Using ODBC in the core

About

FreeSWITCH can be configured to use ODBC to connect to a remote database instead of using the default SQLite databases.

MySQL is not supported natively, you still need ODBC and this how-to (No native Support of MySQL is planned for the close future)

Setup unixODBC

To use unixODBC you will need to install:

- unixODBC >= 2.3
- An ODBC driver for your chosen database.

Several popular databases are supported including MySQL, PostgreSQL and others.

Ubuntu

Before you begin you will need to install and configure your chosen database ensuring you’ve set the correct permissions for access.

For additional help configuring your database, see:

- PostgreSQL on Ubuntu
- MySQL on Ubuntu

Note - the following configuration has been tested on Ubuntu 10.04 LTS Server (64-bit) with PostgreSQL 8.4/9.1 and MySQL 5.1.

Install unixODBC

```
sudo apt-get install unixodbc-dev
```

Install the unixODBC driver

You will need to obtain the ODBC driver for your chosen database. Instructions for obtaining the drivers for MySQL or PostgreSQL are provided below.

For MySQL:

```
sudo apt-get install libmyodbc
```

For PostgreSQL

```
sudo apt-get install odbc-postgresql
```

Configure odbc.ini and odbcinst.ini

Add the following with the correct information into your odbc.ini file located at /etc/odbc.ini

```
/etc/odbc.ini for MySQL
```
If you are running on a 32-bit Linux installation, use the following configuration, but change all occurrences of `/usr/lib64/` to `/usr/lib/` in the examples provided.

[freeswitch]
Driver = /usr/lib64/odbc/libmyodbc.so
SERVER = localhost
PORT = 3306
DATABASE = freeswitch
OPTION = 67108864
USER = root
PASSWORD = password

/etc/odbcinst.ini for MySQL

[MySQL]
Description = MySQL driver
Driver = /usr/lib64/odbc/libmyodbc.so
Setup = /usr/lib64/odbc/libodbcmyS.so
UsageCount = 1
FileVersion = 1
Threading = 0

If it is available on your system, you must use Driver = /usr/lib64/odbc/libmyodbc.so in your odbcinst.ini.

Also note, (see bug below on bottom of page): If you are using the FreeSWITCH pooled resources, then you definitely need Threading = 0 for high call volume. What this does is it "disables a global mutex in the ODBC core that only allows 1 concurrent operation per process regardless of the concurrent connections." (i.e. It slows down when you have too many connections trying at one time)

As far as has been tested, when making changes to the odbcinst.ini, you do not need to restart FreeSWITCH to effect the changes.

/etc/odbc.ini for PostgreSQL

Important If you are running on a 32-bit Linux installation, use the following configuration, but change all references to `/usr/lib64/` to `/usr/lib/` in the examples provided.

[freeswitch]
; WARNING: The old psql odbc driver psqlodbc.so is now renamed psqlodbcw.so
; in version 08.x. Note that the library can also be installed under an other
; path than /usr/local/lib/ following your installation.
Driver = /usr/lib64/odbc/psqlodbcw.so
Description=Connection to LDAP/POSTGRESQL
Servername=localhost
Port=5432
Protocol=6.4
FetchBufferSize=99
Username=postgres
Password=password
Database=freeswitch
ReadOnly=no
Debug=0
CommLog=0

/etc/odbcinst.ini for PostgreSQL

[PostgreSQL]
Description = PostgreSQL driver for Linux & Win32
Driver = /usr/lib64/odbc/psqlodbcw.so

Testing your connection

Once you've set up the .ini files and have your ODBC drivers installed, you can test whether your ODBC connection is working correctly by entering:
isql -v freeswitch

If a connection is successfully established you should see something similar to:

```
+---------------------------------------+
| Connected!                            |
| sql-statement                         |
| help [tablename]                      |
| quit                                  |
+---------------------------------------+
```

SQL>

Compile Freeswitch and Set Modules to Use ODBC

You will need to compile/recompile Freeswitch so that ODBC is detected and included within the installation. You will then need to edit the freeswitch configuration to utilize the ODBC connection you just defined.

See Compile FreeSWITCH with ODBC Support below for more information.

PostgreSQL 8.4 on Debian Squeeze

Note: We've experienced some major stability issues with the unixODBC-debian-package 2.2.14p2-1 from the repository. FreeSWITCH would die anywhere between a few days to weeks. Upgrading to a manually compiled unixODBC 2.3.0-pre fixed this problem and has been tested stable for the last 2 months. --[[User:Peletiah|Peletiah]] 13:04, 4 February 2012 (UTC)

Install PostgreSQL and ODBC

```
apt-get install postgresql unixodbc odbc-postgresql
```

You also have to install "unixodbc-dev", which by default depends on libodbinstq1c2 which would install x11-common and other packages for GUI-Support. To avoid this, you have to create a fake libodbinstq1c2-package, see instructions below. If you don't care, do "apt-get install unixodbc-dev" and continue with odbc-configuration.

Fake libodbinstq1c2-package with equivs

If you don't want to install X11 on a headless server, create a fake-libodbinstq1c2-package with equivs. Please read the documentation at http://www.debian.org/doc/manuals/apt-howto/ch-helpers.en.html to make sure you don't damage your system.

```
install equivs:
apt-get install equivs
cd /tmp

equivs-control libodbinstq1c2

edit the control-file so it looks like this:

Section: libs
Priority: optional
# Homepage: <enter URL here; no default>
Standards-Version: 3.6.2

Package: libodbinstq1c2
Version: 2.2.14p2-1
Description: Qt-based ODBC configuration library
This package contains the libodbinstq library, a library used by some Qt-based GUI configuration tools for managing ODBC drivers and ODBC DSNs. NOT NEEDED FOR UNIXODBC-DEV
```
build the fake-package:

```bash
equivs-build libodbcinstq1c2
install the package that has been created:
dpkg -i libodbcinstq1c2_2.2.14p2-1_all.deb
and finally, install unixodbc-dev without X11 :-) 
apt-get install unixodbc-dev
```

**odbc-configuration**

Create `/etc/odbcinst.ini` by command

```bash
odbcinst -i -d -f /usr/share/psqlodbc/odbcinst.ini.template
```

Create `/etc/odbc.ini` Like this

```ini
[freeswitch]
Description = PostgreSQL Unicode
Driver = PostgreSQL Unicode
Trace = No
TraceFile = /tmp/psqlodbc.log
Database = freeswitch
Servername = 127.0.0.1
UserName = 
Password = 
Port = 5432
ReadOnly = Yes
RowVersioning = No
ShowSystemTables = No
ShowOldColumn = No
FakeOldIndex = No
ConnSettings =
```

Edit `/etc/odbcinst.ini`

```ini
[PostgreSQL ANSI]
Description = PostgreSQL ODBC driver (ANSI version)
Driver = /usr/lib/odbc/psqlodbca.so
Setup = /usr/lib/odbc/libodbcpsqlS.so
Debug = 0
CommLog = 0
UsageCount = 0
Threading = 0

[PostgreSQL Unicode]
Description = PostgreSQL ODBC driver (Unicode version)
Driver = /usr/lib/odbc/psqlodbcw.so
Setup = /usr/lib/odbc/libodbcpsqlS.so
Debug = 0
CommLog = 0
UsageCount = 0
Threading = 0
MaxLongVarcharSize=65536
```

In order to not have postgres log everything it does change CommLog to 0

Test the connection with

```bash
isql freeswitch
```

CentOS 5.2
Install unixODBC and the MySQL ODBC Connector

```bash
yum install unixODBC-devel mysql-connector-odbc
```

- Make symlink from `/usr/lib64/libmyodbc3.so` to `/usr/lib64/libmyodbc.so`
- Run `odbcinst -j`. This will show you list of config files.
- Uncomment the MySQL sample driver configuration in `/etc/odbcinst.ini`.

Add the following with the correct information into your `odbc.ini` file located at `/etc/odbc.ini`:

```ini
[freeswitch]
Driver = MySQL
SERVER = localhost
PORT = 3306
DATABASE = myDatabase
OPTION = 67108864
Socket = /var/lib/mysql/mysql.sock
```

**Note:** If you are connecting your freeswitch server to a remote MySQL database, you can take out the last line from the above setting. Take off the "Socket = /var/lib/mysql/mysql.sock" line.

**Note:** OPTION allows you to set client specific FLAGS, in the example 67108864 (FLAG_MULTI_STATEMENTS) is set - See [http://dev.mysql.com/doc/refman/5.0/en/connector-odbc-configuration-connection-parameters.html](http://dev.mysql.com/doc/refman/5.0/en/connector-odbc-configuration-connection-parameters.html) for all flags. The number represents the addition of all flag numbers that you want enabled.

**Note:** On CentOS it is OPTION (without S), instead of OPTIONS.

For some tips on setting up your dsn up in unixODBC see [[Mod_spidermonkey_odbc#unixodbcc]]

- Test your ODBC setup by running the utility `isql`
  ```bash
  isql maxpowersoft_odbc myUser myPass
  ```

FreeBSD

Install the following ports and follow instructions as detailed within the CentOS section

- `/usr/ports/databases/mysql-connector-odbc`
- `/usr/ports/databases/unixODBC`

Go to [[Mod_spidermonkey_odbc#General_Configuration]] for instructions to configure unixODBC

Create the FreeSWITCH Database

1. Create a database in mysql or pgsql (Note: Do NOT use utf8_bin collation! Especially important with IP failover configurations)
2. Create a user
3. Do not worry about creating the actual tables. FreeSWITCH will take care of this part itself if they do not exist.

MySQL Client 4.x and MySQL Server 5.x

If you're getting an error about authentication method not being supported, use old password format in db.
UPDATE mysql.user
SET password=OLD_PASSWORD('somepassword')
WHERE user='someuser'
AND host='somehost';

Compile FreeSWITCH with ODBC support

UnixODBC support will be autodetected by ./configure, and if found, will be compiled into FreeSWITCH.

ODBC DSN Format

See ODBC DSN

Problems

ODBC not available

Message: "mod_sofia: ODBC IS NOT AVAILABLE!" means "FreeSWITCH is compiled without ODBC support."

Message: "[unixODBC][Driver Manager] Data source name not found, and no default driver specified"

Either:

a) Double check that the name of the [dsn] is exactly the same in your config files
- OR -

b) Try linking your ODBC config files to your FreeSWITCH install directory.

Link ini files to FreeSWITCH Directory

```
mkdir /usr/local/freeswitch/etc
ln -s /etc/odbcinst.ini /usr/local/freeswitch/etc/odbcinst.ini
ln -s /etc/odbc.ini /usr/local/freeswitch/etc/odbc.ini
```

Idle in transaction

There's a known issue with PostgreSQL and ODBC where processes hang due to locking

When this happens (calls appear "hanging", FS-console doesn't respond), "ps ax" will show PostgreSQL processes in limbo:

```
root@localhost:~# ps ax|grep postgres|grep free
15753 ? Ss 0:00 postgres: freeswitch_db freeswitch_db 127.0.0.1(43846) idle in transaction
15759 ? Ss 0:00 postgres: freeswitch_db freeswitch_db 127.0.0.1(43847) idle
15760 ? Ss 0:00 postgres: freeswitch_db freeswitch_db 127.0.0.1(43848) UPDATE waiting
```

A known workaround is to add the following parameters to "odbcinst.ini" in the respective driver-sections(ANSI, Unicode - see odbc-configuration above for a full example):

```
Threading = 0
MaxLongVarCharSize=65536
```

See Bug #3693 for additional information