macOS Manual Installation

About

This page describes the process of installing, updating and removing FreeSWITCH™ and its prerequisites on macOS using the Terminal application instead of the macFI. This page is part of the FreeSWITCH™ macOS documentation.

Knowledge

What You Should Already Know
You must know your way around the macOS Finder, Dock and Textedit. You should also be familiar with the standard macOS folders (directories) such as Applications, Utilities, Downloads, Users, etc.

New To You
A new folder (directory) named /usr/local, it's part of the Linux/UNIX standard directories. We'll also use the Linux/Unix commands: cd, mkdir, ls, cp, mv, chmod, chown, and sudo. You don't need to learn about them unless you change the steps or directories used in this document.

Overview

The manual installation steps below involve downloading and installing the following software:

- Apple Command Line Tools and Compilers that provide the tools needed to generate everything else
- Prerequisite packages that FreeSWITCH™ needs to function
- FreeSWITCH™ itself

It takes one to two hours to have the sample FreeSWITCH™ configuration running on macOS. There's extra information for technical people who want it, *don't be concerned if you don't understand it.*

Preparation

Backup

All of the software this procedure installs can be removed easily. However, to restore to a point where nothing was added or changed, a backup is required.

Terminal Application Preferences

The Terminal application is used extensively during installation. The following changes are highly recommended:

1. Copy Terminal in the Applications/Utilities folder into the Dock for quick access.
2. Start the Terminal application by clicking it in the dock. Then open its preferences panel.
3. In Settings–>Profiles–>Window change the Window Size to larger values such as 140x48 to reduce having to resize the window often.

Installation Directories

Show Invisible Files and Directories in the Finder

macOS hides many Unix directories and files from the Finder including the directories created by this procedure. These files and directories must be unhidden so they appear on the desktop, Finder windows and file dialogs. Additional information and scripts for showing invisible files are in the macOS Customization guide. View hidden files by running the lines below in Terminal:

```
defaults write com.apple.finder AppleShowAllFiles YES
killall Finder
```

Directories Overview and Security

All FreeSWITCH™ and prerequisite files are placed into the macOS provided /usr/local directory which is normally empty and intended for additional software such as FreeSWITCH™. Having a single location allows easy removal. This procedure adds the following subdirectories to /usr/local:

- FreeSWITCH™ source located in /usr/local/src
- FreeSWITCH™ runtime located in /usr/local/freeswitch
- FreeSWITCH™ prerequisite source and runtime directories in /usr/local

The /usr/local directory is protected by System Integrity Protection (aka rootless) and cannot be removed or changed. Subdirectores of /usr/local can be added or removed using administrator authority. The subdirectories can then have their security changed so that non-administrators can modify them.

Create /usr/local Subdirectories

The user name (owner) and group name of the /usr/local subdirectories created by this procedure must match the user and group FreeSWITCH™ will run under. This is typically the main user name of the Mac. Use the mkdir command to create the freeswitch and src directories and the chown command to change the owner. Copy/paste these commands in Terminal, sudo will prompt for the administrator password:

```
cd /usr/local                              # Change to the /usr/local directory
sudo mkdir freeswitch src                  # Create the FreeSWITCH™ runtime and source directories
sudo chown -R `id -u`:`id -g` freeswitch src  # Change the owner of the two new directories to yours
```

> ⊘ If you would like to see the ownership and permissions of directories and files use the "ls -l" command in Terminal.
>
> Since /usr/local is accessed often, it should be placed in the Finder "FAVORITES" sidebar for quick access. Use the Finder to navigate to the /usr/local folder and drag it under "FAVORITES".
>
> Extra info: /usr/local/bin is defined in the default macOS path. The path is viewed by running "echo $PATH" in Terminal. If you don't understand this, don't worry, it doesn't affect anything.

Prerequisites

Additional programs are required to generate and run FreeSWITCH™. They are installed using *Homebrew*, a *package manager* and its *repository* of products for macOS. Homebrew saves time since there is no need to locate packages and determine the version required for each macOS release. Homebrew's Git repository is maintained by volunteers.

macOS Developer Tools

Generation of FreeSWITCH™ and its prerequisites require the macOS Developer **C**ommand **L**ine **T**ools that provide LLVM, Clang, Git and other tools required to generate non-GUI applications. The Xcode Software Developer Kit used to develop macOS GUI applications already includes the CLT. This is not something you need to worry about because if:

1. Xcode and CLT are *not* already installed **Homebrew will download and install the Command Line Tools during Homebrew installation.**
2. You are using the *latest* macOS *and* Xcode is installed, the correct CLT is already installed.
3. You are using a macOS prior to the latest release and Xcode is installed you may have the wrong CLT. That's because when Apple upgrades Xcode, only the latest release of macOS CLT are usually supplied. If macOS is not the latest release, the CLT may not match. **Homebrew will determine if the correct Command Line Tools are already installed. If not, it will download and install them during Homebrew installation.**

Install the Homebrew Package Manager and Prerequisites

The Homebrew package manager and prerequisites takes several minutes to install.

Install Homebrew

Homebrew installation is simple, the install command is found at the Homebrew home page and is listed below.

1. Open Terminal if not already opened, paste and run the ruby command from any directory.
2. Run the command "brew doctor" to verify homebrew works, this command could take a couple of minutes.

```
ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
brew doctor
```

Install Required Homebrew Packages

The basic FreeSWITCH™ prerequisite packages are installed by running the following command (copy/paste the entire line) in Terminal from any directory:

```
brew install autoconf automake curl ffmpeg jpeg ldns libpq libsndfile libtiff libtool lua openssl opus pcre pkg-config speex speexdsp sqlite yasm signalwire/homebrew-signalwire/flite signalwire/homebrew-signalwire/libks signalwire/homebrew-signalwire/signalwire-c
```

⚠️ Ignore any *keg-only* messages, they mean an alias (link) for the package was not placed into /usr/local/bin because brew found that macOS has another version installed. This is not a problem because the FreeSWITCH™ installation process searches Homebrew directories to find prerequisite packages.

🛇 **Already had some prerequisites installed?**

If non-Apple compilers are installed, or prerequisites were installed without homebrew, there *may* be problems.

Download FreeSWITCH™

macOS is now ready to download and install FreeSWITCH™. First, change to the directory the source will be downloaded into:

```
cd /usr/local/src
```

Selecting a Version

The primary FreeSWITCH™ releases are: **Current public release**, recommended for production systems. **Current branch**, contains in-test fixes since current public release, intended as the next maintenance release. **Master development**, containing the latest fixes and features, intended as the next major release. Additional Information is located at Source Options.

Download Current Public Release

Download the current public release of FreeSWITCH™ from [freeswitch-files](#), then decompress, rename and move the directory to /usr/local/src. You can copy the long line below and paste it into a Terminal command line. This single line performs: 1- Determines the latest public release available. 2- Downloads the compressed source file. 3- Decompresses the file into a folder in the current directory (should be /usr/local/src/). 4- Renames the folder to **freeswitch**.

```
FSfile=$(curl -s https://files.freeswitch.org/freeswitch-releases/ | grep -oE "freeswitch-[0-9]*\.[0-9]*\.[0-9]*\.-release\.tar\.bz2" | tail -n 1) && echo Downloading $FSfile && curl https://files.freeswitch.org/freeswitch-releases/$FSfile | tar -xj && mv ${FSfile/.tar.bz2//} freeswitch
```

If there are problems with the current public release, the developers may request that you install the current branch or master development for diagnosis.

Download Current Branch

The git command creates the **freeswitch** directory in src (/usr/local/src/freeswitch) and downloads the FreeSWITCH™ source files, replace # with the release number.

```
git clone –b v1.# https://github.com/signalwire/freeswitch.git
```

Download Master Development

The git command creates the **freeswitch** directory in src (/usr/local/src/freeswitch) and downloads the FreeSWITCH™ master development source files.

```
git clone https://github.com/signalwire/freeswitch.git
```

Install FreeSWITCH™

Generate Installation and Configuration Files

First, change to the newly created freeswitch directory containing the source:

```
cd freeswitch
```

Perform the next line *only if current branch or master development were download*. It creates many additional installation files based on the operating system. For instance, modules.conf used to add optional functionality to FreeSWITCH™. The command will run several minutes:

```
./bootstrap.sh   # <- Not for Current public release.
```

This step determines if the correct prerequisites are installed and sets [compilation](#) options. It will run several minutes:

```
./configure
```

Add Text–To–Speech Module

Additional functionality may be required as your FreeSWITCH™ needs become more complex. Optional modules are used to add new functions to FreeSWITCH™ by editing the modules.conf file. The FLITE Text To Speech functionality is recommended for a new system since some of the sample tests require it. The easiest way to add the FLITE module is to paste and run the following line into Terminal:

```
  perl -pi -e 's{#asr_tts/mod_flite}{asr_tts/mod_flite}' /usr/local/src/freeswitch/modules.conf
```

If you prefer to manually edit the modules.conf file use this procedure:

1. Use the Finder to navigate to the /usr/local/src/freeswitch directory.
2. Scroll to and double-click modules.conf in the freeswitch directory, the first time requires application selection, Textedit should be used.
3. Search for "flite" and remove the # to uncomment.
4. Save the file.

⊗ If this is your first FreeSWITCH™ don't add more than the FLITE module. Some modules may have prerequisites in addition to what is listed in this procedure.

Compile, Create Additional Source Directories and Check for Errors

This step compiles the programs and creates additional directories and files based on modules.conf. While still in /usr/local/src/freeswitch compile FreeSWITCH™ with the "make" command, it runs for several minutes:

```
make
```

If there are no errors proceed, otherwise seek assistance from the FreeSWITCH-users mailing list

Generate FreeSWITCH™ Runtime

This creates the /usr/local/freeswitch runtime library and executable programs.

```
make install
```

Download and Install Sounds

There are mandatory sound prompts and optional music on hold files. All must be downloaded to run the sample IVR. There are four versions available:

- sounds-install moh-install (8 kHz)
- hd-sounds-install hd-moh-install (16 kHz)
- uhd-sounds-install uhd-moh-install (32 kHz)
- cd-sounds-install cd-moh-install (48 kHz)

The cd sounds are recommended since all the sampling rates are provided resulting in fewer problems. Invoke the following command:

```
make cd-sounds-install cd-moh-install
```

Cleanup

This optional step can be performed to save space by removing the temporary files created by make:

```
make clean
```

Test FreeSWITCH™!

The time has come to start and test FreeSWITCH™ using the macOS Testing and Diagnostics instructions.

Updating FreeSWITCH™

The FreeSWITCH™ public release is updated by downloading and installing the latest maintenance level.

Master is updated with new features and fixes frequently. The current branch is updated with fixes only. To update master or the current branch, download updates, compile FreeSWITCH™ and create updated runtime binaries, run the following two commands:

```
cd /usr/local/src/freeswitch
git pull && make current
```

Updating Prerequisites

To update Homebrew and its formulas run "brew update" in Terminal, this should be run prior to upgrading packages. To upgrade all packages, run "brew upgrade", or to upgrade a single package run "brew upgrade *package-name*" command.

Removal

Removing FreeSWITCH™

Consider backing up modified or new configuration files you created! FreeSWITCH™ is completely removed by deleting the /usr/local/freeswitch and /usr/local/src/freeswitch directories using the Finder (drag to trash) or in Terminal:

```
rm -r /usr/local/freeswitch
rm -r /usr/local/src/freeswitch
```

Remove Homebrew and Prerequisites

Packages installed by Homebrew are listed using the "brew list" command. They are removed using the "brew uninstall *package-name*" command. To remove packages and Homebrew run the following in Terminal:

```
rm -rf /usr/local/Cellar /usr/local/.git && brew cleanup
sudo rm -r /Library/Caches/Homebrew   # (macOS prompts for the administrator password)
```

Remove the Command Line Toole

1. Move the /Library/Developer directory to the trash. Or delete it in Terminal (macOS prompts for the administrator password):

```
sudo rm -r /Library/Developer
```

Remove Everything Else

If the /usr/local directory existed prior to using this procedure it has to be examined to determine if the contents are deletable. If the instructions on this page were followed, **and the /usr/local directory was created during this procedure**, everything can be removed to roll back the system prior to using this procedure by performing these steps:

1. Drag everything in the /usr/local directory to the trash. Or delete it in Terminal (macOS prompts for the administrator password):

```
sudo rm -r /usr/local/*
```

> ⊘ NEVER delete the /usr/local directory! It's very difficult to recreate, it must be restored from a backup. This is due to System Integrity Protection (aka rootless) security.

Hide Invisible Files and Directories in the Finder

To reverse the show hidden files step, open the Terminal application and run each line below. All hidden files will disappear from the desktop, Finder and file dialogs.

```
defaults write com.apple.finder AppleShowAllFiles NO
killall Finder
```