

## **HowTo:**

# **Install and Configure ChurchInfo on an Ubuntu LAMP server**

Version 0.1 (author information at end of doc)

### **What's ChurchInfo?**

According to the ChurchInfo web site at <http://www.churchdb.org> , **ChurchInfo** is a free church database program to help churches track members, families, groups, pledges and payments. This is an updated version of Infocentral, with enhancements to track pledges and payments by family, automate donations by bank draft and credit card, and facilitate an annual pledge drive.

### **What's a LAMP server?**

LAMP stands for Linux, Apache, MySQL and PHP/Perl. Each of these represents a layer or service that provides the functionality needed for this sort of application to work.

### **L is for Linux**

The most basic layer is the “L”, which in this case is Linux. Linux is an Operating System or OS, the basic software that has to run on a system to enable you to communicate with it and tell it what to do. Many different groups and companies have taken Linux and packaged it differently in what are called distributions.

For this server we're using a distribution of Linux called **Ubuntu**, one of the more popular distributions of Linux available today. You can learn more about it here: <http://www.ubuntu.com> . Ubuntu comes in a desktop version that includes a very attractive Graphical User Interface (GUI) and desktop apps. The desktop version takes a fairly powerful PC (not as much as Windows, but it still requires some power.)

It will seem a bit counter-intuitive that a *server* version would require less power than a *desktop* version, but it makes sense if you consider that most of the code that uses the resources (processor cycles, RAM and hard drive space) on the desktop version relates to the overhead of presenting an attractive GUI. Servers don't need to do all of that video processing. They're more like trucks than race cars, and what we're installing here is a server -- all we need to do for this server we can do from a command line. That may sound a bit intimidating, but if we stick together we can get through this and learn some things along the way.

### **A is for Apache**

The “A” in LAMP is the Apache web server. Apache is the most widely used web server on the Internet, and there are versions that run on Linux, Windows, and other

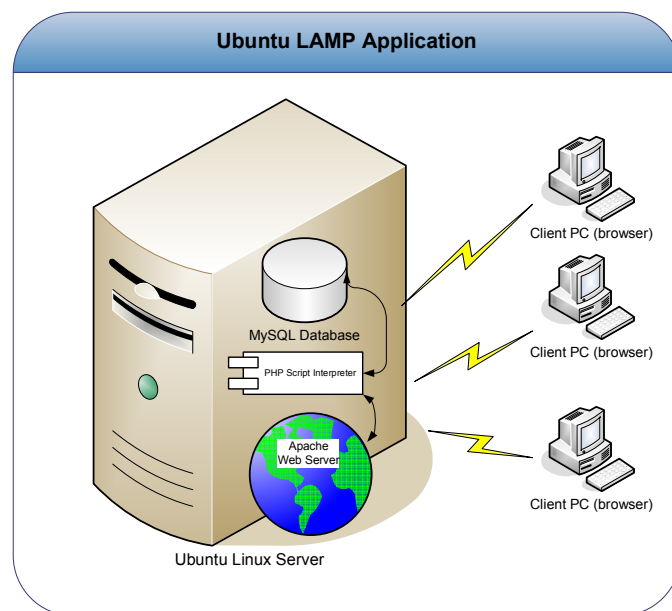
OS platforms. What Apache does is provides a way for you to get to files and applications on this server using a browser. Apache can serve up html files, which are the sorts of files used for web pages on the Internet. But it can do more than this as well, but let's not get ahead of ourselves.

## M is for MySQL

The “M” in LAMP is MySQL. Like Apache, MySQL is free and open source software. It is what is called a Relational Database server, meaning it is a program that holds information organized in databases. This is the back end engine that runs ChurchInfo, and it can host other sorts of data as well.

## P is for PHP

Finally, the “P” in LAMP is PHP, which is a scripting language that adds muscle to the lowly html file. With PHP, web pages can do much more than just display text and images. PHP allows web pages to connect to databases on the back end and provide dynamic data – which means that php creates html code to display information on the fly. It is the piece that holds the others together, communicating between your browser, the Apache web server, and the MySQL database on the back end. The “P” in LAMP can also mean Perl or Python, both scripting languages that can do the same sorts of things as PHP, but for the purposes of this project we're going to focus on PHP.



How a LAMP application works: From your Client PC, you enter the URL of a PHP file on the LAMP server. The Apache web server sees the PHP script in the file that's been called, so instead of displaying the text of the PHP script, it executes it on the server. The script generates what you see in your browser dynamically, pulling information from the database to populate the web page that's presented to you.

## **ChurchInfo is a PHP application**

ChurchInfo consists of a set of PHP files that sits on the Apache server to talk to the database on the back end, and a specially designed structure for the database called a *schema*. Because its components run on a server, ChurchInfo is not a desktop application like a spreadsheet where you have the program and the data file on your desktop PC. It's what's known as a *three-tier client-server application*. A desktop machine with a browser is the client, and the application code (in this case, ChurchInfo) resides in a directory Apache can get to – this is the application server. The database server on the back end is the third tier, and it holds structured data in different sets that are related to each other.

One of the beautiful things about a client-server application like ChurchInfo is that you can have multiple client machines with browsers accessing the application and data simultaneously. The database server handles incoming requests so that the clients don't step on each other.

## **Infrastructure assumptions**

This is a client/server application. This means that you have to have a network infrastructure to install it. At a bare minimum you must have at least one client machine that can run a browser. For most of you this is going to be a Windows desktop, and I'm going to assume that throughout this document.

You also need a network. You can run this application on a local network with only your client and a switch or hub, but in order to do downloads and such you will need a broadband internet connection. Dialup is simply not fast enough to pull down CD image files. If you don't have this sort of connectivity in your church, you have a couple of alternatives: You can download the files onto the PC of a member who has a broadband connection or you can use a library or Internet café connection. There may be wifi networks at Starbucks or other local coffee shops. You can put the downloaded files on a USB drive (a 1-2GB flash drive costs about \$20 at time of this writing, and that is big enough to hold all the files you need.)

A final requirement is a CD burner. If you don't have one, a member will need to burn the install CD for Ubuntu for you. You can request an install CD from Ubuntu for free, but I recommend using a member's broadband connection and CD burner, as Ubuntu's noncommercial side is in the same boat as your church: it exists as a community and doesn't make money.

Ideally you have a client PC with a burner that is hooked to the Internet through a broadband router that is also the switch for your local area network. If you have broadband (DSL or Cable), then a router can be had for as little as \$25.00 from mail order or even Wal-Mart, and I'm pretty sure you have one of those nearby (though in

some rural stores routers may be a special order item.) Your router does another important thing for you – it hands out network addresses using a protocol called DHCP. If you don't have a router that includes an integrated DHCP server, consider getting one.

### **Anticipate some challenges**

If you've never installed Linux before, the thought of building your own server can be daunting. Relax – you will be building this on a PC you don't need to keep any data on. If you fail or make mistakes, you'll be able to start over again. Each mistake you make is an opportunity to learn.

Nevertheless, installing a Linux web server is something that does require some technical inclination. You can probably do this yourself as long as you don't have any problems. Problems can come up in many areas – the PC you use to do your build might have some hardware that Ubuntu doesn't recognize, for instance, or there may be hardware problems (not unheard of on used hardware.) A single document like this one isn't going to be able to anticipate every possible contingency. But I can list a few places where you can get help:

- Local high school computer labs
- Local college computer science departments
- Linux user groups
- The Internet:
  - <http://www.ubuntuforums.org> Ubuntu discussion forums
  - <http://www.churchdb.org/> Look for the forums that support churchinfo
  - <http://www.thelinuxlink.net/14c/> The Linux for Christians Mailing list

### **Minimum System Requirements**

To run ChurchInfo or any LAMP application you will need a server that has at least a 500Mhz processor and 512MB of RAM. Your hard drive should be at least 20GB – not because ChurchInfo takes up that much space, but if you are using a drive that's smaller, it is probably so old it may not be reliable.

This system is going to be a server – it will always be on, and you want a system that will be reliable. An older PC will work for you, but the newer and more robust your hardware is, the more confidence you will have in it. We will be making a way to back up your data so it's safe, but as a general rule, you want to use a decent machine.

You don't need a powerful video card. The server install of Ubuntu is command line only, and won't use the power of a video card. You don't need a sound card on the server either, as the server will not have any function that uses it. Basically you need a processor, RAM, and a hard drive, and you want something that's not going to fall apart. Don't attempt to use a system that had hardware problems – you need healthy hardware. Beyond that, a very modest system is adequate for the urposes of this app.

## Tools to help

If you are experienced at running Linux already, you can probably skip this part. But chances are you're using a Windows desktop, and the server that will host ChurchInfo will be on a network where there are Windows desktops. We're going to set up this server to run headless, that is, without a keyboard, monitor and mouse attached. You'll want a keyboard and monitor for the initial install (no mouse needed.) But afterward you will be able to run this server without them – this will take up less space and use less power.

Running a headless server, though, you will need to be able to access it remotely. The ChurchInfo app is accessed through your browser, but you want to be able to do two things with your server – you want to log into it using a terminal, and you want to be able to access the file system.

First you need a way to log in. We'll be installing a terminal server on your Ubuntu box called SSH, and there's a free terminal client you can download called PuTTY. It's available here:

<http://www.chiark.greenend.org.uk/~sgtatham/putty/>

Or you can simply type Putty into your favorite search engine and find a download location. You want the free SSH/Telnet client. Use the installer for Windows that installs everything except PuTTYtel. This is the client you will use to log into your LAMP server from your Windows box.

There are two ways to access the file system on your Ubuntu Box – you can serve up the files from it using a free package called Samba. This is a bit tricky to configure, and we won't be using it for the purposes of this project (but you may want to consider using your Ubuntu box as a file server later on.) We will be using a free Windows program called WinSCP that allows you to drag and drop files to your server just as if it were another Windows box. WinSCP is available here:

<http://winscp.net/eng/index.php>

Or you can enter WinSCP in your favorite search engine.

Both of these tools install like any other Windows application, and remember you're installing them on your Windows desktop. They don't go on the Linux server.

## Get started: Download Ubuntu Linux Server

Download the ISO disk image for Ubuntu 7.04 Feisty Fawn server here:

<http://www.ubuntu.com> Click on the "Server Edition," and on the page that comes up, click "Download Ubuntu Server Edition." On the next page, you'll need to click the radio button for Ubuntu 7.04 Server Edition.

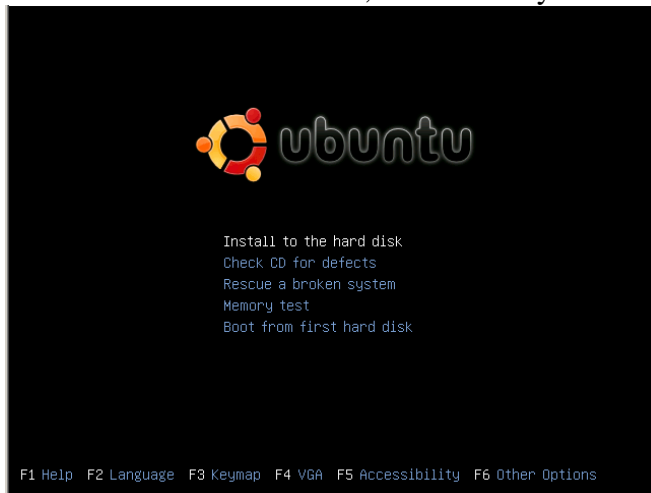
After you download the server CD, burn the image to a CD using your favorite CD burning software (Nero, Roxio or other). If you don't know how to burn an ISO image to

a CD, first try the help button in your burning software. Failing that, use Google or get a techie to help.

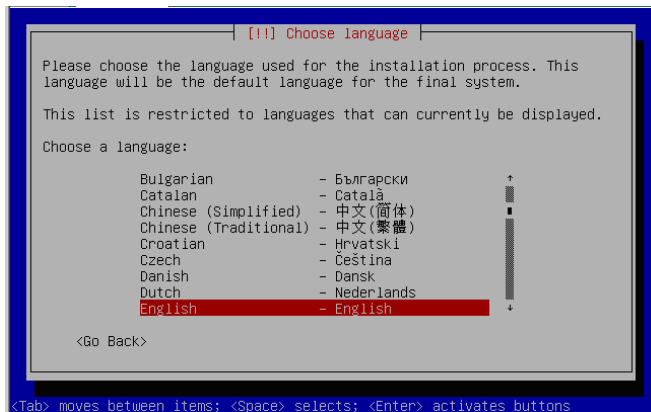
## Installing Ubuntu LAMP server

Boot the server using the CD you created. The installation process **WILL FORMAT THE HARD DRIVE!** Don't do this on a system that has any data on it you intend to keep! Use the server you intend to dedicate to ChurchInfo and later on perhaps file services.

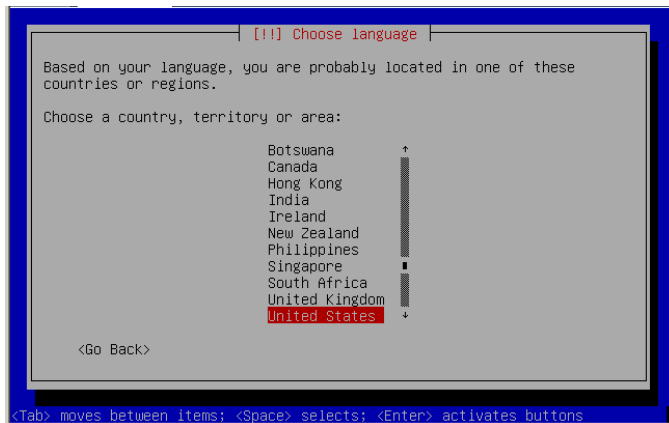
After the initial boot, here's what you'll see:



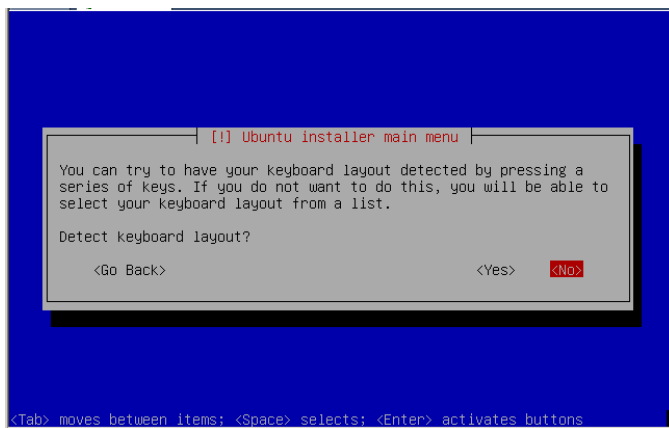
Choose Install to the Hard Disk. The drive will churn a bit, and from here on out the Ubuntu install is text based rather than being on a pretty GUI screen like the boot screen.



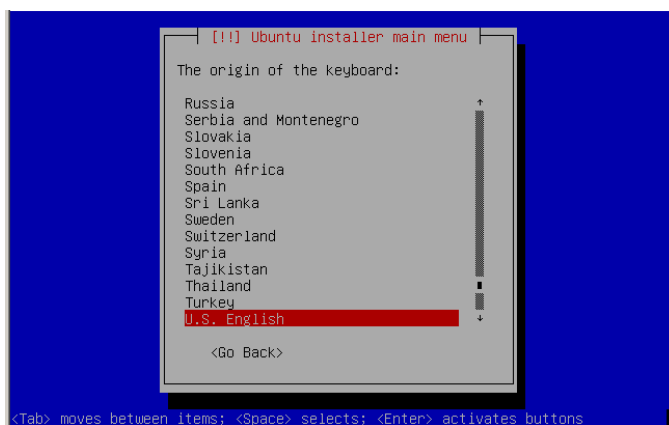
Choose English (assuming that's the language you want to use – the steps are the same for any language, but since the instructions are in English, we'll assume that's what you're using).



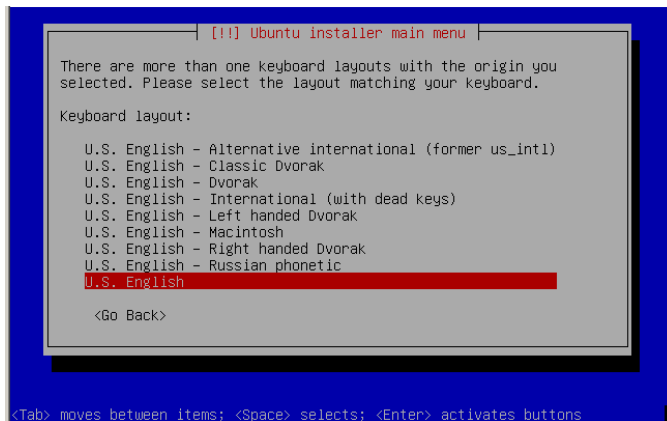
Again, assuming you're in the US, choose that as your country.



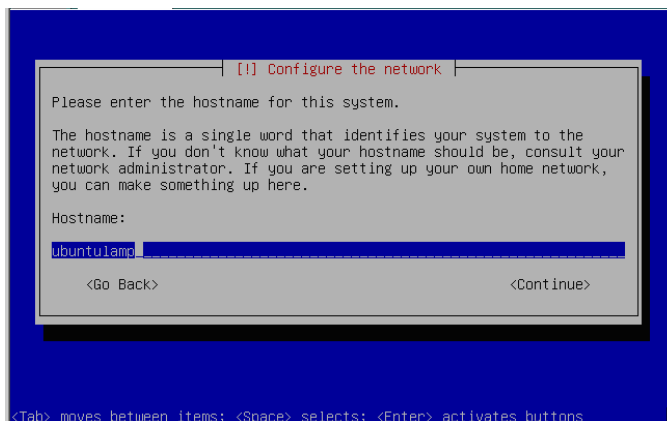
Don't bother having Ubuntu detect your keyboard layout. Use your arrow keys to select No and press enter.



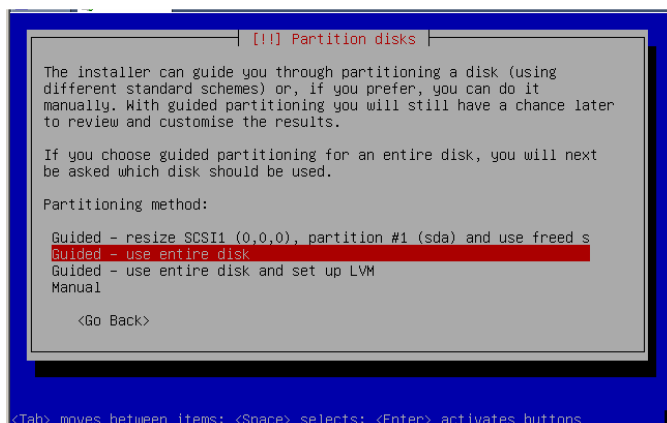
Your keyboard layout is likely going to be US English, as shown here above.



Because there are multiple keyboards for US English, you need to select once more.

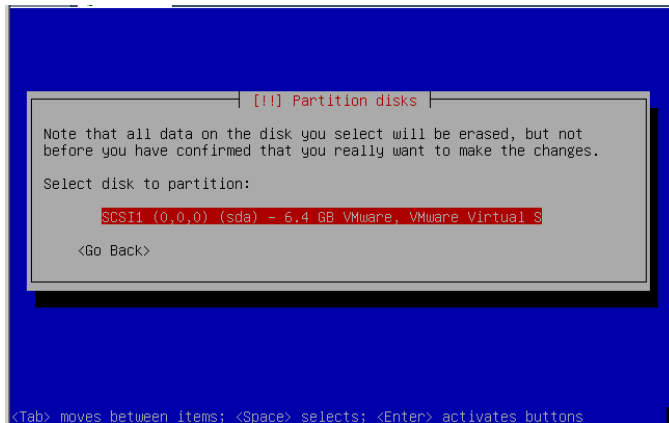


In this step you'll need to enter the desired host name for your server as shown above. The host name I've selected for these instructions is *ubuntulamp*. Not very creative, but that's what we'll use here and to the end of these instructions.

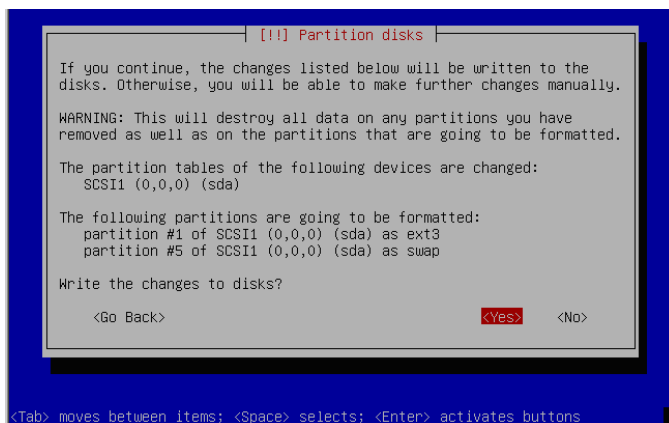


This is going to be a dedicated server, so on the Disk Partitioner select “Guided – use entire disk” as shown above.

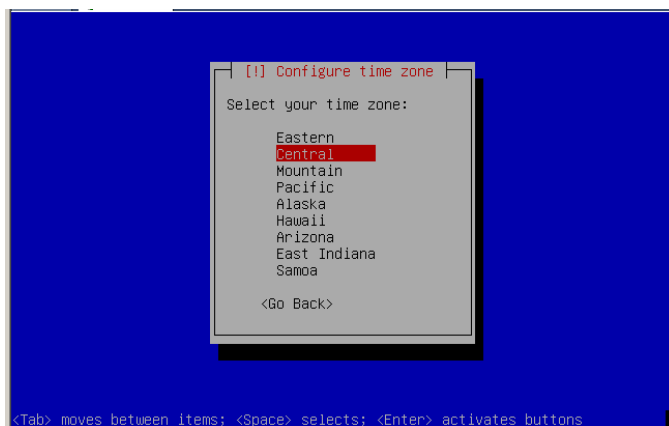




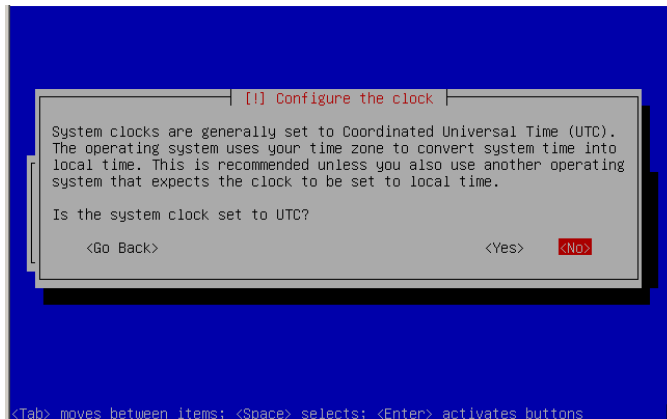
Your disk information will likely look different, but if you have a single hard drive on your system, select that and press Enter.



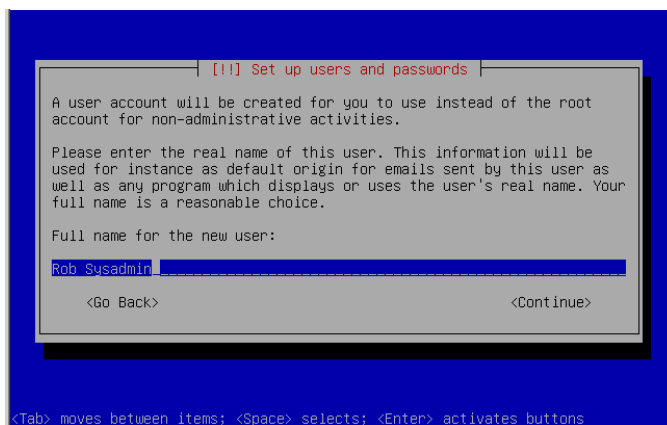
Above you see the first cool warning screen: You're about to destroy all your data! That is, all the old data on your hard drive. As Martin Luther said, "Sin boldly!" Select Yes and hit enter.



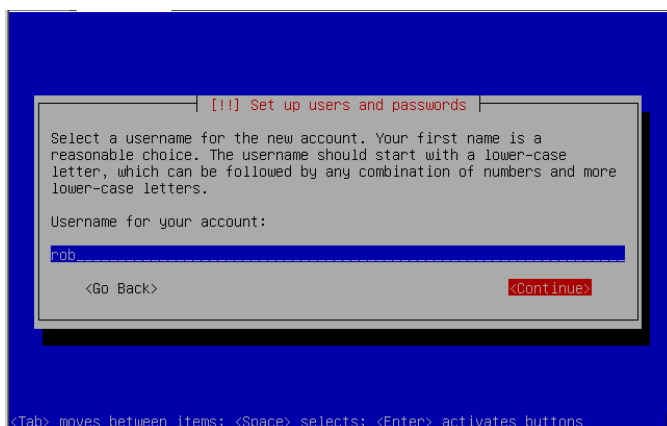
Choose your time zone as shown above. Mine happens to be Central.



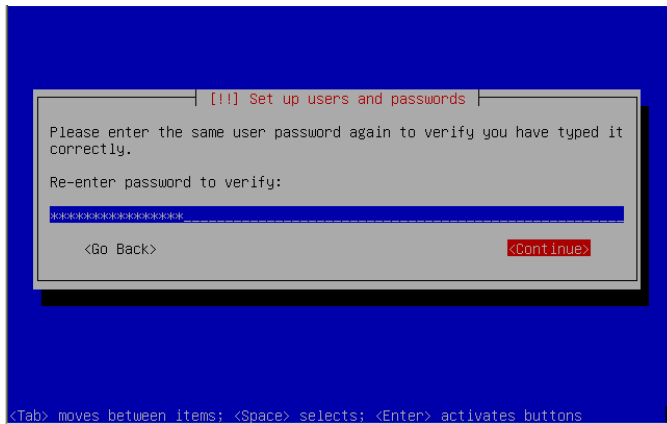
Your system clock is probably NOT set to UTC. If you're not sure, select No. You can always correct this later.



When you install Ubuntu, it will create one user account for you based on what you enter here. This user will be able to use elevated privileges. The full name isn't as important, but it's convenient to have a full name defined. Enter yours here as shown above.

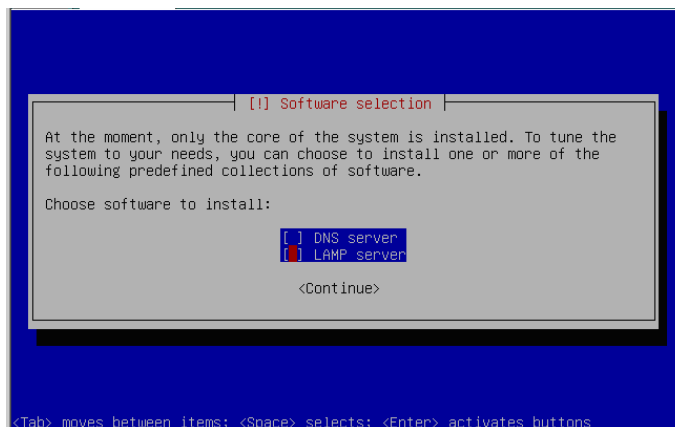


The username you enter here is really important. This is the username you'll use to log onto the server. I'm using the name rob – enter your username as shown above.



I've skipped the first password screen and am showing the confirmation screen. But you will need to enter a password and confirm it. Please remember what this is. If in doubt, use password – you can change this later easily, in fact you will probably want to change the password periodically. It's up to you, but periodic password changes are a best practice.

After you enter and confirm your password the Base system install takes place here. The drive will churn and you will see things scrolling past on the screen. The next step is important – pay attention.



When prompted as to what kind of server you wish to install, use your down arrow key to select LAMP. Tap your space bar to make the selection (an asterisk will appear in the brackets to the left of LAMP). Then tab or arrow down to Continue and hit enter.

Ubuntu will continue to install and finally come to a screen that looks like the one below.

```
* Mounting local filesystems... [ OK ]
* Activating swapfile swap... [ OK ]
* Configuring network interfaces... [ OK ]
* Setting up console font and keymap... [ OK ]
* Starting system log daemon... [ OK ]
* Starting kernel log... [ OK ]
Ubuntu 7.04 ubuntulamp tty1
ubuntulamp login: * Starting MySQL database server mysqld [ OK ]
* Checking for corrupt, not cleanly closed and upgrade needing tables.
g OpenBSD Secure Shell server... * Startin [ OK ]
* Starting deferred execution scheduler atd [ OK ]
* Starting periodic command scheduler crond [ OK ]
* Starting web server (apache2)... [ OK ]
* Running local boot scripts (/etc/rc.local) [ OK ]
```

Kind of intimidating isn't it? This screen has no prompt, so you can't do anything. No worries – Linux has multiple terminal windows available. If you will hit ctrl+alt+F2, you'll get a login prompt.

```
Ubuntu 7.04 ubuntulamp tty2
ubuntulamp login: _
```

Enter the account name you put in during installation (mine's "rob"). When prompted enter your password.

```
Ubuntu 7.04 ubuntulamp tty2
ubuntulamp login: rob
Password:
Last login: Tue May 1 11:10:24 2007 from 192.168.5.1 on pts/0
Linux ubuntulamp 2.6.20-15-server #2 SMP Sun Apr 15 07:41:34 UTC 2007 i686

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
rob@ubuntulamp:~$ _
```

## Logging in and using superuser privileges

You're logged in! Now all you have is a black screen with a prompt in the format of (yourusername)@(yourhostname):~\$

Note: In Linux, all commands are *case sensitive*. Please pay attention to this whenever entering commands on the Linux server.

The default prompt ends in a dollar sign. That tells you that you're logged in as an ordinary user. In order to install stuff, however, you need to be root. The root user has the keys to the kingdom on a Linux box. Hackers know this, too, so many of their exploits aim at trying to break into the root user account. For this reason, Ubuntu wisely disables the root user account. Instead you will use a command called sudo that elevates your user to the privileges of root, using a command called sudo (which I suppose is an abbreviation for superuser do – do this as superuser.) When you precede a command with sudo, you execute the command as root. The first time you do this in any ten-minute period, Ubuntu prompts you for your password. That's your password, not the root user password. After you enter it once, you can sudo as many times as you want to for about ten minutes. After that, Ubuntu will have you authenticate again.

If you don't want to enter sudo prior to every install command, simply enter sudo -s. After you enter your password, you will see the prompt change from a dollar sign to a hash mark (#). This indicates that you're root now. In this way you have all the privileges of root without the security vulnerabilities of having an actual root user account for hackers to try to break into.

## Install SSH

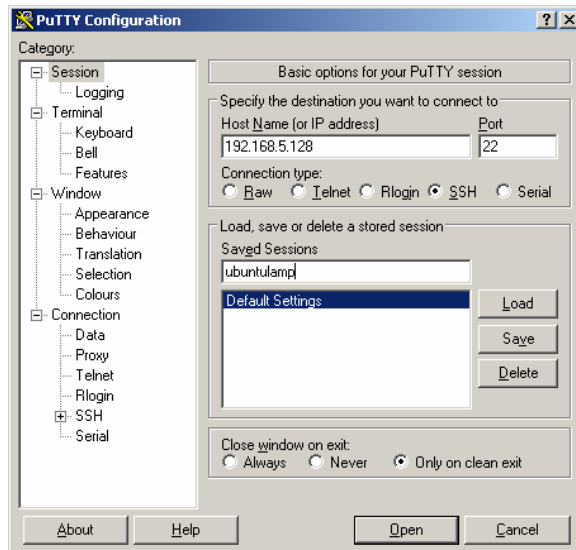
A second minor irritation about Ubuntu is that it doesn't install any remote login facility by default. We need to install the Secure Shell Server, which we can do with a single command. Enter this line as shown:  
sudo apt-get install ssh

```
rob@ubuntulamp:~$ sudo apt-get install ssh_
```

I've already explained what sudo does, so you should expect the password prompt that will come up. Now you know how to install a package and you've used sudo. The above command will install the secure shell server on your system. Secure Shell runs as a program that loads automatically on bootup. This sort of program is called a daemon, from the Greek *δαιμων*. Don't take any theological significance from this – it just means a process that works tirelessly without interaction.

## Log in remotely using SSH

Now that SSH is installed, let's get those tools set up we installed on the Windows desktop earlier. Bring up PuTTY and you'll see a window like this:



Enter your server's address under Host Name (or IP address) as shown above. If you don't know the server's address handed out by your router, enter this command at the command prompt: `ifconfig`

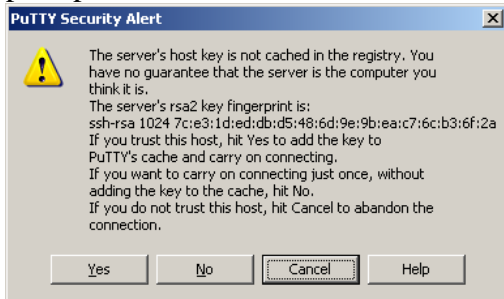
```
rob@ubuntulamp:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:96:BF:BA
          inet addr:192.168.5.128  Bcast:192.168.5.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe96:bfba/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:54 errors:0 dropped:0 overruns:0 frame:0
          TX packets:62 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:7571 (7.3 KiB)  TX bytes:8580 (8.3 KiB)
          Interrupt:17 Base address:0x1400

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

Note on the second line of the response is the entry “inet addr: 192.168.5.128” in the example above. This means my server's IP address is 192.168.5.128. Yours will be different. Use your address wherever you see mine in this document.

Once you know your server's address, enter it as the address in PuTTY, then type the host name under Saved Sessions, and click the Save button. This will keep the session settings for you so you don't have to retype the IP address every time you use PuTTY.

After you click Save, click the Open button at the bottom of the screen. If SSH is working on the server, the very first time you attempt to connect, you will first get a prompt like this one:



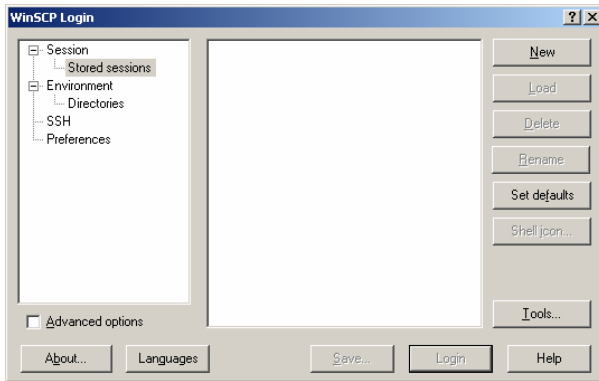
Click Yes. You should not see this screen again. Afterward, you should get a login prompt like this:



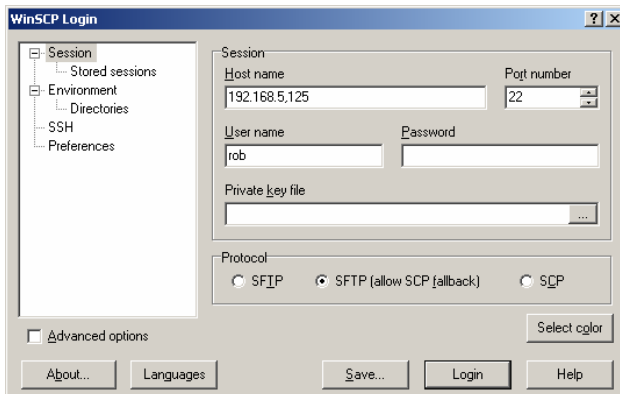
Enter your user name, then enter your password when prompted. If SSH is working, you will have a shell prompt just as if you were on the console of the server. Once you've gotten this far, you can put away the monitor and keyboard on your server. From here on out, you can run headless on the server and do everything from the comfort of your desktop machine, which presumably is where you have a comfy chair and other amenities. Every time you use PuTTY to log into your server, you enter commands on your desktop and they run on the server, just exactly as if you were logged in on the server console. Neat, huh?

## Using WinSCP

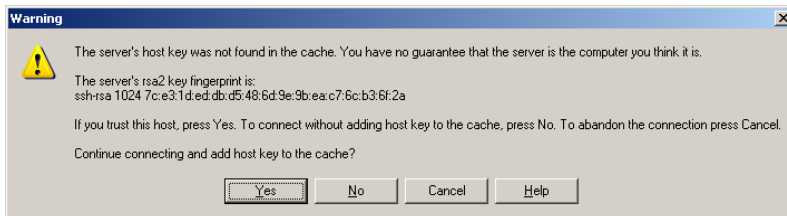
Next, let's get WinSCP going as well, so we can move files to and from our Linux server. You already downloaded and installed WinSCP, if you followed the previous instructions. Open WinSCP (you don't need to close PuTTY). You'll see a window like this one:



Click New, and you'll see the screen depicted below. Enter the information as shown for your server. Note that we're entering the address and User name only – don't enter the password here for security's sake. The port number is already entered for you.

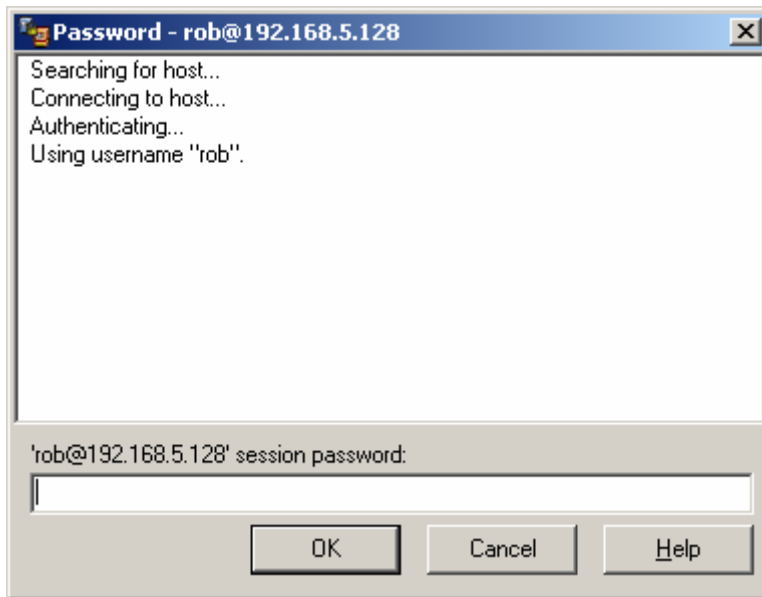


Now click Save, then click Login. Since WinSCP works like PuTTY, using the SSH protocol, it will also want to cache a key file the very first time you log in. Before you get logged in, it will prompt you with a window like this one:

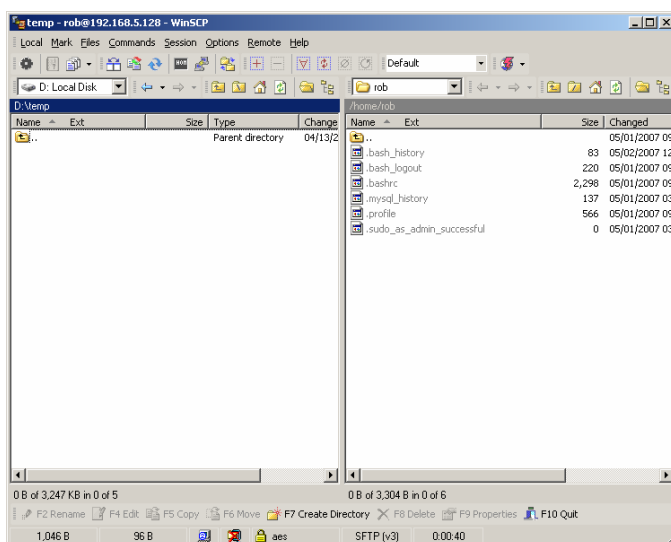


Click Yes, and you'll be prompted with a login window like this one:





Enter your password, and you'll be logged in. You'll see a two-sided file system browser like the one shown here:



## Time for a break

You've done a lot of work – you installed your server, you installed some client tools and have tested them, and now your head is probably swimming and may be aching as well. You may be wondering what you got yourself in for – “I had no idea there would be so many steps. Why can't someone just make something easy to use?”

That's a reasonable question. ChurchInfo isn't that complicated, not really, but it assumes that you have an infrastructure to install it on. We're into 17 pages of instructions here, and we haven't even begun to install the application we are aiming for.

Nevertheless, you should be pleased by this step. You've built a Linux server, it's up and running, and you've successfully connected to it remotely, using the tools you installed on another system. That's a real accomplishment! Your infrastructure is in place and you have a foundation to build on.

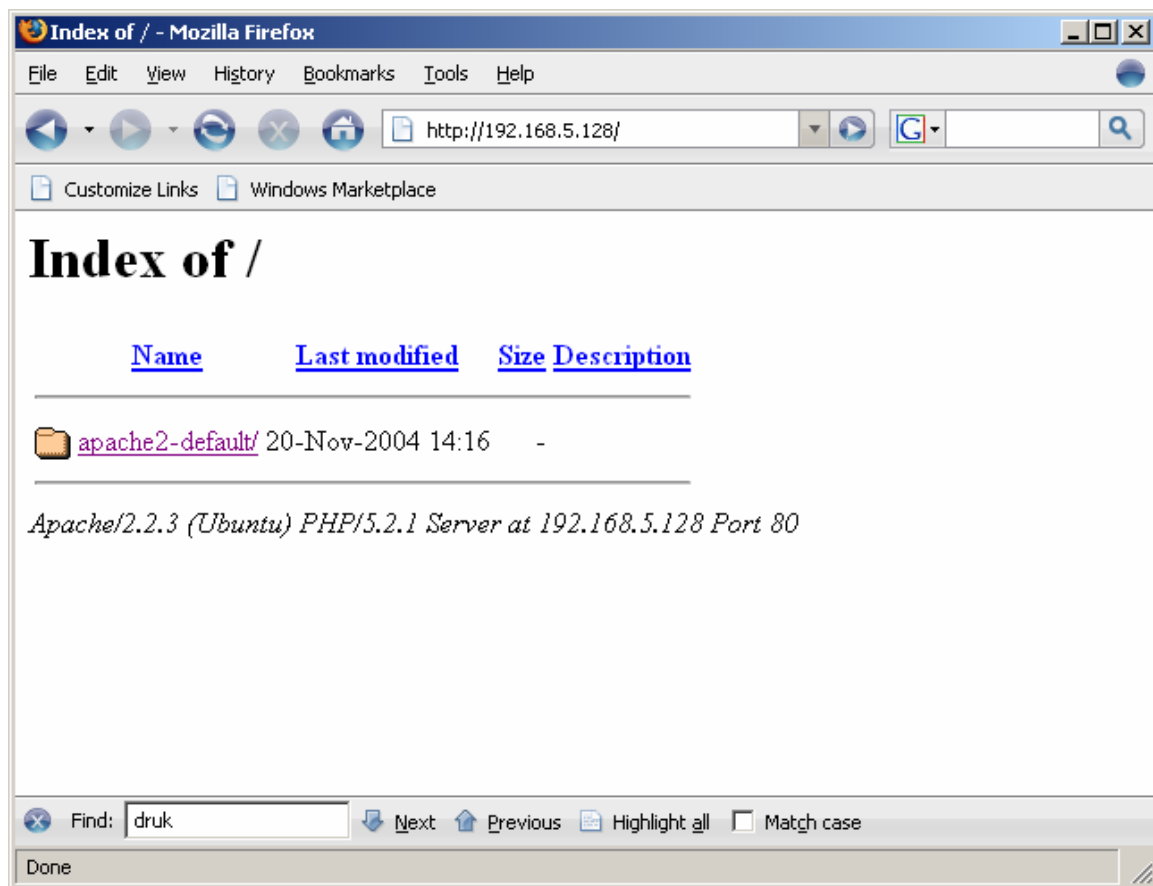
Step out of the work area for a well-deserved break. Eat a meal. Have something to drink. Relax and get feeling refreshed before you continue – we still have quite a few things to do.

## Testing the LAMP Components

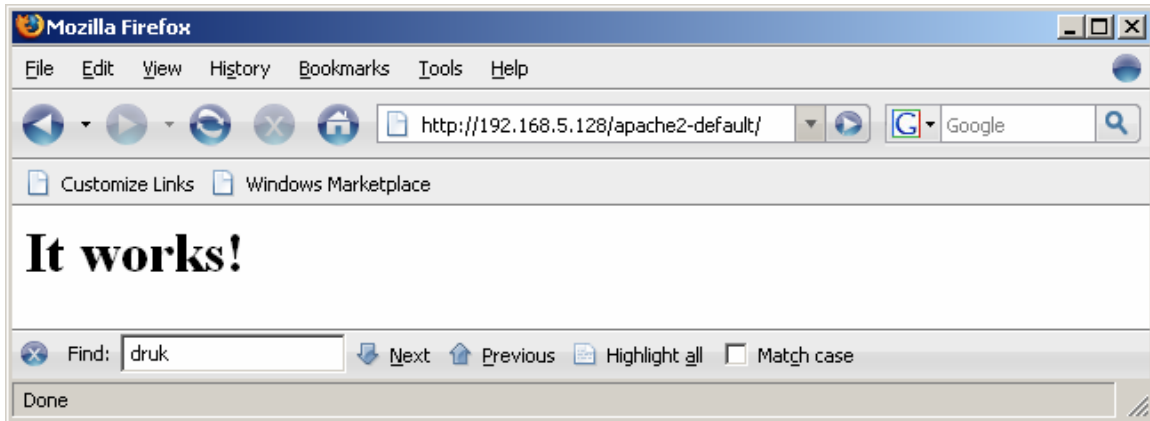
How was your break? Hopefully you've caught your breath and had some time to reflect on what you've accomplished. If you got past the previous steps, we already know that your Linux server is up and working. That takes care of the "L" in LAMP. Before we proceed we need to ensure that all the other components of our infrastructure are ready.

### Apache

Test Apache – go to your desktop machine on the same network with your server. Open a browser (Firefox or Internet Explorer) and type in the address for your Ubuntu box in the URL line and hit enter. You should see a screen like this one:



Click on the folder that says apache-2-default and you should see this:



Apache is working. If it's not, use the Ubuntu forums or have your techie look at the system. This is a basic beginning set of instructions.

## MySQL

We installed SSH on your Ubuntu server and PuTTY on your desktop, and now we're going to use the latter to test and configure MySQL. On your Windows desktop, open up PuTTY and log into your new server.

If the LAMP stack installed correctly, MySQL should already be running. To see if it is, we're going to ask Linux to list the processes running by typing in the command

```
ps -A
```

```
rob@ubuntulamp:~$ ps -A
```

This will list all the services running on your box. Look for a couple of lines like this:

```
3606 ?          00:00:00 mysqld_safe
3648 ?          00:00:00 mysqld
```

The numbers will be different on your system, but we want to see `mysqld_safe` and `mysqld` running as processes. If they're not running, I suspect that Apache wasn't running either, and you didn't install the LAMP server right. I will assume that they are both running.

MySQL is running on your server, but we have not configured it yet. That's what we need to do now – the main thing we need to do is set a password for the root user of the database (not to be confused with the root user of the Linux server. I know the terminology is confusing – just remember that each component of the stack operates in its own compartment, so to speak, even though they are all loaded on the same machine.) In order to configure MySQL we have to log into the database server. The way we do that is by entering the command at the terminal:

```
mysql -u root
```

This will launch the MySQL client as database user root. The initial install sets up the database root user account with no password, so the client goes straight into interactive mode with the database server, as shown:

```
rob@ubuntulamp:~$ mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6
Server version: 5.0.38-Ubuntu_0ubuntu1-log Ubuntu 7.04 distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

Note how the prompt changed. Now you're not in a shell for the OS, but you're at a console for MySQL. We need to set a root password to make sure your databases will be safe. All MySQL commands must end with a semicolon. If you forget to type it at the end of the line, don't worry. Just enter a semicolon on the next line and hit enter. Everything preceding will be accepted as if it's all on one line.

At the MySQL prompt enter this:

```
mysql> use mysql;
```

```
rob@ubuntulamp: ~
rob@ubuntulamp:~$ mysql -u

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 7
Server version: 5.0.38-Ubuntu_0ubuntu1-log Ubuntu 7.04 distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>
```

Next, enter this command to set the root password.

```
mysql> UPDATE user SET Password=PASSWORD('yourpassword')
WHERE user ='root';
```

MySQL will respond with:

```
Query OK, 3 rows affected (0.01 sec)
Rows matched: 3  Changed: 3  Warnings: 0
```

Two more commands to run now:

```
mysql>GRANT ALL PRIVILEGES ON *.* TO 'root'@'localhost'
IDENTIFIED BY 'yourpassword' WITH GRANT OPTION;
```

Type the above line as one line or simply hit enter after 'localhost'. The line executes when you enter the last line ending with a semicolon. Repeat this line to add other users – I recommend you add an admin user and your own user name. The only thing different needs to be the other username substituted for root and the password for that username substituted for *yourpassword*.

```
mysql> FLUSH PRIVILEGES;
```

This will result in:

```
Query OK, 0 rows affected (0.00 sec)
```

After a moment MySQL will kill the session (the word “Aborted” will appear in the prompt and send you back to an OS shell. Now you can log back in using the root password to check and see if you’re working:

```
rob@ubuntulamp:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 5.0.38-Ubuntu_0ubuntul-log Ubuntu 7.04
distribution

Type 'help;' or '\h' for help. Type '\c' to clear the
buffer.
```

```
mysql>
```

If you got this far, MySQL is working and your root database user is set up. We’ve checked out the L, A, and M components. The next one is easy.

## PHP

To see if PHP is working we need to put a simple script file on our web server, and we’ll do this from the command line in PuTTY.

I’ve already noted that it is well outside the scope of this document to teach you how to run a Linux server, and I provided links to some good introductory sources to teach you more. In your SSH session, though, you will note that while you’ve in a shell on the server, the shell is currently in your user home directory. Since my username is rob, my shell is currently located in my home directory which is /home/rob. Enter this command in your own shell:

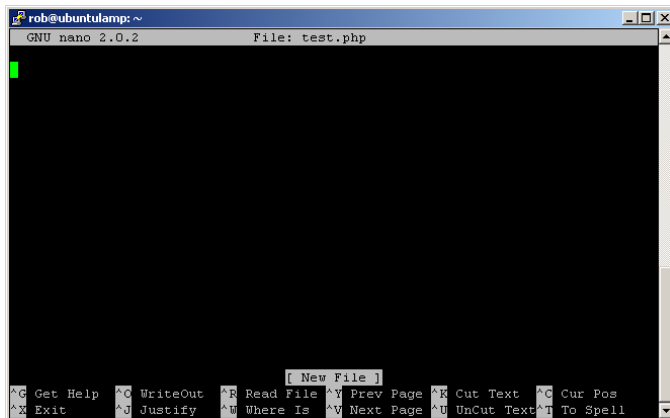
```
pwd
```

The response should tell you your present working directory (which is the breakout of the abbreviation pwd.) In my case it's /home/rob. Yours will be /home/*yourusername*.

We're going to use a little editor called nano to create a text file that will help us test our PHP server. Enter the command

```
nano test.php
```

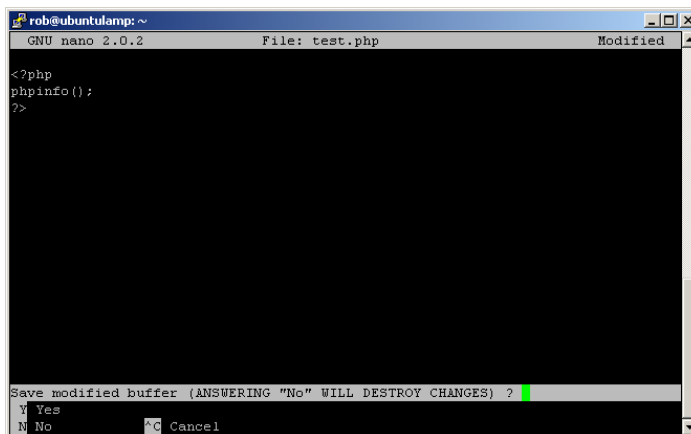
This should open up a screen like this:



Nano is sort of like Notepad on your Windows box, but has a bit more functionality. You're going to enter the following lines. Just type away:

```
<?php
phpinfo();
?>
```

Then hit ctrl+x and you'll see the following screen:



As you may have guessed already, you want to hit Y and enter. This will write out the file test.php with the content you entered. Next we need to copy this file from your user home directory to the web directory on your server. Enter this command:

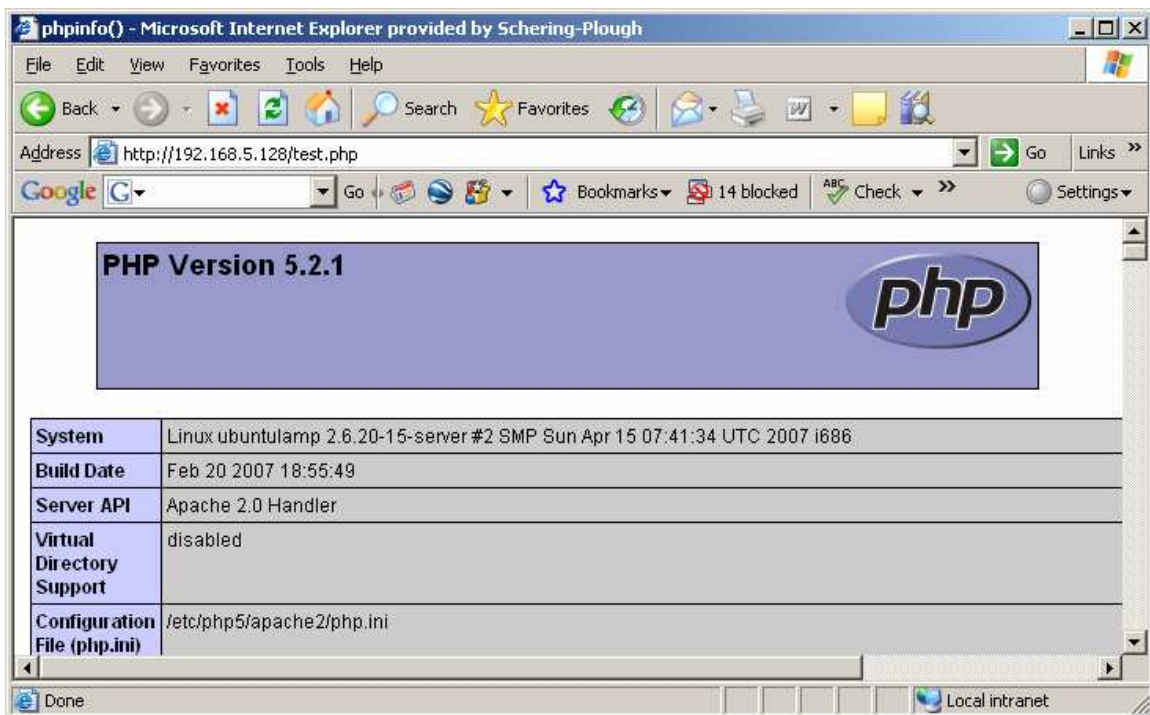
```
sudo cp test.php /var/www
```

Sudo will ask you to enter your password. What you've done is copy (cp) the file test.php to a directory called /var/www which is the root directory of your web server. You had to use sudo because by default only root has write permissions in the /var/www directory. Simple enough?

Now open your browser, and put the address of your server into the URL line in the address bar, followed by a forward slash (right below your right pinkie, next to the right-hand shift key) and the name of the file we created, test.php:

```
http://192.168.5.128/test.php
```

If PHP is working properly on your server, you should see a whole screen of interesting, if arcane, information. A subset is shown here:



If this didn't work, the first thing to check is the file we created – test.php. Is it written exactly as in the example? In your SSH session, enter the commands:

```
cd /var/www  
ls
```



Note that your prompt has changed to show your new directory as /var/www. Is the file test.php there? Type the following command:

```
cat test.php
```

Does the file look exactly like the one below?

```
<?php
phpinfo();
?>
```

If the file is there and the contents are correct, but you still don't see the PHP information web page as shown, something else is wrong. You must get PHP working before you can go further – use the resources indicated earlier to try to get it working. Use forums and the mailing lists – the community will help you, but it may take some time.

Hopefully, though, your web page comes back with the information shown. This web page tells you just a bit about the power of PHP. You typed a tiny little file with one command in it: `phpinfo()`; When you put that file in your browser and called it up from the Apache server on your Ubuntu LAMP box, you got back a whole plethora of information. Looking back to how the LAMP stack works, your Apache server sees the PHP file, and instead of showing you the contents of the file, instead it executes the file, and PHP dynamically generates a web page that displays in your browser, just like an HTML page. In fact, if you will go to the View menu in your browser and click Page Source, you will see the HTML code that PHP generated. That code was all generated by the PHP program on the fly. Pretty cool, huh? Now you begin to glimpse the power of PHP on a LAMP server.

Your server is all checked out now. All the components are working. In the following section we are going to install one more tool before we install ChurchInfo. This tool will enable us to access the MySQL database server directly from the browser. That way we won't have to type in SQL codes at the command line terminal – and I'm pretty sure you are happy about that even if you may not fully know why!

Keep going! We're almost there!

## PHPMyAdmin

We could simply install ChurchInfo at this point and be done with it, but that wouldn't give us any tools for backing up our data, at least not any tools that would be useful for anyone other than an experienced Linux systems administrator. MySQL is a very powerful database system, especially considering the purchase price, which I need not remind you is free.

MySQL is a Relational Database Management System or RDBMS. The SQL portion of the name is an acronym that stands for Structured Query Language. This is a standardized language used to write queries for any database – Microsoft MSSQL, Oracle, Sybase, DB2/UDB, Informix are just a few of the commercial RDBMS programs out there that use Structured Query Language. In the world of Free Software Postgres and MySQL lead the pack, and of these MySQL is probably the most common.

You've already entered some SQL queries at the MySQL command line – remember when you entered the password for the root user? When you flushed the privileges? That was SQL language, which is fairly specialized and arcane as a language. Unless you are really excited about becoming a database administrator or DBA (more alphabet soup – sorry) then you probably would rather not learn SQL. That's where a tool like PHPMyAdmin comes in.

PHPMyAdmin is a graphical tool you get to in a browser that enables you to do much of the administration of a MySQL server. Installing is easy.

Method 1:. Make sure your Ubuntu server is on a network that has access to the Internet, and enter:

```
sudo apt-get install phpmyadmin
```

Enter your password when prompted and Ubuntu will install PHPMyAdmin for you.

Method 2: If you're not on the internet, you can get PHPMyAdmin from any web browser. Go to this URL: <http://www.phpmyadmin.net> .

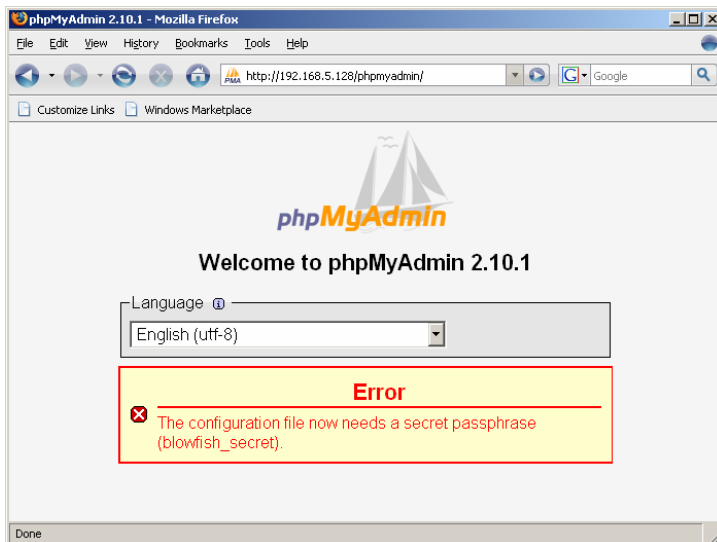
- Follow the links to download the latest version of phpmyadmin – get the zip file.
- Extract it on your Windows desktop, then change the directory name to phpmyadmin.
- Use WinSCP3 and copy the entire directory to your home directory on the Ubuntu LAMP server.
- Login to your LAMP server using PuTTY and enter the following commands:
  - `sudo -s` [enter password when prompted to elevate permissions]
  - `cd /var/www`
  - `mkdir phpmyadmin`

- `cd phpmyadmin`
- `cp -r /home/(yourusername)/phpmyadmin/* .`
- `cd ..`
- `chmod -R 775 phpmyadmin`

Whichever method you used, at this point PHPMyAdmin is installed but not configured on your Ubuntu server. You already have a password set on your MySQL server, you just need to tell PHPMyAdmin you want to use it.

On your desktop, bring up the browser and type in the address of your server followed by a forward slash and *phpmyadmin*:

<http://192.168.5.128/phpmyadmin> If all is well you should see an error message like this:



This is good. It means that PHPMyAdmin is set up properly and all you need is a good config file. At this point I will go against all of the documentation wikis and forum messages and give you a shortcut for a good solid configuration:

Go back to your PuTTY shell session and enter the following commands:

```
sudo -s [enter password if prompted; prompt will change to a hash mark #]
```

```
cd /var/www/phpmyadmin [changes to the phpmyadmin directory]
```

```
cp config.sample.inc.php config.inc.php [copies the file
config.sample.inc.php to config.inc.php]
```

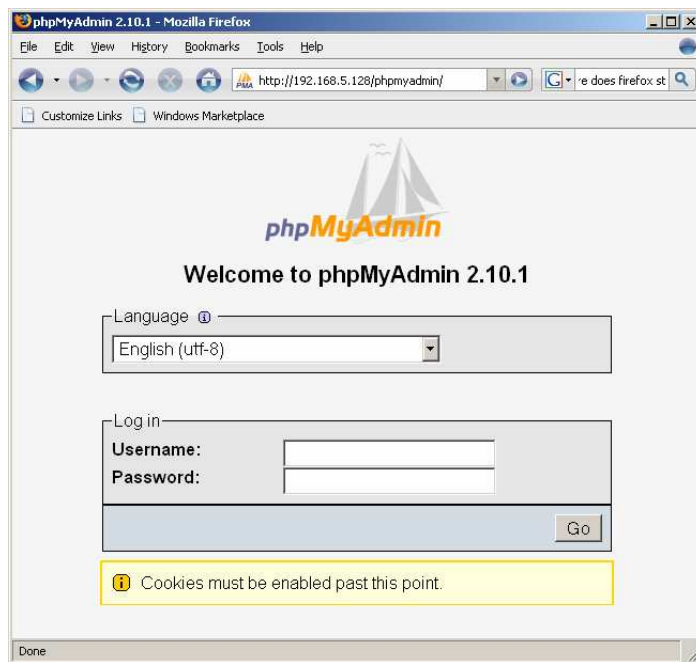
```
nano config.inc.php
```

This will bring up the little screen editor you used before. Use your arrow keys to go down to the line that reads:

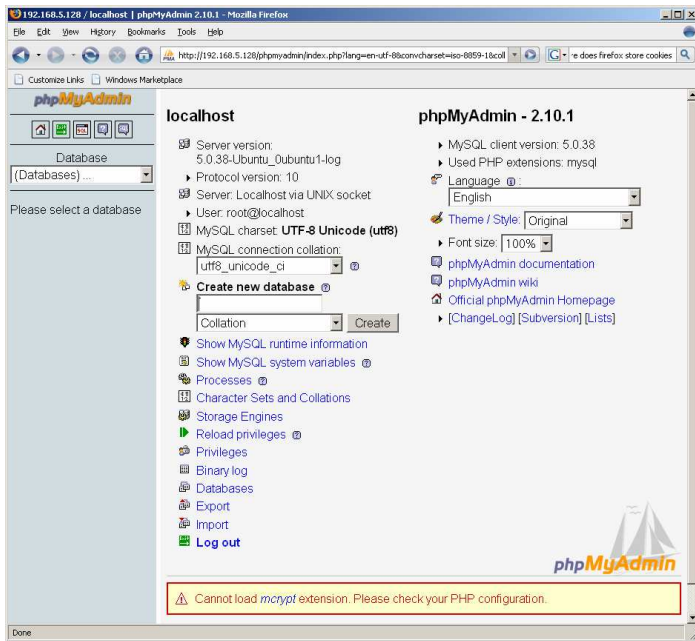
```
$cfg['blowfish_secret'] = ''; /$
```

Using your arrow keys, cursor over so that your cursor is in between the two single quote marks to the right of the equal sign. Type in a good long line of gibberish between the two single quote marks – use numbers, letters, whatever. Just make it fairly random. Make sure you don't overwrite the quote marks. When you're done, type ctrl+X, Y, and enter. This will edit your config file.

Now go to your browser, and reload. You should be greeted with a screen like this one:



Enter root as your username and the root password you set up for MySQL you set up earlier, and you're in! The PHPMyAdmin page will look like this:



There's a lot here. Note that on the initial setup you will probably get an error like the one shown at the bottom of the screen here. It will not affect your functionality for the purposes of ChurchInfo. All you need to do to correct the problem is install mcrypt, but I'm not going to tell you how to do this. Rather I'm going to let you use your web search engine and the support tools I've told you about to get the answer for yourself when it's convenient. This is just a little challenge for you – as I said, if you ignore the error it won't affect you in any serious way.

Congratulations – you have installed and configured a very complex PHP application. If you've gotten this far, it's time to take another break. Next, we will actually install ChurchInfo, the application you wanted to install all along.

If you're a pastor installing this application, you will recall your journey through seminary. Do you remember those first classes when there was so much alien terminology and you felt like you were floundering to try to keep up? You didn't think you would ever be able to learn all the things you needed, did you? But you did.

At this point you're probably feeling very much the same way – even if you didn't go to seminary you can probably identify. Don't be discouraged. You've accomplished a lot, and you have a working infrastructure you can use to manage your church. Now let's get ready to install ChurchInfo.

## ChurchInfo

Unlike some of the other components we've installed, there is no package available that we can simply ask Ubuntu to install for us. ChurchInfo is a niche application and as yet no one has (to my knowledge at the time of this writing) created a package for it because there has not been widespread call to have it loaded into the application repositories for Ubuntu.

The good news is, that doesn't really matter. Because ChurchInfo is a PHP application, simply downloading the files and loading them on our server will take us most of the way to installation.

## Download ChurchInfo

In an Internet browser on your client PC, go to <http://www.churchdb.org> and click the download link. Presently it's the first menu item in the left hand frame. This will take you to ChurchInfo's SourceForge site (SourceForge is a clearinghouse for all kinds of Free Software projects. It provides a place where developers can host source code and interact as a community to develop applications like ChurchInfo). SourceForge should open directly to the download tab on the project (if not, it is the seventh tab from the left as this is written.) At the bottom of the page you will see a large green Download link. Click it.

This will open up a page that shows available files for the release. You'll see a gzip version and a zip version. To avoid having to walk you through extracting the files on the Linux command line, I am electing here to have you download the zip version. Click on the file link for the zip file, and choose Save to Disk.

Once you have the file downloaded, right-click on the file and extract it using whichever extractor you use (zip extractors have been built into Windows since XP, so you should have no worries here.) Extract the files into a folder that will contain only the extracted files – I'd suggest using churchinfo as the folder name. This will make the installation task easier.

Having extracted the files, now copy the folder over to your Linux server. You'll use WinSCP for this. Start WinSCP and make a connection to your Ubuntu box. It will take a moment to connect, then you'll see a file browser with your home directory on the server on the right and (probably) your My Documents folder on your Windows desktop on the left. On the left side browse through your file system until you can find the churchinfo folder containing the extracted files of the ChurchInfo app. Drag it over into your home folder on the Linux server. (Alternatively, you can drag files from an Explorer window directly into the directory on the Linux server. WinSCP supports this.)

Now that you have the files on your LAMP server, you need to make a copy of the folder under your WWW root directory. To do this you'll use PuTTY. Open up a terminal session using PuTTY. Once you're logged in, enter the following commands:

```
cd /var/www [change directory to the WWW root directory]
```

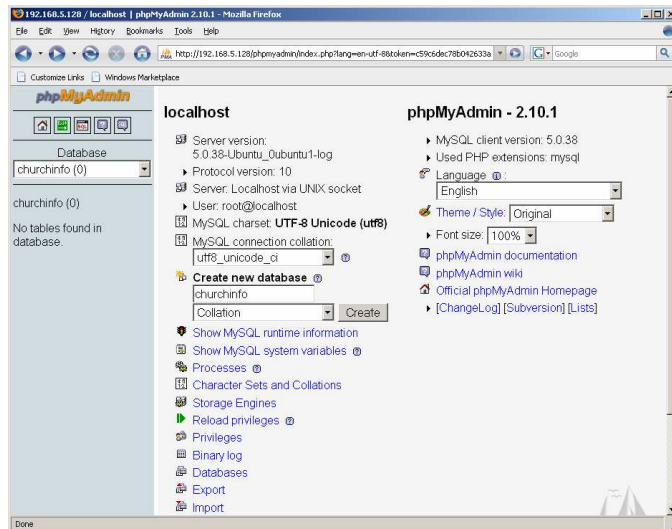
```
sudo -s [elevates to superuser privileges.]
```

```
cp -R /home/(yourusername)/churchinfo . [note the space before  
the final period – this will copy your churchinfo directory over to the web root.]
```

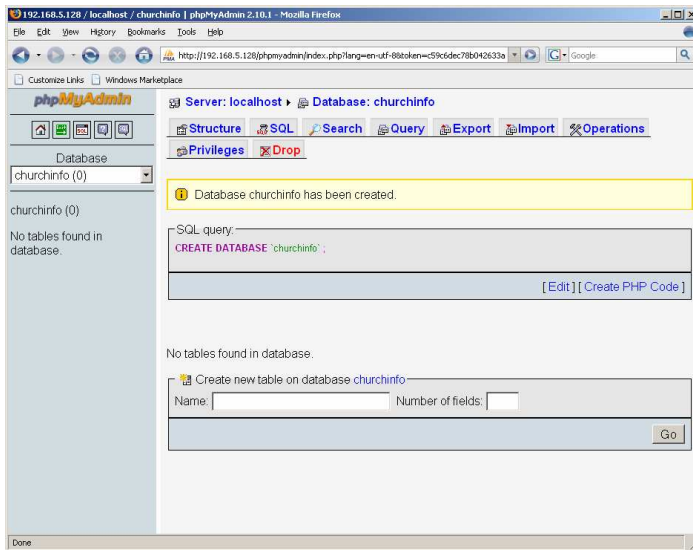
## Create the Database

Now we're going to work with PHPMYAdmin a bit. Back on your desktop, bring up the browser and log into PHPMYAdmin by entering your server address followed by /phpmyadmin . Log in as root and enter your password. We're going to use PHPMYAdmin to create the ChurchInfo database.

In the middle of your PHPMYAdmin window, you will see a text box under the words Create New Database. Type churchinfo into the text box and click the Create button.



PHPMYAdmin will think for a minute then return this screen. Your database is created!



You’ve created the database but right below the creation message you can see the message “No tables found in database.” Let’s take a minute to talk about how data is stored in databases.

First of all, a database will store different categories of data. One category of data will be stored in a *table*. A table is a list of entities in the category and a number of bits of information about each entity. Each entry for one specific entity is called a *record*, and the bits of information that describe the entity are *fields*. ChurchInfo is primarily a database with information about people, so let’s use that as an example.

Imagine a table called People. Each record is a person. There are various different things we can describe about a person – name, address, birthdate, and so forth. One of the principles of database design is to try to break down the bits of information that describe an entity in the database to the point where they can’t be broken down any more. Information that can’t be broken down any more is called atomic. There’s no danger of radiation here, this is just a synonym for indivisible.

For instance, let’s think of a name: John Paul Jones. We can break this name up into first name, a middle name and a last name. If Mr. Jones lives at 4357 Washington Avenue, Jackson, Tennessee 37456, we would want to break down the record into Street address, city, state, and zip code rather than lumping the whole address together.

In each record, each bit of atomic information is recorded in a field. The table is made up of records, also called *rows*, and fields, also called *columns*. A spreadsheet is a good representative of a table. But we want something more powerful – in a Relational database, we take different sorts of tables and relate them to one another. For instance, in ChurchInfo, people are related to families and to groups.

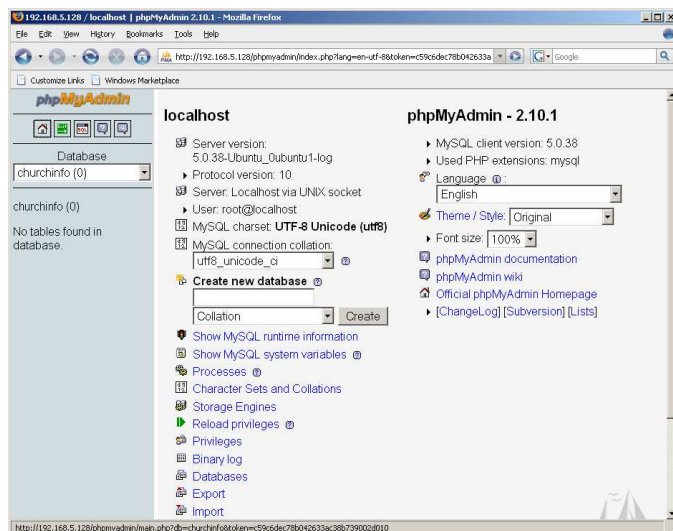
## Install the Database Schema



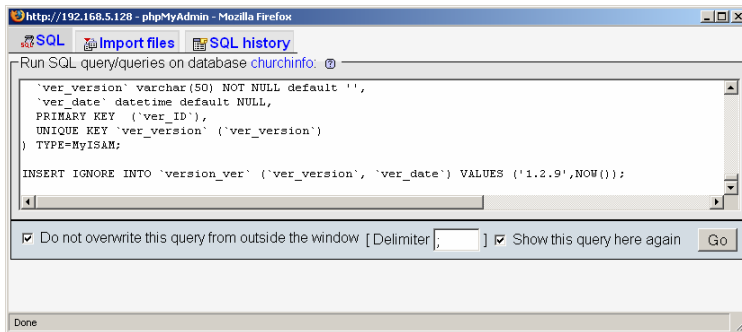
Taken together, a complete set of tables that relate to one another with their specific structure of fields and records is called a *database schema*. So to look at what we have right now, we have a database that is not only empty, but lacks has no schema defined. We're going to install the schema using something called a script written in Structured Query Language – remember that from before? This is the SQL in MySQL. The developers of ChurchInfo have a script already written for us. On your client desktop (you can leave the server alone for this step), look in the folder where you extracted the ChurchInfo files for a subdirectory called SQL. Open that folder.

In the SQL folder you'll see a file called Install.sql. Double-click on that file, and it should open up in Notepad. Hit ctrl+a to select all of the text (or click Edit, Select All). Then hit ctrl+c to copy the text to the clipboard (or click Edit, Copy).

Now go back to your browser, where you created the database for ChurchInfo. In the left side of the window, click on the leftmost icon that looks like a house. This is the PHPMYAdmin Home page. Next, if churchinfo isn't in the Database selection box, hit the dropdown list and select it so the PHPMYAdmin page looks like this:

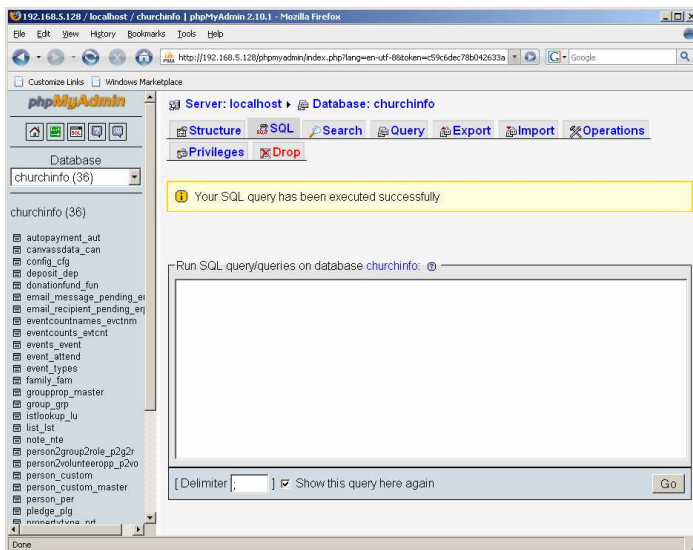


With the churchinfo database in the Database window, click on the SQL icon (Third from the left). This will pop up a browser window that should have tabs for SQL, Import Files, and SQL history across the top. Click in the text area below “Run SQL query/queries on database churchinfo: (if it doesn't say churchinfo, go back to the home page and make sure churchinfo is selected.) Hit ctrl+v to paste the data from the clipboard into the query window. It should now look like this:



If you can see the tail end of the query in your window, use the scroll bar to make sure that all the text from the install.sql script is present by comparing with the text open in Notepad. When you are satisfied that the text is the same, click Go at lower right. PHPMYAdmin will process the scripts very quickly, creating the database schema.

Use Alt+Tab to switch back over to the main browser window. You should see a screen that looks like this:



In the middle you see the report that the SQL query executed successfully and on the left side you see a list of the tables that were created. Good job! Your database still contains no data, but it does have a specified schema now. It's a real database.

## Edit Include File for Database Use

We're almost up and running. Go back to your PuTTY SSH session, and enter the following commands:

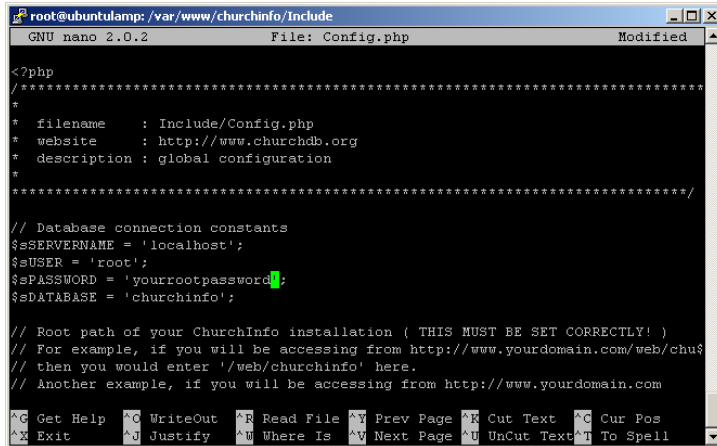
```
cd /var/www/churchinfo/Include
```

[changes to the directory containing the config file we want to edit. Note the capital "I" in Include. Remember Linux filenames and commands are case sensitive]

nano Config.php

[Opens the file Config.php using the nano editor. Remember the upper case C in Config.php]

Edit your file to reflect your root user for MySQL and the password for that user.



```
root@ubuntulamp: /var/www/churchinfo/Include
GNU nano 2.0.2      File: Config.php      Modified
<?php
/*****
 *
 * filename      : Include/Config.php
 * website      : http://www.churchdb.org
 * description   : global configuration
 *
 *****/

// Database connection constants
$sSERVERNAME = 'localhost';
$sUSER = 'root';
$sPASSWORD = 'yourrootpassword';
$sDATABASE = 'churchinfo';

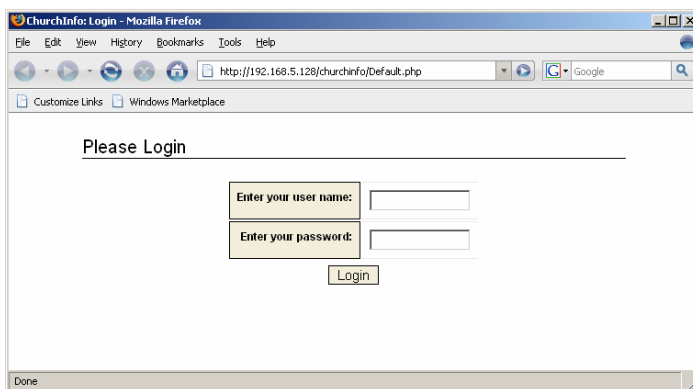
// Root path of your ChurchInfo installation ( THIS MUST BE SET CORRECTLY! )
// For example, if you will be accessing from http://www.yourdomain.com/web/chu$
// then you would enter '/web/churchinfo' here.
// Another example, if you will be accessing from http://www.yourdomain.com

^G Get Help  ^O WriteOut  ^R Read File  ^V Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is   ^N Next Page  ^U UnCut Text ^M To Spell
```

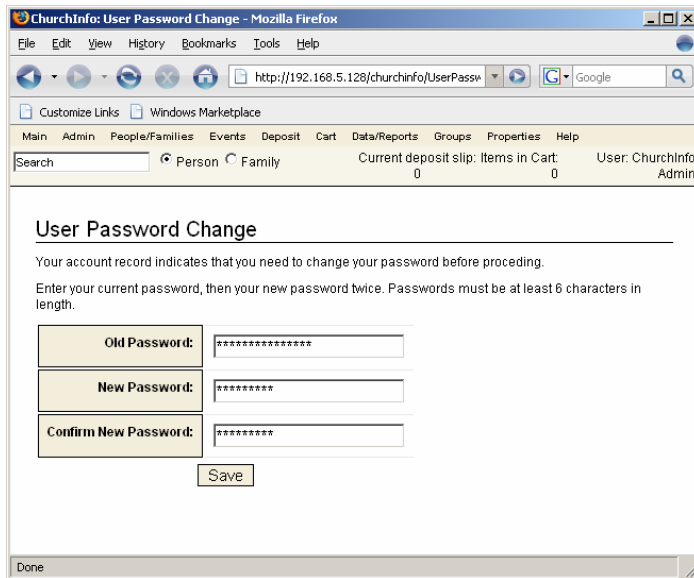
Hit ctrl+X, Y, and Enter to exit.

## Smoke Test

ChurchInfo should be installed now. Time to test. Go to the browser on your client desktop and enter the address for your Ubuntu server followed by a forward slash and churchinfo in the format <http://192.168.5.128/churchinfo> . If you see the screen below, all is well:



Congratulations! You have installed a church management system on your own Linux server! Log into Churchinfo using a user name of Admin (remember – case sensitive!) and a password of churchinfoadmin. Click the Login button to log in. The first order of business is security – ChurchInfo will have you change the default password right away. Enter churchinfoadmin as the old password, then enter the new password you want to use (make sure it's something you can remember.)



Click Save, and your password is changed. ChurchInfo will ask if you wish to change the stored password for root. For now you can say no. It's enough that you've gotten this far and you have a working church management system installed.

It's been quite a journey to get this far, hasn't it? Hopefully you've learned some things along the way. Now it's time to relax a bit and reflect on the journey – something that is worthwhile in our journey of faith.

## Documentation Notes

As stated at the start, this is version 0.1 of this document for installing ChurchInfo and using it. Using this document should enable you to get your server, support tools, and the ChurchInfo app up and running if everything goes perfectly.

This document is a work in progress, and I will be working on polishing up the text and dealing with more contingencies as the work continues. Eventually it is my hope to work with Mike Wilt, the developer for ChurchInfo, to provide some first rate documentation to the project in the form of a book that will be as freely downloadable and reproducible as the application itself.

If you would like to help with the documentation or would like to interact, please contact me at the email address below:

robinson dot mitchell at gmail dot com

The email address is not hyperlinked to help avoid spam. I would be happy to hear from you regarding whether this document has been helpful. Shortly I will try to make it available on a web site, but it's still a very early work in progress.

**This document (as ugly and unreadable as it is)**

**is copyrighted© 2007 by Robinson W. Mitchell.**

It is released under the Gnu Free Documentation License, the full text of which is available here: <http://www.gnu.org/licenses/fdl.txt> . The full text is reproduced below in a font that ought to be called old redeye.

GNU Free Documentation License  
Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc.  
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA  
Everyone is permitted to copy and distribute verbatim copies  
of this license document, but changing it is not allowed.

## 0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

## 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal,

commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

## 2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further

copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

### 3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

### 4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice

- giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
  - H. Include an unaltered copy of this License.
  - I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
  - J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
  - K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
  - L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
  - M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
  - N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
  - O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## 5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single



copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

## 6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

## 7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

## 8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

## 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

#### 10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

#### ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (c) YEAR YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.2
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
A copy of the license is included in the section entitled "GNU
Free Documentation License".
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.