

Table des matières

1	Description de l'application.....	2
2	Installation de Pharmed.....	3
2.1	Installation des pré-requis :.....	3
2.2	Initialisation de la base :.....	3
2.3	Génération des archives :.....	4
2.4	Initialisation de JBoss :.....	4
2.5	Sur un client :.....	4
2.6	Lancement :.....	5
3	Déploiement d'une nouvelle version.....	6
3.1	Sur la machine de développement :.....	6
3.2	Transfert des archives.....	6
3.3	Déploiement :.....	6
3.4	Relancer le service.....	6
4	Exploitation des serveurs.....	7
4.1	Base de données.....	7
4.1.1	Lancement et arrêt de la base MaxDB.....	7
	Démarrage.....	7
	Arrêt.....	7
4.2	Sauvegarde.....	7
4.2.1	Sauvegarde manuelle.....	9
	Creation d'un média de BACKUP.....	9
	Lancement d'une sauvegarde.....	10
4.2.2	Restauration d'une sauvegarde.....	10
	Restauration d'un backup sur une instance existante.....	10
	Restauration d'un backup sur une nouvelle instance.....	12
5	Annexes	13
5.1	Script de démarrage de Jboss.....	13
5.2	Mise en place de l'update automatique des clients.....	15

1 Description de l'application

Architecture 3 tiers :

- un serveur d'application, nommé PHARMED et ayant pour IP IP_PHARMED
- un serveur de base de données nommé BD_SRV
- un client Java

2 Installation de Pharmed

2.1 Installation des pré-requis

Sur le serveur d'application **PHARMED**

- Créer un répertoire \$MYCVS_HOME
- Récupérer les sources dans \$MYCVS_HOME par

```
$> cvs co PRM dbtools aair-to-pharmed
```

- PRM : application Pharmed
- dbtools : outil de génération de code (EJBs, filtres, build.xml,...) présent dans Pharmed cette génération est effectuée à partir des fichiers database.xml et finder.xml présent à la racine de PRM.
- aliaforms (présent dans PRM): permet de générer l'interface graphique du client lors de l'exécution. Les interfaces sont décrites par des fichiers xml et les commandes associées aux actions sont définies par des scripts jpython.
- Installer le JDK 1.4.1 ou supérieur dans \$JAVA_HOME

Attention : certains problèmes sont apparus sous Windows avec la jdk1.5

le serveur Jboss 3.0.6 ne fonctionne pas avec la jdk1.5

- Installer Ant dans \$ANT_HOME à partir de

```
http://mir2.ovh.net/ftp.apache.org/dist/ant/binaries/apache-ant-1.6.5-bin.tar.gz
```

- Installer JBoss 3.0.6 dans \$JBOSS_HOME à partir de

```
http://prdownloads.sourceforge.net/jboss/jboss-3.0.6.zip?download
```

- **Attention** : l'application ne tourne pas avec Jboss > 3.0.6

2.2 Initialisation de la base

Sur le serveur de base de données **BD_SRV** :

- Installer les paquets maxdb par

```
$> apt-get install maxdb-server-7.5.00 maxdb-server maxdb-loadercli maxdb-dbmcli maxdb-lserver maxdb-scripts
```

maxdb-scripts est un paquet réalisé par Aliacom qui crée des scripts facilitant la création des bases de données (comme *createmaxdb*).

- Créer une base de données PHARMED en tant qu'utilisateur système dba par

```
$> createmaxdb PHARMED 1000 100
```

- Configurer maxdb pour démarrer automatiquement la base pour ça ajouter dans /etc/default/maxdb-server :

```
DATABASES="PHARMED"
```

- Démarrer maxdb par

```
$> /etc/init.d/maxdb-xserver start  
$> /etc/init.d/maxdb-server start
```

- La structure de la base est vide. Initialiser la base par \$CVSHOME/shema.sql en tant que sdb

```
$> dbmcli -u dbm,dbm -d PHARMED -i shema.sql
```

- Initialiser les données de la base par \$CVSHOME/initdb.sql en tant que sdb

```
$> dbmcli -u dbm,dbm -d PHARMED -i initdb.sql
```

- Insérer les procédures stockées présentes dans \$MYCVS_HOME/PRM/bd.sapdb:

```
$> loadercli -n localhost -d PHARMED -u dummy,dummy -b uad_procs.sql  
$> loadercli -n localhost -d PHARMED -u dummy,dummy -b protocole_procs.sql  
$> loadercli -n localhost -d PHARMED -u dummy,dummy -b produit_procs.sql  
$> loadercli -n localhost -d PHARMED -u dummy,dummy -b delivrance_ind_procs.sql
```

2.3 Génération des archives

Sur le serveur d'application **PHARMED** :

- Dans \$MYCVS_HOME/PRM, lancer

```
./generate_n_build.sh  
ant deployment
```

2.4 Initialisation de JBoss

Sur le serveur d'application **PHARMED** :

- Depuis \$MYCVS_HOME/PRM/deployment/server/pharmed/lib

```
$> cp $MYCVS_HOME/PRM/dist-server/aliacom.jar \  
$MYCVS_HOME/PRM/dist-client/fop.jar \  
$MYCVS_HOME/PRM/dist-client/jython.jar .
```

- Depuis le répertoire \$JBASS_HOME/server

```
$> ln -s $MYCVS_HOME/PRM/deployment/server/pharmed .
```

2.5 Sur un client

- Installer une JDK 1.4.x

- Créer un répertoire Pharmed/
- Copier dans Pharmed/jars/ le contenu de \$MYCVS_HOME/PRM/deployment/client/
- Copier dans Pharmed/ le contenu de \$MYCVS_HOME/PRM/dist-client/

2.6 Lancement

Sur le serveur d'application **PHARMED** :

- Vérifier que les locales sont bien réglé sinon exporter les variables nécessaires:

```
pharmed:/home/pharmed/jboss/server# locale
LANG=fr_FR
LC_CTYPE="fr_FR"
LC_NUMERIC="fr_FR"
LC_TIME="fr_FR"
LC_COLLATE="fr_FR"
LC_MONETARY="fr_FR"
LC_MESSAGES="fr_FR"
LC_PAPER="fr_FR"
LC_NAME="fr_FR"
LC_ADDRESS="fr_FR"
LC_TELEPHONE="fr_FR"
LC_MEASUREMENT="fr_FR"
LC_IDENTIFICATION="fr_FR"
LC_ALL=
```

Depuis le répertoire \$JBASS_HOME/bin:

- Les rôles de l'application sont définis grâce au fichier roles.xml présent à la racine de PRM il faut indiquer le fichier à utilisé lors du lancement de jboss:

```
$> run.sh -c pharmed -Droles.path=$MYCVS_HOME/PRM/roles.xml
```

Tester l'application depuis le serveur **PHARMED** :

- Depuis \$MYCVS_HOME/PRM/deployment/client

```
$> $JAVA_HOME/bin/java -classpath \
  `ls *.jar ../../dist-client/*.jar | tr '\n' ':'` . \
  -jar ../../dist-client/aaair-client.jar
127.0.0.1
```

Tester l'application depuis un client sur une autre machine :

- Depuis Pharmed/

```
$> java -classpath `ls *.jar jars/*.jar | tr '\n' ':'`. -jar aair-client.jar IP_PHARMED
```

3 Déploiement d'une nouvelle version

3.1 Sur la machine de développement

```
cd cvs/PRM
ant deployment

Buildfile: build.xml

aliacom-lib-build:
.....
aair-dist:
dist:
dist:

deployment:
  [copy] Copying 42 files to /u/bellau/cvs/PRM/deployment/client
  [copy] Copying 2 files to /u/bellau/cvs/PRM/deployment/pharmed/deploy

BUILD SUCCESSFUL
Total time: 17 seconds
```

3.2 Transfert des archives

On dépose sur le serveur par ssh / rsync

```
bellau@boa:~/cvs/PRM/deployment$
rsync -r * pharmed@IP_PHARMED:~/deployment
```

3.3 Deploiement

```
grosminet:~/pharmed/deployment/pharmed/deploy# ls
aliacom.jar  pharmedmq-service.xml  scheduler-service.xml
CVS         sapdb-service.xml      Server.jar
grosminet:~/deployment/pharmed/deploy# cp Server.jar aliacom.jar /pharmed/deploy/
```

3.4 Relancer le service

Un script updateClient met à jour automatiquement le client (cf Annexes).

Arreter et redémarrer jboss (cf Annexes)

```
/etc/init.d/pharmed.sh stop  
(et/ou killall java)  
/etc/init.d/pharmed.sh start
```

4 Exploitation des serveurs

4.1 Base de données

4.1.1 Lancement et arrêt de la base MaxDB

On utilise les scripts de démarrage installés par les paquets Debian. Deux services sont utilisés : le serveur de connexions réseau et le serveur de base de données.

Sur le serveur **BD_SRV** :

Démarrage

```
# /etc/init.d/maxdb-xserver start  
Starting MaxDB communication server: x_server.  
  
# /etc/init.d/maxdb-server start  
Starting MaxDB databases: PHARMED
```

Arrêt

```
# /etc/init.d/maxdb-server stop  
Stopping all MaxDB databases: PHARMED  
# /etc/init.d/maxdb-xserver stop  
Stopping MaxDB communication server: x_server.
```

4.2 Sauvegarde

La sauvegarde est réalisée directement par la base de données sur le lecteur de bande. L'opération étant gérée par la crontab, un mail est envoyé par la cron, sur le compte *root*.

On peut utiliser la commande **mutt** pour regarder les logs de la sauvegarde.

Une sauvegarde réussie présente une sortie similaire à :

```
OK
Returncode          0
Date                20041216
Time                00105249
Server              sapdb
Database             PHARMED
Kernel Version      Kernel    7.5.0    Build 019-000-000-000
Pages Transferred   1600
Pages Left          0
Volumes             1
Medianame           file_backup
Location            /backup/backup
Errortext
Label               DAT_000000002
Is Consistent       true
First LOG Page      181329
Last LOG Page
DB Stamp 1 Date     20041216
DB Stamp 1 Time     00105249
DB Stamp 2 Date
DB Stamp 2 Time
Page Count          1578
Devices Used        1
Database ID         sapdb:PHARMED_20040930_145159
Max Used Data Page  0
```

Un échec dans la sauvegarde (absence de bande ou bande défectueuse) présente en général une sortie plus courte :

```
...
Returncode          - NUMBER
...
```

En général, la vue de la liste de message permet d'avoir une vue globale des sauvegardes : les messages de 3 lignes sont des échecs, celles de 25 lignes correspondent à des sauvegardes réussies.

4.2.1 Sauvegarde manuelle

Cette section décrit la sauvegarde et la restauration manuelles d'une instance

Creation d'un média de BACKUP

Tout d'abord il faut se connecter à l'instance de la base.

Pour cela on utilisera l'utilitaire dbmcli, en tant qu'utilisateur **sapdb** :

```
dbmcli -u dbm,dbm -d PHARMED -n localhost
```

- -u : utilisateur et mot de passe de l'administrateur;
- -d : nom de l'instance (**PHARMED**);
- -n : adresse du serveur MaxDB;

Création du médium de backup :

Attention : si le médium existe déjà (ce qui est le cas), l'opération échoue si on définit un nouveau médium. Regardez la liste des média existants par :

```
dbmcli on PHARMED>medium_getall
OK
data      /var/tmp/sapdb_data.PHARMED      FILE      DATA      0      8      YES      NO
20030502140223  20030502140223
log       /var/tmp/sapdb_log.PHARMED        FILE      AUTO       0      8      YES      NO
20030502140223  20030502140223
file_backup  /backup/backup      FILE      DATA      0      8      YES      NO
20030502143320  20030502143320
tape_backup /dev/st0            TAPE      DATA      64000    8      NO       NO
20030502144641  20030502144641
aliacom_backup /tmp/pharmed.db    FILE      DATA      0      0      YES      NO
20060105173346  20060105174301
```

```
dbmcli on PHARMED>medium_put file_backup /backup/backup FILE DATA 0 0 YES
OK
---
dbmcli on PHARMED>
```

Dans l'exemple le nom du médium de sauvegarde est *file_backup*.

Le fichier de sauvegarde sera écrit dans */backup/backup*

Lancement d'une sauvegarde

```
dbmcli on PHARMED>util_connect dbm,dbm
OK

---

dbmcli on PHARMED>backup_start file_backup
OK
Returncode          0
Date                20041216
Time                00105249
Server              sapdb
Database            PHARMED
Kernel Version      Kernel   7.5.0   Build 019-000-000-000
Pages Transferred   1600
Pages Left          0
Volumes            1
Medianame          file_backup
Location           /backup/backup
Errortext
Label              DAT_000000002
Is Consistent      true
First LOG Page     181329
Last LOG Page
DB Stamp 1 Date    20041216
DB Stamp 1 Time    00105249
DB Stamp 2 Date
DB Stamp 2 Time
Page Count         1578
Devices Used       1
Database ID        sapdb:PHARMED_20040930_145159
Max Used Data Page 0
```

La sauvegarde est disponible là : */backup/backup*

4.2.2 Restauration d'une sauvegarde

Restauration d'un backup sur une instance existante

Installation et exploitation PHARMED

```
dbmcli on PHARMED>db_cold
OK
---
dbmcli on PHARMED>util_connect dbm,dbm
OK
---
dbmcli on PHARMED>recover_start test_backup DATA
OK
Returncode          0
Date                20041216
Time                00105736
Server              sapdb
Database            PHARMED
Kernel Version      Kernel   7.5.0   Build 019-000-000-000
Pages Transferred   1600
Pages Left          0
Volumes            1
Medianame           test_backup
Location            /tmp/test
Errortext
Label               DAT_000000002
Is Consistent       true
First LOG Page      181329
Last LOG Page
DB Stamp 1 Date     20041216
DB Stamp 1 Time     00105249
DB Stamp 2 Date
DB Stamp 2 Time
Page Count          1578
Devices Used        1
Database ID         sapdb:PHARMED_20040930_145159
Max Used Data Page  0
---
dbmcli on PHARMED>db_warm
```

```
OK
```

```
---
```

- *db_cold* permet de mettre la base dans un état d'administration.
- *util_connect* permet de se connecter en mode utilitaire. Enfin
- *recover_start* permet de lancer la restauration de la base.
- Finalement on remet la base dans l'état de fonctionnement normal en appelant *db_warm*.

Restauration d'un backup sur une nouvelle instance

Pour créer la base, on utilise la procédure normale. Puis on crée le médium de backup (voir *Création d'un média de BACKUP*).

```
dbmcli on PHARMED2>db_cold
```

```
OK
```

```
---
```

```
dbmcli on PHARMED2>util_connect dbm,dbm
```

```
OK
```

```
---
```

```
dbmcli on PHARMED2>db_activate recover test_backup
```

```
OK
```

Returncode	0
Date	20041216
Time	00112455
Server	sapdb
Database	PHARMED
Kernel Version	Kernel 7.5.0 Build 019-000-000-000
Pages Transferred	1600
Pages Left	0
Volumes	1
Medianame	test_backup
Location	/tmp/test
Errortext	
Label	DAT_000000002

```
Is Consistent          true
First LOG Page         181329
Last LOG Page
DB Stamp 1 Date        20041216
DB Stamp 1 Time        00105249
DB Stamp 2 Date
DB Stamp 2 Time
Page Count             1578
Devices Used           1
Database ID            sapdb:PHARMED2_20040930_145159
Max Used Data Page     0

---
dbmcli on PHARMED2>db_warm
OK
---
```

5 Annexes

5.1 Script de démarrage de Jboss

Dans /etc/init.d/ : Pharmed.sh:

```
#!/bin/sh

export JAVA_HOME=/usr/java/j2sdk1.4.2
export LANG=fr_FR
export JBOSS_HOME=/home/pharmed/jboss
export PATH=$PATH:$JBOSS_HOME/bin:$JAVA_HOME/bin:$JAVA_HOME/jre/bin

RETVAL=0

# See how we were called.
case "$1" in
```

```
start)
    cd $JBASS_HOME/bin
    if [ -f /var/run/jboss.pid ] ; then
        echo "Pharmed est deja démarré "
    else
        sh runPharmed.sh > /dev/null 2>&1 &
        #
        echo $! > /var/run/jboss.pid
        sleep 30
        until [ `tail -n3 $JBASS_HOME/server/pharmed/log/server.log | grep "Started in" |
wc -l` -eq 1 ] ; do
echo "Waiting start"
        sleep 5
        done
        echo "Pharmed démarré avec succès "
    fi
    ;;

stop)
    echo -n "Arret pharmed: "
        #echo -n "JBoss stoping:"
    if [ -f /var/run/jboss.pid ] ; then
        kill -15 `cat /var/run/jboss.pid`
        rm -rf /var/run/jboss.pid
        echo "Pharmed arret
    else
        echo "Pharmed n'est pas démarré"
    fi
    ;;

restart)
    echo -n "Restarting jboss daemon: "
    $0 stop
    sleep 2
    fi
    ;;

stop)
    echo -n "Arret pharmed: "
```

```
        #echo -n "JBoss stoping:"
    if [ -f /var/run/jboss.pid ] ; then
        kill -15 `cat /var/run/jboss.pid`
        rm -rf /var/run/jboss.pid
        echo "Pharmed arret"
    else
        echo "Pharmed n'est pas démarré"
    fi
    ;;

restart)
    echo -n "Restarting jboss daemon: "
    $0 stop
    sleep 2
    $0 start
    ;;
esac
```

dans \$JBOSS_HOME/bin: runPharmed.sh

```
export JAVA_HOME=/usr/java/java
export JAVA_OPTS='-server -Xmx768m'
export LC_ALL='fr_FR.iso88591'
export LANG='fr_FR.iso88591'
sh      run.sh      -Dpython.cachedir=/tmp      -Djava.awt.headless=true      -Dpython
.path=/pharmed/deploy/Server.jar -Droles.path=/home/pharmed/roles.xml -c pharmed &
echo $! > /var/run/jboss.pid
```

5.2 Mise en place de l'update automatique des clients

Coté serveur il est nécessaire de mettre en place un serveur web (apache par exemple) avec un accès sur le dossier : PRM_HOME/deployment/.

Coté client il est nécessaire de réaliser un script bat ou bash afin de lancer l'update avant le lancement du client:

Installation et exploitation PHARMED

```
javaw -jar updateAgent.jar IP_SERVEUR_JBOSS/deployment/client  
javaw -jar aair-client.jar IP_SERVEUR_JBOSS
```

updateAgent.jar est disponible dans PRM_HOME/deployment/