

# Ch. 3. Démarrage rapide sous UNIX

De Apache OpenOffice Wiki

< FR | Documentation | HSQLDB Guide

## Sommaire

- 1 But de ce chapitre
- 2 Installation
- 3 Installer une instance de base de données persistante et un serveur Hsqldb
- 4 Accès à la base de données
- 5 Créer des comptes additionnels
- 6 Shutdown
- 7 Exécuter Hsqldb comme un démon du système
  - 7.1 Portabilité du script init de hsqldb
  - 7.2 Procédure d'installation du script init
  - 7.3 Dépannage du script Init

## Chapitre 3. Démarrage rapide sous UNIX

Comment installer et utiliser rapidement Hsqldb sous UNIX et Mac OS X

**Blaine Simpson**

HSQLDB Development Group

<blaine.simpson@admc.com>

**\$Date: 2005/07/25 23:20:53 \$**

## But de ce chapitre

Ce chapitre explique comment installer, exécuter et utiliser rapidement Hsqldb sous UNIX.

HSQLDB a beaucoup de fonctionnalités optionnelles puissantes. Je n'en couvrirais que très peu. Ce chapitre couvre ce que je pense être l'installation la plus commune sous UNIX : Exécuter une base de données multi-utilisateurs dotée d'une persistance des données permanente. (Par la suite j'explique que les données sont stockées sur disque pour être persistantes durant les fermetures et démarrages successifs de la base de données). J'explique également comment exécuter Hsqldb comme un démon du système. (system

daemon)

## Installation

Allez à la page <http://sourceforge.net/projects/hsqldb> et suivez le lien ("files") "Download". Vous voulez la dernière version. Celle-ci portera le numéro de version le plus élevé sous le titre "Hsqldb". Vérifiez s'il existe la distribution de la version courante dans le format que vous souhaitez.

Si vous voulez un rpm, vous devez trouver la version en cours de HSQLDB comme décrit dans le paragraphe précédent. Ensuite, sur la page <http://www.jpackage.org/> cliquez sur "hsqldb" dans "free section" et vérifiez si ils ont déjà construit la version courante de HSQLDB. **Hopefully, the JPackage folk will document what JVM versions their rpm will support (currently they document this neither on their site nor within the package itself).** (Je ne peux vraiment pas documenter comment télécharger d'un site complètement hors de mon contrôle).



### Note :

Il peut très bien arriver que certains des formats de fichiers évoqués ci-dessous ne soient pas en fait délivrés. Si c'est le cas, c'est que nous ne nous sommes pas occupés de le construire.

L'installation binaire dépend du format de paquets que vous avez téléchargé.

### *Installation depuis un fichier .pkg.Z*

L'utilisation de ce packaging est réservé à un administrateur Solaris. C'est un paquet System V ([http://fr.wikipedia.org/wiki/UNIX\\_System\\_V](http://fr.wikipedia.org/wiki/UNIX_System_V)). Téléchargez puis décompressez ce paquet avec `uncompress` ou `gunzip`.

```
uncompress filename.pkg.Z
```

Vous pouvez lire à propos de ce paquet en exécutant :

```
pkginfo -l -d filename.pkg
```

Connectez vous comme administrateur et exécutez `pkgadd` pour installer :

```
pkgadd -d filename.pkg
```

### *Installation depuis un fichier .rpm*

C'est un paquet Linux rpm. Après que vous ayez téléchargé le rpm, vous obtenez de l'information en exécutant :

```
rpm -qip /path/to/file.rpm
```

Vous pouvez installer ou mettre à jour les rpm's en exécutant :

```
rpm -Uvh /path/to/file.rpm
```

Commande Administrateur. Les utilisateurs de Suze voudront préserver Yast des paquets installés en exécutant le rpm à partir de Yast : `yast2 -i /path/to/file.rpm`.

### ***Installation depuis un fichier .zip***

Vous devez extraire le fichier zip dans le dossier parent du nouveau dossier racine de HSQLDB. (Vous n'avez pas besoin de créer le dossier **HSQLDB\_HOME**, le processus d'extraction le fera correctement pour vous)

```
cd parent/of/new/hsqldb/home
unzip /path/to/file.zip
```

Tous les fichiers de l'archive zip seront extraits dans les sous-répertoires du nouveau dossier hsqldb.

Parcourez les fichiers installés. (Dans le dossier hsqldb issue du zip. Ou bien avec les utilitaires de votre distribution) (**Under hsqldb for zip file installations. Otherwise, use the utilities for your packaging system**). Le fichier le plus important du système hsqldb est hsqldb.jar, situé dans le dossier lib.

#### **Important**



Pour le propos de ce chapitre, on admet HSQLDB\_HOME comme étant le dossier parent du répertoire lib contenant hsqldb.jar. C'est à dire que si votre chemin vers hsqldb.jar est /a/b/hsqldb/lib/hsqldb.jar, celle de HSQLDB\_HOME est /a/b/hsqldb.

Si la description de votre distribution précise que le fichier hsqldb.jar fonctionne avec votre version de Java, l'installation est finie. Sinon vous devez construire un nouveau fichier hsqldb.jar.

Si vous avez suivi les instructions ci dessus et ne savez toujours pas quelle version de Java supporte votre hsqldb.jar, lisez HSQLDB\_HOME/readme.txt et HSQLDB\_HOME/index.html. Si malgré tout vous n'avez pas trouvé vous pouvez

tester votre `hsqldb.jar` et voir s'il fonctionne, ou bien construire le votre.

Pour utiliser le fichier `hsqldb.jar` fourni, passez directement à la prochaine section de ce document. Ou alors construisez un nouveau fichier `hsqldb.jar`.

### Procédure 3.1. Construire `hsqldb.jar`

1.

Si Ant n'est pas déjà installé, téléchargez la dernière version binaire stable à <http://ant.apache.org> . Placez vous dans le dossier pour l'installation et extrayez l'archive avec :

```
unzip /path/to/file.zip
```

ou

```
tar -xzf /path/to/file.tar.gz
```

ou

```
bunzip2 -c /path/to/file.tar.bz2 | tar -xzf -
```

Tous les fichiers seront installés dans un nouveau sous-dossier nommé `named apache-ant- + version`. Si vous le souhaitez vous pouvez renommer ce dossier après l'extraction.

2.

Définissez la variable d'environnement `JAVA_HOME` sur votre dossier racine de votre distribution Java JRE ou SDK, comme :

```
export JAVA_HOME; JAVA_HOME=/usr/java/j2sdk1.4.0
```

Le chemin est complètement dépendant de la distribution UNIX / Linux. Les distributions Java au format rpm de Sun s'installent normalement dans `/usr/java/quelquechose`. Les distributions Java de Sun en paquets System V (incluant celles de Solaris) s'installent normalement dans `/usr/quelquechose`, avec un lien symbolique de `/usr/java` vers la version par défaut (donc pour Solaris l'adresse de `JAVA_HOME` est couramment `/usr/java`).

3.

Supprimez le fichier existant `HSQLDB_HOME/lib/hsqldb.jar`.

4.

Placez vous dans le dossier `HSQLDB_HOME/build`. Assurez vous que le sous-dossier `bin` de la racine de Ant soit dans votre recherche de chemins. Exécutez la commande suivante :

```
ant hsqldb
```

Ceci construira une nouvelle archive HSQLDB\_HOME/lib/hsqldb.jar.

Voir l'annexe A : Construire HSQLDB si vous voulez construire quelque chose d'autre qu'une archive hsqldb.jar avec tous les réglages par défaut.

## Installer une instance de base de données persistante et un serveur Hsqldb

(Setting up a Hsqldb Persistent Database Instance and a Hsqldb Server)

Si vous avez installé un paquet spécifique à un système d'exploitation, vous devez déjà avoir une instance de base de données et un serveur pré-configuré. Vérifiez que votre package comprenne un fichier nommé `server.properties` (utilisez votre gestionnaire de fichiers). **If you do, then I suggest that you still read this section while you poke around, in order to understand your setup.**

1. Connectez vous à la base de données sous un nom d'utilisateur UNIX. Si cette base de données est destinée à de multiples utilisateurs, ou est un système de production (ou l'émulation d'un système de production), vous devez dédier un utilisateur UNIX à cette fin. Dans les exemples, le nom d'utilisateur est `hsqldb`. Dans ce chapitre, il est fait référence à cet utilisateur comme le `HSQLDB_OWNER`, puisque cet utilisateur possèdera l'instance des fichiers et processus de la base de données.

Si le compte n'existe pas, créez le. Sur tous les System V UNIX et le plupart des systèmes hybrides (incluant Linux), vous pouvez, en tant qu'administrateur, exécuter quelque chose comme :

```
useradd -c 'HSQLDB Database Owner' -s /bin/bash -m hsqldb
```

(Les utilisateurs de variantes BSD peuvent utiliser une commande similaire `pw useradd hsqldb...`).

2. Devenez le `HSQLDB_OWNER`. Copiez le fichier exemple `HSQLDB_HOME/src/org/hsqldb/sample/sample-server.properties` dans le dossier racine `HSQLDB_OWNER` et renommez le en `server.properties`.

```
# Fichier de configuration Hsqldb Server.
# Voir le chapitre Considérations avancées du guide HSQLDB.

server.database.0 file:db0/db0
# Je suggère que, pour chaque file: database définie, vous ajoutiez la
# propriété de connexion "ifexists=true" après que l'instance de base de
# données soit créée (ce qui arrive au premier démarrage du serveur).
# Ajoutez ";ifexists=true" à l'URL file: comme ceci :
# server.database.0 file:db0/db0;ifexists=true
```

Puisque la valeur de la première base de données (server.database.0) commence par file:, l'instance de la base de données sera répandue (persisted) à un ensemble de fichiers dans le dossier défini et dont les noms commencent par le nom spécifié. Définissez le chemin ou vous voulez (les chemins relatifs font référence au dossier contenant le fichier .properties). Pour voir comment spécifier d'autres instances de bases de données de types divers, comment ajuster les réglages pour le port d'écoute et bien d'autres choses, veuillez lire le chapitre Considérations avancées.

3. Définissez et exportez la variable d'environnement CLASSPATH conformément à la valeur de HSQldb\_HOME (comme décrit plus haut) plus "/lib/hsqldb.jar", comme :

```
export CLASSPATH; CLASSPATH=/path/to/hsqldb/lib/hsqldb.jar
```

Dans le dossier racine HSQldb\_OWNER, exécutez :

```
nohup java org.hsqldb.Server &
```

Ceci démarrera le processus Serveur en tâche de fond, et créera votre nouvelle instance de base de données "db0". Continuez quand vous voyez le message affichant HSQldb server... is online. nohup. Soyez seulement certain que la commande ne quittera pas quand vous sortirez du shell courant. (à moins que ce ne soit ce que vous vouliez faire).

## Accès à la base de données

Copiez le fichier HSQldb\_HOME/src/org/hsqldb/sample/sqltool.rc dans le dossier racine de HSQldb\_OWNER. Utilisez chmod pour rendre ce fichier accessible en lecture / écriture seulement pour HSQldb\_OWNER.

```
# $Id: sqltool.rc,v 1.22 2007/08/09 03:22:21 unsaved Exp $
# This is a sample RC configuration file used by SqlTool, DatabaseManager,
# and any other program that uses the org.hsqldb.util.RCData class.
# You can run SqlTool right now by copying this file to your home directory
# and running
#   java -jar /path/to/hsqldb.jar mem
# This will access the first urlid definition below in order to use a
# personal Memory-Only database.
# "url" values may, of course, contain JDBC connection properties, delimited
# with semicolons.
# If you have the least concerns about security, then secure access to
# your RC file.
# See the documentation for SqlTool for various ways to use this file.
```

```
# A personal Memory-Only (non-persistent) database.
urlid mem
url jdbc:hsqldb:mem:memdbid
```

[https://wiki.openoffice.org/wiki/FR/Documentation/HSQLDB\\_Guide/ch03](https://wiki.openoffice.org/wiki/FR/Documentation/HSQLDB_Guide/ch03)

```

#password hiddenpwd
#driver com.mysql.jdbc.Driver
# Note that "databases" in SQL Server and Sybase are traditionally used for
# the same purpose as "schemas" with more SQL-compliant databases.
-----
# Template for a Microsoft SQL Server database
#urlid msprojsvr
#url jdbc:microsoft:sqlserver://hostname;DatabaseName=DbName;SelectMethod=Cursor
# The SelectMethod setting is required to do more than one thing on a JDBC
# session (I guess Microsoft thought nobody would really use Java for
# anything other than a "hello world" program).
# This is for Microsoft's SQL Server 2000 driver (requires mssqlserver.jar
# and msutil.jar).
#driver com.microsoft.jdbc.sqlserver.SQLServerDriver
#username myuser
#password hiddenpwd
-----
# Template for a Sybase database
#urlid sybase
#url jdbc:sybase:Tds:hostname:4100/dbname
#username blaine
#password hiddenpwd
# This is for the jConnect driver (requires jconn3.jar).
#driver com.sybase.jdbc3.jdbc.SybDriver
-----
# Template for Embedded Derby / Java DB.
#urlid derby1
#url jdbc:derby:path/to/derby/directory;create=true
#username ${user.name}
#password any_noauthbydefault
#driver org.apache.derby.jdbc.EmbeddedDriver
# The embedded Derby driver requires derby.jar.
# There's also the org.apache.derby.jdbc.ClientDriver driver with URL
# like jdbc:derby://<server>[:<port>]/databaseName, which requires
# derbyclient.jar.
# You can use \= to commit, since the Derby team decided (why???)
# not to implement the SQL standard statement "commit"!!
# Note that SqlTool can not shut down an embedded Derby database properly,
# since that requires an additional SQL connection just for that purpose.
# However, I've never lost data by not shutting it down properly.
# Other than not supporting this quirk of Derby, SqlTool is miles ahead of ij.
-----

```

Nous utiliserons l'exemple de définition urlid "localhost-sa" du fichier de configuration. L'URL JDBC de cet urlid est `jdbc:hsqldb:hsqldb://localhost`. Ceci est l'URL pour l'instance de base de données par défaut d'un serveur HSQLDB exécuté sur le port par défaut de l'hôte local. Consultez le chapitre "Considérations avancées" pour en savoir plus sur les URLs de connexion à d'autres instances et d'autres serveurs.

Exécuter `SqlTool`.

```
java -jar path/to/hsqldb.jar localhost-sa
```

Si vous obtenez un prompt, tout va bien. Si la sécurité vous concerne un tant soit peu, (If security is of any concern to you at all,) vous devez changer le mot de passe privilégié dans la base de données. Utilisez la commande `SET`



PASSWORD pour changer le mot de passe de l'administrateur système (SA's password).

```
set password "NouveauMotDePasse";
```

Quand vous avez fini, sortez avec la commande \q.

Si vous avez changé le mot de passe de l'administrateur système, vous devez le changer dans le fichier sqltool.rc accordément.

Vous pouvez bien sûr également accéder à la base de données avec un programme client JDBC. Voyez l'annexe B "First JDBC Client Example (<http://hsqldb.org/doc/guide/apb.html>)". Vous aurez besoin de modifier votre chemin de classe pour inclure hsqldb.jar et la / les classe(s) client. Vous pouvez également utiliser d'autres programmes client HSQLDB, comme par exemple `org.hsqldb.util.DatabasesManagerSwing`, un client doté d'une interface graphique et d'un but similaire à `SqlTool`.

Vous pouvez exécuter les clients JDBC depuis n'importe quel compte normal d'utilisateur UNIX, incluant `sqlTool`, tant que le compte a un accès en lecture aux fichiers `hsqldb.jar` et `sqltool.rc`. Voyez le chapitre `SqlTool` pour savoir où mettre `sqltool.rc`, comment exécuter les fichiers SQL, et d'autres fonctionnalités de `SqlTool`.

## Créer des comptes additionnels

Connectez vous à la base de données comme administrateur du système (ou un compte d'utilisateur avec des droits admin.) et utilisez la commande `CREATE USER` pour créer de nouveaux comptes pour votre instance de base de données. Les comptes HSQLDB sont spécifiques à une instance de base de données et non au serveur.

Pour la version en cours de HSQLDB, seulement les utilisateurs ayant le rôle de DBA (d'administrateur de base de données) peuvent créer ou posséder leurs propres objets de base de données. Les membres du groupe DBA ont tous les privilèges. Les non-membres du groupe DBA peuvent se voir accorder des privilèges, mais ne pourront jamais créer ou posséder leurs propres objets de base de données. (Prochainement, les membres non-DBA pourront créer des objets s'ils en ont la permission dans le schéma cible). Quand vous créez une base de données HSQLDB, elle ne compte qu'un utilisateur-- SA (Administrateur Système), un compte DBA, muni d'une chaîne vide pour le mot de passe. Vous pouvez définir un mot de passe (comme décrit plus haut). Vous pouvez créer autant d'autres utilisateurs que vous voulez. Pour faire d'un utilisateur un membre DBA, vous pouvez ajouter l'option "ADMIN" à la commande `CREATE USER`, ou accorder (GRANT) le rôle DBA au compte après l'avoir créé.

Si vous créez un utilisateur ni avec l'option ADMIN ni en lui accordant le rôle

DBA, il pourra lire les tables de données du dictionnaire (**data dictionary tables**), mais sera incapable de créer ou posséder ses propres objets. Il n'aura les droits que du pseudo-utilisateur PUBLIC. Pour lui permettre plus de choses, même les droits pour lire des objets, vous pouvez lui accorder (GRANT) des permissions pour des objets spécifiques, lui accorder des rôles (qui englobent un jeu de permissions), ou lui accorder le rôle DBA lui-même.

Bien que seulement les personnes avec un compte de base de données puissent faire quoi que ce soit avec cette dernière, il est souvent pratique de permettre à des utilisateurs d'autres bases de données de voir les données dans vos tables. Pour optimiser la performance, réduire la contention, et minimiser l'administration, il est souvent meilleur d'attribuer la commande SELECT au groupe PUBLIC sur chaque objets nécessitant un accès public (à l'exception significative de toutes données que vous voudriez garder secrètes).

## Shutdown

Fermez proprement votre base de données lorsque vous en avez fini avec son instance. Vous devez vous connecter comme administrateur système ou autre utilisateur avec des droits d'administrateur, bien sur. Avec SqlTool, vous pouvez taper la ligne de commande suivante :

```
java -jar path/to/hsqldb.jar --sql shutdown localhost-sa
```

Vous n'avez pas besoin de vous préoccuper de l'arrêt du serveur car il le fait automatiquement quand toutes les instances de bases de données servies sont fermées.

## Exécuter Hsqldb comme un démon du système

(Running Hsqldb as a System Daemon)

Vous pouvez bien sur exécuter HSQLDB par inittab sur les distributions UNIX System V, mais habituellement un script init est plus pratique et gérable. Cette section explique comment définir et utiliser nos scripts init UNIX. Nos scripts init sont réservés à l'administrateur. (Ce qui ne veut pas dire que le **Server** s'exécutera comme root-- ce n'est généralement pas le cas).

Le but principal de ce script init est de démarrer un serveur muni des instances de bases de données spécifiées dans votre fichier `server.properties` ; et d'en fermer toutes ces instances et les **urlids** (identifiants URL) que vous devriez (c'est une option) lister dans votre fichier de config du script init. (in your init script config file). Ces **urlids** doivent tous avoir leur entrée dans un fichier `sqltool.rc`. (**These urlids must all have entries in a sqltool.rc file.**). Si, à cause d'un pare-feu, vous voulez exécuter une serveur Web plutôt qu'un serveur, soyez certain du bon état de santé du serveur Web par les réglages de `webserver.properties`, vérifiez vos URLs dans le fichier `sqltool.rc`, et ajustez

TARGET\_CLASS dans le fichier de configuration. (En suivant les exemples commentés dans le fichier de configuration, vous pourrez démarrer un nombre quelconque de listeners de serveurs et/ou de serveurs Web, avec ou sans encryption TLS) - TLS = Transport Layer Security ou couche de sécurité du transport ([http://fr.wikipedia.org/wiki/Transport\\_Layer\\_Security](http://fr.wikipedia.org/wiki/Transport_Layer_Security)).

Après avoir effectué les réglages du script init, l'administrateur pourra quand il voudra l'utiliser pour démarrer et arrêter HSQLDB. (Par ex. pas seulement au démarrage et à l'arrêt du système).

## Portabilité du script init de hsqldb

Le premier critère de design du script init est sa portabilité. Il n'affiche pas de jolis messages en couleurs de démarrage et d'arrêt comme il est commun dans les derniers modèles de distributions Linux et HP-UX (<http://fr.wikipedia.org/wiki/HP-UX>) ; et il ne garde pas de fichiers d'état du sous-système ni n'utilise de fonctions de Démarrage/Arrêt fournies par nombre de distributions, Parce que ces fonctionnalités ne sont pas toutes portables.

Mis à part (**Offsetting**) ces limitations, ce script fait bien son travail sur une variété de plates-formes UNIX testées, et peut facilement être modifié pour s'accommoder d'autres distributions. Bien que vous n'ayez pas d'intégration poussée avec les démons spécifiques au système d'exploitation, les interfaces graphiques d'administrateur, etc., vous avez là un script bien testé et se comportant bien qui donne de bons retours d'utilisateurs. (**you do have a well tested and well behaved script that gives good, utilitarian feedback.**)

## Procédure d'installation du script init

Titre original : *Init script Setup Procedure* (<http://hsqldb.org/doc/guide/ch03.html#N10780>)

La stratégie visée ici est d'obtenir un script init exécutant tout d'abord votre simple serveur ou serveur Web (comme spécifié par TARGET\_CLASS). Une fois ce fonctionnement obtenu, vous pouvez personnaliser la JVM active en y intégrant des serveurs additionnels, vos propres applications (encapsulation), (**embedding**) ou même outrepasser le comportement de HSQLDB avec vos propres classes prépondérantes. (**or even overriding HSQLDB behavior with your own overriding classes.**)



### 1.

Copiez le script init hsqldb du dossier HSQLDB\_HOME/bin vers celui où résident vos scripts sur votre distribution de UNIX. Les chemins d'accès les plus communs sont /etc/init.d ou /etc/rc.d/init.d sur des versions type UNIX System V, /usr/local/etc/rc.d pour les

versions UNIX BSD, et /Library/StartupItems/hsqldb sur OS X (Vous devez créer le répertoire pour ce dernier).

## i

### 2.

Regardez le commentaire tout en haut du script init qui liste les chemins d'accès recommandés pour le fichier de configuration sur diverses plates-formes UNIX. Copiez le fichier de configuration exemple HSQLDB\_HOME/src/org/hsqldb/sample/sample-hsqldb.cfg dans un des chemins de la liste (de votre choix). Éditez le fichier de configuration conformément aux instructions qu'il contient.

```
# $Id: sample-hsqldb.cfg,v 1.16 2005/07/24 18:33:13 unsaved Exp $

# Sample configuration file for HSQLDB database server.
# See the "UNIX Quick Start" chapter of the Hsqldb User Guide.

# N.b.!!!! You must place this in the right location for your type of UNIX.
# See the init script "hsqldb" to see where this must be placed and
# what it should be renamed to.

# This file is "sourced" by a Bourne shell, so use Bourne shell syntax.

# This file WILL NOT WORK until you set (at least) the non-commented
# variables to the appropriate values for your system.
# Life will be easier if you avoid all filepaths with spaces or any other
# funny characters. Don't ask for support if you ignore this advice.

# Thanks to Meikel Bisping for his contributions. -- Blaine

JAVA_EXECUTABLE=/usr/bin/java

# Unless you copied a hsqldb.jar file from another system, this typically
# resides at $HSQLDB_HOME/lib/hsqldb.jar, where $HSQLDB_HOME is your HSQLDB
# software base directory.
HSQLDB_JAR_PATH=/opt/hsqldb/lib/hsqldb.jar

# Where the file "server.properties" resides.
SERVER_HOME=/opt/hsqldb/data

# What UNIX user the server will run as.
# (The shutdown client is always run as root or the invoker of the init script).
# Runs as root by default, but you should take the time to set database file
# ownerships to another user and set that user name here.
HSQLDB_OWNER=hsqldb

# The HSQLDB jar file specified in HSQLDB_JAR_PATH above will automatically
# be in the class path. This arg specifies additional classpath elements.
# To embed your own application, add your jar file(s) or class base
# directories here, and add your main class to the INVOC_ADDL_ARGS setting
# below.
#SERVER_ADDL_CLASSPATH=/usr/local/dist/currencybank.jar

# We require all Server/WebServer instances to be accessible within
# $MAX_START_SECS from when the Server/WebServer is started.
# Defaults to 60.
# Raise this is you are running lots of DB instances or have a slow server.
MAX_START_SECS=200
```

```

# Time to allow for JVM to die after all HSQLDB instances stopped.
# Defaults to 1.
#MAX_TERMINATE_SECS=0

# These are "urlid" values from a SqlTool authentication file
# ** IN ADDITION TO THOSE IN YOUR server.properties OR webserver.properties **
# file. All server.urlid.X values from your properties file will automatically
# be started/stopped/tested. $SHUTDOWN_URLIDS is for additional urlids which
# will stopped. (Therefore, most users will not set this at all).
# Separate multiple values with white space. NO OTHER SPECIAL CHARACTERS!
# Make sure to quote the entire value if it contains white space separator(s).
# Defaults to none (i.e., only urlids set in properties file will be stopped).
#SHUTDOWN_URLIDS='sa mygms'

# SqlTool authentication file used only for shutdown.
# The default value will be sqltool.rc in root's home directory, since it is
# root who runs the init script.
# (See the SqlTool chapter of the HSQLDB User Guide if you don't understand
# this).
#AUTH_FILE=/home/blaine/sqltool.rc

# Set this to either 'WebServer' or 'Server'. Defaults to Server.
# The JVM that is started can invoke many classes (see the following item
# about that), but this is the Server that is used (1) to check status,
# (2) to shut down the JVM, (3) to get urlids for #1 from the
# server's server/webserver.properties file.
#TARGET_CLASS=WebServer
# Note that you don't specify the org.hsqldb package, since you have no
# choice in the matter (you can only run org.hsqldb.Server or
# org.hsqldb.WebServer). If you specify additional classes with
# INVOC_ADDL_ARGS (described next), you do need to specify the
# full class name with package name.

# This is where you specify exactly what your HSQLDB JVM will run.
# The class org.hsqldb.util.MainInvoker will run the TARGET_CLASS
# specified above with any arguments supplied here + any other classes
# and arguments. Every additional class (in addition to the TARGET_CLASS)
# must be preceded with an empty string, so that MainInvoker will know
# you are giving a class name. MainInvoker will invoke the normal
# static main(String[]) method of each such class.
# By default, MainInvoker will just run TARGET_CLASS with no args.
# Example that runs just the TARGET_CLASS with the specified arguments:
#INVOC_ADDL_ARGS='-silent false'
# Example that runs the TARGET_CLASS plus a WebServer:
#INVOC_ADDL_ARGS='' org.hsqldb.WebServer'
# Note the empty string preceding the class name.
# Example that starts TARGET_CLASS with an argument + a WebServer +
# your own application with its args (i.e., the HSQLDB Servers are
# "embedded" in your application). (Set SERVER_ADDL_CLASSPATH too).:
#INVOC_ADDL_ARGS='-silent false "" org.hsqldb.WebServer "" com.acme.Stone --env'
# Example to run a non-TLS server in same JVM with a TLS server. In this
# case, TARGET_CLASS is Server which will run in TLS mode by virtue of
# setting TLS_KEYSTORE and TLS_PASSWORD above. The "additional" Server
# here overrides the 'tls' and 'port' settings:
#INVOC_ADDL_ARGS='' org.hsqldb.Server -port 9002 -tls false"
# Note that you use nested quotes to group arguments and to specify the
# empty-string delimiter.

# For TLS encryption for your Server, set these two variables.
# N.b.: If you set these, then make this file unreadable to non-root users!!!!
# See the TLS chapter of the HSQLDB User Guide, paying attention to the
# security warning(s).
# If you are running with a private server cert, then you will also need to
# set "truststore" in the your SqlTool config file (location is set by the
# AUTH_FILE variable in this file, or it must be at the default location for
# HSQLDB_OWNER).
#TLS_KEYSTORE=/path/to/jks/server.store
#TLS_PASSWORD=password

# Any JVM args for the invocation of the JDBC client used to verify DB

```

```
# instances and to shut them down (SqlToolSprayer).  
# This example specifies the location of a private trust store for TLS  
# encryption.  
# For multiple args, put quotes around entire value.  
#CLIENT_JVMARGS=-Djavax.net.debug=ssl  
  
# Any JVM args for the server.  
# For multiple args, put quotes around entire value.  
#SERVER_JVMARGS=-Xmx512m
```

### i 3.

Soit vous copiez le fichier `sqltool.rc` du dossier `HSQLDB_OWNER` à la racine de votre dossier (into root's home directory), où vous renseignez la valeur de `AUTH_FILE` par le chemin absolu du fichier `sqltool.rc` du dossier `HSQLDB_OWNER`. Ce fichier est lu (**for stops**) directement par root (l'administrateur), même si vous ne lencez pas `hsqldb` en tant qu'administrateur (par le réglage de `HSQLDB_OWNER` dans le fichier de configuration). Si vous faites une copie de ce fichier, prenez garde et utilisez `chmod` pour restreindre les permissions de la nouvelle copie. (Le script `init` renforce (enforces) maintenant les permissions sur ce fichier).

### i 4.

Éditez votre fichier `server.properties`. Pour chaque `server.database.X` définie, établissez une propriété de nom `server.urlid.X` à l'`urlid` pour un utilisateur ayant des droits d'administrateur pour cette instance de la base de données.

#### Exemple 3.1. **server.properties** fragment

```
server.database.0=file://home/hsqldb/data/db1  
server.urlid.0=localhostdb1
```



#### **Attention !**

Soyez certain d'ajouter une `urlid` pour chacune des instances de bases de données. Dans le cas contraire le script `init` ne connaîtra jamais les bases de données devenues inaccessibles et retournera des diagnostics faux.

Pour cet exemple, vous devrez définir l'`urlid` `localhostdb1` dans

votre fichier `sqltool.rc`.

### Exemple 3.2. Exemple `sqltool.rc` stanza

```
urlid localhostdb1
url jdbc:hsqldb:hsqldb://localhost
username sa
password secret
```

## i

### 5. Vérifiez que le script `init` fonctionne.

Exécutez :

```
/path/to/hsqldb
```

en tant qu'administrateur afin de voir les arguments que vous pouvez utiliser. Notez que vous pouvez exécuter

```
/path/to/hsqldb status
```

à tout moment pour vérifier que votre serveur HSQLDB s'exécute.

Réexécutez le script avec chacun des arguments possibles pour réellement bien le tester. Si quelque chose ne fonctionne pas correctement, veuillez vous reporter à la section Dépanner le script `init`.

## i

### 6.

Faites que votre script `init` s'exécute au démarrage et à l'arrêt du système. Si vous êtes sur une variante UNIX contenant `/etc/rc.conf` ou `/etc/rc.conf.local` (comme les variantes BSD ou Gentoo), vous devez spécifier `"hsqldb_enable"` à `"YES"` dans l'un ou l'autre de ces fichiers. (exécutez `cd /etc; ls rc.conf rc.conf.local` pour voir si vous avez un de ces fichiers). Pour les bons vieux systèmes UNIX qui utilisent le style d'initialisation System V (For good UNIXes that use System V style init), vous devez définir soit des liens en dur ou de simples liens soit manuellement ou avec des outils de gestion (tels `chkconfig` ou `insserv`), ou des interfaces graphiques (comme `run level editors`).

Ce paragraphe est réservé aux utilisateurs de Mac OS X. Si vous

avez suivi les instructions ci-dessus, votre script init doit figurer dans `/Library/StartupItems/hsqldb/hsqldb`. Copiez maintenant le fichier `StartupParameters.plist` du répertoire `src/org.hsqldb/sample` de votre distribution HSQLDB vers le répertoire contenant le script init. Aussi longtemps que ces deux fichiers résident dans le dossier `/Library/StartupItems/hsqldb`, votre script init est actif (pour des raisons de portabilité, il ne tient pas compte d'un réglage dans le dossier `/etc/hostconfig`). Vous pouvez l'exécuter au démarrage en tapant

```
-----  
SystemStarter {start|stop|restart} Hsqldb  
-----
```

Hsqldb est le nom du service. Lisez la page `man SystemStarter`. Pour désactiver le script init, effacez le du dossier `/Library/StartupItems/hsqldb`. Difficile à croire, mais les utilisateurs de Mac m'ont rapporté que pendant l'arrêt du système les icônes de démarrage (Startup Items) ne fonctionnaient pas du tout. Par le fait, si vous ne voulez pas que vos données soient corrompues, soyez certains d'exécuter `"SystemStarter stop Hsqldb"` avant d'éteindre votre Mac.

---

Suivez les exemples dans le fichier de configuration pour ajouter des classes additionnelles au serveur classpath de la JVM et pour exécuter des classes additionnelles dans votre JVM. (Voyez les items `SERVER_ADDL_CLASSPATH` et `INVOC_ADDL_ARGS`).

## Dépannage du script Init

(Troubleshooting the Init Script)

Recherchez les processus contenant la chaîne `hsqldb`, et essayez de vous connecter à la base de données depuis chaque client. Si le script init démarre votre base de données avec succès, mais reporte incorrectement qu'il ne l'a pas fait, c'est que vous avez un problème soit avec la spécification de(s) `urlid(s)` ou avec l'installation de `SqlTool`. Si votre base de données ne démarre vraiment pas, sautez au paragraphe suivant. Vérifiez que le(s) `urlid(s)` listé(s) dans les fichiers `server.properties` ou `webserver.properties` sont corrects. Et vérifiez que vous pouvez exécuter `SqlTool` en tant qu'administrateur pour vous connecter aux instances. (Pour ce dernier test, utilisez l'interrupteur `--rcfile` pour régler `AUTH_FILE` dans le fichier de configuration du script init).

Si votre base de données ne démarre toujours pas, alors vérifiez que vous pouvez accéder au compte du propriétaire de cette base et la démarrer. La commande `su NOM_UTILISATEUR -c ...` ne fonctionnera pas sur la plupart des systèmes UNIX à moins que l'utilisateur concerné n'ait un vrai shell de connexion. (unless the target user has a real login shell). Par conséquent, si



vous essayez de resserrer la sécurité en désactivant le shell de connexion de cet utilisateur, vous "casserez" le script init. Si ces solutions envisagées ne réussissent pas, alors déboguez le script init ou parcourez l'aide, comme décrit ci-dessous.

Pour déboguer le script init, exécutez le en mode verbose pour voir exactement ce qui se passe (et peut-être en pas à pas pour les commandes suspectes). Pour exécuter un script init (en fait, n'importe quel script shell sh) en mode verbose, utilisez sh avec l'interrupteur -x ou -v, par exemple :

```
sh -x path/to/hsqldb start
```

Lisez la page man de sh si vous ne connaissez pas la différence entre -v et -x.

Si vous voulez une aide au dépannage, utilisez les listes/forums de HSQLDB ou envoyez moi un e-mail à blaine dot simpson [at] admc dot com. Si vous m'envoyez ce mail, soyez sur de joindre le numéro de révision de votre script init hsqldb (il est tout en haut dans la ligne commençant par "# \$Id:"), et le résultat de la commande suivante :

```
sh -x path/to/hsqldb start > /tmp/hstart.log 2>&1
```

Récupérée de « [https://wiki.openoffice.org/w/index.php?title=FR/Documentation/HSQLDB\\_Guide/ch03&oldid=240618](https://wiki.openoffice.org/w/index.php?title=FR/Documentation/HSQLDB_Guide/ch03&oldid=240618) »

Catégorie : FR/HSQLDB Guide

- 
- Dernière modification de cette page le 6 juillet 2018 à 20:44.
  - Content is available under ALv2 unless otherwise noted.