

LedgerSMB on Ubuntu

John O’Gorman (john@og.co.nz)

13 May 2020

1 Intro

This identifies the actions needed to install LedgerSMB v1.7 on Ubuntu v 19.10. What works on Ubuntu should also work on Debian.

Last week we found the solution to our local-lib problems thanks to Eric Huelsmann. We needed to provide the LedgerSMB directory (containing cpanfile) to the cpanm command line.

We are now trying to setup the Virtual Host for LedgerSMB in lsmb.localhost. It emerges that apache on Ubuntu is very different from on Suse. We have now got the vhost stuff working OK and are about to attack the SSL and Rewrite stuff. We are getting there!

The main thing we have learned is that in Ubuntu Apache2 has hard-coded some of its acceptance of DocumentRoot assignments to /var/www and seemingly refuses to allow /srv/ etc and other things to be accessed. We think we have got round this by

1. making a symbolic link from /opt/ledgersmb to /var/www/ledgersmb/ and
2. setting lsmb.localhost’s DocumentRoot to /var/www/ledgersmb. This seems to be OK for the moment.

We will now work on invoking SSL and finally Rewrite modules.

1.1 Debian

We started out attempting to install LedgerSMB on Debian. We purchased a new laptop computer an Acer Aspire 3 with a 17” screen expecting that we would have no problems adjusting its BIOS to accept dual booting using PXE to install Debian. We were wrong! After solving some of our problems we found that Debian could not (or we could not) get the Debian to configure the graphics correctly for X. So we abandoned that and installed Ubuntu in its place. That seems to have worked fine.

1.2 Ubuntu

Ubuntu is similar (we think) to Debian. Our experience of Linux in recent years is of Suse in its various forms. An important difference is that on Suse we install packages using Yast and Zypper on packages with a .rpm format (Redhat Package Manager). Ubuntu does not have these and in their place uses Aptitude (a graphical interface) and apt-get (for command line use) on packages with a .deb format (initiated by Debian).

1.3 LedgerSMB

LedgerSMB is an opensource double-entry bookkeeping application which

- is browser based. With Linux this defaults to Firefox with support for Dojo javascript
- uses postgresSQL as its database engine

- is written using perl and javascript. The developers are slowly working at changing from perl in the backend to javascript so that the browser can carry the load of execution at the frontend via Javascript

In the long term the developers intend to reorganise the software to adopt a model-view-controller design pattern which will simplify design and error capture.

1.4 Prerequisites

The following are needed: postgresQL, LaTeX, perl, cpanminus, starman, Plack, apache, and a browser compatible with Dojo 1.15

- Firefox which comes with Dojo support built-in
 - Apache a local webserver for Firefox
- PostgreSQL relational database engine
- LaTeX typesetting program
- Perl programming language
 - local::lib a way to hold private copies of perl modules
 - cpanminus an improved way to access CPAN (Comprehensive Perl Archive Network). It will install prerequisite modules if asked (via the `-installdeps` argument).
- starman a perl server gateway interface (PSGI) between webserver and browser
- Plack likewise

You may already have some/many/all of these on your distro of Linux. You can test them using the 2nd command(s) and ignore the install command.

1.5 Versions

	min version	command	output on Ubuntu
Firefox	3.6	firefox -v	Mozilla Firefox 75.0
Apache	2	apachectl -v	Apache/2.4.41 (Ubuntu)
PostgreSQL	9.4	psql -v	11.7
perl	5.14	perl -v	5.28.1
latex		latex -v	pdfTeX 3.14159265-261.30.20
Starman		starman -v	0.4015

The README.md file gives the versions needed for prerequisites

2 Installation

There is a file: README.md in the ledgerSMB directory or website which gives necessary information on how to install ledgerSMB.

On Linux systems you usually need to be logged in as root or be able to run the command: sudo. On Ubuntu it seems that you cannot login as root. To become root you must enter the command:

```
sudo su
```

It will prompt you for your own password. Then you can have a session as root until you type:

```
exit
```

Alternatively for a single command you can run the command: `sudo` followed by the command. In the example commands below I have implied you must be root by prefixing the example commands with the `#` symbol.

Where it is OK to run a command as an ordinary user the prefix is `$`.

You do not type either the `#` or the `$` when entering the command.

For each prerequisite below, we have provided 2 shell commands: the 1st is the install command, the 2nd is a test to confirm the install worked. To assist in finding files that should be installed at different stages you will benefit from installing the command: `locate`

```
# apt-get install mlocate
$ which locate
```

I have also used my own home directory in examples: `/home/john`

I have also used the directory `/opt` to install from the `lsmb` tarball.

You make you own choices for these 2 decisions.

2.1 firefox

```
# apt-get install firefox
$ firefox -v
```

2.1.1 apache

```
# apt-get install apache
$ apachectl -v
```

2.2 postgresql

```
$ aptitude
```

Search for `postgresql` and select all desired related aps such as `DBI`, `DBD`, `DBD-Pg` etc

```
$ psql --version
$ locate DBD.pm
$ locate DBI.pm
$ locate Pg.pm
$ grep postgres /etc/passwd
```

2.3 perl

```
# apt-get install perl
$ perl -v
```

2.3.1 make

```
apt-get install make
make -v
```

2.4 gcc

```
apt-get install gcc
gcc --version
```

2.5 cpanm

```
apt-get install cpanminus
cpanm -v
```

2.6 local::lib

```
# apt-get install liblocal-lib-perl
$ locate lib.pm
```

The local/lib.pm module allows you to have a private installation of cpanm modules (perhaps because they have differences from the official versions)

To do this change to the home directory (in my case /home/john). Then run the shell command:

```
$ perl -Mlocal::lib
```

Its output will be something like this:

```
PATH="/home/john/perl5/bin${PATH:+:${PATH}}"; export PATH;
PERL5LIB="/home/john/perl5/lib/perl5${PERL5LIB:+:${PERL5LIB}}"; export PERL5LIB;
PERL_LOCAL_LIB_ROOT="/home/john/perl5${PERL_LOCAL_LIB_ROOT:+:${PERL_LOCAL_LIB_ROOT}}";
export PERL_LOCAL_LIB_ROOT;
PERL_MB_OPT="--install_base \"/home/john/perl5\""; export PERL_MB_OPT;
PERL_MM_OPT="INSTALL_BASE=/home/john/perl5"; export PERL_MM_OPT;
If this seems OK then run the command again and capture its output in a file (in the example
= envlocallib).
```

```
$ perl -Mlocal::lib > envlocallib
```

Use the shell . (dot) command to execute the commands in envlocallib

```
$ . ./envlocallib
```

If you use a csh command for your shell then replace the 1st dot . with source:

```
$ source ./envlocallib
```

To check this type:

```
env | grep PERL
echo $PATH
```

Run the command as user john (or whoever)

```
$ perl -MCPAN -Mlocal::lib -e 'CPAN::install(LWP)'
ls -R perl5
```

2.7 LyX and LaTeX

LedgerSMB does not need LyX but we could not live without it. It is a graphical frontend to LaTeX.

```
$ aptitude
```

Install LyX and LaTeX and any dependants

```
$ lyx -version
```

2.8 LedgerSMB

To install ledgersmb:

- Decide where: e.g. /opt
- Create a user: e.g. lsmbadmin
- download the tar.gz file from website: ledgersmb.org
- run the tar xvfz command
- If necessary run command `chown -r ledgersmb`

2.8.1 /opt

If /opt does not exist, create it:

```
$ sudo su
# mkdir /opt
```

2.8.2 user lsmbadmin

```
# useradd lsmbadmin
# groupadd lsmb
# exit # from sudo
```

2.8.3 Download ledgersmb-1.7.11

Browse the website: ledgersmb.org

Go to download/f/

Select the version you want.

Double click the filename, choose file.

Run the command:

```
$ cd Downloads
# tar xvfz ledgersmb.tar.gz --directory /opt
# ls -lR /opt
```

2.9 Starman and Plack

```
$ cd /home/john
$ cpanm --quiet --notest \
    --with-feature=starman \
    --with-feature=latex-pdf-ps \
    --with-feature=latex-pdf-images \
    --installdeps /opt/ledgersmb

$ which starman
$ which plackup
$ locate Plack.pm
$ locate Starman
```

The `cpanm` command will look in the `ledgersmb` directory, find the file: `cpanfile` which contains a list of target modules each with a list of their prerequisites. It will then, if all goes well, install them into your `~/perl5` directory.

This is a sample of some of the lines in the `cpanfile`

```

#!/perl
requires 'perl', '5.18.0';
requires 'CGI::Emulate::PSGI';
requires 'CGI::Parse::PSGI';
requires 'Config::IniFiles';
requires 'DBD::Pg', '3.3.0';
requires 'DBI', '1.635';
requires 'Data::UUID';
requires 'DateTime';
-----
requires 'PGBObject', '1.403.2';
# PGBObject::Simple 3.0.1 breaks our file uploads
requires 'PGBObject::Simple', '>=3.0.2';
requires 'PGBObject::Simple::Role', '2.0.2';
requires 'PGBObject::Type::BigFloat', '1.0.0';
requires 'PGBObject::Type::DateTime', '1.0.4';
requires 'PGBObject::Type::ByteString', '1.1.1';
requires 'PGBObject::Util::DBMethod';
requires 'PGBObject::Util::DBAdmin', '1.0.1';
-----
recommends 'Math::BigInt::GMP';
feature 'latex-pdf-ps', "PDF and PostScript output" =>
sub {
requires 'LaTeX::Driver', '0.300.2';
requires 'Template::Latex', '3.08';
requires 'Template::Plugin::Latex', '3.08';
requires 'TeX::Encode';
};

```

You can see that the latex-pdf-ps is defined here. It isn't in the CPAN archives and that's why it took me 3 days to find it! You have to put the path to the ledgersmb directory at the end of the cpanm commands or cd there and put . as the trailing argument, otherwise it searches the CPAN archives.

I defy anyone to find this in official cpanm documentation!

3 Configuration

3.1 Postgresql

The installation of postgresQL will have resulted in a Linux super-user: postgres which has total control of the postgresql clusters.

3.1.1 Create lsmb administrator postgresql user

It is recommended that we create a separate postgresQL user to administer the ledgerSMB: This user needs:

- right to create databases
- no superuser rights (e.g. to drop clusters)

- right to login
- right to add postgresql users (called roles)
- a password to authenticate logins

Enter the command:

```
# sudo -u postgres createuser \
    --no-superuser --createdb \
    --login --createrole \
    --pwprompt lsmbadmin
```

This will prompt twice for the lsmbadmin's new password:

```
Enter password for new role: ****
Enter it again: ****
```

Test this:

```
# su - postgres -c 'createdb mytestdb'
psql -h localhost -U lsmbadmin \
    -d mytestdb -c 'select version()'
```

This should succeed and display something like:

```
PostgreSQL 11.7 (Ubuntu ..... ) 20191008, 64-bit
```

You can drop the empty database mytestdb

```
# su - postgres -c 'drop mytestdb'
```

3.2 Configure database access in pg_hba

Find the file pg_hba

```
# locate pg_hba
```

This produces output like:

```
/etc/postgresql/11/main/pg_hba.conf
```

The last lines of this file are:

```
# IPv4 local connections:
host all all 127.0.0.1/32 md5
host all all 192.168.168.0/24 md5
# IPv6 local connections:
host all all ::1/128 md5
# Allow replication connections from localhost, by a user with the
# replication privilege.
local replication all peer
host replication all 127.0.0.1/32 md5
host replication all ::1/128 md5
```

LedgerSMB recommends that these lines be altered to the following:

```

local all postgres peer
local all all peer
host all postgres 127.0.0.1/32 reject
host all postgres ::1/128 reject
host postgres,template0,template1 lsmb_dbadmin 127.0.0.1/32 md5
host postgres,template0,template1 lsmb_dbadmin ::1/128 md5
host postgres,template0,template1 all 127.0.0.1/32 reject
host postgres,template0,template1 all ::1/128 reject
host all all 127.0.0.1/32 md5
host all all ::1/128 md5

```

In order to complete an installation as quickly as possible we are skipping this section. We will return to this section 3.2 to implement the security options later when we have LedgerSMB working.

3.2.1 Restart PostgreSQL

```
$ service postgresql restart
```

3.3 Starman

3.3.1 Create user ledgersmb

```

# useradd -d /non-existent -r -U \
    -c "LedgerSMB/Starman service system user" \
    ledgersmb

```

The -d /non-existent argument was ignored!
 Instead use the argument: -M or --no_create-home

3.3.2 Copy file ledgersmb_starman.service

Find file in tarball. In our case its in /opt/ledgersmb/doc/conf/systemd.
 Copy it to /etc/systemd/system

```

# cd /opt/ledgersmb/doc/conf/systemd
# cp ledgersmb_starman.service /etc/systemd/system
# cd /etc/systemd/system
# vi ledgersmb_starman.service

```

Edit it changing WorkingDirectory= /opt/ledgersmb
 Edit it changing Environment=PERL5lib=/home/john/perl5/lib/perl5
 Edit it changing ExecStart=/usr/local/bin

3.3.3 Enable Starman and start it

```

$ systemctl enable ledgersmb_starman
$ service ledgersmb_starman start
$ journalctl -u ledgersmb_starman.service \
    --since="today" -l -e

```

Output was:


```
2020/04/27-14:56:55 Starman::Server (type Net::Server::PreFork)
starting! pid(8190)
Resolved [localhost]:5762 to [127.0.0.1]:5762, IPv4
Binding to TCP port 5762 on host 127.0.0.1 with IPv4
Setting gid to "997 997 997"
```

Alleluia!

3.4 Apache

Apache2 is the application which most of the world's websites use to serve data to browsers. The version we found on Ubuntu is apache2.4.41 which is radically different to what we have become familiar with.

Its configuration is read from directory: `/etc/apache2/` which contains the following files: `apache2.conf`, `envvars`, `magic`, `ports.conf`. It also contains the following pairs of directories: `conf-available` and `conf-enabled`, `mods-available` and `mods-enabled`, and `sites-available` and `sites-enabled`.

3.4.1 The files in `/etc/apache2`

3.4.1.1 `apache2.conf` We have removed the comment lines from the file and inserted a directory block for `/opt/ledgersmb`.

```
-----
PidFile ${APACHE_PID_FILE}
Timeout 300
KeepAlive On
MaxKeepAliveRequests 100
KeepAliveTimeout 5
# These need to be set in /etc/apache2/envvars
User ${APACHE_RUN_USER}
Group ${APACHE_RUN_GROUP}
HostnameLookups Off
ErrorLog ${APACHE_LOG_DIR}/error.log
LogLevel warn
IncludeOptional mods-enabled/*.load
IncludeOptional mods-enabled/*.conf
Include ports.conf
<Directory />
Options FollowSymLinks
AllowOverride None
Require all denied
</Directory>
<Directory /usr/share>
AllowOverride None
Require all granted
</Directory>
<Directory /var/www/>
Options Indexes FollowSymLinks
AllowOverride None
Require all granted
</Directory>
<Directory /opt/ledgersmb/>
Options Indexes FollowSymLinks
AllowOverride None
Require all granted
```

```

</Directory>
#<Directory /srv/>
# Options Indexes FollowSymLinks
# AllowOverride None
# Require all granted
#</Directory>
AccessFileName .htaccess
<FilesMatch "\.ht">
Require all denied
</FilesMatch>
LogFormat "%v:%p %h %l %u %t \"%r\" %>s %0 \"%{Referer}i\" \"%{User-Agent}i\"" vhost_combined
LogFormat "%h %l %u %t \"%r\" %>s %0 \"%{Referer}i\" \"%{User-Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %0" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent
# Include of directories ignores editors' and dpkg's backup files,
# see README.Debian for details.
IncludeOptional conf-enabled/*.conf
IncludeOptional sites-enabled/*.conf
-----

```

3.4.1.2 envvars This configures the environment variable values which will be used by apache2:

We have removed the comments.

```
# envvars - default environment variables for apache2ctl
```

```

unset HOME
if [ "${APACHE_CONFDIR##/etc/apache2-}" != "${APACHE_CONFDIR}" ] ; then
SUFFIX="-${APACHE_CONFDIR##/etc/apache2-}"
else
SUFFIX=
fi
export APACHE_RUN_USER=www-data
export APACHE_RUN_GROUP=www-data
export APACHE_PID_FILE=/var/run/apache2${SUFFIX}/apache2.pid
export APACHE_RUN_DIR=/var/run/apache2${SUFFIX}
export APACHE_LOCK_DIR=/var/lock/apache2${SUFFIX}
export APACHE_LOG_DIR=/var/log/apache2${SUFFIX}
export LANG=C
export LANG

```

To benefit from the www-data values you will need to install lynx if it isn't on your system:

```
apt-get install lynx
```

3.4.1.3 magic You don't need to look at this file. It was used by the file command to identify what sort of file by looking at the top line(s).

3.4.1.4 ports.conf

```

# If you just change the port or add more ports here, # you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf
Listen 80

```

```

<IfModule ssl_module>
Listen 443
</IfModule>
<IfModule mod_gnutls.c>
Listen 443
</IfModule>

```

The Ubuntu apache2 v 2.4.41 seems more intolerant of multiple Listen statements in various .conf files. Get prepared to comment out listen statements in the ledgersmb supplied config files if error messages imply listen problems.

3.4.2 etc/apache2/ directories

These directories are in pairs which deal with conf(iguration)s, mod(ule)s, and sites. The idea is that

- you put your conguration files into the appropriate -available file: confs-available, mods-available, and sites-available.
- you use the command a2enconf, a2enmod, or a2ensite to enable the intended .conf file. The a2en- commands create a symbolic link of your nominated file into the -enabled directory
- you run the command: apachectl restart

This simplifies the configuration but there are a lot of interrelated complications in the new layout.

So we recommend that you make implementation changes 1 at a time, test them using the command: apachectl configtest, fix any reported errors, and then move on to the next change.

3.4.3 ledgersmb.conf

Find the file: /opt/ledgersmb/doc/conf/ledgersmb.conf.default

Copy it to /opt/ledgersmb

e.g.

```

# cd /opt/ledgersmb/doc/conf
# cp ledgersmb.conf.default \
    /opt/ledgersmb.ledgersmb.conf
# vi /opt/ledgersmb/ledgersmb.conf

```

Make changes to values:

3.4.4 ledgersmb apache-vhost.conf

There is a file in the ledgersmb tarball which is a template for adding a virtual host to the Apache configuration. It is

```

/opt/ledgersmb/doc/conf/apache-vhost.conf:
# This is a 'vhost' definition file example for use with Starman/LedgerSMB
# reverse proxying.
#
# Please replace the following parameters:
#
# * WORKING_DIR
# * YOUR_SERVER_NAME

```

```

# * SSL_KEY_FILE
# * SSL_CERT_FILE
# * SSL_CHAIN_FILE
#
#
# this block also requires mod_ssl and mod_rewrite to be enabled
# Comment out the 'Listen' and/or 'NameVirtualHost' when Apache complains
Listen 443
# NameVirtualHost is ignored by Apache 2.4
NameVirtualHost *:443
<VirtualHost *:443>
ServerName YOUR_SERVER_NAME
DocumentRoot WORKING_DIR/UI
# If you own a publicly exposed server, consider submitting it
# to the SSL security tests available at
# https://www.ssllabs.com/ssltest/
SSLEngine On
SSLCertificateFile SSL_CERT_FILE
SSLCertificateKeyFile SSL_KEY_FILE
SSLCertificateChainFile SSL_CHAIN_FILE
<Location "/">
SSLRequireSSL
</Location>
RewriteEngine On
# Rewrite '/' URL to /login.pl script
RewriteRule "^/$" "/login.pl" [R=301,L]
# "hidden" files (those starting with a dot), don't exist
RewriteRule "^/\." - [R=404,L]
# configuration files (those ending in '.conf'), don't exist
RewriteRule "\.conf$" - [R=404,L]
# Rewrite non-static content to the application backend
RewriteCond "%{REQUEST_FILENAME}" !-f
RewriteCond "%{REQUEST_FILENAME}" !-d
RewriteRule "^/(.*)" "http://localhost:5762/$1" [P]
ProxyPassReverse "/" "http://localhost:5762/"
</VirtualHost>

```

To explain the RewriteRule statements:

- The syntax is: Pattern Substitution Code
- Pattern is a Regular Expression (RE) where
 - (...) delimits a string returned as \$1,\$2,.. and \$0 = whole string
 - [...] delimits a character sequence where
 - * + = 1 or more chars
 - * ? = 0 or more chars
 - * [0-9] + denotes 1 or more chars in range 0-9
 - * \ backslash escapes any of the preceding RE meanings
 - * . means any 1 char
 - * * means zero or more of preceding RE
 - * + means 1 or more of preceding RE
 - within an RE ^ means the beginning of the line
 - within an RE \$ means the end of the line

To explain the RewriteCond: It means execute the following RewriteRule when the RewriteCond arguments are true

- RewriteCond %{FILENAME} !-f means FILENAME is not a file
- RewriteCond %{DIRNAME} !-d means DIRNAME is not a dir
- %1-9 means back references to REs in order

Flags e.g. [FLAG] or [Flag1, FLAG2, etc]

- F = Forbidden. Send HTML code 403
- N =Next. Restart Rewrite rule
- R = Requested file has moved code 302
- L = Last don't process any more Rewrite rules
- NC = No Case i.e. case insensitive
- P = Proxy. Apache grabs remote content in substitution
- R = Redirect. R=401, or R=301 permanent

3.4.5 SSL

I suggest that initially you comment out all the lines relating to SSL. When you have the ledgersmb working without the security you can re-edit the file:lsmb-vhost and uncomment the SSL references.

You can create and certify your own ssl files using openssl commands.

The example below creates for og the files og.key, og.csr, and og.crt:

- .key = key
- .csr = certificate signing request
- .crt = certificate

```
openssl genrsa -aes128 -out og.key 2048
openssl rsa -text -in og.key
openssl req -new -key og.key -out og.csr
openssl req -text -in og.csr -noout
openssl req -new -x509 -days 365 -key og.key \
    -out og.crt
openssl x509 -text -in og.crt -noout
```

3.4.6 Install the apache-vhost.conf file

Copy the file to /etc/apache2/sites-available/ledgersmb.conf.

Edit the file changing:

1. WORKING_DIR to /opt/ledgersmb
2. YOUR_SERVER_NAME to lsmb.localhost

```

# This is a 'vhost' definition file example for use with Starman/LedgerSMB
# reverse proxying.
#
# Please replace the following parameters:
#
# * WORKING_DIR
# * YOUR_SERVER_NAME
# * SSL_KEY_FILE
# * SSL_CERT_FILE
# * SSL_CHAIN_FILE
#
#
# this block also requires mod_ssl and mod_rewrite to be enabled
# Comment out the 'Listen' and/or 'NameVirtualHost' when Apache complains
Listen 443
# NameVirtualHost is ignored by Apache 2.4
NameVirtualHost *:443
<VirtualHost *:443>
ServerName YOUR_SERVER_NAME
DocumentRoot WORKING_DIR/UI
# If you own a publicly exposed server, consider submitting it
# to the SSL security tests available at
# https://www.ssllabs.com/ssltest/
SSLEngine On
SSLCertificateFile SSL_CERT_FILE
SSLCertificateKeyFile SSL_KEY_FILE
SSLCertificateChainFile SSL_CHAIN_FILE
<Location "/">
SSLRequireSSL
</Location>
RewriteEngine On
# Rewrite '/' URL to /login.pl script
RewriteRule "^/$" "/login.pl" [R=301,L]
# "hidden" files (those starting with a dot), don't exist
RewriteRule "^/\." - [R=404,L]
# configuration files (those ending in '.conf'), don't exist
RewriteRule "\.conf$" - [R=404,L]
# Rewrite non-static content to the application backend
RewriteCond "%{REQUEST_FILENAME}" !-f
RewriteCond "%{REQUEST_FILENAME}" !-d
RewriteRule "^/(.*)" "http://localhost:5762/$1" [P]
ProxyPassReverse "/" "http://localhost:5762/"
</VirtualHost>

```

Edit the copy changing

- WORKING_DIR to /opt/ledgersmb (or maybe /var/www/ledgersmb)
- YOUR_SERVER_NAME to lsmc.localhost # eileen
- Comment out all references to SSL (for now)
- Comment out the Listen 443 statement
- Comment out NameVirtualHost *:443

You may have noticed that files: login.pl and setup.pl are missing from the tarball. Attempts to point your browser to these result in triggering the RewriteRule above to invoke the browser Proxy via the 5762 port which redirects via Starman to /opt/LedgerSMB/lib/PSGI.pm

We owe this knowledge to Erik Huelsmann.

3.4.7 Useful diagnostic commands

```
# apachectl configtest # or
# apachectl -t # same as above
# apachectl status # this only works if you have lynx
# apachectl -X -D DUMP_VHOSTS # for debugging vhosts
# apachectl -V # which lists the compiled defaults of apache2
```

There is also a command `apache2ctl` which is a symbolic link to `apachectl`.

3.4.8 Activate the vhost

Apache provides pairs of commands to enable (`a2en-`), disable (`a2dis-`), followed by `mod` (`module`), `site`, and `conf` for `.conf` files

```
# a2ensite ledgersmb.conf
# a2enmod rewrite
# a2enmod proxy
# a2enmod proxy_http
```

These commands work by creating or removing symlinks from files in `xxxx-available` to the same names in `xxxx-enabled` where `xxxx` = `conf`, `mod`, or `site` all in directory `/etc/apache2`. You will probably need all of the above commands to activate the stuff needed by `ledgersmb`.

3.4.9 Restart apache

```
# apachectl restart
```

You can also use `apachectl` to control the `apache2` http server:

```
# apachectl start
# apachectl stop
# apachectl restart
# apachectl graceful-stop
```

The above commands obviate the need for using `service` or `systemctl` commands with `ledgersmb`.

3.4.10 Simplify the vhost config file

We recommend that you take the approach of creating a simplified version of the `vhost.conf` file

- by commenting out or removing `ssl` statements
- iterate through the process of
 - running `# a2ensite lsmb-vhost.conf` to enable it
 - running `# apachectl configtest` to check for errors
 - fixing the error(s). You may have to enable other modules
 - pointing your browser to `http://lsmb.localhost`
 - If OK then progressively uncomment the removed portions and repeat the process until you have a version of the full `vhost.conf` file working.

3.5 Run LedgerSMB

Point your browser to `lsmb.localhost/`
or maybe `lsmb.localhost/login.pl`
or maybe `lsmb.localhost/setup.pl`
or maybe `lsmb.localhost:5762/login.pl`
or maybe `lsmb.localhost:5762/setup.pl`
or maybe `/login.pl`
or maybe `/setup.pl`