7.0 Zero Footprint HSQLDB

Has HSQLDB been superceded by PostGreSQL? if yes, should we delete this page? HL.

I am not entirely happy with the stop and start scripts and also not done picking apart the example provided by hsqldb and explaning the reasoning behind our own custom script.

Introduction

Explain what HSQLDB is about.

Setup

To fill in, though it's pretty simple.

Starting

If the admin password has changed, the /opt/hsqldb/sqltool.rc must be edited.

The password in the sqltool.rc file is unencrpyted. This password should be encrypted, the file protected, or the script modified to prompt for the password. We still need to come up with an optimal solution.

Script,

```
#!/bin/bash
cd /opt/hsqldb

# Add date to nohup.out file which is automatically created by the nohup
command to capture any output from commands sent to it
echo $(date) >nohup.out

# Assuming that java is the the class path.
# Execute Java program calling class path where -cp identifies classpath
and,
# Creates a database file called BonsaiResourceBundleDatabase if it does
not exist relative to the current directory
# -dbname.0 designates that the database will be called
BonsaiResourceBundleDatabase
# Sends Java program to background and to not terminate via nohup command.
nohup java -cp lib/hsqldb.jar org.hsqldb.Server -database.0
file:data/BonsaiResourceBundleDatabase -dbname.0
BonsaiResourceBundleDatabase &

# Without this sleep, the script will require that the user hit enter to
return back to the command line
sleep 1
echo "hsqldb started"
```

From the HSQLDB manual it provides this example,

```
# Make an environment variable called CLASSPATH
export CLASSPATH
# Set CLASSPATH to include the hsqldb.jar file. Java will look at CLASSPATH
list to know what to execute.
CLASSPATH=/path/to/hsqldb/lib/hsqldb.jar
# Executes Java program calling class and sends to background.
nohup java org.hsqldb.Server &
```

Stopping

Script,

```
cd /opt/hsqldb
java -jar lib/hsqldb.jar --sql "shutdown;" --rcFile ./sqltool.rc
localhost-sa
```

versus HSQLDB manual.

Backup

Start from really simple, using tar, to using script and then advanced using replication.

Adding Users

There is a need to have multiple users for administration of the HSQL database.

The administrator account is "SA" and has all privileges including create, read and write.

The staff users have read and write access but can not alter tables.

Finally an application account needs to be created.

This section needs more work. Consider moving the discussion on database accounts into a general area.

To add a user, use the following syntax,

```
CREATE USER sheeley password "<password>"
GRANT SELECT ON APP_COMM TO sheeley;
GRANT INSERT ON APP_COMM TO sheeley;
GRANT UPDATE ON APP_COMM TO sheeley;
GRANT SELECT ON BUNDLE TO sheeley;
GRANT INSERT ON BUNDLE TO sheeley;
GRANT UPDATE ON BUNDLE TO sheeley;
GRANT SELECT ON RESOURCE TO sheeley;
GRANT INSERT ON RESOURCE TO sheeley;
GRANT UPDATE ON RESOURCE TO sheeley;
```

Changing Password

The syntax command for changing the password

```
SET PASSWORD password ;
```

Changes the password of the currently connected user. Empty password can be set using ""

In the ursis appication, the location that the properties file is inside a jar file located in
/opt/apache/tomcat/webapps/ursis/WEB-INF/lib/BrandingEngine.jar

Inside the BrandingEngine.jar is branding_config.properties, which sets the path of the properties file.
BrandingPropertiesLocation=/opt/apps/BrandingEngine/resources/T6/branding_env.properties

For the account that general users use, set the username and password in the branding_env.properties file. The LDAP browser also relies on these credentials so you must edit the following file.

/opt/apps/ums/resources/T5/ldapProperties.properties (which is symbolically linked)

The password encryption tool is in source control C:\SVN\tools\gtb-encryption\encryption.bat

Resources

http://hsqldb.org/doc/2.0/guide/unix-chapt.html - start here and simplify these instructions.