

Install Gentoo

Table of contents

1 Préparation du système.....	2
2 Préparation des répertoires web.....	2
3 Configuration de PostgreSQL.....	2
4 Installation du driver JDBC.....	3
5 Configuration de OpenLDAP.....	4
6 Préparation du build.....	5
7 Installation du Middleware.....	8
8 Installation de la partie cliente.....	9
9 Test final.....	10

Ce document décrit une installation sur une machine unique équipée de la distribution gentoo. C'est un bon complément au document [Install Howto](#) même si vous n'utilisez pas cette distribution.

1. Préparation du système

La machine s'appelle **barbalala**

Voici les versions des paquets installés par emerge :

- dev-java/ant-core 1.6.2-r2
- dev-java/ant-tasks 1.6.2-r5
- net-www/apache 2.0.54-r4
- net-nds/openldap 2.2.28
- dev-db/postgresql 8.0.3

Dans la mesure où l'arbre portage ne nous propose pas encore les bonnes versions, et qu'on a pas le temps de se pencher sur les ebuilds, nous installons ces derniers à la main :

- JBoss 4.0.0 : installé dans /opt/jboss-4.0.0
- JDK 1.5 : installé dans /opt/jdk1.5.0_05

2. Préparation des répertoires web

Je créé 2 répertoires pour les besoin de l'application. Ces répertoires devront être accessible via HTTP :

```
mkdir /var/www/localhost/htdocs/emis-update
mkdir /var/www/localhost/htdocs/emis-att
```

Je m'assure que le répertoire de stockage des pièces jointes sera accessible en écriture par le user qui lance jboss.

```
chown -R jboss /var/www/localhost/htdocs/emis-att
```

3. Configuration de PostgreSQL

Nous avons suivi l'excellent tutorial http://gentoo-wiki.com/HOWTO_Configure_Postgresql

Préparation des répertoires pour les bases

```
mkdir -p /var/lib/postgresql/data
chown -R postgres:postgres /var/lib/postgresql/data
```

Initialisation des répertoires de bases

```
su - postgres
initdb -D /var/lib/postgresql/data
exit
```

Démarrage du serveur

```
barbalala root # /etc/init.d/postgresql start
* Starting PostgreSQL...
[ ok ]
barbalala root #
```

Création du user 'emis'

```
barbalala root # createuser -U postgres emis -W
Le nouvel utilisateur a-t'il le droit de créer des bases de données ? (y/n)
o
Le nouvel utilisateur a-t'il le droit de créer des utilisateurs ? (y/n) n
Mot de passe :
CREATE USER
```

Warning:

Attention à l'astuce, j'ai dû répondre 'o' et non 'y' pour que mon utilisateur ait effectivement les droits pour créer une base de données.

Création de la base 'emis'

```
barbalala root # createdb -U emis -W emis
Mot de passe :
CREATE DATABASE
```

Note:

Si vous avez déjà créé une base 'emis' lors d'un précédent déploiement, supprimez-la (`dropdb -U emis -W emis`) puis recréez-la.

4. Installation du driver JDBC

Je télécharge la version JDBC3 pour la version 8.0 sur <http://jdbc.postgresql.org/download/postgresql-8.0-312.jdbc3.jar>

Je copie le jar dans le répertoire des libs de jboss :

```
cp postgresql-8.0-312.jdbc3.jar /opt/jboss-4.0.0/server/default/lib/
```

5. Configuration de OpenLDAP

De très bons conseils en suivant <http://www.gentoo.org/doc/en/ldap-howto.xml>

Modification du fichier de configuration

```
vi /etc/openldap/slapd.conf
```

Voici les extraits concernés :

```
include          /etc/openldap/schema/core.schema
# on ajoute aussi les schemas necessaires :
include          /etc/openldap/schema/cosine.schema
include          /etc/openldap/schema/nis.schema
include          /etc/openldap/schema/inetorgperson.schema

suffix           "dc=my-domain,dc=com"
rootdn           "cn=Manager,dc=my-domain,dc=com"
rootpw           secret
```

Note:

J'ai tout laissé par défaut pour le suffix et rootpw mais il va sans dire que vous devrez adapter cela !

Warning:

Veillez à bien importer tous les schémas nécessaires.

Initialisation de l'annuaire

Nous devons créer un fichier LDIF pour initialiser l'annuaire.

```
barbalala root # touch base.ldif
barbalala root # vi base.ldif
```

Voici son contenu :

```
dn:dc=my-domain,dc=com
objectClass: top
objectClass: dcObject
objectClass: organization
o: my-domain
dc: my-domain
```

Nous l'ajoutons avec la commande ldapadd :

```
barbalala root # ldapadd -D "cn=Manager,dc=my-domain,dc=com" -W -f
base.ldif
Enter LDAP Password:
adding new entry "dc=my-domain,dc=com"
ldap_add: Added

barbalala root #
```

6. Préparation du build

Je récupère le snapshot sur la page de [download](#).

```
barbalala root # wget
http://emis.sourceforge.net/installation/emis-snapshot-20050928.zip
barbalala root # unzip -q emis-snapshot-20050928.zip
barbalala root # cd emis-snapshot
barbalala emis-snapshot # vi build.properties
```

Je modifie mon build.properties selon ma configuration, voici son contenu :

```
# ne pas modifier
$cvs.host=none
# indique d'utiliser les fichiers locaux (snapshot)
# la suite est inutile :
#$cvs.dir=/var/lib/cvs
#$cvs.user=cvsuser
#$cvs.type=extssh

# le répertoire de base du jboss
$jboss.home=/opt/jboss-4.0.0

# l'url jndi depuis un client
$jboss.jndi.url=jnp://barbalala:1099

# le path où construire le site d'update (pour la machine qui fait le
build)
$update.path=/var/www/localhost/htdocs/emis-update

# l'url d'update (pour un client)
$update.url=http://barbalala/emis-update

# le path ou déposer les pièces jointes (pour le serveur applicatif)
$attachments.path=/var/www/localhost/htdocs/emis-att

# l'url ou sont trouvables les pièces jointes (pour un client)
$attachments.url=http://barbalala/emis-att

# le host qui héberge postgresql (pour le serveur applicatif et le builder)
$database.host=localhost
$database.name=emis
```

```

$database.user=emis
$database.pwd=emisPwd

# chemin où trouver le driver jdbc (pour les tâches SQL)
$database.driver.path=/opt/jboss-4.0.0/server/default/lib/postgresql-8.0-312.jdbc3.jar

# indique a Hibernate s'il doit créer la base au démarrage (create|no)
$hibernate.Hbm2ddlAuto=no

# le host qui héberge le LDAP (pour le serveur applicatif)
$ldap.host=localhost:389

# l'url LDAP pour un client
$ldap.url=ldap://barbalala:389

# Un compte LDAP ayant des droits
$ldap.security.principal=cn=Manager,dc=my-domain,dc=com
$ldap.security.credentials=secret

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

# Base de l'arbre
$ldap.baseDn=dc=my-domain,dc=com

# Branche où seront recherchés les utilisateurs
$ldap.personsDn=ou=People,dc=my-domain,dc=com

# Branche où seront recherchés les groupes
$ldap.unitsDn=ou=Units,dc=my-domain,dc=com

# Branche applicative
$ldap.appsDn=ou=Apps,dc=my-domain,dc=com

# OU contenant les queues workflow
$ldap.queuesDn=ou=queues,ou=emis,ou=Apps,dc=my-domain,dc=com

# OU contenant les queues de route
$ldap.routesDn=ou=routes,ou=emis,ou=Apps,dc=my-domain,dc=com

# OU contenant les features RCP
$ldap.featuresDn=ou=features,ou=emis,ou=Apps,dc=my-domain,dc=com

# Pour l'exemple, les paramètres du compte de messagerie
$example.mail.name=devteam
$example.mail.user=devteam@tiscali.fr
$example.mail.password=*****
$example.mail.pop3.host=pop.tiscali.fr
$example.mail.smtp.host=mail.tiscali.fr
$example.mail.from=Example <devteam@tiscali.fr>

```

Je spécifie le JAVA_HOME dans le build.sh. Voici le contenu de celui-ci :

Attention

Assurez vous de bien utiliser le JDK 1.5 lors de la compilation. Si la variable JAVA_HOME ne pointe pas vers celui-ci ou qu'elle n'existe pas, décommentez et adaptez la ligne correspondante dans le build.sh

```
#!/bin/sh

# Replace and uncomment if necessary
#
export JAVA_HOME=/opt/jdk1.5.0_05
# export ANT_HOME=/opt/java/libs/apache-ant-1.6.2
# PATH=$PATH:$ANT_HOME/bin

export
CLASSPATH=$CLASSPATH:cvs/net.sf.emis.libstore/ant/ant-contrib-1.0b1.jar

ant $1
```

Je rends le build.sh exécutable :

```
barbalala emis-snapshot # chmod +x build.sh
```

Je lance la target de test :

```
barbalala emis-snapshot # ./build.sh testAll
Buildfile: build.xml

testAll:
  [echo] Checking properties
  [echo] Testing write access in jBoss deploy directory :
/opt/jboss-4.0.0/server/default/deploy
  [touch] Creating /opt/jboss-4.0.0/server/default/deploy/.test
  [delete] Deleting: /opt/jboss-4.0.0/server/default/deploy/.test
  [echo] Testing write access in update site directory :
/var/www/localhost/htdocs/emis-update
  [touch] Creating /var/www/localhost/htdocs/emis-update/.test
  [delete] Deleting: /var/www/localhost/htdocs/emis-update/.test
  [echo] Testing write access in attachments storage directory :
/var/www/localhost/htdocs/emis-att
  [touch] Creating /var/www/localhost/htdocs/emis-att/.test
  [delete] Deleting: /var/www/localhost/htdocs/emis-att/.test
  [echo] Testing DB connectivity to host localhost
  [sql] Executing commands
  [sql] 2 of 2 SQL statements executed successfully

  [echo] All tests are OK. You can launch builds.

BUILD SUCCESSFUL
Total time: 2 seconds
```

Tout à l'air OK, je peux lancer les builds.

7. Installation du Middleware

Initialisation de la base de données :

```
barbalala emis-snapshot # ./build.sh initDb
```

Déploiement de la source de données :

```
barbalala emis-snapshot # ./build.sh deployDs
```

Initialisation LDAP

Je lance la target de préparation des fichier LDIF :

```
barbalala emis-snapshot # ./build.sh prepareLdif
```

ANT me dis que les fichiers sont disponibles dans le répertoire 'ldif'. Un rapide coup d'oeil :

```
barbalala emis-snapshot # ls ldif/
groups.ldif ous.ldif user.ldif
barbalala emis-snapshot # head ldif/ous.ldif
```

```
# La branche People
dn:ou=People,dc=my-domain,dc=com
ou: People
objectClass: top
objectClass: organizationalUnit
```

OK, tout à l'air bien. Comme mon LDAP est tout fraîchement installé, j'importe la totale :

```
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
```

Attention

Si votre LDAP est déjà peuplé et utilisé, il est probable que vous ayez déjà une branche People où sont rangés tous vos utilisateurs et un branche Group où sont rangés vos groupes. Dans ce cas pensez à commenter les entrées pour People et Units.

Installation de l'application J2EE :

```
barbalala emis-snapshot # ./build.sh installEar
```


Lancement de JBoss :

Dans une autre console , je m'assure que les variables d'environnement JBOSS_HOME et JAVA_HOME sont correctes.

```
barbalala root # export JBOSS_HOME=/opt/jboss-4.0.0
barbalala root # export JAVA_HOME=/opt/jdk1.5.0_05/
barbalala root # /opt/jboss-4.0.0/bin/run.sh
```

En surveillant les logs jboss on devrait voir apparaitre, entre autre :

```
20:16:09,977 INFO [WrapperDataSourceService] Bound connection factory for
resource adapter for ConnectionManager
'jboss.jca:service=DataSourceBinding,name=net.sf.emis/jdbc/emisDS to JNDI
name 'java:net.sf.emis/jdbc/emisDS'
```

et encore :

```
20:16:36,805 INFO [EARDeployer] Started J2EE application:
file:/opt/jboss-4.0.0/server/default/deploy/net.sf.emis.ear
```

D'autre part, il va y avoir un truc qui parle régulièrement, c'est le Retriever qui est lancé. C'est bon signe, mais pour l'instant il ne trouve pas d'alias à aller POPer. On va arranger ça.

Déploiement de l'exemple

```
barbalala root # ./build.sh deployExample
```

Après ce dépoitement, on doit voir apparaitre dans la console JBoss :

```
20:20:33,313 INFO [MailService] Mail Service bound to
net.sf.emis/javamail/Devteam
```

S'il y a des emails sur ce compte, alors il vont être retirés et ca va beaucoup parler.

8. Installation de la partie cliente

On lance le packaging de l'application RCP :

```
barbalala emis-snapshot # ./build.sh installProduct
```

A la fin, ANT m'indique ou je peux trouver les packages, je note pour plus tard :

```
[echo] Zipped product packages are available at :
[echo]           http://barbalala/emis-update/product/emis-linux.zip
[echo]           http://barbalala/emis-update/product/emis-windows.zip
```

Création du site d'update :

```
barbalala emis-snapshot # ./build.sh installSite
```

9. Test final

Le moment est venu de tester l'installation.

Sur une autre machine, équipée de X et d'un JRE 1.5, ma station de travail par exemple, je récupère le package linux à l'adresse indiquée par ANT lors du build :

`http://barbalala/emis-update/product/emis-linux.zip`

Je dézippe, je rend exécutable, je link un jre 1.5, le lance :

```
$ unzip -q emis-linux.zip
$ cd emis
$ ls
configuration emis plugins startup.jar
$ chmod +x emis
$ ln -s ~/java/sdk/jdk1.5.0_04/jre jre
$ ./emis
```

Au lancement, vous devriez avoir une invite de log comme celle là :

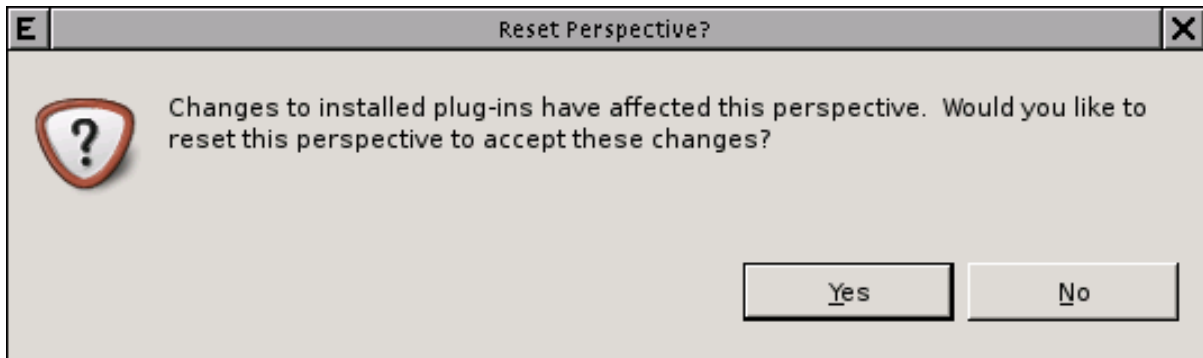
The image shows a standard Linux-style login dialog box. The title bar contains the letter 'E' on the left and a close button 'X' on the right. The main text reads 'Veillez saisir vos identifiant.' Below this, there are two input fields: 'Login:' with the text 'user' and 'Password:' with the text '*****'. At the bottom right, there are two buttons labeled 'OK' and 'Cancel'.

Invite de login

Puisque j'ai insérer le fichier user.ldif dans le LDAP, j'utilise le compte de test (user/password) pour m'authentifier.

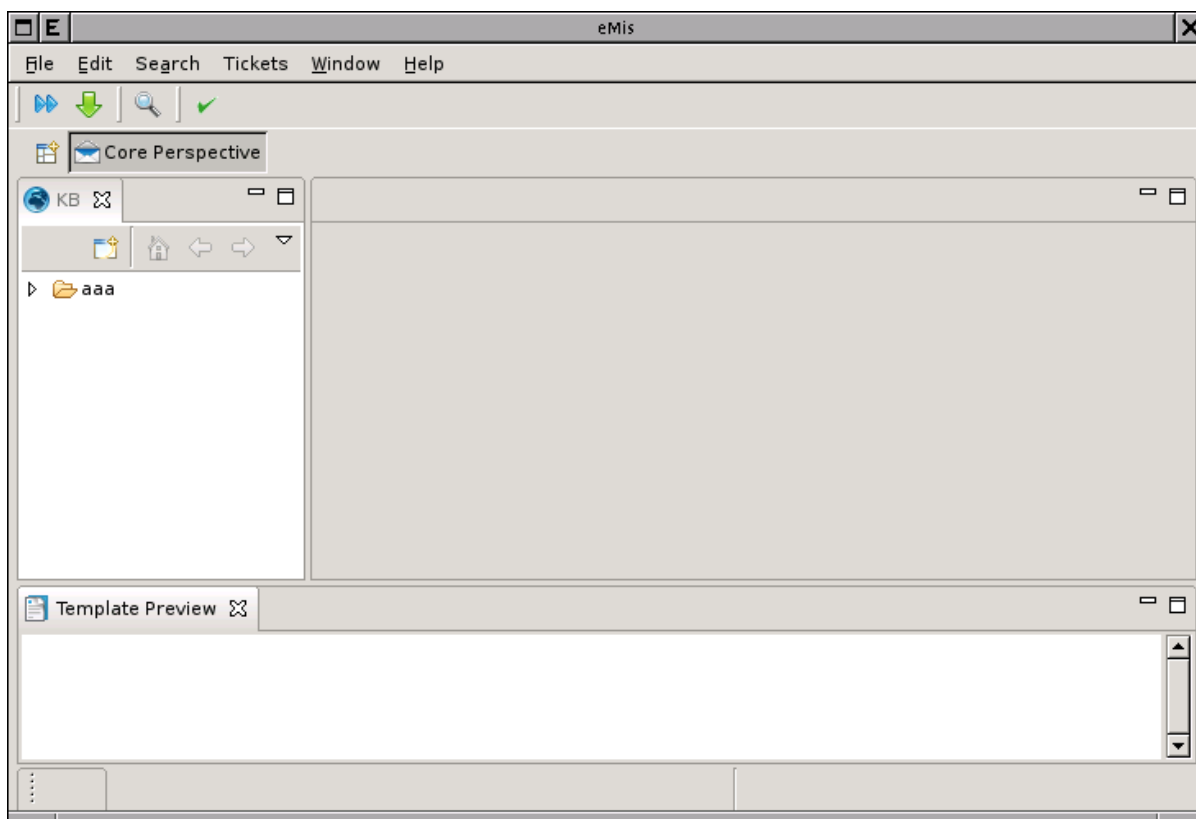
Le premier lancement est long : après vous avoir authentifié, le launcher télécharge et installe les features auxquelles vous avez droit depuis le site d'update (avec le user.ldif, c'est la totale).

Après ces opérations, vous devez avoir une fenêtre de la sorte :



Validation de la perspective

Validez, après quoi vous devriez voir apparaître ceci :



Premier lancement

Pour comprendre les bases et vous familiariser avec l'interface, je vous invite à consulter le document [Premiers Pas](#).