

JGenea - Guide Utilisateur

**L'application généalogique 100% java
par Templ**

JGenea - Guide Utilisateur: L'application généalogique 100% java

par Templ

Publié \$Date\$

Ce guide d'utilisation est destiné aux personnes désirant utiliser JGenea pour leurs travaux généalogiques (constitution de leur généalogie, exploitation de dépouillements, génération de documentation au format html & pdf...). Il détaille tous les cas d'utilisations de cette application.

Bien qu'entièrement écrit en Java, aucune connaissance en programmation java n'est nécessaire; il est cependant préférable de posséder néanmoins des bases d'informatique.

Si vous remarquez des erreurs ou des points non suffisamment détaillés, n'hésitez pas à me le faire savoir pour améliorer ce guide d'utilisation. Celui-ci sert de point de départ; par la suite, il est recommandé d'utiliser les forums de discussions pour d'éventuelles questions, suggestions.

Table des matières

I. Introduction	
1. Comment utiliser JGenea?	
Utiliser JGenea en mode autonome.	3
Utiliser JGenea en mode client / serveur.	3
Utiliser JGenea en mode client / serveur avec une interface web.	3
II. Installation de JGenea	
2. Installation de JGenea Ihm	
Récupération JGenea Ihm	6
Installation JGenea Ihm dans le détail	6
3. Installation de JGenea Web	
Récupération JGenea Web	9
Installation JGenea Web dans le détail	9
III. Configuration de JGenea	
4. Configuration de JGenea Ihm	
Configuration des bases de données dans JGenea Ihm	12
Configuration des accès aux images dans JGenea Ihm	13
Configuration des templates html dans JGenea Ihm	13
Configuration des échanges GEDCOM dans JGenea Ihm	13
5. Configuration de JGenea Web	
Configuration des bases de données dans JGenea Web	14
IV. Exportations, sauvegardes et importations de données dans JGenea	
6. Exportations de données	
Exportation sql de données	18
Exportation gedcom de données	18
7. Sauvegardes des données	
Sauvegardes de la base de données	20
Sauvegardes par une exportation sql	20
Sauvegardes par une exportation gedcom	20
8. Importations de données	
Importation sql de données	21
Importation gedcom de données	21
V. Fonctionnalités de JGenea	
9. Fonctionnalités de JGenea Ihm	
Fonctionnalité 1	24
10. Fonctionnalités de JGenea Web	
Fonctionnalité 1	25
VI. Cas concrets	
11. Réaliser sa généalogie	
12. Traiter des dépouillements	
13. Gérer ses documents	
14. Partager ses recherches grâce à Internet	
A. Installation complète & configuration de JGenea Web sur un Linux Debian	
Installation du jdk 1.3 de Blackdown	32
Installation de Tomcat / JBoss	32
Configuration de JBoss / Tomcat pour JGenea Web	32
Configuration de JBoss / Tomcat avec PJA (go.sh)	33
Installation de JGenea Web	34
Scripts d'administration de JBoss / Tomcat	34
B. Installation complète & configuration de Postgresql sur un Linux Debian	
Installation de Postgresql 7.1	37
Installation du driver jdbc pour Postgresql	38
Fonctionnement de JGenea avec Postgresql	38
Scripts d'administration de Postgresql	40

Liste des exemples

2.1. script de lancement de l'ihm sous Unix / Linux	6
2.2. script de lancement de l'ihm sous Windows NT	7
2.3. script de lancement de l'ihm sous Windows 9x	7
2.4. script de Hypersonic en mode serveur sous Windows NT	7
2.5. script de Hypersonic en mode serveur sous Windows NT	7
4.1. fichier Conf.properties	12
4.2. fichier Conf.properties	13
5.1. fichier Db.properties	14
5.2. fichier struts-config.xml	15
A.1. fichier /etc/apt/sources.list	32
A.2. Partie de jboss.jcml à changer (jdbc)	33
A.3. Ligne de commande de lancement de JBoss / Tomcat avec PJA (extrait go.sh)	33
A.4. Script de lancement de JBoss go.sh	34
A.5. Script de lancement de JBoss jboss (à mettre dans /etc/init.d)	35
B.1. fichier /etc/apt/sources.list	37
B.2. Fichier de configuration des connexions à Postgresql (/etc/postgresql/pg_hba.cong)	38
B.3. Fichier de configuration des bases de données de JGenea Ihm (conf/properties/Conf.properties)	39
B.4. Fichier de configuration des bases de données à afficher pour JGenea Web (conf/properties/Db.properties)	39
B.5. Fichier de configuration des bases de données dans Stuts pour JGenea Web (conf/xml/struts-config.properties)	39
B.6. Script de lancement de Postgresql (/etc/init.d/postgresql)	40
B.7. Script de lancement de Postgresql (/usr/lib/postgresql/bin/postgresql-startup)	41

Partie I. Introduction

JGenea est une application de généalogie écrite en java, opensource. Elle est en fait composée de sous-parties: un outil graphique pour constituer votre arbre généalogique, importer / exporter vos données, les visualiser de manière interactive, rechercher , générer des documentations sur vos recherches dans différents formats (html, pdf) et gérer vos recherches généalogiques (courriers, recherches sur le web...) et un outil de publication web.

- L'outil graphique 1 est basé sur les Swings Java (bibliothèque graphique java) et permet de saisir, d'importer & exporter, de visualiser & rechercher, de générer de manière simple et graphique vos informations généalogiques.
- L'outil web 2 est basé sur Struts (framework applicatif java) et permet un accès simple, rapide et contrôlé à vos informations généalogiques.

¹Cet outil est appelé dans tout le document ainsi que sur le site du projet (téléchargement, cvs...) *JGenea Ihm*.

²Cet outil est appelé dans tout le document ainsi que sur le site du projet (téléchargement, cvs...) *JGenea Web*.

Table des matières

- 1. Comment utiliser JGenea? 3
 - Utiliser JGenea en mode autonome. 3
 - Utiliser JGenea en mode client / serveur. 3
 - Utiliser JGenea en mode client / serveur avec une interface web. 3

Chapitre 1. Comment utiliser JGenea?

Il y a plusieurs manières d'utiliser JGenea suivant les besoins que l'on a. Elle fonctionne de pair avec une base de données relationnelles pour tout ce qui concerne les données: par défaut nous utilisons Hypersonic [<http://hsqldb.sourceforge.net>], une base relationnelle opensource écrite en java. Il est cependant possible d'utiliser n'importe quelle base relationnelle supportant le sql.

Une base de type Hypersonic est fourni avec la distribution. Elle est pré-remplie avec les informations concernant les pays, départements, communes et types d'actes. Celle-ci peut être employée comme base uniquement utilisable par JGenea Ihm et ne peut donc recevoir qu'une connexion à la fois. Elle peut également être utilisée en mode client / serveur et donc être partagée par un ou plusieurs JGenea Ihm (en local ou sur une machine distante) et par un ou plusieurs JGenea Web. Nous verrons de manière contrète dans le paragraphe Configuration comment paramétrer ces différents cas.

JGenea (Ihm & Web) peut être configuré pour pointer sur plusieurs bases de données. Dans ce cas, au lancement (pour JGenea Ihm) et pendant la phase d'authentification (pour JGenea Web), il est permis de choisir sur quelle base de données, on désire se connecter.

Utiliser JGenea en mode autonome.

Ce type d'utilisation consiste à utiliser la base pré-remplie de la distribution en mode exclusif. Cette base est chargée au démarrage de JGenea Ihm. Pour le moment, il n'est pas recommandé d'utiliser ce mode car le temps de chargement est long.

En effet, vu la taille des tables des départements et communes, Hypersonic, dans cette version met plus de 30 secondes à la charger. Cependant on m'a assuré que dans la version 1.71 de ce moteur de base de données, ce temps serait divisé par dix et deviendrait donc plus acceptable.

Dans ce mode, JGenea Web n'est pas utilisable. Il est donc fortement conseillé pour le moment d'utiliser JGenea en mode client / serveur qui permet de ne charger les données que la première fois.

Utiliser JGenea en mode client / serveur.

Ce type d'utilisation consiste à utiliser la base pré-remplie de la distribution en mode client / serveur ou une autre base de données fonctionnant comme serveur (postgresql, oracle...). Cela a le grand avantage de pouvoir partager les données entre plusieurs applications et de ne charger les données qu'une seule fois. De plus les applications n'ont pas besoin d'être nécessairement sur une même machine.

Cette utilisation implique d'être conscient des problèmes de sécurité qu'elle sous-entend. En effet, tout le monde peut se connecter sur la base (après une identification au niveau de la base de données): le port est ouvert sur l'extérieur. Il faut donc penser à protéger son réseau vis-à-vis de la base (au moyen d'un firewall par exemple).

Utiliser JGenea en mode client / serveur avec une interface web.

Ce type d'utilisation consiste à se placer dans le cas d'une base en mode client / serveur. On fait alors pointer JGenea Web sur cette base comme source de données.

JGenea Ihm est alors utilisée comme interface d'administration de l'interface web au niveau des accès et des droits. Il permet également de gérer les données affichées par JGenea Web.

Partie II. Installation de JGenea

L'installation de JGenea est relativement facile. Le prérequis est évidemment une machine virtuelle java. Les tests et les développements ont été réalisés avec le jdk 1.3.1 de Sun disponible sur le site de Sun [<http://java.sun.com/j2se/1.3/>]. Cette partie ne traitera pas de l'installation de jdk. Cependant en annexe, se trouve les étapes à suivre pour installer le jdk de BlackDown sur un Linux Debian.

Par la suite, nous considérerons que `JDKHOME` représente le répertoire d'installation de la machine virtuelle java. Dans tous les scripts suivants, elle devra être remplacée.

Java étant multiplateforme, JGenea peut être installé autant sur une machine windows que sur une machine unix / linux. Les tests et développements ont été réalisés aussi bien sur des machines Windows 9x / NT que sur des machines Linux Debian

Pour utiliser JGenea Web, il est également nécessaire d'installer un serveur applicatif java supportant les servlets, les jsps et Struts. Pour les tests et développements, il a été utilisé la distribution Tomcat 4.0.1 / JBoss 2.2.4 disponible sur le site de JBoss [<http://jboss.org>]. La mise en oeuvre de ce serveur sera détaillé par la suite.

Table des matières

- 2. Installation de JGenea Ihm
 - Récupération JGenea Ihm 6
 - Installation JGenea Ihm dans le détail 6
- 3. Installation de JGenea Web
 - Récupération JGenea Web 9
 - Installation JGenea Web dans le détail 9

Chapitre 2. Installation de JGenea Ihm

Récupération JGenea Ihm

La distribution de JGenea Ihm est disponible sur le site [http://sourceforge.net/project/showfiles.php?group_id=47631]. Il est recommandé d'utiliser la dernière version de l'application.

La distribution est structurée de la manière suivante:

- *bin/*: répertoire contenant les scripts de lancement de l'application (ihm, console, explorateur gecom et mini serveur web).
- *conf/*: répertoire contenant les fichiers de configuration de l'application ainsi que les templates html.
- *lib/*: répertoire contenant les bibliothèques java (jgenea, hypersonic, iText).
- *db/*: répertoire contenant la base pré-remplie.

Installation JGenea Ihm dans le détail

Après avoir dezippé la distribution de JGenea Ihm, il est nécessaire d'éditer tous les scripts de lancement pour pointer vers la bonne machine virtuelle java. Si vous êtes sous Windows 9x / NT, il vous faut éditer les fichiers ayant l'extension bat et, si vous êtes sous Linux, il vous faut éditer les fichiers ayant l'extension sh.

Voici la liste des scripts de lancement.

```
bash$ ls -l *.bat
total 4
-rw-r--r-- 1 templ 244 Jul 18 22:04 console.bat
-rw-r--r-- 1 templ 217 Jul 18 22:04 gedihm.bat
-rw-r--r-- 1 templ 220 Jul 18 22:04 ihm.bat
-rw-r--r-- 1 templ 150 Jul 18 22:04 webservice.bat
bash$ ls -l *.sh
total 4
-rw-r--r-- 1 templ 265 Jul 18 22:04 console.sh
-rw-r--r-- 1 templ 233 Jul 18 22:04 gedihm.sh
-rw-r--r-- 1 templ 239 Jul 18 22:04 ihm.sh
-rw-r--r-- 1 templ 142 Jul 18 22:04 webservice.sh
```

Sous Linux, les scripts de lancement sont de la forme suivante. Il suffit de changer la valeur de la variable d'environnement JAVAHOME par le répertoire d'installation de votre jdk.

Exemple 2.1. script de lancement de l'ihm sous Unix / Linux

```
#!/bin/sh

export JAVAHOME=/usr/java/jdk1.3.1
export CLASSP=../lib/jgenea.jar:../lib/jgenea-fusion.jar:../lib/iText.jar:
../lib/hsqldb.jar:../lib/jakarta-regexp-1.2.jar:../conf/

$JAVAHOME/bin/java -classpath $CLASSP org.genealogie.ihm.Main
```

Exemple 2.2. script de lancement de l'ihm sous Windows NT

```
set JAVAHOME=d:\jdk1.3.1_01
set CLASSP=..\lib\jgenea.jar;..\lib\jgenea-fusion.jar;..\lib\iText.jar;
..\lib\hsqldb.jar;..\lib\jakarta-regexp-1.2.jar;..\conf\
%JAVAHOME%\bin\java -classpath %CLASSP% org.genealogie.ihm.Main
```

Attention

Sous Windows 9x, les scripts bat renvoie des erreurs car il ne prend pas bien en compte les variables d'environnement. Il faut donc modifier tous les fichiers ayant l'extension bat de la manière suivante.

Exemple 2.3. script de lancement de l'ihm sous Windows 9x

```
d:\jdk1.3.1_01\bin\java -classpath ..\lib\jgenea.jar;..\lib\jgenea-fusion.jar;
..\lib\iText.jar; ..\lib\hsqldb.jar;..\lib\jakarta-regexp-1.2.jar;..\conf\
org.genealogie.ihm.Main
```

Note

Pour lancer, la base Hypersonic de la distribution en mode serveur, il suffit de lancer runServer.bat ou runServer.sh (suivant votre os) après avoir positionné comme précédemment la valeur de la variable d'environnement JAVAHOME par le répertoire d'installation de votre jdk.

Exemple 2.4. script de Hypersonic en mode serveur sous Windows NT

```
set JAVAHOME=d:\jdk1.3.1_01
set CLASSP=..\lib\hsqldb.jar
%JAVAHOME%\bin\java -classpath %CLASSP% org.hsqldb.Server %1 %2 %3 %4 %5 %6 %7 %8 %9
```

Exemple 2.5. script de Hypersonic en mode serveur sous Windows NT

```
#!/bin/sh
export JAVAHOME=/usr/java/jdk1.3.1
export CLASSP=../lib/hsqldb.jar
$JAVAHOME/bin/java -classpath $CLASSP org.hsqldb.Server %1 %2 %3 %4 %5 %6 %7 %8 %9
```

Voici la ligne commande pour démarrer la base Hypersonic jgenea sur le port 66 sous Windows:

```
bash$ cd db
bash$ runServer -port 66 -database jgenea
Server 1.6 is running
Press [Ctrl]+[C] to abort
```

Voici la ligne commande pour démarrer la base Hypersonic jgenea sur le port 66 sous Unix / Linux:

```
bash$ cd db
bash$ ./runServer.sh -port 66 -database jgenea
Server 1.6 is running
Press [Ctrl]+[C] to abort
```

Chapitre 3. Installation de JGenea Web

Récupération JGenea Web

La distribution de JGenea Web est disponible sur le site [http://sourceforge.net/project/showfiles.php?group_id=47631]. Il est recommandé d'utiliser la dernière version de l'application.

La distribution est structurée de la manière suivante:

- *jgenea.war*: fichier contenant tous les fichiers de l'application web JGenea ainsi que toutes les bibliothèques et taglibs de Struts.

Installation JGenea Web dans le détail

Après avoir dezippé la distribution de JGenea Web, il est nécessaire de déployer l'application web sur le serveur applicatif java. Nous prenons ici comme serveur d'application Tomcat 4.0.1 / JBoss 2.2.4 mentionné précédemment. Pour ce faire, il suffit de copier *jgenea.war* dans le répertoire *jboss/deploy*. Le déploiement est automatique!

Attention

Lors du déploiement, le serveur applicatif va chercher à se connecter à la base de données configuré par défaut pour créer le pool de connection. Par défaut, est configurée une base Hypersonic en mode client / serveur utilisant le port 66.

Note

Nous verrons en annexe comment installer, JBoss / Tomcat sur une machine Linux Debian pour rendre sa généalogie accessible sur le Web (avec des notions de droits d'accès).

Partie III. Configuration de JGenea

Table des matières

- 4. Configuration de JGenea Ihm
 - Configuration des bases de données dans JGenea Ihm 12
 - Configuration des accès aux images dans JGenea Ihm 13
 - Configuration des templates html dans JGenea Ihm 13
 - Configuration des échanges GEDCOM dans JGenea Ihm 13
- 5. Configuration de JGenea Web
 - Configuration des bases de données dans JGenea Web 14

Chapitre 4. Configuration de JGenea Ihm

Configuration des bases de données dans JGenea Ihm

La configuration des bases de données dans JGenea Ihm se fait dans le fichier `Conf.properties`.

Les deux premières clés servent à lister les bases disponibles (`bases`) en leur associant un descriptif sommaire (`desc-bases`). Ensuite pour chaque clé, on définit les propriétés nécessaires à la connexion.

Elles sont les suivantes:

- `[cle].typeJdbc`: permet de définir le type de dbase de données (par exemple, `hsql` pour Hypersonic).
- `[cle].driverJdbc`: permet de définir la classe du driver jdbc qui sera utilisé pour se connecter à la base. Pour Hypersonic, la valeur sera `org.hsqldb.jdbcDriver`.
- `[cle].urlJdbc`: permet de définir l'url jdbc qui sera utilisé pour se connecter à la base. Elle spécifie pour une url réseau la machine où se trouve la base et le port sur lequel elle écoute. Pour Hypersonic, la valeur sera de la forme `jdbc:hsqldb:hsql://machine:port`.
- `[cle].loginJdbc`: permet de définir l'utilisateur utilisé pour se connecter à la base.
- `[cle].passwordJdbc`: permet de définir le mot de passe de l'utilisateur spécifié pour se connecter à la base.

Exemple 4.1. fichier `Conf.properties`

```
(...)  
# DB  
bases=test,test1  
desc-bases=Base hsql,Base hsql1  
  
# Connexion  
# Hsql  
test.typeJdbc=hsql  
test.driverJdbc=org.hsqldb.jdbcDriver  
test.urlJdbc=jdbc:hsqldb:../db-hsql/jgenea  
test.loginJdbc=sa  
test.passwordJdbc=  
  
# Hsql  
test1.typeJdbc=hsql  
test1.driverJdbc=org.hsqldb.jdbcDriver  
test1.urlJdbc=jdbc:hsqldb:hsql://localhost:66  
test1.loginJdbc=sa  
test1.passwordJdbc=  
(...)
```

Configuration des accès aux images dans JGenea

Ihm

La configuration relative à l'accès aux images dans JGenea Ihm se fait dans le fichier `Conf.properties`.

Les informations de ce fichier de configuration servent à la fois au mini serveur web en spécifiant son "document root" (`root`) et à l'interface graphique en spécifiant l'url de base du répertoire des images partagées (`IMAGES`).

Elles sont les suivantes:

- `[cle].IMAGES`: permet de définir l'url de base du répertoire des images partagées.
- `[cle].root`: permet de spécifier pour le mini serveur web son "document root".

Exemple 4.2. fichier `Conf.properties`

```
(...)  
#IMAGES=http://localhost:77/  
IMAGES=http://localhost/images  
root=d:/g.bin/imgs/  
(...)
```

Configuration des templates html dans JGenea Ihm

Depuis la version 1.0 de JGenea Ihm, la génération html se fait au moyen de fichiers xsl.

Configuration des échanges GEDCOM dans JGenea Ihm

A venir...

Chapitre 5. Configuration de JGenea Web

Comme cela a été vu précédemment, JGenea Web est constitué d'un unique fichier (`jgenea.war`). Ce fichier contient les librairies java, les pages jsp, les images et les fichiers de configuration. Donc, pour pouvoir changer la configuration, il faut extraire tous ces fichiers inclus.

Pour ce faire, il suffit d'utiliser la commande `jar` disponible avec le jdk précédemment installé. Elle est au même endroit que les commandes `java` et `javac`.

Pour créer le fichier, on utilise l'option `cvf`; pour extraire les fichiers contenus, l'option `xvf` et pour lister les fichiers contenus `tvf`.

```
bash$ jar cvf jgenea.war *
bash$ jar tvf jgenea.war
bash$ jar xvf jgenea.war
```

Configuration des bases de données dans JGenea Web

La configuration des bases de données dans JGenea Web se fait à deux endroits: dans le fichier `WEB-INF/classes/properties/Db.properties` et dans le fichier `WEB-INF/struts-config.xml`.

Le premier fichier contient deux clés permettant de lister les pools de connexion disponibles (`bases`) ainsi qu'un descriptif des données vers lesquelles elles pointent (`bases-desc`). Les descriptifs doivent être relativement courts car ils sont affichés dans un menu déroulant sur la page d'authentification du site.

Exemple 5.1. fichier `Db.properties`

```
# DB
bases=Perso,Test1
bases-desc=Ma base,Base hsql1
```

Attention

Les identifiants de pools de connexion de la clé `bases` doivent correspondre à l'attribut `key` des définitions pools dans le fichier `struts-config.xml`.

Le deuxième fichier contient la définition des pools de connexion pour le framework Struts. Ils sont définis en début de fichier dans la balise `data-sources`.

Un pool est défini par la balise `data-source` (`key` représente l'identifiant du pool. Les propriétés relatives au pool sont définies au moyen de sous-balises `set-property` ayant l'attribut `property` différent suivant la propriété.

Pour un fonctionnement correct, il faut définir les propriétés suivantes:

- `autoCommit`: permet de déterminer si les modifications sur la base sont validées automatiquement ou non. Il est conseillé de le laisser à `false`.

- *description*: permet de décrire la base. Il peut être renseigné ou non. Ce paramètre n'a pas trop d'importance.
- *driverClass*: permet de définir la classe du driver jdbc qui sera utilisé pour se connecter à la base. Pour Hypersonic, la valeur sera `org.hsqldb.jdbcDriver`.
- *url*: permet de définir l'url jdbc qui sera utilisé pour se connecter à la base. Elle spécifie pour une url réseau la machine où se trouve la base et le port sur lequel elle écoute. Pour Hypersonic, la valeur sera de la forme `jdbc:hsqldb:hsqldb://machine:port`.
- *user*: permet de définir l'utilisateur utilisé pour se connecter à la base.
- *password*: permet de définir le mot de passe de l'utilisateur spécifié pour se connecter à la base.
- *minCount*: permet de définir le minimum de connexions utilisées dans le pool.
- *maxCount*: permet de définir le maximum de connexions utilisées dans le pool.

Exemple 5.2. fichier struts-config.xml

```
<data-sources>
  <data-source key="Perso">
    <set-property property="autoCommit" value="false" />
    <set-property property="description" value="Hsql" />
    <set-property property="driverClass" value="org.hsqldb.jdbcDriver" />
    <set-property property="maxCount" value="4" />
    <set-property property="minCount" value="2" />
    <set-property property="password" value="" />
    <set-property property="url" value="jdbc:hsqldb:hsqldb://localhost:66" />
    <set-property property="user" value="sa" />
  </data-source>
  <data-source key="Test1">
    <set-property property="autoCommit" value="false" />
    <set-property property="description" value="Hsql" />
    <set-property property="driverClass" value="org.hsqldb.jdbcDriver" />
    <set-property property="maxCount" value="4" />
    <set-property property="minCount" value="2" />
    <set-property property="password" value="" />
    <set-property property="url" value="jdbc:hsqldb:hsqldb://localhost:67" />
    <set-property property="user" value="sa" />
  </data-source>
</data-sources>
```

Attention

Si un pool est mal défini ou n'arrive pas à se connecter à la base de données correspondantes, JGenea Web ne se chargera pas correctement et ne pourra fonctionner. Les messages d'erreur apparaîtront dans les logs de la sortie standard du serveur applicatif.

Partie IV. Exportations, sauvegardes et importations de données dans JGenea

Toutes les actions qui suivent, peuvent être exécutées soit à partir de l'interface graphique, soit à partir de la console. Il est conseillé de faire ces manipulations à partir de la console. C'est pourquoi tous les exemples suivants sont réalisés à partir de la console.

Table des matières

- 6. Exportations de données 18
 - Exportation sql de données 18
 - Exportation gedcom de données 18
- 7. Sauvegardes des données 20
 - Sauvegardes de la base de données 20
 - Sauvegardes par une exportation sql 20
 - Sauvegardes par une exportation gedcom 20
- 8. Importations de données 21
 - Importation sql de données 21
 - Importation gedcom de données 21

Chapitre 6. Exportations de données

Exportation sql de données

Comme JGenea utilise une base de données relationnelles supportant le langage sql, il est possible d'exporter la totalité de sa base dans ce format dans une optique de sauvegarder ses données.

L'exportation au format sql la totalité de sa base se fait grâce à la commande suivante:

```
> export -type sql -repertoire ../dump
Etat: #####
Duree: 3s 14ms
>
```

Ceci va créer un sous-répertoire jj-mm-aaaa (où jj est le jour courant sur deux chiffres, mm est le mois courant sur deux chiffres et aaaa est l'année courante sur quatre chiffres) et écrit dedans un fichier par table exportée.

Le contenu du répertoire généré 02-06-2002 est:

```
bash$; ls 02-06-2002/
total 32
-rw-r--r-- 1 templ 131 acces.out.20-06-2002.sql
-rw-r--r-- 1 templ 65 acces_commune.out.20-06-2002.sql
-rw-r--r-- 1 templ 0 acces_departement.out.20-06-2002.sql
-rw-r--r-- 1 templ 122 acces_famille.out.20-06-2002.sql
-rw-r--r-- 1 templ 0 acces_pays.out.20-06-2002.sql
-rw-r--r-- 1 templ 73 acces_type_document.out.20-06-2002.sql
-rw-r--r-- 1 templ 118 538 acte.out.20-06-2002.sql
-rw-r--r-- 1 templ 450 document.out.20-06-2002.sql
-rw-r--r-- 1 templ 585 etat_recherche.out.20-06-2002.sql
-rw-r--r-- 1 templ 396 famille.out.20-06-2002.sql
-rw-r--r-- 1 templ 7 751 liaison_acte.out.20-06-2002.sql
-rw-r--r-- 1 templ 0 liaison_actes.out.20-06-2002.sql
-rw-r--r-- 1 templ 0 liaison_acte_document.out.20-06-2002.sql
-rw-r--r-- 1 templ 70 liaison_document.out.20-06-2002.sql
-rw-r--r-- 1 templ 0 liaison_documents.out.20-06-2002.sql
-rw-r--r-- 1 templ 17 673 liaison_famille.out.20-06-2002.sql
-rw-r--r-- 1 templ 100 liaison_recherche_adresses.out.20-06-2002.sql
-rw-r--r-- 1 templ 0 liaison_recherche_personne.out.20-06-2002.sql
-rw-r--r-- 1 templ 0 liaison_recherche_web_personne.out.20-06-2002.sql
-rw-r--r-- 1 templ 33 889 mariage.out.20-06-2002.sql
-rw-r--r-- 1 templ 0 page_supplementaire.out.20-06-2002.sql
-rw-r--r-- 1 templ 289 343 personne.out.20-06-2002.sql
-rw-r--r-- 1 templ 4 537 recherche.out.20-06-2002.sql
-rw-r--r-- 1 templ 2 203 recherche_adresse.out.20-06-2002.sql
-rw-r--r-- 1 templ 4 979 recherche_web.out.20-06-2002.sql
-rw-r--r-- 1 templ 1 400 recherche_web_adresse.out.20-06-2002.sql
-rw-r--r-- 1 templ 722 recherche_web_adresse_repertoire.out.20-06-2002.sql
-rw-r--r-- 1 templ 224 855 tables.out.20-06-2002.sql
-rw-r--r-- 1 templ 183 type_document.out.20-06-2002.sql
-rw-r--r-- 1 templ 448 type_recherche.out.20-06-2002.sql
```

Attention

Si deux exportations sql se font le même jour, la second écrase la première.

Exportation gedcom de données

JGenea est capable de générer des fichiers GEDCOM 5.5 comprenant soit la base complète soit les ascendants d'une personne donnée.

L'exportation au format gedcom de la totalité de sa base se fait grâce à la commande suivante:

```
> export -type gedcom -fichier export_complet_gecom.ged
Etat: #####
Duree: 3s 44ms
>
```

Toutes les données gedcom se trouveront alors dans le fichier précisé dans l'attribut fichier, ici export_complet_gecom.ged.

L'exportation au format gedcom de l'ascendance de la personne spécifiée se fait grâce à la commande suivante:

```
> personne -nom durand
| Nom & prenom
|-----
| Durand Jean ( 34 )
| Durand Marie ( 12 )
| Durand Joseph ( 21 )
|-----
>
> export -type gedcom -fichier export_ascendance_gecom.ged -id 12
Etat: #####
Duree: 3s 44ms
>
```

Toutes les données gedcom se trouveront alors dans le fichier précisé dans l'attribut fichier, ici export_ascendance_gecom.ged.

Attention

Si deux exportations gedcom se font avec le même nom de fichier d'exportation, la seconde écrasera la première.

Chapitre 7. Sauvegardes des données

Sauvegardes de la base de données

Il existe pour chaque base de données des façons spécifiques pour sauvegarder les données de celle-ci. Nous nous intéresserons ici au cas d'Hypersonic puisqu'il s'agit de la base utilisée par défaut avec JGenea.

Comme nous l'avons vu précédemment, les fichiers des données de la base Hypersonic se trouve dans la distribution dans le répertoire db.

Voici un aperçu de ce que contient ce répertoire:

```
bash$ ls db/
-rw-r--r-- 1 templ          20 jgenea.data
-rw-r--r-- 1 templ          75 jgenea.properties
-rw-r--r-- 1 templ      3 402 161 jgenea.script
-rwxr-xr-x 1 templ          97 runServer.sh
```

Le simple fait de copier les quatre fichiers (`jgenea.backup`, `jgenea.data`, `jgenea.properties`, `jgenea.script`) dans un répertoire permet de sauvegarder la base. Si un problème survient (destruction des fichiers, erreurs de manipulation...), il est possible en copiant ces fichiers à leur emplacement d'origine de récupérer les données.

Sauvegardes par une exportation sql

La meilleure façon de faire des sauvegardes avec JGenea est d'utiliser l'exportation sql. En effet elle date les exportations et peut les placer dans n'importe quel répertoire (un répertoire contenant toutes les sauvegardes par exemple).

Voici un aperçu de ce que contient le répertoire de sauvegardes dump:

```
bash$ ls dump/
-rw-r--r-- 1 templ      120 020 10-05-2002/
-rw-r--r-- 1 templ      131 045 12-05-2002/
-rw-r--r-- 1 templ      132 021 08-06-2002/
-rw-r--r-- 1 templ      147 573 15-06-2002/
```

On remarque que des sauvegardes ont été faites les 10/05/2002, 12/05/2002 08/06/2002 et 15/06/2002. Il est possible de régénérer la base en faisant une importation sql.

Sauvegardes par une exportation gedcom

La sauvegarde par fichier gedcom consiste à archiver des exportations au format gedcom et de les importer ensuite.

Chapitre 8. Importations de données

Attention

Avant la moindre importation, il est nécessaire de vider la base d'importation. Ceci se fait via la console avec la commande viderbase.

Importation sql de données

Importation gedcom de données

Partie V. Fonctionnalités de JGenea

Table des matières

- 9. Fonctionnalités de JGenea Ihm24
 - Fonctionnalité 124
- 10. Fonctionnalités de JGenea Web25
 - Fonctionnalité 125

Chapitre 9. Fonctionnalités de JGenea lhm

Fonctionnalité 1

A venir...

Chapitre 10. Fonctionnalités de JGenea Web

Fonctionnalité 1

A venir...

Partie VI. Cas concrets

Table des matières

11. Réaliser sa généalogie	
12. Traiter des dépouillements	
13. Gérer ses documents	
14. Partager ses recherches grâce à Internet	

Chapitre 11. Réaliser sa généalogie

Chapitre 12. Traiter des dépouillements

Chapitre 13. Gérer ses documents

Chapitre 14. Partager ses recherches grâce à Internet

Annexe A. Installation complète & configuration de JGenea Web sur un Linux Debian

Le but de ce paragraphe est de fournir un guide d'installation complet de JGenea Web (jdk, jboss / tomcat, scripts d'administration) sur une machine Linux Debian.

Attention

Cette partie sous-entend que votre système Linux Débian a correctement été installé et que vous possédez une liaison internet.

Installation du jdk 1.3 de Blackdown

L'installation va se faire au moyen de la commande apt-get qui sert à installer des packages de manière simple. Il faut cependant configurer le fichier /etc/apt/sources.list pour qu'il puisse accéder aux packages du jdk

Exemple A.1. fichier /etc/apt/sources.list

```
# See sources.list(5) for more information, especialy
# Remember that you can only use http, ftp or file URIs
# CDROMs are managed through the apt-cdrom tool.
deb http://http.us.debian.org/debian stable main contrib non-free
deb http://non-us.debian.org/debian-non-US stable/non-US main contrib non-free
deb http://security.debian.org stable/updates main contrib non-free

# jdk1.3
deb ftp://metalab.unc.edu/pub/linux/devel/lang/java/blackdown.org/debian woody non-free
```

L'installation se fait ensuite grâce à la commande suivante:

```
bash$ apt-get update
bash$ apt-get install j2sdk1.3
```

Une fois cela terminé, vous avez accès directement à java, javac...

Installation de Tomcat / JBoss

L'installation de Tomcat (version 4.0.1) / Jboss (version 2.4.4) n'est pas très compliqué. Après avoir récupéré la distribution sur le site de jboss [<http://jboss.org>], il suffit de le tarrer. Nous n'utiliserons pas les scripts de démarrage fournis, mais les scripts de la section Scripts d'aministration de JBoss / Tomcat.

Configuration de JBoss / Tomcat pour JGenea Web

Dans le cas de l'utilisation d'une base Hypersonic, il est possible de la lancer et de l'arrêter en même temps que JBoss / Tomcat. Toute la configuration se fait dans le fichier conf/catalina/jboss.conf.

Tous les pools de connexion préconfigurer (par exemple DefaultDS) sont inutiles et peuvent donc être supprimés car les pools sont à configurer au niveau de struts.

Attention

Par défaut dans la distribution, le port par défaut de la base Hypersonic pré-remplie est 66.

Exemple A.2. Partie de jboss.jcml à changer (jdbc)

```
<!-- ===== -->
<!-- JDBC -->
<!-- ===== -->

<mbean code="org.jboss.jdbc.JdbcProvider" name="DefaultDomain:service=JdbcProvider">
  <attribute name="Drivers">org.hsqldb.jdbcDriver</attribute>
</mbean>

<mbean code="org.jboss.jdbc.HypersonicDatabase"
      name="DefaultDomain:service=Hypersonic">
  <attribute name="Port">66</attribute>
  <attribute name="Silent">>true</attribute>
  <attribute name="Database">jgenea</attribute>
  <attribute name="Trace">>false</attribute>
</mbean>

( ... )

<!-- ===== -->
<!-- TOMCAT -->
<!-- ===== -->

<!-- Uncomment to add embedded catalina service -->
<mbean code="org.jboss.web.catalina.EmbeddedCatalinaServicesX"
      name="DefaultDomain:service=EmbeddedTomcat">
  <attribute name="Port">443</attribute>
</mbean>
```

Configuration de JBoss / Tomcat avec PJA (go.sh)

PJA [<http://www.eteks.com/pja/>] est une api java permettant d'afficher et de manipuler des images via un programme java fonctionnant sur une machine ne possédant pas de display X11 (dans le cas d'unix / linux) ou quand les ressources GDI viennent à manquer sous Windows.

Il suffit de modifier légèrement la ligne de commande de lancement de JBoss / Tomcat et d'ajouter les deux fichiers jar (pja.jar et pjatools.jar) dans le classpath de ce serveur applicatif.

Exemple A.3. Ligne de commande de lancement de JBoss / Tomcat avec PJA (extrait go.sh)

```
( ... )
JAVA_HOME=/usr/lib/j2sdk1.3
( ... )
```

```
JBOSS_CLASSPATH=$JBOSS_CLASSPATH:$JBOSS_HOME/lib/crimson.jar;$JBOSS_HOME/lib/pja.jar;
                                $JBOSS_HOME/lib/pjtools.jar
JAXP=-Djavax.xml.parsers.DocumentBuilderFactory=org.apache.crimson.
                                jaxp.DocumentBuilderFactoryImpl
JAXP="$JAXP -Djavax.xml.parsers.SAXParserFactory=org.apache.crimson.
                                jaxp.SAXParserFactoryImpl"
PJA=-Dawt.toolkit=com.eteks.awt.PJAToolkit
PJA="$PJA -Djava.awt.graphicsenv=com.eteks.java2d.PJAGraphicsEnvironment"
XBOOTPJA=-Xbootclasspath:$JAVA_HOME/jre/lib/rt.jar:$JBOSS_HOME/lib/pja.jar

case "$1" in
  start)
    shift
    #echo -n "JBoss starting:"
    if [ -f /var/run/jboss.pid ] ; then
      print_status running
    else
      $JAVACMD $HOTSPOT $JAXP $PJA $XBOOTPJA -classpath $JBOSS_CLASSPATH
                                org.jboss.Main catalina > /dev/null 2>&1 &
      echo $! > /var/run/jboss.pid
      until [ `tail -n 1 $JBOSS_HOME/log/server.log | grep "JBoss-2.4.4 Started" |
                                wc -l` -eq 1 ] ; do
        sleep 5
      done
      print_status success
    fi
  ;;
  (...)

```

Installation de JGenea Web

Pour installer JGenea Web, il suffit de copier le fichier `jgenea.war` issu de la distribution dans le répertoire `deploy/` de JBoss / Tomcat. Le déploiement de l'application web se fait automatiquement.

Note

Les logs de JBoss / Tomcat sont visibles dans le le fichier `log/server.log`

Scripts d'administration de JBoss / Tomcat

Exemple A.4. Script de lancement de JBoss `go.sh`

```
#!/bin/sh

#source function library.
. /etc/init.d/functions

TOMCAT_HOME=/applis/JBoss-2.4.4_Tomcat-4.0.1/catalina
JBOSS_HOME=/applis/JBoss-2.4.4_Tomcat-4.0.1/jboss
JAVA_HOME=/usr/lib/j2sdk1.3

JAVACMD=$JAVA_HOME/bin/java

JBOSS_CLASSPATH=$JBOSS_CLASSPATH:$JBOSS_HOME/bin/run.jar:$JAVA_HOME/lib/tools.jar
JBOSS_CLASSPATH=$JBOSS_CLASSPATH
```

Installation complète & configuration de JGenea Web sur un
Linux Debian

```
HOTSPOT=`java -version 2>&1 | grep HotSpot`"x"
if [ "$HOTSPOT" != "x" ]; then
    HOTSPOT="-server"
else
    HOTSPOT=""
fi

JBOSS_CLASSPATH=$JBOSS_CLASSPATH:$JBOSS_HOME/lib/crimson.jar
JAXP=-Djavax.xml.parsers.DocumentBuilderFactory=org.apache.crimson.
                                     jaxp.DocumentBuilderFactoryImpl
JAXP="$JAXP -Djavax.xml.parsers.SAXParserFactory=org.apache.
                                     crimson.jaxp.SAXParserFactoryImpl"

case "$1" in
    start)
        shift
        #echo -n "JBoss starting:"
        if [ -f /var/run/jboss.pid ] ; then
            print_status running
        else
            $JAVACMD $HOTSPOT $JAXP -classpath $JBOSS_CLASSPATH org.jboss.Main
                                     catalina > /dev/null 2>&1 &
            echo $! > /var/run/jboss.pid
            until [ `tail -n 1 $JBOSS_HOME/log/server.log |
                    grep "JBoss-2.4.4 Started" | wc -l` -eq 1 ] ; do
                sleep 5
            done
            print_status success
        fi
        ;;

    stop)
        shift
        #echo -n "JBoss stoping:"
        if [ -f /var/run/jboss.pid ] ; then
            kill -15 `cat /var/run/jboss.pid`
            rm -rf /var/run/jboss.pid
            until [ `tail -n 1 $JBOSS_HOME/log/server.log |
                    grep "Shutdown complete" | wc -l` -eq 1 ] ; do
                sleep 5
            done
            print_status success
        else
            print_status stopped
        fi
        ;;

    run)
        shift
        $JAVACMD $HOTSPOT $JAXP -classpath $JBOSS_CLASSPATH org.jboss.Main catalina
        ;;

    status)
        shift
        if [ -f /var/run/jboss.pid ] ; then
            print_status status_running
        else
            print_status status_stopped
        fi
        ;;

esac

exit 0
```

Exemple A.5. Script de lancement de JBoss jboss (à mettre dans /etc/init.d)


```
#!/bin/sh

export JAVA_HOME=/usr/lib/j2sdk1.3

export JBOSS_HOME=/applis/JBoss-2.4.4_Tomcat-4.0.1/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 $JBOSS_HOME/bin
    echo -n "Starting JBoss/Tomcat daemon: "
    $JBOSS_HOME/bin/go.sh start
    ;;

  stop)
    echo -n "Stopping JBoss/Tomcat daemon: "
    cd $JBOSS_HOME/bin
    $JBOSS_HOME/bin/go.sh stop
    ;;

  status)
    echo -n "Status JBoss/Tomcat daemon: "
    $JBOSS_HOME/bin/go.sh status
    ;;

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

Annexe B. Installation complète & configuration de Postgresql sur un Linux Debian

Le but de ce paragraphe est de fournir un guide d'installation complet de Postgresql 7.1 sur une machine Linux Debian.

Attention

Cette partie sous-entend que votre système Linux Débian a correctement été installé et que vous possédez une liaison internet.

Installation de Postgresql 7.1

L'installation va se faire au moyen de la commande apt-get qui sert à installer des packages de manière simple. Il faut cependant configurer le fichier `/etc/apt/sources.list` pour qu'il puisse accéder aux packages de postgresql

Exemple B.1. fichier `/etc/apt/sources.list`

```
# See sources.list(5) for more information, especialy
# Remember that you can only use http, ftp or file URIs
# CDROMs are managed through the apt-cdrom tool.
deb http://http.us.debian.org/debian stable main contrib non-free
deb http://non-us.debian.org/debian-non-US stable/non-US main contrib non-free
deb http://security.debian.org stable/updates main contrib non-free

#jdk1.3
deb ftp://metalab.unc.edu/pub/linux/devel/lang/java/blackdown.org/debian woody non-free

# postgresql 7.1
deb http://people.debian.org/~elphick/postgresql/pg7.1/potato/ ./
```

L'installation se fait ensuite grâce à la commande suivante:

```
bash$ apt-get update
bash$ apt-get install postgresql
```

Une fois cela terminé, un compte postgres a été créé sur la machine, les différents outils pour démarrer, créer & supprimer des bases, se connecter à postgresql ont été installés et un script de démarrage `/etc/init.d/postgresql` a été créé.

Note

Pour que Postgresql soit lancée au démarrage de Linux, il suffit d'utiliser `update-rc.d`.

```
bash$ update-rc.d postgresql defaults
```

Cette commande va créer les liens nécessaires.

Les applications suivantes font partie de postgresql:

- *postmaster*: permet de lancer postgresql.
- *createdb*: permet de créer une nouvelle base de données.

Attention

Pour pouvoir utiliser les accents en français, il faut utiliser l'option `-E UNICODE`.

- *dropdb*: permet de supprimer une base de données existantes.
- *psql*: permet de se connecter à une base de données pour exécuter des commandes sql. `\q` permet de quitter l'application, `\i` permet d'exécuter le contenu sql d'un fichier, `\dd` permet de lister tous les objets (dont les tables) de la base, `\d` permet de lister les éléments d'un objet (en particulier d'une table).

Attention

Les programmes précédemment cités ne peuvent être lancés qu'avec des utilisateurs autres que root.

Installation du driver jdbc pour Postgresql

Les drivers jdbc pour Postgresql peuvent être récupérés sur le site suivant dédié: <http://jdbc.postgresql.org/>. Etant donné que Postgresql 7.1 et un jdk 1.3 sont installés, nous utiliserons le driver jdbc suivant: `jdbc7.1-1.2.jar` [<http://jdbc.postgresql.org/download/jdbc7.1-1.2.jar>].

Aucune installation particulière n'est nécessaire, il suffit uniquement de placer le fichier jar précédent dans le class-path de l'application pour pouvoir l'utiliser.

Attention

Comme jdbc utilise le réseau et que java ne supporte pas les sockets unix, postmaster doit être lancé avec l'option `-i` (postmaster se lance sous l'utilisateur postgres).

```
bash$ postmaster -i -s
```

Des modifications sont alors à apporter dans le fichier `pg_hba.conf` pour autoriser les connexions à la base de données.

Exemple B.2. Fichier de configuration des connexions à Postgresql (`/etc/postgresql/pg_hba.conf`)

```
local          all                                peer sameuser
host           all          127.0.0.1      255.255.255.255  trust
host           all          192.168.1.1   255.255.255.0   trust
```

Fonctionnement de JGenea avec Postgresql

Pour faire fonctionner Jgenea avec Postgresql, les étapes suivantes sont à suivre:

- *classpath*: il faut ajouter dans tous les scripts de lancement de jgenea, jdbc7.1-1.2.jar au classpath de la commande java de lancement. postgresql.
- *classpath jboss*: il faut ajouter jdbc7.1-1.2.jar au classpath de jboss en le plaçant dans le répertoire lib/ext/ de cleui-ci. Le chargement des jars de ce répertoire est dynamique. postgresql.
- *jgenea ihm*: il faut configurer l'accès à la base: type (psql), driver (org.postgresql.Driver), url jdbc (jdbc:postgresql://machine:port/database), login (postgres) et mot de passe (le mot de passe de postgres).

Exemple B.3. Fichier de configuration des bases de données de JGenea Ihm (conf/properties/Conf.properties)

```
(...)  
# DB  
bases=test1,test2  
desc-bases=Base hsql,Base Postgresql  
  
# Hsql  
test1.typeJdbc=hsql  
test1.driverJdbc=org.hsqldb.jdbcDriver  
test1.urlJdbc=jdbc:hsqldb:hsql://localhost:66  
test1.loginJdbc=sa  
test1.passwordJdbc=  
  
# Hsql  
test2.typeJdbc=psql  
test2.driverJdbc=org.postgresql.Driver  
test2.urlJdbc=jdbc:postgresql://localhost:5432/jgenea  
test2.loginJdbc=postgres  
test2.passwordJdbc=postgres  
(...)
```

- *jgenea ihm*: il faut configurer l'accès à la base: type (psql), driver (org.postgresql.Driver), url jdbc (jdbc:postgresql://machine:port/database), login (postgres) et mot de passe (le mot de passe de postgres).

Exemple B.4. Fichier de configuration des bases de données à afficher pour JGenea Web (conf/properties/Db.properties)

```
# DB  
bases=Test1,Test2  
bases-desc=Base hsql1,Base postgresql
```

Exemple B.5. Fichier de configuration des bases de données dans Stuts pour JGenea Web (conf/xml/struts-config.properties)

```
(...)  
<data-sources>  
  <data-source key="Test1">  
    <set-property property="autoCommit" value="false" />
```

```
<set-property property="description" value="Hsql" />
<set-property property="driverClass" value="org.hsqldb.jdbcDriver" />
<set-property property="maxCount" value="4" />
<set-property property="minCount" value="2" />
<set-property property="password" value="" />
<set-property property="url" value="jdbc:hsqldb:hsql://localhost:66" />
<set-property property="user" value="sa" />
</data-source>

<data-source key="Test2">
  <set-property property="autoCommit" value="false" />
  <set-property property="description" value="Postgresql" />
  <set-property property="driverClass" value="org.postgresql.Driver" />
  <set-property property="maxCount" value="4" />
  <set-property property="minCount" value="2" />
  <set-property property="password" value="postgres" />
  <set-property property="url" value="jdbc:postgresql://localhost:5432/jgenea" />
  <set-property property="user" value="postgres" />
</data-source>
</data-sources>
(...)
```

Attention

Il faut pour utiliser Postgresql créer le modèle de données. Ceci est possible grâce à la console et la commande install et, attention, doit être fait après avoir créé la base jgenea avec createdb.

```
> install -repertoire REPERTOIRE_SCRIPTS -type psql -pays fr
```

Scripts d'administration de Postgresql

Le script suivant, livré avec Postgresql, permet de lancer, arrêter et redémarrer Postgresql. Il est nécessaire de le modifier pour rajouter l'option -i, nécessaire pour l'utilisation de jdbc.

Exemple B.6. Script de lancement de Postgresql (/etc/init.d/postgresql)

```
#!/bin/sh
#
# NOTE TO SYSTEM ADMINISTRATORS #
# To stop postgresql running, do
#   ln -sf /bin/false /usr/lib/postgresql/bin/can_i_run
# To re-enable it, do
#   rm /usr/lib/postgresql/bin/can_i_run

startup () {
  . /etc/postgresql/postmaster.conf
  touch ${POSTGRES_LOG:=/var/log/postgres.log}
  chown postgres.postgres $POSTGRES_LOG
  su - postgres -c /usr/lib/postgresql/bin/postgresql-startup
}

POSTMASTER=/usr/lib/postgresql/bin/postmaster

case "$1" in
  start)
    startup
    ;;
```

```
stop)
    echo Stopping PostgreSQL postmaster.
    start-stop-daemon --stop --verbose --exec ${POSTMASTER} || exit 0
    ;;
restart)
    echo Restarting PostgreSQL postmaster.
    start-stop-daemon --stop --oknodo --verbose --exec ${POSTMASTER}
    sleep 3
    startup
    ;;
force-reload)
    echo Reload not supported by PostgreSQL - restarting postmaster.
    start-stop-daemon --stop --oknodo --verbose --exec ${POSTMASTER}
    sleep 3
    startup
    ;;
reload)
    echo PostgreSQL does not support a reload option.
    exit 2
    ;;
*)
    echo "Usage: /etc/init.d/postgresql {start|stop|restart}"
    exit 1
    ;;
esac
exit 0
```

Exemple B.7. Script de lancement de Postgresql (/usr/lib/postgresql/bin/postgresql-startup)

```
#!/bin/sh
## WARNING TO MAINTAINERS - DO NOT EDIT this file in debian/. It will be
## overwritten by debian/rules. Edit the .in template instead.
set -e
#
# Initialisation
#
# (This should set up values for POSTGRES_HOME and POSTGRES_DATA, which
# say where the library and database are. It should also set up DATEFORMAT
# which governs whether the backend sends back dates in European or
# American format, and the logging options.)
#
# This is NOT a configuration file.
# If you wish to change any values, the proper place to do so is in
# /etc/postgresql/postmaster.init. ANY CHANGES YOU MAKE TO THIS FILE
# WILL GET OVERWRITTEN BY THE NEXT INSTALLATION.
# To stop postgresql running, do
# ln -sf /bin/false /usr/lib/postgresql/bin/can_i_run

check_version () {
    # compare the database format with the format expected by the software
    dbformat=`cat ${PGDATA}/PG_VERSION 2>/dev/null`
    if [ A${dbformat} != A${version} ]
    then
        if [ -z "${dbformat}" ]
        then
            echo The database framework has not yet been created. Use
            echo initdb to do this.
        else
            echo The database is in an older format that cannot be read by
            echo version ${version} of PostgreSQL.
            echo
            if [ -f /usr/lib/postgresql/dumpall/default_encoding ]
```

```
        then
            echo "The postinstallation script should attempt to
                    upgrade the database"
            echo "automatically.  If it fails, it must be done by hand."
        else
            echo Run postgresql-dump to dump the old database and to reload
            echo it in the new format.
        fi
        echo "*** READ /usr/share/doc/postgresql/README.Debian.migration.gz
                    FIRST! ***"
    fi
    echo
    echo The version ${version} postmaster cannot be started until
    echo this is done.
    exit 255
fi
}

obsolete_config () {
    echo "
    ***** Obsolete configuration files *****

    postmaster.init and pg_options in /etc/postgresql/ are obsolete.

    Please update your configuration files by integrating your site-
    specific changes with the new format in postmaster.conf and
    postgresql.conf and delete or rename the old configuration files.

    PostgreSQL cannot be started until you have done this.

    *****
    "
    exit 1
}

# Refuse to proceed if the old config files are still there
if [ -f /etc/postgresql/postmaster.init -o -f /etc/postgresql/pg_options ]
then
    obsolete_config
    exit 1
fi

. /etc/postgresql/postmaster.conf

export LANG PGDATESTYLE

if [ -z "${PGDATESTYLE}" ]
then
    PGDATESTYLE=ISO,European
fi

PGDATA=${POSTGRES_DATA:-/var/lib/postgres/data}
PGLIB=/usr/lib/postgresql
export PGLIB PGDATA
POSTMASTER=${PGLIB}/bin/postmaster
version=7.1      # Note: this is not necessarily the same as the
                 # software version.

# Give administrators a means of not allowing postgresql to start:
# Link /usr/lib/postgresql/bin/can_i_run to /bin/false
if [ -x /usr/lib/postgresql/bin/can_i_run -a "$1" != dump_schema ]
then
    if ! /usr/lib/postgresql/bin/can_i_run
    then
        echo PostgreSQL is temporarily disabled
        echo
        echo To enable it, remove /usr/lib/postgresql/bin/can_i_run \
        echo link it to /bin/true\ ) and run \ ` /etc/init.d/postgresql start\ `
        echo
        exit 1
    fi
fi
```

Installation complète & configuration de Postgresql sur un Linux Debian

```
# Make sure we have a database directory
if [ ! -d ${PGDATA} ]
then
    echo No readable database directory for postgresql
    exit 3
fi

if [ ! -d ${PGDATA}/base ]
then
    echo There is no PostgreSQL database framework in $PGDATA.
    echo Run initdb as the postgres user to create it
    exit 3
fi

# First, check that the database is the right version
check_version

# Make sure that we don't try to start if the executable is missing.
if [ ! -x ${POSTMASTER} ]
then
    echo No postmaster executable for postgresql
    exit 3
fi

if [ -n "$PGPORT" ]
then
    OPTIONS="-o '-p $PGPORT'"
fi

# Make sure there is no UNIX-socket file lurking around; this socket
# can get left behind if there is a system crash.
if [ -S /var/run/postgresql/.s.PGSQL.${PGPORT:=5432} ]
then
    rm /var/run/postgresql/.s.PGSQL.${PGPORT}
fi

# Ready to go: stand clear...
echo Starting PostgreSQL postmaster.
cd ${POSTGRES_HOME}
echo /usr/lib/postgresql/bin/pg_ctl start -D ${PGDATA} -i
    -l ${POSTGRES_LOG:=/var/log/postgres.log} ${OPTIONS}
/usr/lib/postgresql/bin/pg_ctl start -D ${PGDATA} -i -l ${POSTGRES_LOG} ${OPTIONS}

# If possible, increase the size of the kernel file table
fmaxfile=/proc/sys/kernel/file-max
if [ ! -r $fmaxfile ]
then
    fmaxfile=/proc/sys/kernel/fs/file-max
    if [ ! -r $fmaxfile ]
    then
        fmaxfile=
    fi
fi

if [ -n "$fmaxfile" ]
then
    fmax=`cat $fmaxfile`
    if [ $fmax -lt ${KERNEL_FILE_MAX:=1032} ]
    then
        echo ${KERNEL_FILE_MAX} > $fmaxfile
    fi
fi

# Give it a chance to get going
sleep 5

exit 0
```