 'lib' de l'instance jboss
- l'utilisateur qui lance jboss doit avoir les droits suivant :
  - ecriture sur le répertoire de stockage des pièces jointes.

3.2. Apache

Nous avons besoin de 2 sites web :

- Un premier pour diffuser les pièces jointes : Soit le répertoire /var/www/localhost/htdocs/emis-att sur la machine MAINHOST, accessible en web par http://MAINHOST/emis-att
- Un second pour héberger toutes features de l'application cliente et servir de serveur d'update. Soit le répertoire /var/www/localhost/htdocs/emis-update accessible par http://MAINHOST/emis-update

```
$attachments.path=/var/www/localhost/htdocs/emis-att
$attachments.url=http://MAINHOST/emis-att

$update.path=/var/www/localhost/htdocs/emis-update
$update.url=http://MAINHOST/emis-update
```

3.3. JBoss

JBoss est installé dans /usr/share/jboss

```
$jboss.home=/usr/share/jboss
```

Pour une installation par défaut, sans modification de la configuration :

```
$jboss.jndi.url=jnp://MAINHOST:1099
```

Nous devons installer le driver JDBC pour postgresql. On peut trouver ces drivers ici : http://jdbc.postgresql.org/download.html Télécharger la bonne version et placer le jar dans $jboss.home/server/default/lib

```
$database.driver.path=/usr/share/jboss/server/default/lib/postgresql-8.0-312.jdbc3.jar
```

3.4. PostgreSQL

Créer la base et créer l'utilisateur propriétaire de cette base (accès complet).

```
$database.host=DBHOST
$database.name=emis
$database.user=emis
$database.pwd=emispwd
```

Le paramètre ci dessous indique à Hibernate (couche de mapping OR) de ne pas créer la base au démarrage.

```
$hibernate.Hbm2ddlAuto=no
```

3.5. OpenLdap

Voici la liste des paramètres que nous devons connaitre.

- Le nom du server LDAP, pour le serveur d'application

```
$ldap.host=DIRHOST:389
```

- Le nom du serveur LDAP, depuis un client

```
$ldap.url=ldap://DIRHOST:389
```

- La base de l'arbre

```
$ldap.baseDn=dc=my-domain,dc=com
```

- Les paramètres du Manager pour se connecter et avoir des droits d'ecritures

```
$ldap.security.principal=cn=Manager,dc=my-domain,dc=com
$ldap.security.credentials=secret
```

- Un user fantôme pour ajouter dans les groupes créés

```
# Un compte par défaut pour ajouter aux groupes
$ldap.nobody=cn=nobody
```

- La base ou on trouve les comptes utilisateurs

```
$ldap.personsDn=ou=People,dc=my-domain,dc=com
```

- La base ou on trouve les groupes d'utilisateurs

```
$ldap.unitsDn=ou=Units,dc=my-domain,dc=com
```

- La racine de l'unité organisationelle applicative

```
$ldap.appsDn=ou=Apps,dc=my-domain,dc=com
```

- Puis les bases pour les queues, les routes, et les features

```
$ldap.queuesDn=ou=queues,ou=emis,ou=Apps,dc=my-domain,dc=com
$ldap.routesDn=ou=routes,ou=emis,ou=Apps,dc=my-domain,dc=com
$ldap.featuresDn=ou=features,ou=emis,ou=Apps,dc=my-domain,dc=com
```

3.6. Compte de test

Pour tester l'application, vous pouvez déterminer un compte email depuis lequel les emails seront retirés et intégrés dans emis. Utilisez un compte de test, ou créez en un pour l'occasion. Vous devrez renseigner les paramètres habituels pour ce genre d'opération :

- Le nom de la ressource dans le système

```
$example.mail.name=example
```

- Le nom d'utilisateur

```
$example.mail.user=example
```

- Le mot de passe

```
$example.mail.password=examplePwd
```

- Les serveurs

```
$example.mail.pop3.host=pop.domain.net
$example.mail.smtp.host=smtp.domain.net
```

- Le nom qui apparaitra comme expéditeur, sous forme encodée (ici 'Example <example@domain.net>')

```
$example.mail.from=Example &lt;example@domain.net&gt;
```

4. Build

Le script build.sh à la racine du répertoire de snapshot lance ant avec une target spécifique (passée en paramètre). Il définit le classpath en y intégrant la bibliothèque ant-contrib nécessaire pour certaines tâches.

La section de définition des variables JAVA_HOME, ANT_HOME et PATH est commentée. Vous pouvez avoir besoin de spécifier ces variables (et donc de les décommenter) si ces elles ne sont pas déjà configurées dans votre environnement.

Warning:

Assurez vous que JAVA_HOME pointe bien vers un jdk version 1.5 minimum.

On s'assure que le build.sh est executable

```
chmod +x build.sh
```

Warning:

Veillez à respecter l'ordre dans lequel les tâches vous sont présentées. Certaines se basent sur des bibliothèques packagées par d'autres ...

4.1. Verification

Une target ant valide certains points nécessaires au bon déroulement du déploiement (droits d'accès en écriture sur certains répertoires, connectivité vers la base de données). Il est fortement recommandé de valider cette étape. Si elle échoue, vérifiez les points précédents.

```
./build.sh testAll
```

Warning:

Ne continuez pas tant que cette tâche n'est pas SUCCESSfull !

4.2. Initialisation LDAP

Cette tâche prépare différents fichier LDIF pour préparer la structure LDAP nécessaire au fonctionnement de l'application. La configuration est basée sur ce qui a été renseigné dans le build.properties.

```
./build.sh prepareLdif
```

Selon la structure de votre propre LDAP, vous aurez peux être besoin d'en commenter certaines entrées avant d'intégrer les fichiers, par exemple si la branche Person existe déjà ...).

Warning:

Veillez cependant à ne pas modifier les entrées elle-mêmes (dn ...). Si nécessaire, faites les modifications dans le build.properties et relancez la tâche de préparation.

Dans le répertoire 'ldif' vous trouverez :

- ous.ldif : déclare les OU (unité organisationnelle).
  Vous pouvez avoir besoin de commenter ou supprimer la partie concernant la branche People (si vos utilisateurs sont déjà rangés dans une OU spécifique), ou la branche Units (si vos groupes sont déjà organisés dans une OU).

Warning:

Il est fortement recommandé de laisser la structure telle qu'elle est à partir de la branche 'emis'.

- groups.ldif : prépare les entrées par défaut (groupes), queue de worflow et features. Vous devez intégrer ces données telles quelles.
- user.ldif : ce fichier permet la création d'un premier utilisateur pour tester l'application. Vous pouvez vous en passer si votre LDAP est déjà peuplé (alors vous avez certainement votre compte !). Cependant, assurez vous que le compte que vous utiliserez soit membre des features 'admin' et 'redaction'. Sans cela, vous aurez une fenêtre grise vide lors du lancement de l'application, décevant ...

Les fichiers doivent être intégrés dans l'ordre suivant :

```
ldapadd -D "cn=Manager,dc=my-domain,dc=com" -W -f ldif/ous.ldif
ldapadd -D "cn=Manager,dc=my-domain,dc=com" -W -f ldif/groups.ldif
ldapadd -D "cn=Manager,dc=my-domain,dc=com" -W -f ldif/user.ldif
```

4.3. Initialisation de la base de donnée

Cette tâche créer les tables et les nourrit avec les entrées par défaut.

```
./build.sh initDb
```

4.4. Déploiement de la source de données

Créé la source de donnée et deploie le service dans le répertoire de déploiement de jboss.

```
./build.sh deployDs
```

Note:

Si vous avez une vue sur les logs jboss, vous devez voir l'activation de cette datasource.

4.5. Installation du middleware

Cette tâche compile la partie server de l'application et déploie le package EAR généré dans le répertoire de déploiement de jboss.

```
./build.sh installEar
```

Note:

Lors de la première installation, si jboss est lancé, des exceptions seront levées lors de l'initialisation de l'application J2EE. Cela vient du fait qu'un certain nombres de classes communes sont déployées dans les libraires commune de l'instance et ne sont donc pas visible avant redémarrage.

4.6. Redémarrage de JBoss

Redémarrez jboss afin que l'application J2EE emis s'initialise correctement.

4.7. Installation des modules d'exemple

Cette tâche déploie une configuration minimale de l'application permettant de d'en découvrir le fonctionnement. Elle déploie un 'alias', c'est à dire un compte email depuis lequel les mails vont être retirés et intégrés. C'est le compte d'exemple dont les paramètres ont été reneignés dans le build.properties.

```
./build.sh deployExample
```

4.8. Compilation et packaging de l'application cliente

Cette tâche génère deux archives :

- emis-linux.zip
- emis-windows.zip

Ce sont les applications cliente pour les deux plateformes cibles. Ces zip se trouvent dans le répertoire 'packages' après le build et sont aussi disponible en téléchargement sur le site d'update.

```
./build.sh installProduct
```

4.9. Préparation du site d'update

L'application livrée au point précédent n'étant qu'un squelette, nous avons besoin d'un endroit où sont déposés les plugins constituant l'application. C'est le rôle du site d'update. Cette tâche génère tous les plugins, les organise en features et génère le site.

```
./build.sh installSite
```

5. Test de l'application cliente

Récupérez l'archive pour le système cible dans le répertoire 'packages' ou bien téléchargez les zip :

- http://MAINHOST/product/emis-linux.zip
- http://MAINHOST/product/emis-windows.zip

Décompressez les archives. L'utilisateur doit avoir des droits d'écritures sur le répertoire 'emis' (téléchargement des features ...).

- Sous linux :
  Vous devrez rendre le programme emis executable et rendre disponible un jre (1.5) à la racine :

  ```
  unzip emis-linux.zip
  cd emis
  ln -s ~/...../jdk1.5.0_04/jre .
  chmod +x emis
  ./emis
  ```

- Sous windows :
  Assurez vous qu'une variable d'environnement JAVA_HOME pointe sur un JRE 1.5, puis double-cliquez sur emis.exe

Vous devez voir apparaitre une fenêtre d'authentification. Saisissez un login et un mot de passe valide dans votre annuaire LDAP. Si vous avez utilisé le fichier emis-user.ldif, utilisez le login 'user' et le mot de passe 'password'.

Lors de son lancement, après authentification, l'application va déterminer les features auquelles vous pouvez accéder selon les droits définis dans le LDAP. Celles-ci sont alors téléchargées depuis le site d'update. Ce temps de téléchargement peut-être assez long (2 minutes, le plugin de vérification orthographiques contient un dictionnaire volumineux), soyez patient. A la fin du téléchargement, une boite de dialogue devrait apparaitre, confirmez.

Pour comprendre les bases, je vous invite à consulter le document Premiers Pas.