LedgerSMB on SUSE

John O'Gorman (john@og.co.nz)

13 May 2020

1 Intro

This identifies the actions needed to install LedgerSMB v1.7 on SUSE Tumbleweed. What works on Tumbleweed should also work with OpenSUSE Leap.

Some time ago we found the solution to our local-lib problems thanks to Eric Huelsman. 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.

The main thing we have learned is that Apache2, unlike its Ubuntu version, which 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, on the SUSE version still expects to use /srv/www/vhosts/ for the DocumentRoot of its vhosts.

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

We will now work on invoking SSL and finally Rewrite modules

1.1 SUSE

On Suse we install packages using Yast (Yet Another Setup Tool) 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).

1.2 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 postgreSQL 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.

1.3 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 or whatever browser you have
- 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.

1.4 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. To beecome root you 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

```
# zypper install mlocate
$ which locate
```

You can save typing by using the command:

```
zypper in mlocate
```

which is functionally equivalent!
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

```
# zypper install firefox
$ firefox - v
```

2.1.1 apache

```
# zypper install apache
$ apachectl -v
```

2.2 postgresql

```
$ yast
```

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

```
# zypper install perl
$ perl -v
```

2.3.1 make

```
# zypper install make
$ make -v
```

2.4 gcc

```
# zypper install gcc
$ gcc --version
```

2.5 cpanm

```
# zypper install cpanminus
$ cpanm -v
```

2.6 local::lib

```
# zypper 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 repalce 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.

```
$ yast
```

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 'PGObject', '1.403.2';
# PGObject::Simple 3.0.1 breaks our file uploads
requires 'PGObject::Simple', '>=3.0.2';
requires 'PGObject::Simple::Role', '2.0.2';
requires 'PGObject::Type::BigFloat', '1.0.0';
requires 'PGObject::Type::DateTime', '1.0.4';
requires 'PGObject::Type::ByteString', '1.1.1';
requires 'PGObject::Util::DBMethod';
requires 'PGObject::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.

On SUSE you will have gained a program: /usr/bin/pg_ctl which allows you to start, restart PostgreSQL without resorting to systemctl or service commands. (This did not happen for us on Ubuntu?).

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

On SUSE there is a command: /usr/bin/pg_ctl which can be used to start, stop, restart instances of PostgreSQL. e.g.

```
$ pg_ctl restart
```

This seems to be missing on Ubuntu so instead you would use:

```
$ service postgresql restart
```

3.3 Starman

3.3.1 Create user ledgersmb

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

The argument: -M or —nocreate-home means not to create a home directory

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 SUSE is apache v 2.4.43 which is radically different from the version apache v 2.4.41 on Ubuntu.

Its configuration is read from directory: /etc/apache2/

3.4.1 directories

/etc/apache2 (on SUSE) contains the following directories: conf.d, sysconfig.d, and vhosts.d.

- conf.d which contains .conf files:
  - git.conf
  - manual.conf
  - nagios.conf
  - php7,conf
  - subversion.doc.conf
- vhosts.d where you put your xxxvhost.conf files for each virtual host
  - vhost.template
  - vhost-ssl.template
- sysconfig.d with files:
  - global.conf
  - include.conf
  - loadmodule.conf
- sslcrl
- sslcrt
- sslcsr
- sslkey
- sslprm

The /etc/apache2/ directory also contains the following reference files: charset.conv, magic, and mime.types. You do not need to visit these.

3.4.2 config files

The remainder of the files in /etc/apache2/ are the configuration files:
There are 18 files all with suffix: .conf

1. default-server.conf - Sets DocumentRoot to srv/www/htdocs-
2. errors.conf
3. global.conf - Sets CustomLog to /var/log/apache2/access_log combined
4. httpd.conf which is the primary file. Do not edit it!
5. listen.conf If using SSL sets Listen 443. Equivalent of Ubuntu/Debian ports.conf
6. loadmodule.conf - lists modules .so (shared object) filenames
7. faults.conf
8. mod_cgit_timeout.conf
9. mod_info.conf
10. config.conf
11. mod_mime-defaults.conf
12. mod_reqtimeout.conf
13. mod_status.conf
14. mod_userdir.conf - sets ~/public_html as directory for URL ~john
15. mod_usertrack.conf
16. protocols.conf
17. server-tuning.conf
18. ssl_global.conf
19. uid.conf Set User to wwwrun and Group to www

Note that there is no apache2.conf, envvars, nor ports.conf.

- httpd.conf is used in place of Ubuntu/Debian: apache2.conf
- listen.conf is used in place of Ubuntu/Debian: ports.conf

3.4.3 No -available or -enabled dirs

The SUSE implementation of Apache2 does not have the following pairs of directories:

- conf-available and conf-enabled,
- mods-available and mods-enabled,
- and sites-available and sites-enabled.

3.4.4 No sites or confs

SUSE's apache does not contain the commands:

- a2ensites and a3dissites
- a2enconf and a2disconf.

3.4.5 a2enmod and a2enflag

SUSE's apache does have

- a2enmod and a2dismod - used to enable/disable modules
- a2enflag and a2disflag -used to enable/disable flags (e.g. SSL, STATUS)

3.4.6 The files in /etc/apache2

3.4.6.1 apache2.conf This file is not part of the SUSE apache distribution. Its place is taken by httpd.conf

3.4.6.2 httpd.conf The apache documentation says not to edit this file.

3.4.6.3 envvars This file is not part of the SUSE apache distribution.
To benefit from the www-data values you will need to install lynx if it isn't on your system:

```
zypper install lynx
```

3.4.6.4 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).

3.4.6.5 ports.conf This file is not part of the SUSE distribution. Its place is taken by listen.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.
So we recommend that you make implementation changes 1 at a time, test them using the command: apachectl -t, fix any reported errors, and then move on to the next change.

3.4.7 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:

3.4.8 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

3.4.9 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.

3.4.9.1 gensslcert SUSE's Apache2 provides gensslcert a bash (Bourne Again Shell) script in /usr/bin. Its author is Peter Poeml <apache@suse.de> and it calls openssl to build for you a self certified SSL set of files into /etc/apache2 in directories: ssl.crt, ssl.csr, ssl.key, and ssl.prm. Note: This file: gensslcert is not implemented on Debian/Ubuntu

3.4.9.2 openssl You can also 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
```

3.4.9.3 Enabling SSL

In SUSE apache2 you will need to do the following:

1. Create an SSL certificate
2. Set ServerName in vhosts block
3. Consider loading module: socache__shmcb
4. enflag SSL

Create an SSL certificate e.g.

```
# gensslcert -n og.gen.nz
```

this will populate the sslxxx dirs in /etc/apache2
Set the ServerName inside VirtualHost block:

```
ServerName og.co.nz
# a2enflag SSL STATUS
```

For the STATUS flag to work you may need to install lynx:

```
zypper install lynx
```

3.4.10 Install the apache-vhost.conf file

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

1. WORKING_DIR to /srv/www/vhosts/ledgersmb # or /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 /var/www/ledgersmb
- YOUR_SERVER_NAME to lsmb.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.

3.4.11 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
# apachectl -S # which lists all relevant config values for virtual hosts
```

To install lynx run the command:

```
zypper install lynx
```

The apachectl command in the shell script written by Marc SLenko (1997). The Debian version has been heavily modified by Stefan Fritsch (2010).
The command apachectl is a symbolic link to apache2ctl.

3.4.12 Activate the vhost

3.4.13 Restart apache

```
# apachectl restart
```

You can also use apachectl to control the apache2 html server:

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

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

3.4.14 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 # apachectl -t 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

Now implement the SSL stuff as

4 Virtual Hosts

We want to create virtual hosts. This is best attacked stepwise. Get the simplest version going 1st.
At each point in the climb to the top we need to check the changes we have just made using the command:

```
# apachectl -t
```

We respond to error messages and repeat until we have a Syntax OK message. Then we run the command:

```
# apachectl restart
```

4.1 Steps:

In our attempts to install LedgerSMB on SUSE we adopted an incremental approach.

1. create an entry in /etc/hosts: 127.0.0.0 og.localhost
2. create a hello.html file in the og.vhost file
3. copy it to index.html (to test)
4. point browser to e.g. og.localhost

5. run command apachectl -S to see what has been configured
6. restart apache: # apachectl restart

Now repeat the process for another virtual host name e.g. og.ssl
When that has worked you can then attempt to turn on SSL as in the section Enabling ssl.
Finally get the Rewrite commands working using the 3rd virtual host: lsmb.localhost.

4.2 In detail

4.2.1 Create hello.html file

Create a file: hello.html:

```
<html>
  <body>
    <h1>It works!</h1>
    <h2> filename is /srv/www/vhosts/og/index.html</h2>
  </body>
</html>
```

4.2.2 Set up 1st vhost

We will create a virtual host e.g. og.localhost

1. Edit the file /etc/hosts: adding 127.0.0.1 og.localhost where og is our virtual host
2. Create a directory in /srv/www/vhosts/ named og
3. Copy your file:hello.html to /srv/www/vhosts/og/index.html
4. Copy the file vhost.template to ogvhost.conf and start editing it
5. When you have changed the template values suitably and are getting the Syntax OK message then point the browser to og.localhost. You should see the message It works! and it will tell you the filename path
6. If this does not work use the command: apachectl -S

4.2.3 Set up 2nd vhost with SSL

Now you can, for example, add SSL support by copying the file vhost-ssl.template to ogss-lvhost.conf. Change your file hello.html so that you replace og with ogssl.

1. Edit the file /etc/hosts: adding 127.0.0.1 ogssl.localhost where ogssl is our next virtual host
2. Create a directory in /srv/www/vhosts/ named ogssl
3. Copy your file:hello.html to /srv/www/vhosts/ogssl/hello.html
   # cp hello.html /srv/www/vhosts/ogssl/index.html
4. Edit the hello.html file in ogssl changing og to ogssl and similarly alter the directory name from og to oggssl
5. In /ogssl copy the file hello.html to index.html
6. Copy the file vhost-ssl.template to ogss-lvhost.conf and start editing it
7. When you have changed the template values suitably and are getting the Syntax OK message then point the browser to ogssl.localhost

8. You should see the message It works! and it will tell you the full filename path to ogssl. If does not work properly run the command:

```
# apachectl -S
```

FIx any errors reported by the apachectl command.

Set up SSL as in the section Enabling SSL.

Repeat configctl -t, make fixes in necessary, apachectl restart

Point browser to ogg.localhost until success.

4.2.4 Set up lsmb with SSL and Rewrite

Repeat the same sequence as with og.localhost and ogssl.localhost for lsmb.localhost.

1. edit /etc/hosts adding 127.0.0.1 lsmb.localhost
2. create a symbolic link from /opt/ledgersmb to /srv/www/vhosts/ledgersmb
3. copy your hello.html file with an extra line indication the 2 directory paths: /srv/www/vhosts/ledgersmb and /opt/ledgersmb
4. copy the new hello.html file to index.html
5. run apachectl -t
6. if Syntax OK run apachectl restart
7. point browser to lsmb.localhost and look for It Works!
8. then copy the full ledgersmb vhost example file into /etc/apache2/vhosts.d/ledgersmb-vhost.conf
9. finger in ears and point browser to lsmb.host

4.3 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