

## Contents

- [1 The Basics](#)
- [2 The Requirements](#)
- [3 The Source](#)
- [4 Build Instructions](#)
- [5 Usage](#)
- [6 Database Schema](#)
- [7 FreeSWITCH Configuration](#)
- [8 Templates](#)
- [9 The Dirty Details](#)
- [10 Warning](#)

## The Basics

This is just another example of mod\_xml\_curl in action, done in C++ rather than a scripted language. In principle it is quite similar to the [Ruby directory example](#).

The program executes without any command-line options and, by default, daemonizes. It simply listens on a specified IP and Port for HTTP requests (typically from mod\_xml\_curl) and responds with the current user directory (as retrieved from a database) in XML format (or HTML, text, etc. format) as laid out in two template files.

All configuration may be done using a configuration file which, currently, is only used for specifying database parameters, listening host/port, and which user to surrender permissions to.

## The Requirements

A Unix system, Qt 4.6+, and a C++ compiler toolchain.

## The Source

<http://osgrid.foxmoxie.net/share/rrcbsdirectory-0.1.tar.bz2> (4.5KB)

## Build Instructions

Run "qmake" in the directory, then "make".

## Usage

Run the resulting "rrcbsdirectory" binary. Find it in your process manager and kill it, (*kill `ps aux | grep rrcbsdirectory | grep -v grep | awk '{print \$2;}'`*) and then edit

## Mod\_xml\_curl\_C++\_Example

### ~/config/RiverRoadCable/DirectoryDaemon.conf

Here's what each key means:

```
[General]
dbtype=QMYSQl                (Or any 'valid' Qt database driver, although the queries are untested)
dbhost=localhost             (Database hostname)
dbbase=sip_users             (Database name)
dbuser=myuser                (Database user, or delete the line for QSQLITE or similar)
dbpass=mysecret              (Database password, or delete the line if no password is required)
tmpl_doc=/etc/fs_document.xml (Path to the 'document body' template)
tmpl_ent=/etc/fs_entry.xml   (Path to the template used for each directory entry)
uid=0                        (After startup, drop root permissions and become this userid.)
gid=0                        (Same as 'uid', but for group id.)
chuser=false                 (If false, we don't do the uid/gid change at all.)
net_host=127.0.0.1           (What IP to listen on.)
net_port=2000                (And which port to listen on.)
```

## Database Schema

The table *sip\_extensions* must be created with the following layout:

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
ext	varchar(32)	NO		NULL	
subid	int(11)	YES		NULL	
callgroup	varchar(32)	YES		NULL	
context	varchar(32)	YES		NULL	
callerid_text	varchar(32)	YES		NULL	
callerid_number	varchar(32)	YES		NULL	
password	varchar(64)	NO		NULL	
vmailpass	varchar(16)	YES		NULL	
active	tinyint(4)	NO		0	

... and the column names have to match, order doesn't matter. The MySQL table creation code is:

```
CREATE TABLE `sip_extensions` (
  `id` int(11) NOT NULL auto_increment,
  `ext` varchar(32) NOT NULL,
  `subid` int(11) default NULL,
  `callgroup` varchar(32) default NULL,
  `context` varchar(32) default NULL,
  `callerid_text` varchar(32) default NULL,
  `callerid_number` varchar(32) default NULL,
  `password` varchar(64) NOT NULL,
  `vmailpass` varchar(16) default NULL,
  `active` tinyint(4) NOT NULL default '0',
  PRIMARY KEY (`id`)
)
```

## FreeSWITCH Configuration

In my implementation, I actually had to remove the existing user directory loading jazz because it was causing me grief (and wasn't used anyway).

```
<configuration name="xml_curl.conf" description="cURL XML Gateway">
  <bindings>
    <binding name="sql_directory">
      <param name="gateway-url" value="http://127.0.0.1:2000/?action=directory" bindings="directo
      <param name="disable-100-continue" value="true"/>
    </binding>
  </bindings>
</configuration>
```

See [conf/autoload\\_configs/xml\\_curl.conf.xml](#) for details, but basically this is the simplest setup.

## Templates

*document.xml* is the main template, entries will be plopped in place of the #ENTRIES# keyword located in it. The 'domain' is also specified here.

*entry.xml* contains a boilerplate of an individual directory entry. Options may be hardcoded into this, but may cause more work for your switch, as it would have superfluous XML to read. If you find yourself putting static options in this template, consider setting them on your dialplan for this context/callgroup instead.

## The Dirty Details

The code is split into 4 parts:

```
startup/initialization
database
directory logic
http-request handler
```

I tried to make the code easy to understand, but still compact and efficient for production use. I know how common it is for an overworked developer to recycle example code into their own production deployments, and if it makes you feel any better, this particular application is used in production.

## Warning

It's important to note that this version has no encryption or authentication whatsoever and could quite easily feed your directory (along with passwords) to any web client hitting the right port. For casual playing around, just set *net\_host* to **localhost**.