

Installation on OS X Alternatives

Archived Page



Archived Obsolete - No Longer Maintained

As of October 7, 2015, this page is no longer maintained by the original author and is marked as archived. It is replaced by:

[macOS Installation](#)

About

This page describes methods of installing FreeSWITCH™ and its prerequisites on OS X without using Homebrew as per the [Installation and Setup on OS X](#). This is for people who do not want to use Homebrew or for packages not provided by Homebrew. The **Homebrew method is highly recommended** since it saves a lot of time, is easy to update, and very easy to remove packages. The options currently illustrated here are:

- A script to install FreeSWITCH™ and required prerequisites
- Install Individual Packages
- A configure file used to “make” everything



Upgraded OS X or Xcode ?

If there are problems installing prerequisites check if the OS X release was upgraded, for instance from 10.7 to 10.8 or 10.8 to 10.9. Also check if Xcode was upgraded, for instance from Xcode 3 to 4, or 4 to 5. It was found that in some cases older upgraded systems had issues due to old parts of a prior OS X or Xcode still intact. In these cases a fresh OS X and Xcode install may be required.



Prerequisites Previously Installed Differently?

If prerequisites were previously installed using different methods and there are problems, you should build OS X from scratch. In tests, using Homebrew or the script below worked every time on freshly installed OS X systems.



First FreeSWITCH™?

If you're installing for the first time please use Homebrew since it has worked every time for all OS X releases on this wiki.



Check Dates on This Page

This page is not updated as often as the main OS X wiki pages describing an installation using Homebrew.

Minimum Required Packages for OS X

In order to build FreeSWITCH™ on OS X the following packages must be installed first:

- **autoconf** - gnu autoconf utility.
- **automake** - gnu automake utility.
- **libtool** - gnu libtool utility. Starting with Lion Apple no longer ships automake with XCODE 4.3.
- **pkg-config** - many build functions invoke pkg-config
- **ldns** - Starting May 2014 FreeSWITCH™ requires ldns. Previous FreeSWITCH™ releases included this package, it is now a prerequisite.
- **libjpeg** - Some FreeSWITCH™ functions invoke SPANDSP, which invokes LIBJPEG. FreeSWITCH™ provides SPANDSP source but not LIBJPEG source.

- **openssl** - Starting Feb 2014 FreeSWITCH™ requires openssl. openssl is deprecated since OS X 10.7 Lion and not usable for FreeSWITCH™ in older OS X releases.
- **curl** - Starting Mar 2014 FreeSWITCH™ 1.4 requires libcurl. Previous FreeSWITCH™ releases included this package, it is now a prerequisite..
- **pcre** - Starting Mar 2014 FreeSWITCH™ 1.4 requires pcre. Previous FreeSWITCH™ releases included this package, it is now a prerequisite.
- **sqlite** - Starting Mar 2014 FreeSWITCH™ 1.4 requires sqlite. Previous FreeSWITCH™ releases included this package, it is now a prerequisite.
- **speex** - Starting Mar 2014 FreeSWITCH™ 1.4 requires libspeex. Previous FreeSWITCH™ releases included this package, it is now a prerequisite.

Install Using A Script



The script below contains code for downloading ldns but the ldns section was not tested yet. It is included here since ldns is required as of May 19 2014. It will be tested later.

The following script was **tested March 19, 2014** on OSX 10.9.2, 10.8.5, and 10.7.5. It assumes that the /usr/local and /usr/local/src directories exists and the proper ownership security set as per the Create /usr Subdirectories Installation on OS X 10.9 Mavericks Create Directories [Create /usr Subdirectories](#). Using this script allows easy removal of FreeSWITCH™ and installed packages. Read the comments for instructions and prompts. Of course, you can also cut/paste lines to run parts of the script in Terminal if desired.

```
#!/bin/sh
# Tested on OS X 10.9.2, 10.8.5, 10.7.5 March 19 2014
#
# This script can be run asis or separated into multiple scripts, and/or edited for different directories,
or run one line at a time.
# Example of how to use this script:
# 1. Create /usr/local and /usr/local/src as documented in the OS X install wiki page
# 2. Save this text on the Desktop as fsinstall.command
# 3. chmod +x ~/Desktop/fsinstall.command
# 4. Run command in terminal: . ~/Desktop/fsinstall.command
#
# Prompts:
# Installation of openssl and pcre will prompt for the admin password during make install
# Installing on OS X 10.7 script will pause prior to freeswitch make to allow removal of mod_v8 from modules.
conf
#
echo "Installing autoconf *****"
cd /usr/local/src/
curl -4 -o autoconf-2.69.tar.gz http://ftp.gnu.org/gnu/autoconf/autoconf-2.69.tar.gz
tar xzfv autoconf-2.69.tar.gz
cd autoconf-2.69
./configure --prefix=/usr/local
make
make install
make clean
cd ..
rm autoconf-2.69.tar.gz
#
echo "Installing automake *****"
cd /usr/local/src/
curl -4 -o automake-1.14.tar.gz http://ftp.gnu.org/gnu/automake/automake-1.14.tar.gz
tar xzfv automake-1.14.tar.gz
cd automake-1.14
./configure --prefix=/usr/local
make
make install
make clean
cd ..
rm automake-1.14.tar.gz
#
echo "Installing libtool *****"
cd /usr/local/src/
curl -4 -o libtool-2.4.2.tar.gz http://gnu.mirror.constant.com/libtool/libtool-2.4.2.tar.gz
tar xzfv libtool-2.4.2.tar.gz
cd libtool-2.4.2
./configure --prefix=/usr/local --program-prefix=g
```

```

make
make install
make clean
cd ..
rm libtool-2.4.2.tar.gz
#
echo "Installing pkg-config *****"
cd /usr/local/src/
curl -4 -o pkg-config-0.28.tar.gz http://pkgconfig.freedesktop.org/releases/pkg-config-0.28.tar.gz
tar xzfv pkg-config-0.28.tar.gz
cd pkg-config-0.28
./configure --prefix=/usr/local --with-internal-glib
make
make install
make clean
cd ..
rm pkg-config-0.28.tar.gz
#
echo "Installing jpeg *****"
cd /usr/local/src/
curl -4 -o jpegsrc.v8d.tar.gz http://www.ijg.org/files/jpegsrc.v8d.tar.gz
tar xzfv jpegsrc.v8d.tar.gz
cd jpeg-8d
./configure --prefix=/usr/local
make
make install
make clean
cd ..
rm jpegsrc.v8d.tar.gz
#
#
echo "Installing ldns *****"
cd /usr/local/src/
curl -4 -o ldns-1.6.17.tar.gz http://www.nlnetlabs.nl/downloads/ldns/ldns-1.6.17.tar.gz
tar xzfv ldns-1.6.17.tar.gz
cd ldns-1.6.17
./configure --prefix=/usr/local
make
make install
make clean
cd ..
rm ldns-1.6.17.tar.gz
#
echo "Installing openssl *****"
cd /usr/local/src/
curl -4 -o openssl-1.0.1f.tar.gz http://www.openssl.org/source/openssl-1.0.1f.tar.gz
tar xzfv openssl-1.0.1f.tar.gz
cd openssl-1.0.1f
./configure --prefix=/usr/local/ darwin64-x86_64-cc
make
echo "-----"
echo "-- enter the admin password if requested --- "
echo "-----"
sudo make install
make clean
cd ..
rm openssl-1.0.1f.tar.gz
echo "----- openssl directory change:"
echo "-- enter the admin password if requested to change owner of /usr/local/lib/pkgconfig --- "
echo "-- /usr/local/lib/pkgconfig must not be root for curl & pcre to install into --- "
echo "-----"
sudo chown $USER:staff /usr/local/lib/pkgconfig
#
echo "Installing curl *****"
cd /usr/local/src/
curl -4 -o curl-7.35.0.tar.gz http://curl.haxx.se/download/curl-7.35.0.tar.gz
tar xzfv curl-7.35.0.tar.gz
cd curl-7.35.0
./configure --prefix=/usr/local
make
make install

```

```

make clean
cd ..
rm curl-7.35.0.tar.gz
#
echo "Installing pcre *****"
cd /usr/local/src/
curl --remote-name ftp://ftp.csx.cam.ac.uk/pub/software/programming/pcre/pcre-8.34.tar.gz
tar xzfv pcre-8.34.tar.gz
cd pcre-8.34
./configure --prefix=/usr/local
make
make install
make clean
cd ..
rm pcre-8.34.tar.gz
#
echo "Installing speex *****"
cd /usr/local/src/
curl -4 -o speex-1.2rc1.tar.gz http://downloads.xiph.org/releases/speex/speex-1.2rc1.tar.gz
tar xzfv speex-1.2rc1.tar.gz
cd speex-1.2rc1
./configure --prefix=/usr/local
make
make install
make clean
cd ..
rm speex-1.2rc1.tar.gz
#
echo "Installing sqlite *****"
cd /usr/local/src/
curl -4 -o sqlite-autoconf-3080401.tar.gz https://sqlite.org/2014/sqlite-autoconf-3080401.tar.gz
tar xzfv sqlite-autoconf-3080401.tar.gz
cd sqlite-autoconf-3080401
./configure --prefix=/usr/local
make
make install
make clean
cd ..
rm sqlite-autoconf-3080401.tar.gz
#
#
#
echo "Installing FreeSwitch *****"
cd /usr/local/src/
git clone https://stash.freeswitch.org/scm/fs/freeswitch.git
cd freeswitch
./bootstrap.sh
./configure CFLAGS="-I/usr/local/include" LDFLAGS="-L/usr/local/lib"
OS_Version=$(sw_vers -productVersion | sed "s:.digit:.*$:g")
if [[ ${OS_Version} == 10.6 ]] || [[ ${OS_Version} == 10.7 ]]; then # mod_v8 does not build in 10.6 and
10.7
echo "-----"
read -p "Comment mod_v8 in modules.conf, then press [Enter] key to continue..."
fi
make
make install
make cd-sounds-install cd-moh-install
make clean

http://www.nlnetlabs.nl/downloads/ldns/ldns-1.6.17.tar.gz

```

Install Packages Individually

You can use the manual technique that requires locating, downloading and following instructions for each package. That requires several commands, knowledge about configuration, takes a lot of time, places programs in directories that are much harder to remove from, and harder to update. It didn't always go smoothly. When downloading, if there are multiple file extensions available, select the file ending in tar.gz if available. You should use the script above as a starting point.



If you're thinking about installing using this method, then later try Homebrew.... **STOP!** Some methods may make it difficult to remove them, and Homebrew will not fully install packages it already finds on the system. That strategy has been tested and it caused many problems. If you want to proceed, make sure to have a backup of your system *before* manually installing packages!

Install Using Makefile

Please note this was not tested by me (Mario G) but was contributed by one of the developers. It may be of interest to some. Since formatting a make file is tricky on the wiki the source can be found [here](#). This provides the latest updates if the file is kept up-to-date by it's author.

Cleanup

Delete installers, compressed files, and other files no longer needed.

MacPorts Packages

There are no plans to support MacPorts built prerequisites or FreeSWITCH™.