

LedgerSMB on Mac OS X

An Installation Guide

Table of Contents

What Is LedgerSMB?	4
Prerequisites	4
Known Issues & Limitations	5
Installation	5
First Use	12
Additional Information	16
Notes	17

What Is LedgerSMB?

From the LedgerSMB website

LedgerSMB is an acronym for Ledger Small Medium Business.

Goal

The LedgerSMB project aims to bring

Highly secure

High performance

High quality

Easily useable

Easily accessible

ERP/accounting software to the small and midsize businesses to use as a foundation to build upon. It includes a flexible framework to support Rapid Application Development for custom extensions. The fact that LedgerSMB is open source - and that the source code is freely available - makes it even more customizable and flexible.

Prerequisites

LedgerSMB takes the approach of building on the foundations of other projects rather than reinventing the wheel. Because of that philosophy, the most complex part of installing LedgerSMB is not installing the application itself, but is getting all of the dependent bits of code in place. There is a long list of pre-requisite code.

- PostgreSQL 9.x - <http://www.postgresqlfromac.com/>
- Perl - installed by default
- Xcode - <http://developer.apple.com/>
- MacText - <http://mirror.ctan.org/systems/mac/mactex/MacTeX.pkg>

Before starting the installation, you will need to have Xcode downloaded at a minimum, but it would ideally have all of the above downloaded.

You will also need about 8gb of disk space for everything, so check your free space before you get started.

Known Issues & Limitations

At this time, the only known limitation is that the TrustCommerce credit card processing module will not install due to issues in the CPAN repository when installing on OS X.

Installation

As a note, these instructions are based upon Mac OS X 10.6.8. Some of the modules are now installed in OS X by default, however, there is no reason not to update them if needed.

Installation should take about 90 minutes on a reasonably current machine. In some cases, it may take longer if your internet connection is slow or your computer is otherwise busy while you are installing.

The easiest path to installation starts with Xcode. Making sure that you have a working command line build environment is critical to the success of installing LedgerSMB, as all of the Perl modules that need to be installed will require that build environment. Prior to Mountain Lion (10.8) any installation of Xcode installed the command line tools. Starting with Mountain Lion, it is necessary to install the Command Line Tools separately. This can be done through the Xcode Preferences -> Downloads dialog.

After installing Xcode, it is time to install PostgreSQL. Again, run the installer. The default options should be fine for setting the environment up.

Next, install the MacTeX package, again accepting the defaults is fine. This will install a ready to run and use LaTeX package. LaTeX is the engine for generating documents and reports out of LedgerSMB, so while this is listed as optional in the LedgerSMB documentation, it adds core functionality to the package and really should be considered a requirement.

Once that is successful, go ahead and reboot to ensure that the environment is all properly configured for both Xcode and PostgreSQL.

At this point, you are ready to dig into the nitty gritty of the installation. Installing all of the

Configuring the Perl Environment

CPAN is the package management system for Perl. Rather than spend hours downloading and building packages by hand, CPAN is here to get the job done with as little user interaction as possible. Each module will ask a few questions and then do it's

job. Please note though, cpan commands and names are case sensitive, and some of the modules may in fact be current depending upon your version of Mac OS X.

Unfortunately, the version of CPAN on 10.6.8 is fairly old, so the first task is to update CPAN. Like all of the Perl module install process, this will be done in Terminal, so start with opening Terminal and entering the following commands.

```
$ sudo cpan
cpan[1]> install cpan
```

You will have to answer several questions to start this installation. There is no need to choose anything but defaults, so simply hitting enter through them will handle everything. When it is complete, exit CPAN using the command exit.

```
cpan[1]> exit
```

Now you are ready to start the long process of installing all of the various Perl modules needed to LedgerSMB. So restart CPAN, start the installs.

Data::Dumper

```
cpan[1]> install Data::Dumper
```

Log::Log4perl

```
cpan[2]> install Log::Log4perl
```

Locale::Maketext

```
cpan[3]> install Locale::Maketext
```

Locale::Maketext::Lexicon

```
cpan[4]> install Locale::Maketext::Lexicon
```

DBI

```
cpan[5]> install DBI
```

MIME::Base64

```
cpan[6]> install MIME::Base64
```

Digest::MD5

```
cpan[6]> install Digest::MD5
```

HTML::Entities

```
cpan[7]> install HTML::Entities
```

DBD::Pg

Depending upon the version of OS X you are installing on, you may wish to exit CPAN to set up the environment to find the version of PostgreSQL you installed. Starting with 10.7, OS X includes the PostgreSQL client libraries. With them in place, you can simply install, but the tests for the install will fail as the server is not present.

On prior versions, the DBD::Pg install will ask for the path to pg_config if it cannot find it. When it does, simply provide the path to your pg_config. If you use the PostgreSQL for Mac installer, the path is as follows: /Library/PostgreSQL/bin/pg_config

```
cpan[8]> install DBD::Pg
```

Math::BigFloat

```
cpan[9]> install Math::BigFloat
```

IO::File

This is probably current, but worth attempting to install just in case

```
cpan[10]> install IO::File
```

Encode

```
cpan[11]> install Encode
```

Locale::Country

This installation typically also requires a large number of dependent or temporary installs, and may require user interaction to say yes to these installations.

```
cpan[12]> install Locale::Country
```

Locale::Language

May be current.

```
cpan[13]> install Locale::Language
```

Time::Local

```
cpan[14]> install Time::Local
```

Cwd

```
cpan[15]> install Cwd
```

Config::Std

```
cpan[16]> install Config::Std
```

MIME::Lite

May require user input to accept several required sub modules, accept the default answers to the questions to complete the install.

```
cpan[17]> install MIME::Lite
```

Template

May require user input to accept several required sub modules, accept the default answers to the questions to complete the install.

```
cpan[18]> install Template
```

Error

```
cpan[19]> install Error
```

CGI::Simple

```
cpan[20]> install CGI::Simple
```

File::MimeInfo

```
cpan[21]> install File::MimeInfo
```

DateTime

```
cpan[22]> install DateTime
```

At this point you have the bare minimum bits to run LedgerSMB, but you can include the following to gain some extra features.

Net::TCLink

Support for TrustCommerce credit card processing. Currently known to fail installation on OS X. Problem appears to be with TCLink itself, not OS X environment.

Parse::RecDescent

Support for the *experimental* scripting engine

```
cpan[23]> install Parse::RecDescent
```

Template::Plugin::Latex

Support for Postscript and PDF output. Requires a full LaTeX installation to be already installed. The MacTeX package is perfect and can be obtained from the pre-requisites link, or <http://mirror.ctan.org/systems/mac/mactex/MacTeX.pkg> (large, 2gb). LaTeX must be installed before the cpan installation can succeed.

```
cpan[24]> install Template::Plugin::Latex
```

XML::Twig

Support for OpenOffice output. Requires a couple of answers to questions, but the defaults are fine.

```
cpan[25]> install XML::Twig
```

Excel::Template::Plus

Support for Excel output.

```
cpan[26]> install Excel::Template::Plus  
cpan[27]> quit
```

That is all of the Perl configuration required. Now it is time to set up the LedgerSMB application itself, starting with the Apache configuration, and then the LedgerSMB application itself.

Configure Apache

Ensure that both Perl and mod_rewrite are configured in your Apache2 configuration. Start by editing the file /etc/apache2/httpd.conf to ensure that the line:

```
LoadModule rewrite_module libexec/apache2/mod_rewrite.so
```

does not have a # at the beginning of the line. If it does, remove that #, save the file and restart Apache. Since we are now working in an area of the system that most Mac users have little experience, we are going to go back to the Terminal to make these changes.

```
$ cd /etc/apache2
$ sudo vim ./httpd.conf
```

Now using the arrow keys, navigate down until you find the mod_rewrite line. In a default configuration, this will be about 100-120 lines into the file. If there is a # in the first position of the mod_rewrite line, move the cursor using the arrow keys to the #, and press the x key to remove the #. Now press the : followed by wq and hit enter to save and exit.

Configure LedgerSMB

At this point, Apache itself is ready and it is time to configure LedgerSMB in the Apache environment. Copy the LedgerSMB folder from wherever you have saved it. If it is still a .tar file, you can readily move it, and expand it.

```
$ sudo cp ~/Downloads/ledgersmb-1.3.21.tar /Library/
$ cd /Library
$ sudo tar xf ledgersmb-1.3.21.tar
$ sudo chown -R root:wheel ledgersmb
$ cd ledgersmb
$ cp ledgersmb-httpd.conf.template ledgersmb-http.conf
$ sudo vim ledgersmb-http.conf
```

Using the arrow keys, replace all instances of WORKING_DIR with the path /Library/ledgersmb/.

Just a reminder, in VIM, use the x key to delete a single character at the location of the cursor. To go into 'edit' mode so you can type, hit the i key. When you have finished typing, press the esc key to exit edit mode to move to the next item to delete and change. If you make a mistake, you can always hit esc, then : followed by q! and enter to quit without saving changes and start over. When you have completed all your changes, hit esc, then : followed by wq and enter to save your changes and exit.

With your changes complete, you are ready to move the ledgersmb-http.conf file into place to tell Apache to include it.

On Mac OS X10.6 or 10.7, this goes into the /etc/apache2/sites directory, on 10.8 however, sites has been renamed other.

10.6.x & 10.7.x

```
$ sudo mv ledgersmb-http.conf /etc/apache2/sites
```

10.8.x

```
$ sudo mv ledgersmb-http.conf /etc/apache2/other
```

Restart apache.

```
$ sudo apachectl restart
```

Create your ledgersmb.conf to tailor to your installation

```
$ cd /Library/ledgersmb  
$ sudo cp ledgersmb.conf.default ledgersmb.conf  
$ sudo vim ledgersmb.conf
```

Navigate down into file to make the changes needed to tailor the installation to your needs. First move to about line 50 and add the PostgreSQL bin path to the path line

```
PATH: /bin:/usr/bin:/usr/local/bin:/usr/local/pgsql/bin:/Library/  
PostgreSQL/bin
```

After that change is made, you also need to adjust your default_db entry to reflect the name of your database. You can set the default_db to blank if you wish the application to prompt the user for a database/company with each login.

```
default_db = yourdatabasename
```

The final change required in the ledgersmb.conf file is the path to the PostgreSQL contrib folder. This line is very close to the end of the file at about line 95.

```
contrib_dir = /Library/PostgreSQL/share/postgresql/extension
```

Configure PostgreSQL

Running the PostgreSQL for Mac installer will get you a ready to run PostgreSQL server. At this point, the only setup that PostgreSQL needs to be ready to work with LedgerSMB is to alter its security settings, and super user password to something that works with LedgerSMB. First off, set a stronger password on the postgres user (the default installation does not have a password).

```
$ /Library/PostgreSQL/bin/psql -U postgres
postgres=# ALTER USER Postgres WITH PASSWORD '<newpassword>';
postgres=# \q
```

Then alter the security configuration to work best with LedgerSMB

```
$ sudo vim /Library/PostgreSQL/data/pg_hba.conf
```

in the pg_hba.conf file, add a line to encourage all local connections to use md5, before the equivalent trust line.

```
# IPv4 local connections:
host    all             all             127.0.0.1/32     md5
host    all             all             127.0.0.1/32     trust
```

with the key element being the last md5 bit. The default is usually trust, which is far less secure than md5.

It is time to restart PostgreSQL, and do the first use configuration.

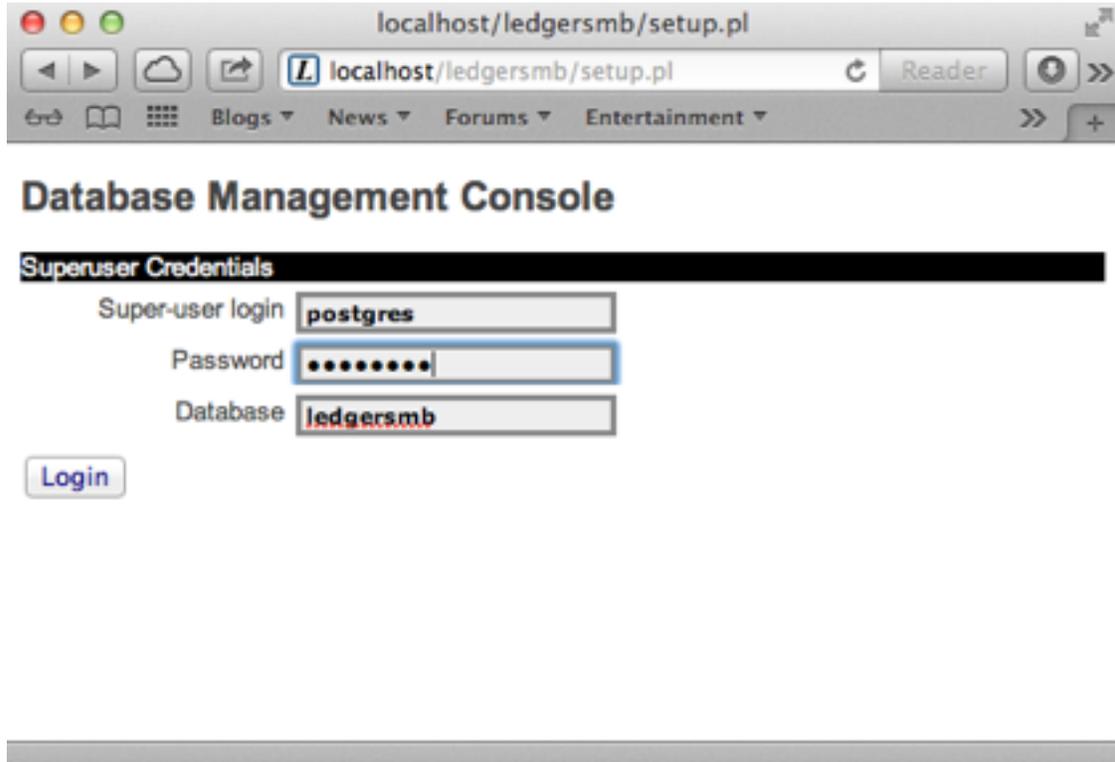
```
$ sudo /Library/StartupItems/PostgreSQL/PostgreSQL restart
```

First Use

Once everything has been installed, it is time to run the internal setup script to finish up the installation and make sure it is ready to use. This is done in a web browser on the server, as by default the web app is limited to local machine usage.

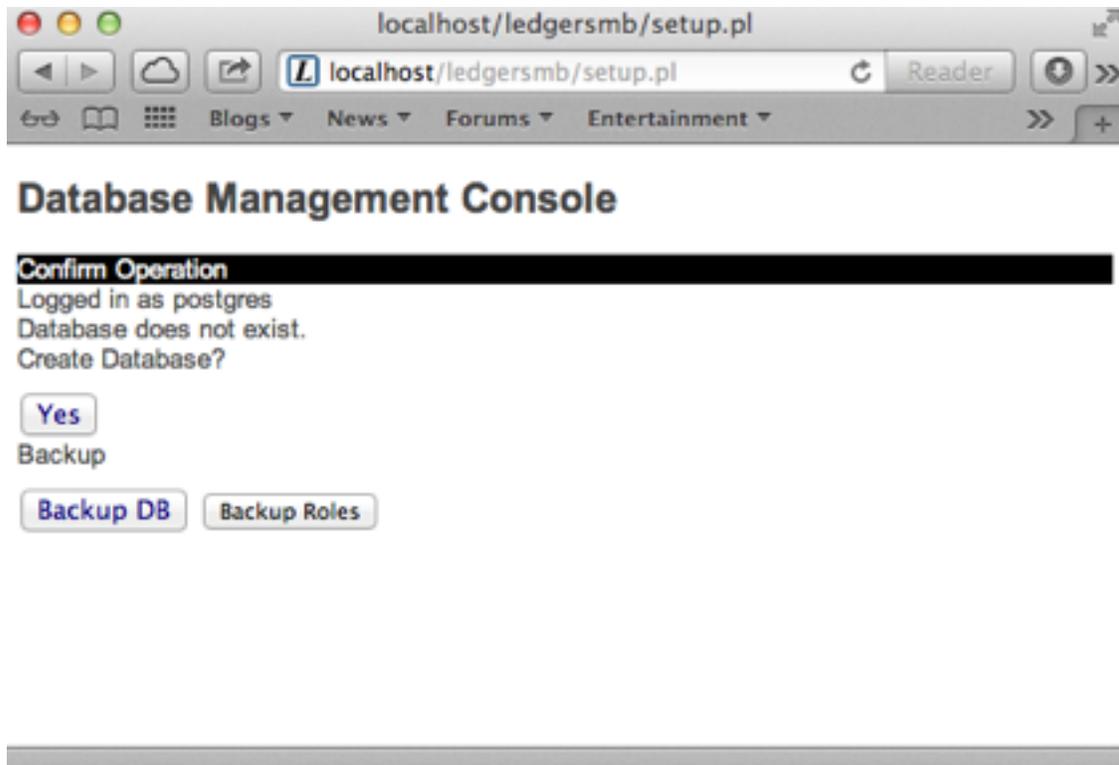
Open a web browser and navigate to <http://localhost/ledgersmb/setup.pl> to start the first use setup.

You should see the following window:



The screenshot shows a web browser window with the address bar containing `localhost/ledgersmb/setup.pl`. The page title is "Database Management Console". Below the title, there is a section titled "Superuser Credentials" with a black background. Underneath, there are three input fields: "Super-user login" with the value "postgres", "Password" with a masked password of ten dots, and "Database" with the value "ledgersmb". A "Login" button is located below the input fields.

Enter in the postgres user as the super user, using the password you set above as the <newpassword> and the database name you wish to use. If it does not exist, it will be created for you. Click login.



If the database doesn't exist, tell the setup to create it and select Next.



Database Management Console

Country Code

[Next](#)

[Skip](#)

Select your country code from the drop down list and select Next.

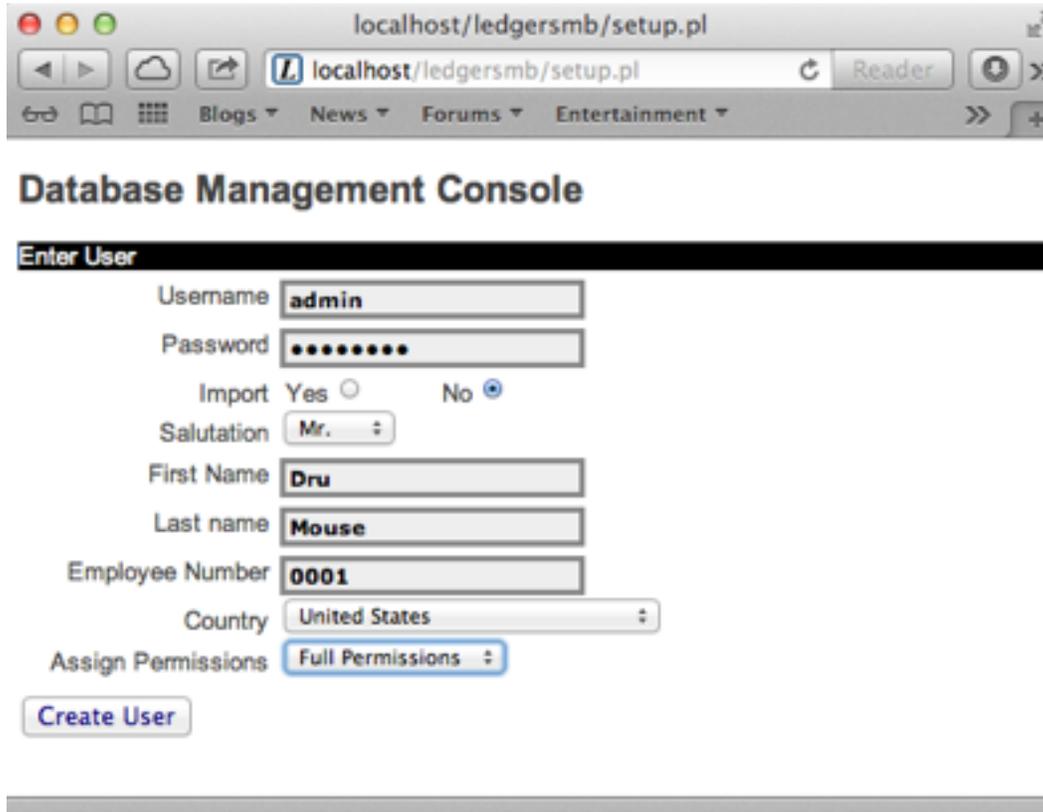


Database Management Console

[Next](#)

[Skip](#)

Select the closest match to your business type from the drop down and select Next.



The screenshot shows a web browser window with the URL localhost/ledgersmb/setup.pl. The page title is "Database Management Console". Below the title is a section titled "Enter User" with a black background. The form contains the following fields and options:

- Username:
- Password:
- Import: Yes No
- Salutation:
- First Name:
- Last name:
- Employee Number:
- Country:
- Assign Permissions:

At the bottom of the form is a button labeled "Create User".

Fill in the user information to create your administration user account and Create User to create your initial user.

And finally, select Start Using LedgerSMB

Additional Information

If you wish to allow access to your LedgerSMB installation from computers other than the localhost, you have to alter the ledgersmb-http.conf file in sites/other to allow this. The default configuration will deny all hosts other than localhost and will look very similar to the this

```
# By default, only allow from localhost. If you change this, please be
# advised that you should use SSL protection on any and all network
# connections that access this application in order to protect usernames
# and passwords.
```

```
Order Deny,Allow
Allow from 127.0.0.1
Allow from localhost
Deny from All
```

Expanding this to allow from a specific network, you can simply add an allow line with a partial address as follows

```
Allow from 192.168.0
```

or by name

```
Allow from mydomainname.local
```

It will be necessary to restart Apache after making any change to this file to take effect.

Notes

If you worked through this and are still having problems, you can reach the author at dru@druware.com. As the packager of PostgreSQL for Mac over the last several years, we have seen all kinds of strange things, we will be happy to try and help as much as we can. We are also available for consulting on PostgreSQL related issues.