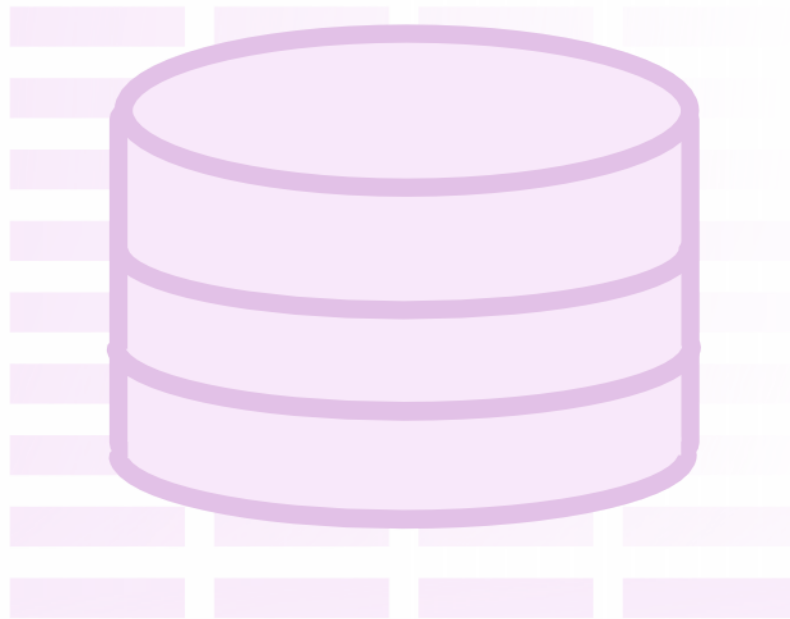


Jean-François Nifenecker

# LIBREOFFICE BASE ET HSQLDB

POUR EN FINIR AVEC LE MODE INTÉGRÉ



*version 0.10a du 14 juin 2014*



## Table des matières

1. Présentation .....	5
1.1. À qui est destiné ce manuel ? .....	5
1.2. LibreOffice / OpenOffice.org / ApacheOpenOffice ? .....	5
1.3. Java .....	6
2. Cohabitation HSQLDB / LibreOffice .....	7
2.1. Que contient une base de données ? .....	7
2.2. LibreOffice Base : trois modes d'exploitation .....	7
2.3. Contenu d'une base HSQLDB accédée à travers LibreOffice Base .....	9
2.4. Quelle configuration choisir ? .....	11
3. Installer HSQLDB .....	13
3.1. Télécharger HSQLDB .....	13
3.2. « Installer » HSQLDB .....	13
3.3. Fonctionnement du moteur HSQLDB .....	14
4. Utiliser Base en mode fichier autonome (mono-poste) .....	15
4.1. Configurer LibreOffice .....	15
4.2. Créer une base externe .....	17
4.3. Problèmes connus .....	21
5. Utiliser Base en mode client-serveur (multi-utilisateurs) .....	23
5.1. Opérations préalables .....	23
5.2. Configurer le serveur .....	23
5.3. Configurer les clients .....	30
6. Transformer une base incorporée en base autonome .....	31
6.1. Préalables .....	31
6.2. Étapes de la migration .....	31
7. Sauvegarde et restauration de bases HSQLDB utilisées avec Base .....	35
7.1. Sauvegarde .....	35
7.2. Restauration .....	36
8. Annexes .....	37
8.1. Divorcer d'avec MS-Access .....	37
8.2. Les limites de Base .....	37
8.3. Glossaire .....	38
9. Auteur et licence .....	39
9.1. Auteur .....	39
9.2. Remerciements .....	39
9.3. Historique des modifications .....	39
9.4. Licence .....	39

# 1. Présentation

*En termes de bases de données, LibreOffice est livré d'emblée avec un moteur HSQLDB intégré, ce qui permet d'exploiter immédiatement le module Base pour créer de petites bases de données personnelles. Cependant, le module Base permet beaucoup plus que cela. Ce module constitue en effet ce que l'on nomme un « frontal » de bases de données (front-end), ce qui lui permet d'offrir l'accès à un grand nombre de types de bases de données. Les moteurs les plus connus sont incontestablement MySQL et PostgreSQL. Cependant, il existe également la possibilité d'accéder à travers Base à des bases de données HSQLDB externes, qu'elles soient mono-poste ou client-serveur.*

Déconnecter Base et le moteur de base de données – nous parlerons ici uniquement de HSQLDB mais le discours peut être extrapolé de manière symétrique aux moteurs externes tels que PostgreSQL ou MySQL/MariaDB – offre de nombreux avantages : la sécurité d'abord mais aussi permet des mises à jour du fichier .odb indépendamment de la base elle-même.

## 1.1. À qui est destiné ce manuel ?

Ce manuel s'adresse à tous ceux qui désirent aller « plus loin » avec la gestion des bases de données tout en continuant d'y accéder au moyen des outils bureautiques de la suite LibreOffice. Il intéressera aussi tous ceux qui sont conscients des problèmes de sécurité que pose l'usage des bases de données en mode incorporé tel qu'il est proposé par défaut sous LibreOffice.

Si nous y avons inséré quelques informations « techniques », celles-ci restent à la portée de tous pour peu que nous ayons un minimum d'appétence pour l'outil informatique.

La présentation qui est faite ici est centrée sur l'emploi du moteur de bases de données HSQLDB. Il va de soi que la connexion de LibreOffice à d'autres moteurs externes utilisant des pilotes spécifiques présente de nombreuses proximités (PostgreSQL, MySQL, MariaDB, Firebird, etc.). En d'autres termes, ce manuel aidera le lecteur à comprendre le « grand dessein » de l'usage de la suite LibreOffice conjointement avec des bases externes.

## 1.2. LibreOffice / OpenOffice.org / ApacheOpenOffice ?

Ce document a été rédigé en s'appuyant sur LibreOffice v.3.3.4 sous Windows XP sp3 ; il a ensuite été complété en utilisant LibreOffice v.4.1.4. Compte tenu de la proche parenté des trois suites LibreOffice (LibO), OpenOffice.org (OOo) et ApacheOpenOffice (AOO), il est très probable que ces explications soient transposables entre elles.

Chaque fois que ce sera utile, nous préciserons les différents cas pouvant se présenter en fonction de la suite bureautique utilisée et du système d'exploitation gouvernant la machine.



### 1.3. Java

Un environnement d'exécution *Java* est nécessaire au fonctionnement de *HSQLDB*.

Si aucun environnement Java n'existe sur la machine, nous devons en installer un. LibO/OOo/AOO supportent également Java. L'environnement conseillé est *Oracle JRE*, à télécharger sur le site d'Oracle (voir le chap. Références). Versions conseillées pour un fonctionnement correct de LibO : JRE v.1.7.0.xx.

# 2. Cohabitation HSQLDB / LibreOffice

*HyperSQL (HSQLDB) est un moteur de bases de données rapide et performant écrit en Java. De fait, ce moteur peut s'intégrer à toute plate-forme (Windows, MacOS, GNU/Linux, etc.) dès lors que le support de Java est assuré.*

---

## **La voie vers d'autres SGBDR**

*La démarche décrite dans ce document permet aussi d'envisager l'utilisation de Base comme frontal pour d'autres SGBDR tels que MySQL ou PostgreSQL. LibreOffice propose des connecteurs spécifiques pour accéder à ces moteurs de bases de données.*

---

### 2.1. Que contient une base de données ?

Une base de données comporte typiquement :

- des tables, leur définition, leur contenu, les contraintes éventuellement posées dessus, c'est-à-dire les données proprement dites ;
- des éléments d'interrogation (requêtes, vues) ;
- des outils d'interface pour l'affichage ou l'impression (formulaires, rapports) ;
- des paramètres d'accès qui permettent aux différents logiciels de dialoguer avec le gestionnaire de bases de données.
- Ces différents éléments seront gérés différemment selon la manière dont nous accéderons à la base.

### 2.2. LibreOffice Base : trois modes d'exploitation

En bref, *Base* permet la gestion de bases de données selon trois modes distincts : le mode incorporé, le mode fichier autonome, le mode client-serveur.

#### **a. Le mode incorporé (embedded mode)**

C'est le mode proposé par défaut. *LibreOffice* gère la base de données à travers un moteur *HSQLDB* et un pilote internes à la suite bureautique<sup>1</sup>. Le corollaire est que toutes les données sont gérées en mémoire.

Le fichier `.odb` contient les données (les tables), les éléments d'interrogation (requêtes et vues) et les éléments d'interface (formulaires et rapports) (figure 1).

---

<sup>1</sup> Ce moteur est présent sous la forme d'un fichier `hsqldb.jar` dans le sous-répertoire `Basis\program\classes` de l'installation de LibreOffice.

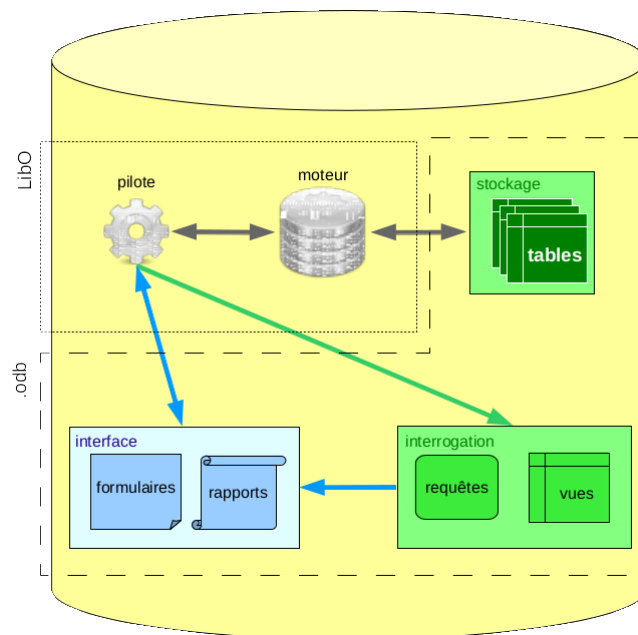


Figure 1 : LibreOffice Base et HSQLDB en mode incorporé

Ce mode de fonctionnement a l'avantage d'être simple d'utilisation (un seul contenant entièrement autonome) mais comporte les inconvénients (1) d'être restreint à une taille compatible avec la mémoire vive disponible et (2) surtout et par conséquent, d'être sujet à des pertes de données – voire de la base toute entière – en cas de problèmes sur le poste pendant l'exécution.

Si la base peut être mise en partage, elle n'est accessible avec les droits complets que par *un seul* poste à la fois. Il s'agit bien d'un usage mono-poste.

### b. Le mode fichier autonome (file mode)

Un moteur *HSQLDB* externe est installé sur le poste et gère les bases de données indépendamment des processus d'interrogation ou d'exploitation. Le fichier `.odb` ne contient que les éléments d'interrogation et d'interface (voir la figure 2).

L'installation demande quelques manipulations complémentaires mais, à l'usage, la base est beaucoup plus robuste et mieux protégée des aléas pouvant survenir sur la machine hôte.

Si la base peut être mise en partage, elle n'est accessible avec les droits complets que par *un seul* poste à la fois. Nous l'utilisons toujours en mono-poste.

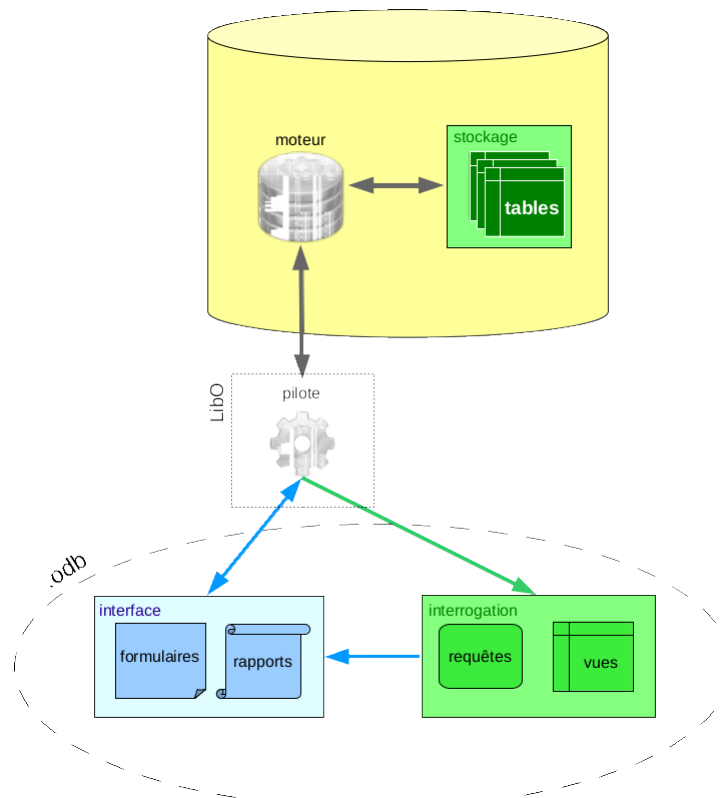


Figure 2 : LibreOffice Base et HSQLDB en mode autonome ou client-serveur

### c. Le mode client-serveur (server mode)

Ce mode est très ressemblant au mode autonome – un moteur *HSQLDB* externe gère les bases de données – mais la base peut être utilisée concurremment par plusieurs postes clients.

#### Sauvegarde

Les données stockées dans nos bases de données deviennent rapidement volumineuses et sont souvent de grande importance dans l'activité de leurs utilisateurs. Dans tous les cas, la sauvegarde régulière des bases de données est une nécessité (le mieux est de l'automatiser).

Les questions relatives à la sauvegarde et à la restauration des bases *HSQLDB* accédées à travers des composants *LibreOffice Base* sont présentées au chapitre 7.

## 2.3. Contenu d'une base HSQLDB accédée à travers LibreOffice Base

Comme nous l'avons vu précédemment, une base *HSQLDB* gérée à travers *LibreOffice Base*, en mode autonome ou client-serveur, est constituée de trois ensembles (voir la figure 2) : la base, les outils d'interrogation et d'interface, les paramètres d'accès au moteur *HSQLDB*.

### a. La base

Ces fichiers sont stockés dans un répertoire dédié à la base de données (voir le chapitre 4.2). Il s'agit d'une série de fichiers dont le nom est celui de la base et portant des extensions variables en fonction de leur usage.



### b. Les outils d'interrogation et d'interface

Ces outils d'interrogation et d'interface (requêtes, formulaires, rapports, etc.) sont définis dans le module *Base*. Ils sont enregistrés dans le fichier `.odbc` qui est créé lors du premier accès à la base de données depuis le module *Base* (ch. 4.2).

Lors de sa création, il est possible d'enregistrer ce fichier « n'importe où ». Cependant, pour des raisons pratiques, il est conseillé d'enregistrer ce fichier dans le même répertoire que la base elle-même.

### c. Les paramètres

Trois familles de paramètres sont importantes pour notre usage : les paramètres d'enregistrement de la base, les paramètres d'accès au moteur de gestion *HSQLDB* et les paramètres d'accès à la base de données.

#### *Paramètres d'enregistrement de la base dans LibreOffice*

L'enregistrement permet à tous les modules de la suite bureautique de connaître l'existence de la base de données et d'y accéder.

L'enregistrement de la base de données dans *LibreOffice* est un paramètre du fichier `registrymodifications.xcu`. Ce fichier se trouve à la racine du profil de l'utilisateur *LibreOffice*. L'information proprement dite est une ligne `DataAccess` qui ressemble à ceci :

```
<item oor:path="/org.openoffice.Office.DataAccess/RegisteredNames"><node
oor:name="org.openoffice.basetest" oor:op="replace"><prop oor:name="Location"
oor:op="fuse"><value>file:///D:/mabase/basetest.odbc</value></prop><prop
oor:name="Name" oor:op="fuse"><value>basetest</value></prop></node></item>
```

#### *Paramètres d'accès au moteur HSQLDB*

Ce sont les paramètres de l'accès au moteur *HSQLDB* externe depuis *LibreOffice* (mode d'accès, configuration du pilote, etc.) (voir les chapitres 4 et 5).

Ces paramètres sont enregistrés dans le fichier de configuration de *LibreOffice* `\config\javasettings_windows_x86.xml`<sup>2</sup>. Ce fichier est stocké dans le profil *LibreOffice* de l'utilisateur dont l'emplacement est variable selon le système sous lequel nous travaillons<sup>3</sup>. La gestion de cette configuration est l'objet du chap. 4.1.

Exemple de contenu d'un fichier `javasettings` (nous avons surligné l'emplacement des paramètres en question) :

```
<?xml version="1.0" encoding="UTF-8"?>
<!--This is a generated file. Do not alter this file!-->
<java xmlns="http://openoffice.org/2004/java/framework/1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<enabled xsi:nil="true"/>
<userClassPath xsi:nil="false">C:\Program Files\hsqldb-
2.2.9\lib\hsqldb.jar;C:\Program Files\hsqldb-2.2.9\lib</userClassPath>
<vmParameters xsi:nil="true"/>
<jreLocations xsi:nil="true"/>
<javaInfo xsi:nil="false" vendorUpdate="2013-05-02" autoSelect="false">
<vendor>Oracle Corporation</vendor>
<location>file:///C:/Program%20Files/Java/jre7</location>
<version>1.7.0_45</version>
```

<sup>2</sup> Sous *GNU/Linux*, le fichier se nomme `javasettings_Linux_X86_64.xml`.

<sup>3</sup> Pour plus d'informations sur le profil de l'utilisateur *LibreOffice*, nous nous reporterons au document qui lui est consacré.





```
<features>0</features>
<requirements>0</requirements>
<vendorData>660069006C0065003A002F002F002F0043003A002F00500072006F006700720061006
D00250032003000460069006C00650073002F004A006100760061002F006A007200650037002F0062
0069006E002F0063006C00690065006E0074002F006A0076006D002E0064006C006C00</vendorDat
a>
</javaInfo>
</java>
```

### Paramètres d'accès à la base de données

Ces paramètres, enfin, sont ceux qui permettent au module base d'accéder à la base de données elle-même. Ils sont enregistrés dans le fichier .odb qui est créé au moment de la construction de la base (chap. 4.2).

## 2.4. Quelle configuration choisir ?

En pratique, quelle option est conseillée en fonction de quel usage ?

Usage	Utilisation conseillée	Commentaires
Essais, maquettes, <b>petites</b> bases de données personnelles	Base + HSQLDB incorporé (mode par défaut)	Sauvegarde très régulière du fichier .odb
Bases de données personnelles évoluées, bases de données professionnelles à utilisateur unique	Base + HSQLDB externe (chap. 4)	Gestion des sauvegardes indispensable (chap 7)
Bases de données professionnelles à utilisateurs multiples	Base + HSQLDB client-serveur (chap. 5)	

# 3. Installer HSQLDB

*Pour un fonctionnement autonome – en présence ou en dehors de toute installation d'une suite bureautique –, nous allons devoir installer la dernière version de HSQLDB.*

## 3.1. Télécharger HSQLDB

*HSQLDB est disponible sur le site dédié (chap. Références), sous la forme d'un fichier compressé (.zip). Au moment où ces lignes sont écrites, c'est la version 2.2.9 qui est la plus récente (certaines copies d'écran ont été réalisées avec la version précédente mais le contexte est inchangé).*

## 3.2. « Installer » HSQLDB

À réaliser sous un compte d'administration.

Il n'y a à proprement parler pas d'« installation » à effectuer : il suffit de décompresser le fichier .zip téléchargé dans le répertoire de notre choix et c'est fait (figure 3).

Par mesure de cohérence avec le système hôte, ici nous avons choisi de décompresser les fichiers *HSQLDB* sous Program Files (*WinXP*).

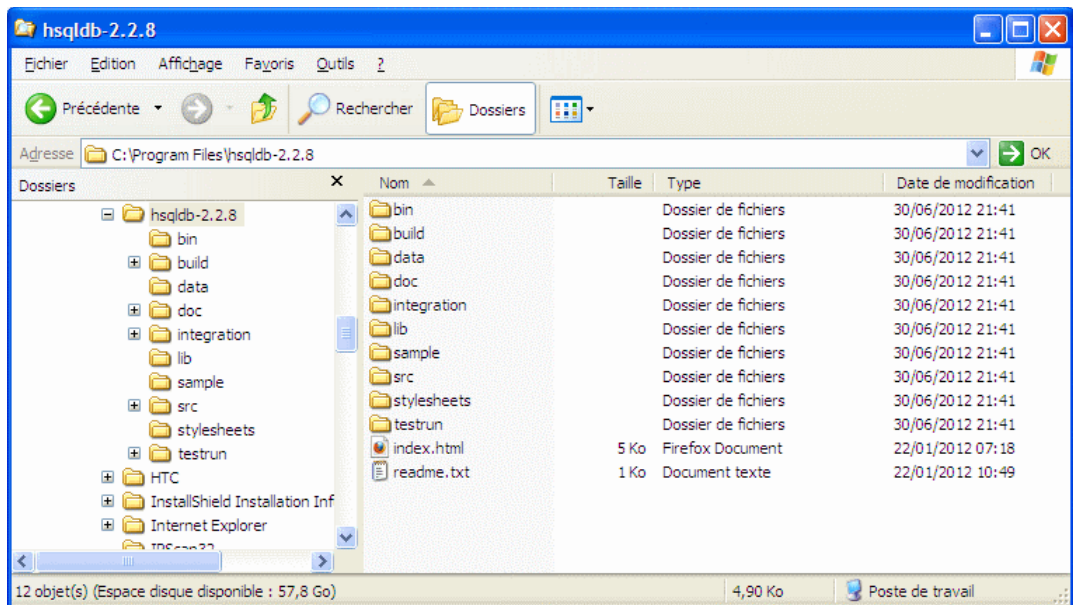


Figure 3 : Le répertoire d'installation de HSQLDB

Pour plus d'informations sur *HSQLDB*, nous nous reportons à la documentation disponible sur le site (voir chap. Références).

Il nous reste à configurer *LibreOffice* afin d'avoir accès aux bases que pourra gérer ce moteur *HSQLDB* externe. Ce paramétrage est présenté aux chapitres qui suivent.



### 3.3. Fonctionnement du moteur HSQLDB

Le moteur *HSQLDB* s'exécute dans une machine virtuelle java (JVM) et écoute les connexions venant du même ordinateur ou des autres machines présentes dans le réseau local. Plusieurs programmes différents peuvent se connecter au serveur pour y lire ou écrire des informations. Les applications clientes – c'est le cas de *LibreOffice Base* – se connectent en utilisant le pilote *HSQLDB JDBC* (*java database connectivity*).

# 4. Utiliser Base en mode fichier autonome (mono-poste)

Nous allons maintenant utiliser le module LibreOffice Base afin de réaliser une base de données non incorporée (dite « autonome » ou « externe »).

---

## Où sont les tables ?

En mode autonome, contrairement au mode incorporé, le fichier `.odt` généré par Base ne comportera **pas** les tables de données (voir la figure 1, page 8). Celles-ci sont gérées par le moteur `HSQLDB` externe que nous venons d'installer et non par le moteur incorporé à LibO. Le processus étant extérieur à la suite bureautique, les données seront bien mieux protégées si des problèmes surviennent.

---

Nous allons voir comment configurer LibO d'abord, puis comment initialiser l'accès aux bases autonomes ensuite. Par défaut, ces accès se font en mono-poste : un seul utilisateur (celui du poste) peut accéder aux données. Il est possible de partager les données mais un seul utilisateur aura plein accès aux données (le premier connecté). Pour un véritable usage multi-utilisateurs, nous nous reporterons au chapitre 5 Utiliser Base en mode client-serveur (multi-utilisateurs).

## 4.1. Configurer LibreOffice

Pour exploiter sous Base le moteur `HSQLDB` externe installé au chapitre précédent, nous devons indiquer les paramètres utiles à LibO pour la connexion au moteur.

---

### Note

La manipulation peut s'effectuer sous tout compte d'utilisateur et s'applique quel que soit le système.

Les paramètres définis ici sont mémorisés dans un fichier `javasettings` sous le profil de l'utilisateur (voir le chap. 2.3.c, page 10).

---

Ce paramétrage se réalise dans LibreOffice via le menu **Outils > Options**, branche **LibreOffice > Avancé**<sup>4</sup> (figure 4).

---

<sup>4</sup> Dans les versions antérieures à la version 4 de LibreOffice, c'est dans **LibreOffice > Java** que ce paramétrage est réalisé. Les options sont les mêmes que celles qui sont indiquées ici.

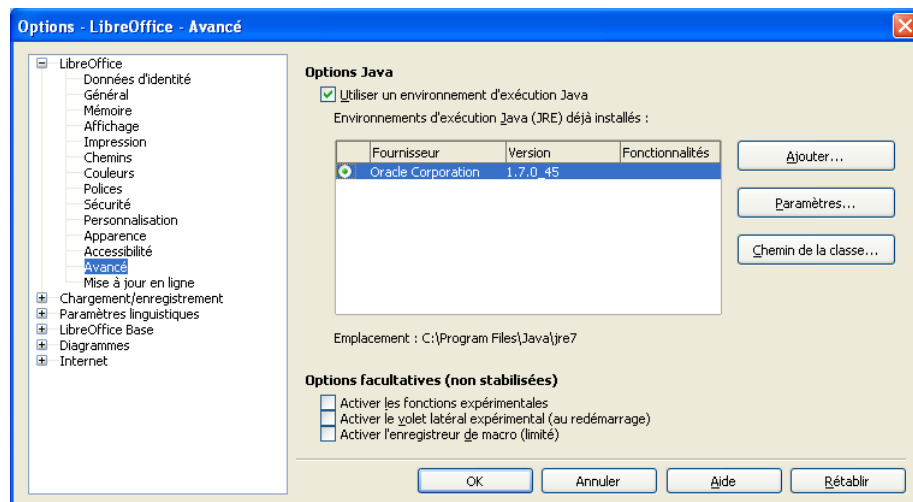


Figure 4 : Option de configuration Java

Nous vérifions que la case **Utiliser un environnement d'exécution Java** est bien cochée.

Si ce n'est fait, nous sélectionnons l'environnement Java à utiliser. Au besoin, le bouton **Ajouter** nous permet de pointer vers l'environnement *Java* installé sur la machine.

Puis nous cliquons le bouton **Chemin de la classe** ce qui nous amène à une nouvelle fenêtre (figure 5).

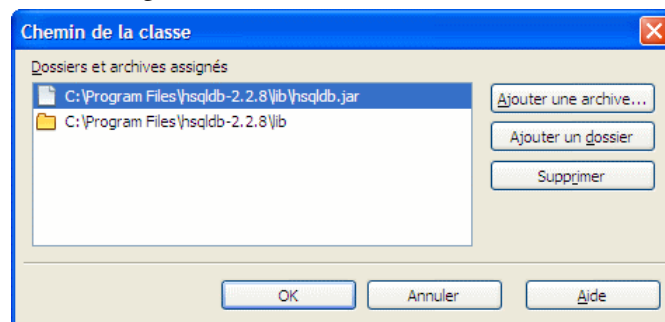


Figure 5 : Paramètres de la classe Java

Nous appelons successivement :

– **Ajouter une archive**

Pointons vers le fichier `hsqldb.jar` installé précédemment. Dans notre exemple, nous avons choisi de décompresser le fichier `.zip` dans `C:\Program Files\hsqldb-2.2.9`. Le fichier `hsqldb.jar` se trouve dans le sous-répertoire `\lib` ;

– **Ajouter un dossier**

Pointons vers le sous-répertoire `\lib` de l'installation ci-dessus.

Le résultat final est présenté à la figure 5.

Nous validons et refermons *LibreOffice* pour valider le nouveau paramétrage.

Désormais *LibreOffice* connaît le moteur *HSQLDB* externe. Il saura s'y connecter lors de l'exploitation de nos bases.



### Où sont mémorisées ces informations ?

Les paramètres définis ci-dessus sont enregistrés dans le fichier `javasettings` du profil utilisateur LibreOffice <sup>5</sup>.

## 4.2. Créer une base externe

Vérifions que LibreOffice sait maintenant gérer des bases HSQLDB externes. Pour ce faire, nous lançons le module *Base*. L'**Assistant Bases de données** s'exécute immédiatement. Nous allons en suivre les quatre étapes afin de configurer l'accès à notre nouvelle base externe.

Pour notre exemple, notre base de données est nommée `mabase` et est installée dans `D:\dgdidi\MesDocuments\databases\mabase`.

### Étape 1 Définir le mode d'accès au moteur de base de données

Nous choisissons **Se connecter à une base de données existante** au moyen du connecteur JDBC (*Java database connectivity*) (figure 6).

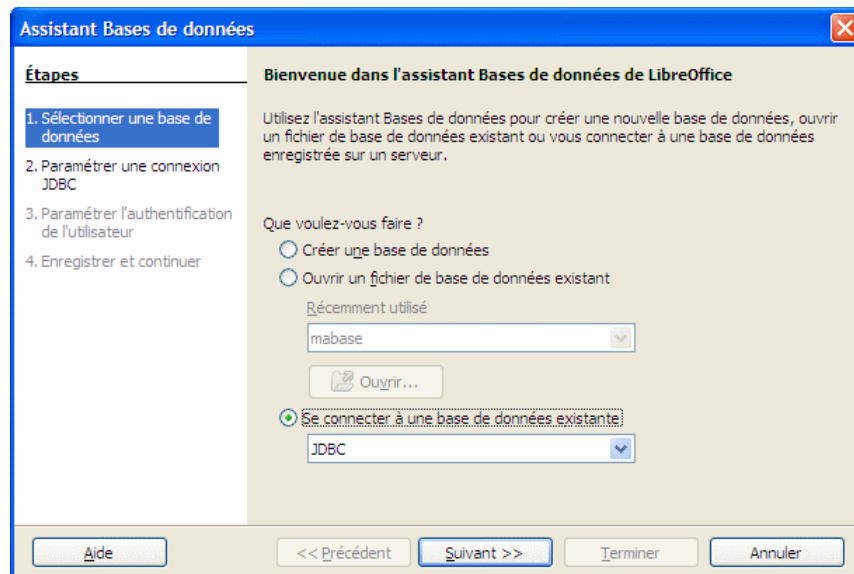


Figure 6 : Assistant Bases de données – Choix du connecteur JDBC

Continuons par **Suivant**.

### Étape 2 Définir l'emplacement de la base

Maintenant, nous devons expliciter l'emplacement de la base (URL) et le pilote JDBC à utiliser (figure 7).

<sup>5</sup> Sous Windows, ce fichier se nomme `javasettings_Windows_x86.xml` ; sous GNU/Linux, le fichier se nomme `javasettings_Linux_X86_64.xml`.

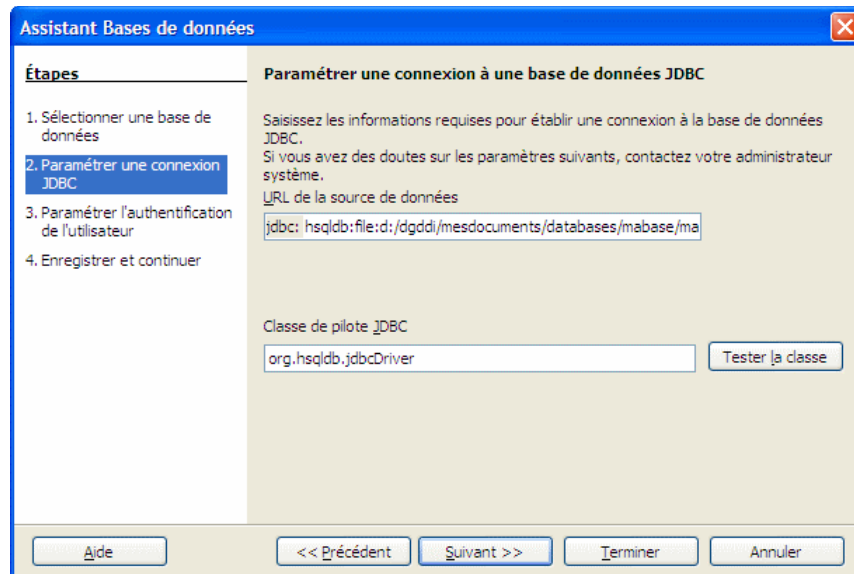


Figure 7 : Assistant Bases de données – Connexion à la base HSQLDB

Dans la rubrique **URL**, nous entrons :

```
hsqldb:file:d:/dgddi/mesdocuments/databases/mabase/basetest;default_schema=true;shutdown=true;hsqldb.default_table_type=cached;get_column_name=false
```

où

- d:/dgddi/mesdocuments/databases/mabase/ désigne **l'emplacement** de la base de données (le **répertoire** qui la contient) ;
- basetest est le **nom** de la base de données ;

Les paramètres qui suivent déterminent le mode de fonctionnement du moteur *HSQLDB* pour cette base :

- default\_schema=true  
Nécessaire pour une connexion depuis *LibO*.
- shutdown=true  
Referme la base de données lorsque la dernière connexion est elle-même fermée. Ce paramètre n'a d'utilité qu'en mode mono-poste.
- hsqldb.default\_table\_type=cached  
Détermine que les tables sont par défaut mises en cache.
- get\_column\_name=false  
(utilisée par mesure de compatibilité avec des versions plus anciennes de *HSQLDB*).

Plus d'informations sur ces paramètres dans la documentation disponible sur le site HSQL (voir chap. Références).

Dans la rubrique **Classe de pilote JDBC**, nous saisissons le nom du pilote à utiliser pour la connexion à la base, soit : `org.hsqldb.jdbcDriver`.

Le bouton **Tester la classe** permet de vérifier la bonne connexion au pilote *JDBC* selon la configuration ci-dessus (figure 8).

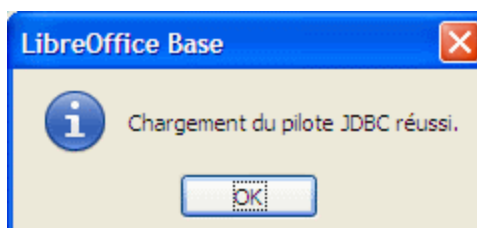


Figure 8 : Test de l'accès au pilote JDBC

Nous continuons par **Suivant**.

### Étape 3 Paramétrer l'authentification

Ici, nous indiquons les paramètres d'authentification pour l'accès à la base via le moteur *HSQLDB* (figure 9).

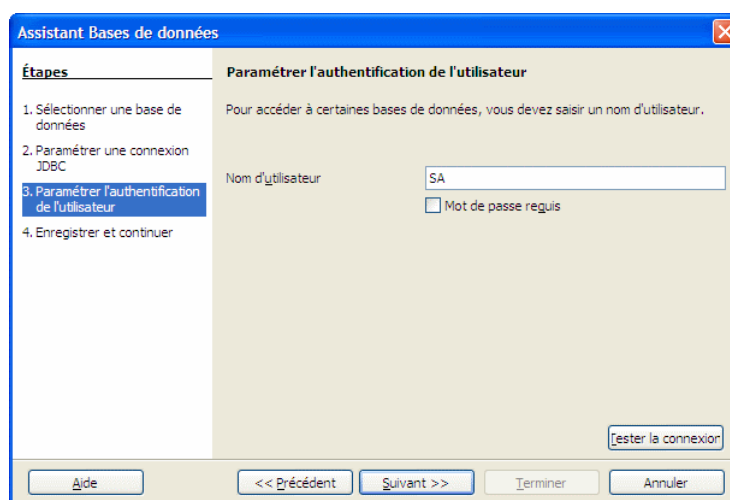


Figure 9 : Assistant Bases de données – Paramètres d'authentification

Le nom d'utilisateur d'une base *HSQLDB* par défaut est *SA* (il est également possible de laisser la rubrique vide), sans mot de passe. Le moteur *HSQLDB* permet de définir des protections par nom d'utilisateur et mot de passe. Ceci sort du cadre de ce document. La documentation *HSQLDB* donne plus de renseignements à ce propos.

Le bouton **Tester la connexion** permet de vérifier la bonne connexion au moteur externe *HSQLDB* (figure 10).

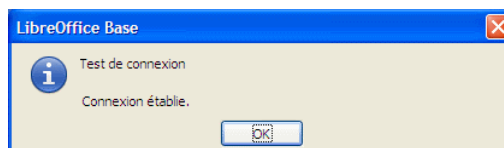


Figure 10 : Résultat du test de la connexion au moteur HSQLDB

Nous passons à la dernière étape par **Suivant**.

### Étape 4 Référencer la base

Enfin, nous devons choisir si nous désirons **référencer** cette base de données dans *LibO*. Ce choix est conseillé afin d'en permettre l'accès ultérieur direct depuis tous les modules de *LibreOffice*, pour du publipostage par exemple.



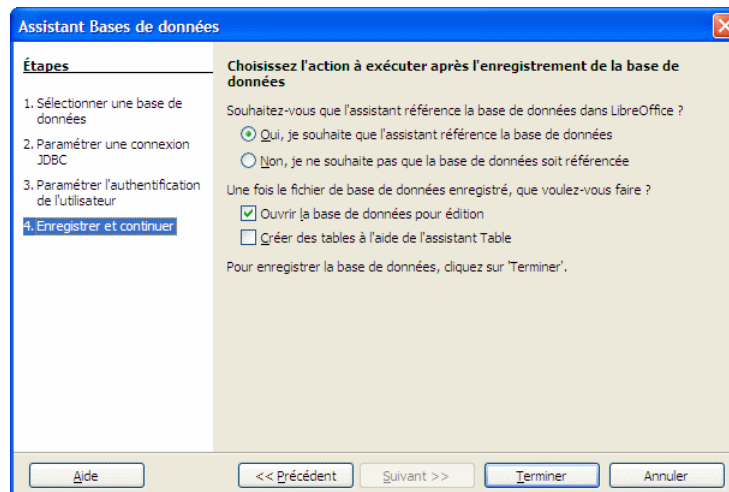


Figure 11 : Assistant Bases de données – Enregistrement de la base

En cliquant **Terminer**, *LibO* demande le nom et l'emplacement pour le fichier .odf de configuration d'accès à la base. *A priori*, n'importe quel nom est valide, cependant il peut être intéressant de nommer le fichier .odf en cohérence avec le nom de la base de données.

Nous conseillons également d'enregistrer ce fichier .odf dans le répertoire de la base elle-même. Nous optimiserons ainsi le système de sauvegarde qui devra être mis en place ensuite, l'ensemble des fichiers constituant la base de données étant regroupés dans ce répertoire unique.

### Sauvegarde ?

Lorsque l'on évoque la gestion de bases de données, encore plus que pour toute autre activité informatique, la sauvegarde est un refrain qui revient toujours. Et pour cause.

La sauvegarde et sa petite sœur la restauration sont présentées au chapitre 7.

Lorsque le module *Base* se lance, nous constatons que sa barre d'état montre les paramètres que nous venons de définir. À gauche, le type de connexion (typiquement JDBC), au centre les paramètres de fonctionnement pour l'accès à la base (voir ci-dessus), à droite le nom de l'utilisateur (SA) (figure 12).



Figure 12 : La barre d'état du module Base après configuration

Ces paramètres, initialisés aux étapes ci-dessus, sont modifiables *a posteriori*, par le menu **Édition > Base de données > Propriétés** ou **Connexion**.

Regardons maintenant le contenu du répertoire de notre base de données (figure 13).

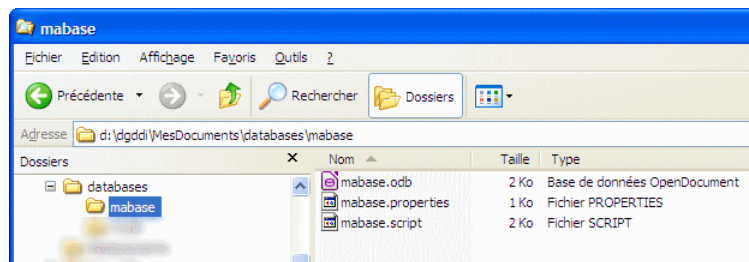


Figure 13 : Le contenu du répertoire de stockage de notre base

Nous y trouvons trois fichiers :

- mabase.odt

Ce fichier appartient à *LibO* et décrit les paramètres d'accès à la base pour la suite bureautique. Ce fichier contiendra également par la suite les divers formulaires, rapports, requêtes et vues que nous serons amenés à réaliser pour gérer la base de données ;

- mabase.properties

- et mabase.script

Ces deux fichiers texte ont été générés par le moteur externe *HSQLDB* sur l'ordre de *Base*. Ils contiennent les paramètres de gestion pour le moteur. *mabase.script* décrit la structure de la base. Pour l'instant son contenu est réduit mais les tables et les relations à venir y seront par la suite détaillés.

Nous pouvons maintenant ajouter des données dans la nouvelle base ainsi créée.

Sauvegardons !

---

*Une base de données est destinée à s'enrichir. Elle constitue un patrimoine précieux et sa perte peut s'avérer lourde de conséquences. C'est pourquoi ici, plus que jamais, la sauvegarde des fichiers est impérative (voir le chapitre. 7).*

---

### 4.3. Problèmes connus

Depuis OOo v.3.3, la création des tables de bases de données *HSQLDB* externes dans *Base* présente deux défauts : les champs autovaleur et les valeurs par défaut des champs numériques ne sont pas enregistrés.

Il y a néanmoins des solutions. Celles-ci font appel à des commandes SQL :

#### a. Champs Autovaleur

Dans *Base*, appelons le menu **Outils > SQL**. Dans la fenêtre qui s'ouvre, nous saisissons une commande SQL comme indiqué ci-dessous.

#### ***Ajout d'une fonction Autovaleur à une colonne clef primaire existante***

```
ALTER TABLE "Nom_Table" ALTER COLUMN "Nom_Colonne" INTEGER
IDENTITY
```

Exemple : ALTER TABLE "T\_Clients" ALTER COLUMN "ID" INTEGER
IDENTITY

#### ***Erreur ?***

*Si la colonne n'existe pas ou n'est pas clef primaire, nous recevons un message d'erreur.*

---



### **Ajout d'une colonne clef primaire avec fonction Autovaleur à une table**

```
ALTER TABLE "Nom_Table" ADD COLUMN "Nom_Colonne" INTEGER  
GENERATED BY DEFAULT AS IDENTITY (START WITH 1)
```

Exemple : ALTER TABLE "T\_Clients" ADD COLUMN "ID" INTEGER  
GENERATED BY DEFAULT AS IDENTITY (START WITH 1)

Nous veillerons à respecter les guillemets.

Notons qu'il est fréquent de nommer ID (« identifiant ») les colonnes clefs primaires non significatives.

Ces commandes mettent la table à jour mais cette mise à jour n'est reflétée dans Base qu'après avoir choisi **Affichage > Actualiser les tables**.

### **b. Valeurs par défaut des champs numériques**

Appelons le menu **Outils > SQL**.

```
ALTER TABLE "Nom_Table" ALTER COLUMN "Nom_Colonne" SET DEFAULT  
'Valeur'
```

Exemple : ALTER TABLE "T\_Clients" ALTER COLUMN "Age" SET DEFAULT  
'20'

Nous veillerons à respecter les guillemets, simples ou doubles.

Cette commande met la table à jour mais cette mise à jour n'est reflétée dans Base qu'après avoir choisi **Affichage > Actualiser les tables**.

#### **Un autre outil pour gérer les bases HSQLDB**

Notons au passage qu'il est possible de gérer la **structure** des bases de données *HSQLDB* externes au moyen d'autres outils, par exemple *SQuirreL* (voir chap. Références). Cette manière de procéder est même conseillée : à chacun selon ses usages. Le module *Base* pour les formulaires, les rapports, les requêtes ou les vues ; *SQuirreL* (ou autre) pour la gestion de la base de données (création de tables ou de champs, modification des structures, indexation, contraintes, etc.).

Cette manière de travailler est très fortement conseillée.

# 5. Utiliser Base en mode client-serveur (multi-utilisateurs)

*Si, en mode autonome, le partage de la base est possible, l'accès concurrent ne l'est pas : le premier utilisateur obtient tous les droits, les suivants peuvent lire mais non modifier les données. Ici, nous allons montrer comment partager les données et y accéder depuis plusieurs postes simultanément (10 postes semble une limite raisonnable pour des bases de données accédées de manière permanente).*

Nous ne reviendrons pas ici sur l'installation du moteur *HSQLDB* présentée au chapitre 3 Installer *HSQLDB*. Ce moteur n'aura à être installé que sur la machine serveur. Les clients utiliseront *Base* et, peut-être, d'autres outils pour accéder aux données de la base partagée.

Nous allons voir comment configurer le serveur, comment configurer *Base* sur le serveur, comment réaliser la connexion, et enfin comment configurer *Base* sur les clients pour qu'ils accèdent à la base de données.

## 5.1. Opérations préalables

Avant toute autre opération, nous nous assurons d'avoir sauvegardé notre base externe.

## 5.2. Configurer le serveur

Un seul ordinateur – le serveur de bases de données – fait fonctionner le serveur *HSQLDB* et rend les données disponibles au monde extérieur. La configuration décrite ici ne doit donc être exécutée qu'une seule fois, sur la machine serveur. Nous décrivons également le paramétrage (facultatif) d'un frontal *Base* sur la machine hôte.

### a. Configurer LibreOffice

#### Étape 1 Installer *HSQLDB*

Installons *HSQLDB* comme indiqué au chapitre 3.

#### Étape 2 Relever les informations

Nous relevons :

- l'adresse IP du poste serveur ou son nom d'hôte  
Sous *Windows*, la commande `ipconfig /all` donne ces informations  
Sous *GNU/Linux*, nous utiliserons la commande `ifconfig`.
- le chemin vers `Java.exe` sur le poste,
- le chemin vers `hsqldb.jar`.



### Étape 3 Créer les scripts de démarrage et d'arrêt

Créons les scripts de démarrage et d'arrêt pour le serveur *HSQLDB*.

Ces scripts automatisent les opérations et nous dispensent d'avoir à entrer les commandes manuellement. Voici deux exemples de scripts, à aménager en fonction des besoins. Nous enregistrerons ces scripts dans un répertoire dédié.

#### Script de démarrage

Ce script sera nommé (par exemple) `Start_HSQLDB_Server.cmd`. En voici la syntaxe générale :

```
java -cp ../lib/hsqldb.jar org.hsqldb.Server -database.0 file:mydb -dbname.0 xdb
```

où

- java invoque l'appel du programme Java ;
- l'option `-cp ../lib/hsqldb.jar org.hsqldb.Server` indique à Java le chemin de la classe à exécuter (le serveur *HSQLDB*) ;
- l'option `-database.0 file:mydb` spécifie à *HSQLDB* la base de données à gérer. `mydb` est le nom de la base (les fichiers de la base ont tous `mydb` comme nom). Ce nom peut comporter des majuscules et des minuscules. En pratique, et selon le point d'où sera émise la commande, on pourra avoir `file:mydb` ou `file:c:/mydatabase/mydb` (attention aux barres de fraction qui ne sont pas inversées sous Windows !) ;
- (facultatif) l'option `-dbname.0 xdb` indique l'alias de la base de données à *HSQLDB*. `xdb` est le nom de l'alias. Seules les lettres minuscules sont permises.

Un script *shell* Windows s'écrit typiquement ainsi :

```
@echo off
set javapath=C:\Program Files\Java\jre6\bin\Java.exe
set jarpath=C:\Program Files\openoffice.org 3\basis\program\classes\hsqldb.jar
"%javapath%" -cp "%jarpath%" org.hsqldb.Server -database.0 file:mydb
```

Ce script doit être adapté à notre situation : les variables `javapath` et `jarpath` contiennent les chemins d'accès aux fichiers `java.exe` et `hsqldb.jar`.

#### Script d'arrêt

Syntaxe générale de la ligne de commande :

```
java -cp ../lib/hsqldb.jar org.hsqldb.util.ShutdownServer
```

Un script *shell* Windows `Stop_HSQLDB_Server.cmd` peut s'écrire ainsi :

```
@echo off
set javapath=C:\Program Files\Java\jre6\bin\Java.exe
set jarpath=C:\Program Files\openoffice.org 3\basis\program\classes\hsqldb.jar
"%javapath%" -cp "%jarpath%" org.hsqldb.util.ShutdownServer
```

Ce script doit être adapté à notre situation : les variables `javapath` et `jarpath` contiennent les chemins d'accès aux fichiers `java.exe` et `hsqldb.jar`.

#### Attention

Le script ci-dessus ne doit **pas** être utilisé pour arrêter une base ou un serveur protégé par mot de passe.

Lorsque le serveur est protégé par un mot de passe (utilisateur SA), la commande `"%javapath%" -jar "%jarpath%" --sql "shutdown;" --inlineRC`



```
url=jdbc:hsqldb:hsqldb://localhost,user=SA
```

demande ce mot de passe avant la fermeture de la base.

Si nous désignons la base de données par un alias à l'ouverture, alors la commande:

```
"%javapath%" -jar "%jarpath%" --sql "shutdown;" --inlineRC
```

```
url=jdbc:hsqldb:hsqldb://localhost/mydb_alias,user=SA
```

demande le mot de passe avant fermeture.

## Étape 4 Démarrer le serveur HSQLDB

Démarrons le serveur *HSQLDB* au moyen du script de démarrage.

### b. Connecter Base à la base existante au moyen de l'Assistant

Nous lançons le module *Base*. L'Assistant Bases de données s'exécute.

#### Étape 1 Mode d'accès au moteur de bases de données

À cette étape, nous définissons le mode d'accès au moteur de base de données.

Nous choisissons **Se connecter à une base de données existante** au moyen du connecteur JDBC (*Java database connectivity*) (figure 14).

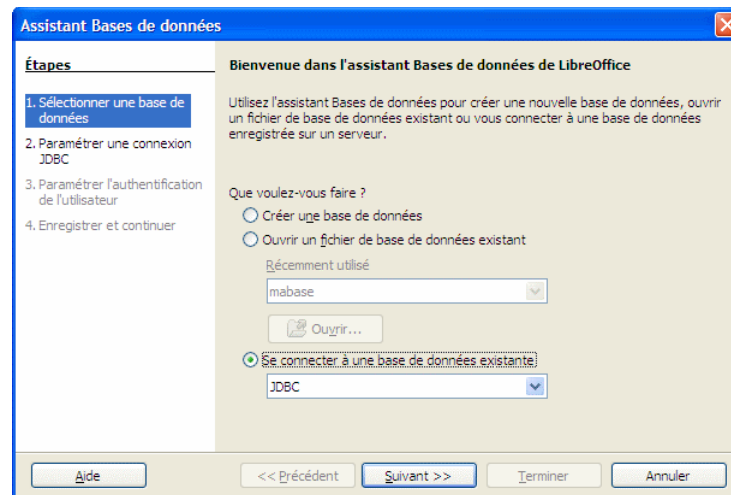


Figure 14 : Assistant Bases de données – Connexion JDBC

Continuons par **Suivant**.

#### Étape 2 Définir l'emplacement de la base

Maintenant, nous devons expliciter l'emplacement de la base (**URL**) et le pilote JDBC à utiliser (figure 17).

Dans la rubrique **URL**, nous entrons :

```
hsqldb:file:hsqldb:hsqldb://localhost/;default_schema=true
```

où

- localhost  
Désigne l'ordinateur local ;
- default\_schema=true  
Est placé là par souci de compatibilité avec d'anciennes versions de *HSQLDB*.

Plus d'informations sur ce paramètre dans la documentation disponible sur le site *HSQLDB* (voir chap. Références).



## Note

Le protocole `hsq1:` et la connexion `//localhost/` spécifiés ci-dessus permettent d'accéder à la première base de données, sans nom d'alias, servie par le serveur HSQLDB sur cet ordinateur.

Nous nous reporterons à la documentation de HSQLDB pour spécifier une base de données par son nom ou par son alias, ou bien pour spécifier un protocole d'accès sécurisé (recommandé), ou encore pour accéder à une base de données sur un autre machine en précisant son adresse IP ou son nom d'hôte.

Par exemple :

```
hsqldb:hsq1s://192.168.1.210/;default_schema=true
```

fournit une connexion sécurisée (`hsq1s` au lieu de `hsq1`) vers un autre ordinateur sur le réseau local (à l'adresse statique `192.168.1.210`), sur le port sécurisé par défaut (554, non spécifié) et fournit l'accès à la première base de données servie par le serveur HSQLDB (le nom d'alias est vide).

Dans la rubrique **Classe de pilote JDBC**, nous saisissons :  
`org.hsqldb.jdbcDriver`

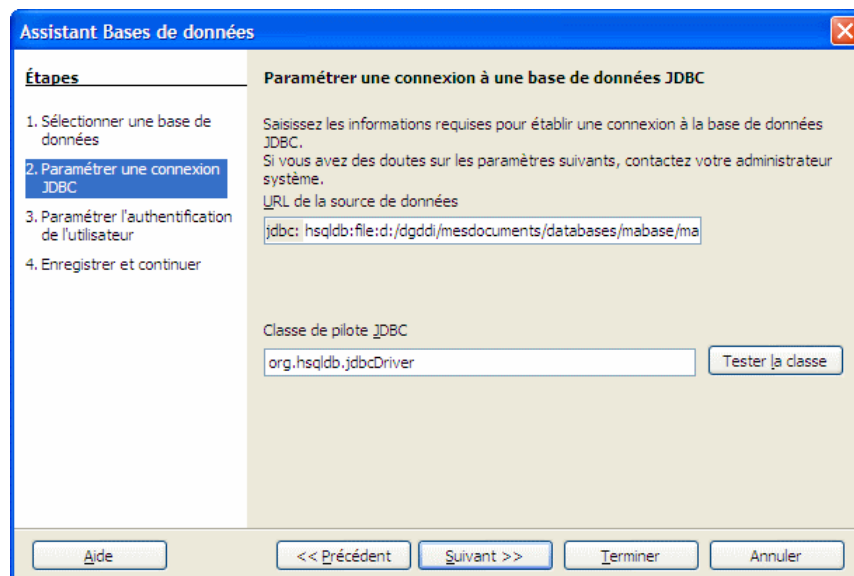
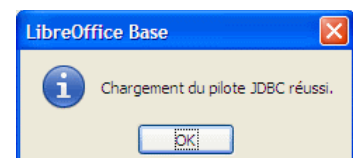


Figure 15 : Assistant Bases de données – Paramètres de la connexion

Le bouton **Tester la classe** permet de vérifier la bonne connexion au pilote `JDBC` selon la configuration ci-dessus (ci-contre).



Nous continuons par **Suivant**.

## Étape 3 Paramétrer l'authentification

Ici, nous indiquons les paramètres d'authentification pour l'accès à la base via le moteur `HSQLDB` (figure 17 ci-dessous).

Le nom d'utilisateur par défaut est `SA`, sans mot de passe (il est également possible de laisser la rubrique vide). Le moteur `HSQLDB` permet de définir des protections par nom d'utilisateur et mot de passe. Ceci sort du cadre de ce document. Nous consulterons la documentation `HSQLDB` pour plus de renseignements.

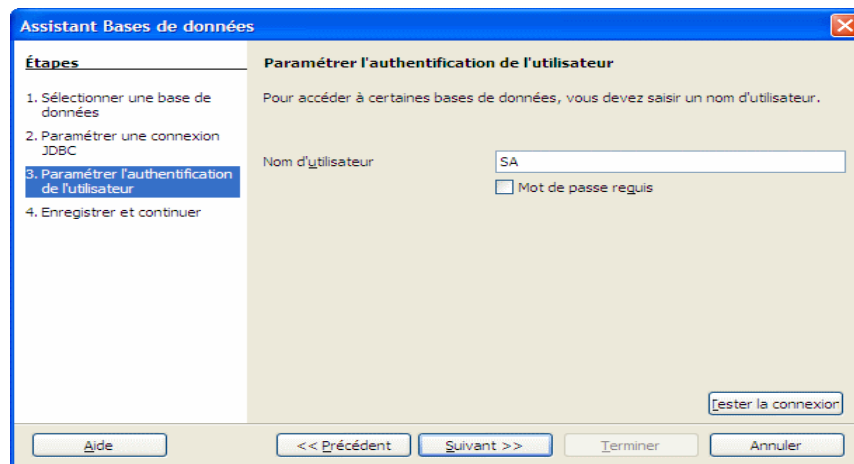
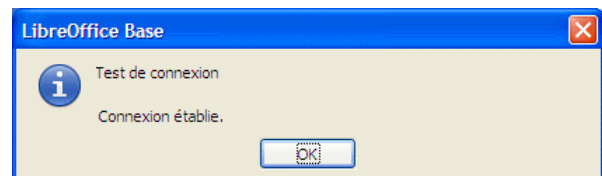


Figure 16 : Assistant Bases de données – Authentification

Le bouton **Tester la connexion** permet de vérifier la bonne connexion au moteur externe *HSQLDB*.



Nous passons à la dernière étape par **Suivant**.

#### Étape 4 Référencer la base

Enfin, nous devons choisir si nous désirons référencer cette base de données dans *LibO* (figure 17). Ce choix est conseillé afin d'en permettre l'accès ultérieur direct depuis les modules de la suite bureautique, pour réaliser des publipostages par exemple.

En mode client-serveur, il est rare que le serveur soit utilisé pour des tâches de gestion – en tout cas, il ne le devrait pas –, aussi le référencement de la base est-il sans doute superflu. Cependant, ce référencement pourra être utile lors de nos tests.

En cliquant **Terminer**, *LibO* demande le nom et l'emplacement pour le fichier .odf de configuration d'accès à la base.



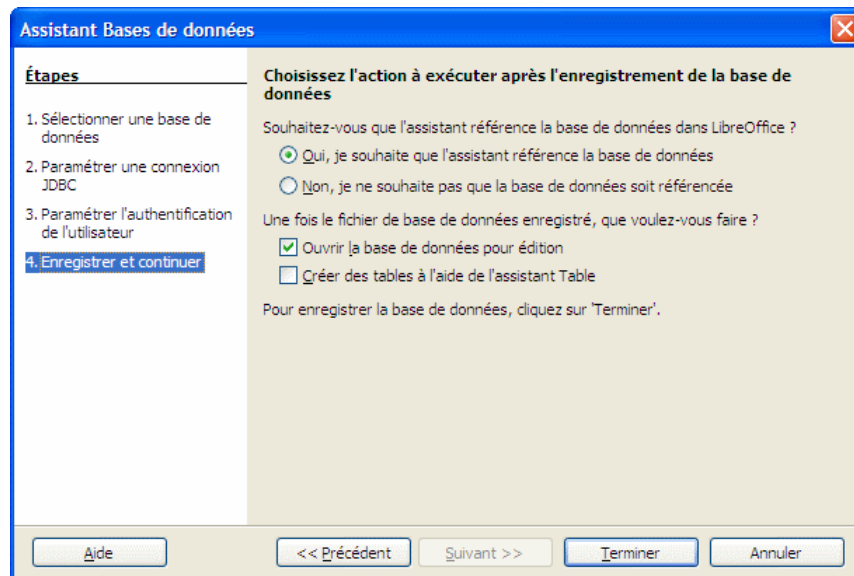


Figure 17 : Assistant Bases de données – Enregistrement

Nous conseillons d'enregistrer ce fichier .odb dans le répertoire de la base elle-même. Nous optimiserons ainsi le système de sauvegarde qui sera mis en place ensuite, l'ensemble des fichiers constituant la base de données étant regroupés dans ce répertoire unique.

Lorsque le module *Base* se lance, nous constatons que la barre d'état montre les paramètres que nous venons de définir. À gauche, le type de connexion (typiquement : JDBC), au centre les paramètres de fonctionnement pour l'accès à la base (xxx), à droite le nom de l'utilisateur (SA).

Ces paramètres sont modifiables *a posteriori*, par le menu **Édition > Base de données > Propriétés** ou **Connexion**.

### Serveur web

Il est possible de rendre visible sur Internet une machine exécutant un serveur HSQLDB en mode serveur web.

Le paramétrage d'un serveur web HSQLDB met en jeu notre routeur sur lequel nous devons ajuster la translation de ports et les paramètres DHCP. Notre serveur web devra disposer d'une adresse et d'un port IP fixes. L'adresse du routeur sur le WAN devra également être une adresse fixe (au besoin nous pourrions utiliser des services tels que DynDNS).

Côté clients, le seul paramétrage requis sera celui de l'URL de la base de données dans le module Base, qui sera de la forme :

```
hsqldb:https://mon_domaine.homeip.net/;default_schema=true.
```

HSQLDB est par nature sécurisé car il s'exécute dans une machine virtuelle Java. Pour une solution « à l'épreuve des balles », voir XXX.

Il est donc possible d'exploiter notre suite bureautique comme frontal pour accéder à une base de données connectée à l'Internet. Ceci parce qu'un accès Internet ou réseau est une fonction du driver qui peut être configuré pour de nombreux scénarios.

## c. Tests et dépannage

Le module *Base* étant démarré, nous vérifions les paramètres d'état en bas de l'écran.



### **Vérification**

Le mode choisi doit y être apparent dans la barre d'état de *Base* : JDBC et `hsqldb:hsql://localhost/default_schema=true`.

Si la barre d'état indique *Base intégrée* et Moteur de base de données HSQLDB, alors nous n'utilisons pas le serveur externe pour le fichier `.odb` courant.

### **Test final**

Une fois l'état de fonctionnement correct confirmé, cliquons sur l'icône **Tables** dans le module *Base*. Si les tables de la base apparaissent sans qu'un message d'erreur soit affiché, alors notre système fonctionne correctement.

### **Erreurs**

Au contraire, si un message d'erreur est affiché, cela peut provenir de l'une quelconque des raisons suivantes :

1. Le serveur *HSQLDB* n'est pas actif ou pas configuré.  
Nous vérifions que `Java.exe` fait partie des tâches en cours d'exécution ;
2. Les rubriques **URL** et **Classe de pilote JDBC** ne sont pas configurées correctement.  
Nous les vérifions par **Édition / Base de données / Propriétés**.
3. La syntaxe utilisée pour désigner la **Classe de pilote JDBC** est très précise. Vérifions qu'aucun espace parasite ne s'est glissé dans ou après le libellé `org.hsqldb.jdbcDriver`
4. Si nous utilisons *Base* et le serveur *HSQLDB* sur la même machine, il est possible que plusieurs occurrences de `Java.exe` soient en cours d'exécution, *Base* et le serveur *HSQLDB* utilisant chacun une occurrence différente.
5. Nous pouvons également vérifier la chaîne de lancement du serveur *HSQLDB*, particulièrement le chemin des différents fichiers. Ces chemins doivent correspondre à ceux des fichiers `Java.exe` et `hsqldb.jar` dans *LibreOffice*. Il est préférable de ne pas entrer de noms d'alias car cela complique les choses, en particulier pour le dépannage, à la fois dans les propriétés de connexion pour le driver sous *Base* et dans la chaîne de démarrage du serveur *HSQLDB*.
6. *LibreOffice* et *OpenOffice.org* sont proposées sous forme d'application portable. Cette version contient du code qui gère automatiquement le chemin vers `Java.exe`. Le résultat est que nos paramètres peuvent être remplacés par ceux de l'application portable. Il semble que seulement deux répertoires soient permis pour `JavaPortable` sous *LibreOffice* et *OpenOffice.org* portable :  
`PortableApps\CommonFiles\Java`  
ou `PortableApps\LibreOfficePortable\App\Java`.
7. Des problèmes d'identification de l'utilisateur : identifiant ou mot de passe erroné.

---

### **Conseil**

*Les questions relatives aux partages réseau ou aux problèmes de permissions réclament un dépannage avancé qui dépasse le cadre de ce document. Chaque fois que c'est possible, il est préférable de tester les composants client-serveur sur la machine serveur avant de déployer sur les clients.*

---



### 5.3. Configurer les clients

Notre base de données est maintenant gérée par un serveur *HSQLDB* fonctionnel sur un machine serveur.

Reste à offrir l'accès à la base de données aux différents postes clients à travers le module *Base* de la suite bureautique. Rien n'est plus simple : il nous suffit de récupérer une copie du fichier `.odb` créé ci-dessus (5.2.b) et de la copier sur la machine cliente. Ce fichier constitue le « frontal » d'accès à notre base de données en réseau.

# 6. Transformer une base incorporée en base autonome

*Nous avons installé un moteur externe – qu’il soit mono-poste ou client-serveur – et nous savons configurer nos bases pour exploiter ce moteur. Nous pouvons transformer (« migrer ») les bases intégrées que nous utilisons jusqu’à maintenant en bases externes, ce qui leur confèrera plus de robustesse et nous donnera plus d’indépendance.*

## 6.1. Préalables

Avant de commencer, assurons-nous que :

- nous disposons d’une copie de **sauvegarde** de notre fichier .odb,  
Un incident peut toujours se produire. Conserver une copie de la base de données dans son état antérieur est une sage précaution ;
- nous avons préparé un **répertoire** pour accueillir la nouvelle base externe.  
Ce répertoire est l’infrastructure de notre future base.

## 6.2. Étapes de la migration

La migration s’effectue en cinq étapes.

### **Étape 1** Refermer Base

Refermons le module *Base* s’il est ouvert.

### **Étape 2** Extraire les fichiers

Extrayons les fichiers relatifs à la base de données du fichier .odb.

Les fichiers .odb sont des archives compressées au format *Zip*. Utilisons un outil de décompression (ex : *7-zip*, voir chap. Références) pour accéder au contenu du fichier .odb.

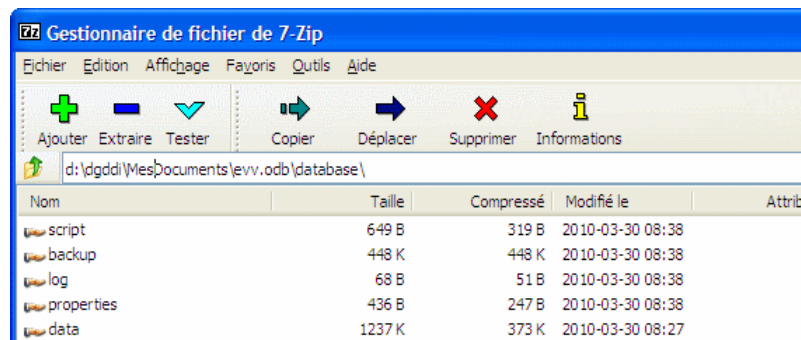


Figure 18 : Le contenu d'une base de données HSQLDB nouvellement créée

Les fichiers qui nous intéressent sont placés dans le répertoire database de l'archive. Ils se nomment `script`, `backup`, `properties` et `data`. Un fichier `log` peut également exister (figure 18).

Nous les extrayons vers le répertoire dédié à la nouvelle base externe (voir chap. 6.1 Préalables). Nous les renommons de manière que leur nom devienne l'extension d'autant de fichiers dont le nom sera celui de la base de données.

Exemple : si la base de données se nomme `BaseCompta`, nous renommons

- `script` → `BaseCompta.script`,
- `backup` → `BaseCompta.backup`,
- `properties` → `BaseCompta.properties`
- et `data` → `BaseCompta.data`.

### Note

Les fichiers `.script`, `.data` et `.log` sont des composants critiques de nos bases de données HSQLDB. Ils doivent être sauvegardés très régulièrement.

Les fichiers `.data` (et `.backup` qui l'accompagne) n'existeront pas si nous utilisons la configuration par défaut de HSQLDB (tables en mémoire). Cependant, lors de notre migration depuis LibreOffice – qui utilise des tables « en cache » – nous trouverons ces deux types de fichiers. Si bien que, à moins de modifier un paramètre dans le fonctionnement de HSQLDB, ce dernier continuera de les créer.

Le fichier `.log` sera détruit après un arrêt « propre » de la base de données ou après réalisation d'un « point de contrôle ». L'exécution d'une des commandes SQL `SHUTDOWN COMPACT` ou `CHECKPOINT DEFrag` réalise ce nettoyage.

### Étape 3 Créer le fichier .odb

Si ce n'est déjà fait, créons le fichier `.odb` correspondant à la nouvelle base autonome ainsi que décrit plus haut au chap.4.2 Créer une base externe.

### Étape 4 Copier/Coller les autres informations

Démarrons maintenant deux instances du module *Base* :

- Dans la première instance (source) nous ouvrons la base incorporée en cours de migration ;
- dans la seconde instance (cible) nous ouvrons la nouvelle base externe en cours de création.

Nous allons maintenant pouvoir basculer l'ensemble des autres composants de notre base incorporée : formulaires, rapports et requêtes par un simple copier/déposer entre les fenêtres des deux instances.



Les macros peuvent être copiées/déposées depuis le menu **Outils > Macros > Gérer les macros** dans *Base*, ou encore être exportées/importées comme bibliothèques (voir le même menu).

---

**Et les tables ?**

*Bien entendu, nous ne touchons pas aux tables elles-mêmes, déjà copiées à l'étape 2.*

---

**Étape 5** Rétablir les relations

Il ne reste plus qu'à rétablir manuellement les relations entre tables dans la nouvelle base externe, de manière à correspondre à celles qui sont définies dans la base incorporée.

Notre base externe est maintenant fonctionnelle !

---

**Presque...**

*Nous remarquons que les propriétés des champs de nos tables sont verrouillées pour l'édition. En revanche, la création, la recopie et la suppression de tables sont toujours possibles par l'intermédiaire des assistants.*

*Ces possibilités nous offrent donc une méthode de contournement à l'édition des propriétés des champs : copions puis collons une table existante vers l'icône **Tables** pour créer une copie renommée. L'Assistant permet également tout changement dans la structure de la table. Il suffit ensuite de supprimer la table initiale puis de renommer la table nouvellement créée par un nouveau Copier/Coller.*

**Attention**

*Gardons à l'esprit que, dès lors que des données figurent dans les tables, le changement de structure peut s'avérer difficile si des conversions de données sont nécessaires.*

*L'usage de commandes SQL (`ALTER TABLE` ou `ALTER COLUMN`) est également possible, tout comme l'édition du fichier `.script`.*

---

# 7. Sauvegarde et restauration de bases HSQLDB utilisées avec Base

À de nombreuses reprises nous avons insisté sur la nécessité de **sauvegarder** les données du serveur. Cette opération doit être effectuée régulièrement. Elle permet de prévoir et palier les aléas auxquels peuvent être soumises la base de données, les informations de connexion voire la machine serveur elle-même. Si nous conservons des versions successives de nos sauvegardes, alors nous serons également à même de retrouver une information d'un état antérieur. Enfin, grâce aux sauvegardes, nous pourrons aisément transférer la base d'un serveur vers un autre, en cas de défaillance ou de mise à jour du matériel.

Voici quelques précisions sur la façon de procéder à cette sauvegarde et à la restauration qui en est la réciproque obligée. Les différents paramètres en jeu ont été présentés au chapitre 2.1.

## 7.1. Sauvegarde

La sauvegarde concerne, bien entendu, chacun des trois ensembles listés ci-dessus.

### a. Sauvegarde de la base de données HSQLDB

Nous enregistrons le répertoire qui contient les fichiers de la base *HSQLDB* sur un support externe.

### b. Sauvegarde des formulaires, requêtes, etc.

De même, nous enregistrons le fichier `.odb` sur un support externe.

---

*Si nous avons pris la précaution d'enregistrer le fichier `.odb` dans le même répertoire que les fichiers de la base de données, la sauvegarde de ces deux éléments peut s'effectuer en une seule opération de copie (point précédent).*

---

### c. Paramètres LibreOffice

La sauvegarde de ces paramètres est facultative : il est toujours possible de les restaurer manuellement sans trop de difficulté.

#### ***Paramètres d'enregistrement de la base***

L'enregistrement de la base de données dans *LibreOffice* est un paramètre du fichier `registrymodifications.xcu` qui se trouve à la racine du profil de l'utilisateur *LibreOffice*. Nous pouvons donc enregistrer ce fichier sur un support externe.



### **Paramètres d'accès au moteur HSQLDB externe**

Nous enregistrons le fichier `javasettings_XXX.xml` de configuration des options *Java* sur un support externe. Ce fichier est stocké dans le sous-répertoire `/config` du profil de l'utilisateur *LibreOffice*.

## 7.2. Restauration

Nous suivons bien entendu le même schéma que pour la sauvegarde.

### a. Restauration de la base de données HSQLDB

Nous recopions les fichiers constituant la base de données dans son répertoire.

*Le changement de nom du répertoire lors de la restauration implique la modification éventuelle des paramètres d'enregistrement de la base dans LibreOffice (voir ci-dessous).*

### b. Restauration des formulaires, requêtes, etc.

Nous restaurons le fichier `.odb` à l'endroit voulu.

*Si nous avons pris la précaution d'enregistrer les fichiers `.odb` dans le même répertoire que les fichiers de la base de données, la restauration de ces deux éléments peut s'effectuer en une seule opération de copie (point précédent).*

*En cas de changement du nom du répertoire de stockage de la base de données, il faut ouvrir le fichier `.odb` et le reconfigurer par **Édition / Base de données / Propriétés**.*

### c. Restauration des paramètres LibreOffice

#### **Paramètres d'enregistrement de la base**

Si nous désirons retrouver l'enregistrement de la base de données dans *LibreOffice* alors il nous faut restaurer le fichier `registrymodifications.xcu` vers la racine du profil de l'utilisateur *LibreOffice*.

#### **Enregistrement manuel**

*Le fichier `registrymodifications.xcu` comporte bien d'autres paramétrages. Sa restauration peut donc avoir des effets indésirables.*

*Si nous ne restaurons pas ce fichier, alors nous devons re-enregistrer la base de données dans LibreOffice, si tel est le besoin. Cette opération s'effectue par **Outils > Options, LibreOffice Base > Bases de données, bouton Nouveau**.*

#### **Paramètres d'accès au moteur HSQLDB externe**

Le fichier `javasettings` doit être restauré dans son répertoire natif, c'est-à-dire dans le profil *LibO* de l'utilisateur, au sous-répertoire `/config`.

#### **To restore or not to restore ?**

*La restauration du fichier `javasettings_XXX.xml` n'a de sens que si la machine cible comporte les **mêmes** versions de *Java* et de *HSQLDB* **et** si les chemins d'accès à ces outils sont constants.*

*Dans les autres cas – changement de version **ou** de chemin d'accès – il sera plus simple de reconstituer la configuration *ex nihilo* telle qu'elle est décrite au ch. 4.1 ou 5.2.*



# 8. Annexes

## 8.1. Divorcer d'avec MS-Access

Pour qui désire donner une nouvelle vie à des portages de bases *MS-Access* sous la forme de solutions multiplate-formes librement distribuées, l'usage de *Base* comme frontal, avec *HSQLDB* en mode client-serveur est une option à prendre en considération.

La migration – sous *Windows* – nous permet d'utiliser *Base* pour nous connecter à une base de données *MS-Access* ou *MS-Access 2007*, ce qui nous fournit une connectivité limitée en lecture/écriture. L'importation des tables *Access* sous *HSQLDB* (ou tout autre moteur) permet de retrouver le contrôle complet sur les données et structures. Une simple manipulation de copier/coller entre instances de *Base* permet d'arriver à nos fins.

En revanche, le portage des autres composants *Access* vers *Base* n'est pas trivial. Nous trouverons une source d'informations ici :

[https://wiki.openoffice.org/wiki/MSA-Base\\_Faq](https://wiki.openoffice.org/wiki/MSA-Base_Faq)

Sous la version *Access 2010*, *Base* ne sait pas importer les formulaires, rapports, requêtes et macros. Et ça ne risque pas de changer avant longtemps. De fait, *Base* n'est pas *Access...* et *Access* n'est pas *Base*

(<https://forum.openoffice.org/en/forum/viewtopic.php?f=13&t=14060#p66326>).

## 8.2. Les limites de Base

Chaque instance de *Base* (le fichier *.odb*) est limité à l'accès à une et une seule base de données.

Cependant, d'autres modules de *LibreOffice* ne souffrent pas de cette limitation, c'est le cas de *Calc*, *Writer* ou *Impress*. Ce qui nous suggère des méthodes de contournement : les contrôles de formulaires (zone de texte, boîte de liste, grille, etc.) créés sous ces modules permettent d'accéder à différentes sources de données (fichiers *.odb*). Et chacun de ces fichiers *.odb* permet un lien transparent avec la base de données correspondante, qu'elle soit incorporée ou autonome.

Références

JRE

*Java Runtime Environment*

L'environnement d'exécution Java permet l'exécution de programmes écrits en Java sur les principales plate-formes *Windows*, *Linux*, *MacOS X*.

Les versions de JRE sont téléchargeables sur le site d'Oracle :

<http://www.java.com/fr/download/>

HSQLDB

Site: <http://hsqldb.org/>

Téléchargement : <http://sourceforge.net/projects/hsqldb/files/>

Documentation : <http://hsqldb.org/web/hsqldbDocsFrame.html>



<i>Squirrel</i>	Outil de gestion de bases de données externes, écrit en Java, donc multi-plateformes. <i>Squirrel</i> peut accéder et gérer tous les SGBD disposant d'un connecteur <i>JDBC</i> . <a href="http://squirrel-sql.sourceforge.net/">http://squirrel-sql.sourceforge.net/</a>
7-zip	7-zip est un outil de gestion d'archives, dit outil de compression. Ce logiciel (libre) gère de très nombreux formats d'archives compressées : Zip, 7z (son format natif), tar, gz, etc. <a href="http://7-zip.org/">http://7-zip.org/</a>

### 8.3. Glossaire

<i>Back-end</i>	Élément principal. En français : dorsal ou principal.
Client	Ordinateur ou logiciel qui reçoit ses informations d'un serveur, en général à travers un réseau.
<i>Embedded mode</i>	Voir Mode incorporé.
Enregistrement (d'une base dans <i>LibreOffice</i> )	Opération par laquelle tous les modules de la suite bureautique <i>LibreOffice</i> ont connaissance de l'existence d'une base et peuvent donc y accéder.
<i>File mode</i>	Voir Mode autonome.
Frontal	Logiciel permettant la communication entre plusieurs applications hétérogènes (SGBDR ↔ Suite bureautique).
Hôte	Machine qui héberge un logiciel.
Mode autonome	En bases de données, mode de fonctionnement dans lequel le logiciel et la base de données sont indépendants.
Mode client-serveur	Mode de fonctionnement dans lequel plusieurs utilisateurs peuvent accéder à une même ressource simultanément.
Mode incorporé	En bases de données, mode de fonctionnement dans lequel le logiciel et la base de données sont intimement liés. On trouve également « mode embarqué ».
Mode multiposte	Mode de fonctionnement dans lequel plusieurs utilisateurs peuvent accéder à une même ressource, simultanément ou non.
Moteur de base de données	Logiciel qui contrôle, lit, enregistre et trie des informations dans une ou plusieurs bases de données.
Pilote de bases de données	Logiciel qui traduit les appels des applications vers un moteur de base de données et réciproquement.
Serveur	Ordinateur ou logiciel qui distribue les informations qu'il détient vers des ordinateurs ou des logiciels clients sur un réseau.

# 9. Auteur et licence

## 9.1. Auteur

Auteur : Jean-François Nifenecker  
jean-françois.nifenecker@laposte.net

## 9.2. Remerciements

Ce document résulte de la compilation et de l'adaptation d'un grand nombre de messages sur les listes de discussions et forums, francophones ou internationaux. Que les auteurs, aussi nombreux que compétents soient ici remerciés pour leur volonté de partage, fidèle aux valeurs du Libre, en particulier Villeroy, DACM et Alex Thurgood.

Merci à **Isabelle Dutailly** pour ses remarques, commentaires et suggestions, mais également pour sa proposition de couverture.

## 9.3. Historique des modifications

Version	Date	Commentaire
0.9	8/6/2014	Première version pour relecture

## 9.4. Licence

Ce document est offert sous les conditions de la licence Creative Commons version 4, Attribution – Partage dans les Mêmes Conditions

<http://creativecommons.org/licenses/by-sa/4.0/>

