2

Building and Installation

FreeSWITCH is open source software. Basically this means that anyone can obtain, read, compile, mangle, fix, or anything that comes to mind, the raw source code of the application. Many users, especially beginners, will find that dealing with source code is somewhat a daunting task, but rest assured, we are doing our best to make this experience as painless as possible. In the future, we will be adding binary packages into various popular Linux distributions, but for the time being, we will explain how to manually obtain and install FreeSWITCH for Unix and Windows. (For the purpose of this chapter, the terms "Unix-like" and "Linux/Unix" refer not only to Unix and Linux but also to FreeBSD and Mac OSX.) Try not to fret if it seems overwhelming. With a little patience and luck, the whole process will go smoothly. It's not entirely unlike a root canal. It's been said that many root canals are pulled off without a hitch and when they go wrong, they go horribly wrong; and that is where the horror stories, which we all hear, come from.

In this chapter, we will discuss how to download and install FreeSWITCH from the source code for Unix-like environments as well as for Windows. We will cover the necessary prerequisites for each operating system. Finally, we will explain how to launch FreeSWITCH and how to run it in the background.

In this chapter, we will cover the following:

* Setting up the FreeSWITCH environment
* Downloading and installing FreeSWITCH
* Launching FreeSWITCH and running it in the background
* Laying the ground work for our FreeSWITCH installation

# Setting up the FreeSWITCH environment

FreeSWITCH, like many other software applications, requires a suitable environment. Primarily that means choosing an appropriate operating system for your hardware and having the proper LAN/WAN connectivity and physical environment.

## Operating System

The first question to consider here is: which operating system should be used? Generally speaking, it is good to use an operating system with which you are comfortable and familiar. One caveat to consider is 32-bit versus 64-bit. Some users have reported problems when running a 32-bit OS on a 64-bit hardware platform. We strongly recommend that you use a 64-bit OS if you have 64-bit hardware.

Those who prefer a Windows environment can use XP, Vista, Windows 7, Server 2003, Server 2008 R2, or Server 2012. Several users have reported good success with production systems running on modern hardware, and using Windows Server 2008.

On the other hand, there is a wide variety of Unix-like operating systems available, many of which are freely downloadable. Most of us have an operating system (Linux, BSD, Solaris, and so on), and distribution (CentOS, Debian, Ubuntu, and so on.) that we prefer to use. The FreeSWITCH developers do not advocate any particular operating system or distribution.

Some have asked which platform is "the best" for FreeSWITCH. There are many factors to consider when choosing a platform on which to run a telephony application. FreeSWITCH is cross-platform, and therefore, it compiles and runs on numerous systems. However, through hard-earned experience we know which operating systems and distributions lend themselves to real-time telephony applications. The bottom line is that you want your system to be stable and reliable. The FreeSWITCH community has overwhelmingly endorsed CentOS 5 and Debian 6 as production-ready Linux distributions.

Note: As of this writing there are concerns with the performance of FreeSWITCH under CentOS 6. We recommend not using FreeSWITCH with CentOS 6 until these issues are resolved.

Keep in mind that “bleeding edge” distributions generally are not appropriate for real-time telephony systems. “Boring and predictable” are preferable to “latest and greatest.”

# Operating system prerequisites

Each operating system has its own set of prerequisites. Make sure that you have met the prerequisites for your platform.

## Linux/Unix

The following items frequently are already installed on your system. Note that a Git client is not required:

* Git: A Git client also gives you access to the current code repository (recommended especially for developers and those who want the latest code)
* GNUMAKE: The GNU version of make
* AUTOCONF: Version 2.60 or higher
* AUTOMAKE: Version 1.9 or higher
* LIBTOOL: Version 1.5.14 or higher
* GCC: Versi on 3.3 or higher
* WGET: Any recent version
* LIBNCURSES: Any recent version
* BZIP2: Any recent version

## Mac OSX

It is strongly recommended that Mac users have, at the very least, OS X Version 10.4. Compiling FreeSWITCH on OS X requires the installation of the Apple XCode Developer Tools. You may download them from http://connect.apple.com. Free registration is required.

Apple has been making some changes in the tools supported on OSX. The FreeSWITCH community does their best to keep people informed of the latest information with respect to building and running FreeSWITCH on OSX. Stay informed by visiting http://wiki.freeswitch.org/wiki/Installation\_and\_Tips\_for\_Mac\_OS\_X.

## Windows

FreeSWITCH in a Windows environment has two primary requirements. They are as follows:

1. Microsoft Visual C++ 2008 or 2010 (or 2008 or 2010 Express Edition)
2. A file decompression utility

FreeSWITCH in Windows is compiled and built using Microsoft Visual C++ (MSVC) or Visual C++ Express Edition (MSVCEE). The Express Edition is free to download though registration is required. It can be obtained at http://www.microsoft.com/Express/VC. The other requirement for Windows is a file decompression utility like WinZip (www.winzip.com) or WinRAR (www.rarlab.com). A free alternative is 7-Zip (www.7-zip.org ). Each of these utilities will add a right-click menu option to Windows Explorer.

Note: The Express Editions of Visual C++ does not support 64 bit targets by default. If you are intending to build 64 bit versions of FreeSWITCH for Windows, it is recommended that you have Professional editions of Visual Studio instead of Visual C++ Express.

# Text editors and XML

Working with FreeSWITCH requires you to have a text editor with which you are comfortable. Regardless of your editor choice, we strongly recommend that you use a text editor that supports XML syntax highlighting. You will find that editing XML configuration files is much easier on the eyes with highlighting turned on.

If you do not already have a preferred editor, then we suggest trying one or two for your platform. Be aware that if you are in a Linux/Unix environment that does not have GUI (Graphical User Interface), then your choices will be fewer. However, there are several excellent text-only editors available:

* Emacs — A text-only editor available for just about any Unix-like environment, including Mac OS X. It can highlight source code, XML, HTML, and more. This is the editor of choice for the FreeSWITCH development team. (A GUI version of Emacs is also available.)
* Vi/Vim — A text-only editor available for just about and Unix-like environment. Like Emacs it can highlight source code and markup languages. (A GUI version of Vim is also available.)
* Notepad++ — A graphical text editor for Windows environment. It supports highlighting of many programming and markup languages. It is a very useful and free text editor for Windows.
* Microsoft Visual Studio/Visual C++ Express — This Integrated Development Environment has a graphical editor that plays out very well with XML files. It supports highlighting and auto-completion of the XML tags and will display a red underline for any improperly closed or edited XML tags and or elements.

# Downloading the source

Most open source projects have their source code divided into two general categories: stable and latest. The FreeSWITCH project recent formed these two branches. Version 1.2.x is the “stable” branch and version 1.3.x is the “latest” branch. You can update to the latest branch at any time if you are using Git. (See the Building From the latest Code section in this chapter.) One other point to keep in mind: binary distributions of FreeSWITCH might be available for your platform. While they are certainly convenient, in our experience it is easier to troubleshoot, update, and customize your FreeSWITCH installation when compiling from the source.

Be sure that your system has Internet access because the build process will occasionally need to download additional files.

The source code can be obtained from the following FreeSWITCH download site: http://files.freeswitch.org. Locate a file named freeswitch-1.2.x.tar.gz (where x is the latest build number), and download it into a local directory on your computer, then decompress it. A typical session in Linux might look like the following:

#>cd /usr/src

#>wget http://files.freeswitch.org/freeswitch-1.2.1.tar.bz2

#>tar jxvf freeswitch-1.2.1.tar.bz2

This will create a new directory that contains the FreeSWITCH source code, ready for you to compile on your system. (From now on, this will be referred to as the FreeSWITCH source directory.)

Windows users should create a new directory and download the source file. See the Compiling FreeSWITCH For Windows section, later in this chapter.

# Building from the latest code

If you prefer to be on the latest version of FreeSWITCH, then you will need a Git client. Use yum, apt, or whichever package manager your distribution has to install Git. In Windows, a popular (and free) is TortoiseGit (code.google.com/p/tortoisegit).

In Linux/Unix environments a typical git checkout and compile session would look like this:

#>git clone git://git.freeswitch.org/freeswitch.git

#>cd freeswitch

#>./bootstrap

#>./configure

#>make install

#>make cd-sounds-install

#>make cd-moh-install

The preceding commands will take some time to complete. You can automate the process a bit by chaining the commands together with the && operator. These commands are discussed in more detail below.

# Compiling FreeSWITCH for Linux/Unix/Mac OS X

The install procedure is essentially the same for Linux, Unix, or Mac OS X. However, make sure that your system has met the prerequisites listed in the previous section.

## Compiling FreeSWITCH

Compiling FreeSWITCH requires just a few steps, although it will take some time depending upon the speed of your system. The basic procedure for compiling FreeSWITCH is as follows:

* Edit modules.conf file to customize which modules are compiled by default
* Run configure script
* Run make and make install to compile and install
* Edit modules.conf.xml to customize which modules are loaded by default
* Install the sound and music files

Following are detailed step-by-step instructions for compiling FreeSWITCH.

### Step 1: Edit modules.conf

The modules.conf file contains a list of the various FreeSWITCH modules that will be configured and compiled. The default modules.conf file has a sensible set of modules pre-selected to be compiled. However, there is one optional module that we will enable now. You should have a new subdirectory named freeswitch-1.2.x, where 1.2.x is the version number. For example, if the latest stable version is 1.2.1 then your source directory will be /usr/src/freeswitch-1.2.1. Follow the steps below:

1. Change directory into the new FreeSWITCH source directory:

#>cd /usr/src/freeswitch-1.2.x

1. Open modules.conf in a text editor. Scroll down to the following line:

#asr\_tts/mod\_flite

1. Remove the # character from the beginning of the line, then save and exit. The mod\_flite module enables FreeSWITCH to use the open source Festival Lite text-to-speech (TTS) engine. (The Flite TTS engine does not produce particularly high quality speech synthesis. However, it is very handy for doing TTS testing.)

After editing modules.conf we are ready to start the build process.

Removing the # character at the beginning of a line in modules.conf will cause the module on that line to automatically be built when issuing the make command. Likewise, adding a # at the beginning of the line will prevent the corresponding module from being built automatically.

### Step 2: Run configure script

Like many open source projects, FreeSWITCH in UNIX-like environments makes use of the now famous configure script. From within the FreeSWITCH source directory, launch the configure script, as follows:

#>./configure -C

The configure script performs many tasks, including making sure that the prerequisites have been met. If a prerequisite has not been met then the configure script will exit, and tell you which dependency has not been met. If this occurs then you must resolve the issue and rerun the configure script. You will need to make sure that all of the prerequisites have been met before the configure script will run to completion. The -C argument tells the configure to create a config.cache file that will be used by subsequent configure scripts with the various libraries included in the source tree.

During the configuration process you will see the configure script run multiple times. FreeSWITCH makes use of many libraries like Apache Portable Runtime (APR) and Perl Compatible Regular Expressions (PCRE). Each of these elements has its own specific configure script that is customized to its own needs.

After some time the configure script finishes and returns you to the system prompt. You will undoubtedly see a lot of output on the screen from the configuration process, but if you do not see any errors then you may proceed to the compilation process.

### Step 3: Run make and make install

The configuration process in the previous step actually creates what is called a Makefile for FreeSWITCH, its libraries, and its various modules. The compilation and installation of FreeSWITCH are both handled by the make utility. First run make, and then run make install. Many users will run them both with one command line, which is as follows:

#>make && make install

Like the configure script, the make process takes a while, and will stop if there are any errors. Usually things go well, and at the end of the compilation and installation you are greeted with the following message:

+-------- FreeSWITCH install Complete ----------+

+ FreeSWITCH has been successfully installed. +

+ +

+ Install sounds: +

+ (uhd-sounds includes hd-sounds, sounds) +

+ (hd-sounds includes sounds) +

+ ------------------------------------ +

+ make cd-sounds-install +

+ make cd-moh-install +

+ +

+ make uhd-sounds-install +

+ make uhd-moh-install +

+ +

+ make hd-sounds-install +

+ make hd-moh-install +

+ +

+ make sounds-install +

+ make moh-install +

+ +

+ Install non english sounds: +

+ replace XX with language +

+ (ru : Russian) +

+ ------------------------------------ +

+ make cd-sounds-XX-install +

+ make uhd-sounds-XX-install +

+ make hd-sounds-XX-install +

+ make sounds-XX-install +

+ +

+ Upgrade to latest: +

+ ---------------------------------- +

+ make current +

+ +

+ Rebuild all: +

+ ---------------------------------- +

+ make sure +

+ +

+ Install/Re-install default config: +

+ ---------------------------------- +

+ make samples +

+ +

+ +

+ Additional resources: +

+ ---------------------------------- +

+ http://www.freeswitch.org +

+ http://wiki.freeswitch.org +

+ http://jira.freeswitch.org +

+ http://lists.freeswitch.org +

+ +

+ irc.freenode.net / #freeswitch +

+ +

+-----------------------------------------------+

If you see a message like the last one then you have successfully compiled FreeSWITCH, and can proceed to the next step. If an error occurs then the compilation process will stop and report it. You will need to correct the problem before you can continue. If the error message is unfamiliar to you then you should contact the FreeSWITCH community using the resources listed in Appendix A - The FreeSWITCH Online Community.

### Step 4: Edit modules.conf.xml

The modules.conf.xml file contains a list of modules that FreeSWITCH will load when it is launched. The default modules.conf.xml file corresponds with the default modules.conf file. The modules that are built by default in modules.conf are also enabled by default in modules.conf.xml. As we enabled mod\_flite to be built in modules.conf, we need to enable mod\_flite in modules.conf.xml, so that it will be loaded automatically when FreeSWITCH starts. As a rule of thumb, any module that you wish to load automatically when FreeSWITCH starts must be enabled in modules.conf.xml.

The modules.conf.xml file is located in the conf/autoload\_configs subdirectory. The default location is /usr/local/freeswitch/conf/autoload\_configs/modules.conf.xml. Open the file in a text editor and locate the following line near the end of the file:

<!-- <load module="mod\_flite"/> -->

Remove the <!-- and --> tags so that it looks like the following:

<load module="mod\_flite"/

Save the file and exit. You are almost ready to start the FreeSWITCH application.

What's the difference between modules.conf and modules.conf.xmlfiles? The modules.conf file is found in the source directory, and is used to control FreeSWITCH modules which are compiled when running make. The modules.conf.xml file is part of the default XML configuration, and is found in the FreeSWITCH autoload\_configs subdirectory. It controls which modules are loaded when FreeSWITCH is launched.

### Step 5: Install sound and music files

Sound and music files are not absolutely required. However, they are highly recommended. Without them you will not have music on hold, nor will features like voicemail and the sample IVR be functional. FreeSWITCH has sample sound and music files available in four different sampling rates. We recommend installing all of them so that you can take advantage of high quality audio connections wherever possible.

To install the sound files just issue the following command in the FreeSWITCH source directory:

#>make cd-sounds-install

To install the music files issue the following command:

#>make cd-moh-install

These commands will download and install the sound and music files in 8 kHz, 16 kHz, 32 kHz, and 48 kHz. FreeSWITCH will use the appropriate sampling rate when playing a sound or music file to a caller.

You are now ready to start FreeSWITCH. The next section covers compiling FreeSWITCH in the Windows environment, so skip down to the section named Starting FreeSWITCH.

# Compiling FreeSWITCH for Windows

As mentioned in the Operating system prerequisites section, FreeSWITCH is built with MSVC or MSVCEE. The steps presented here are specifically for MSVCEE 2010; however, the steps for the various editions of MSVC are essentially the same.

## Important considerations for Windows users

Unless you are a developer you may find that using the FreeSWITCH binary installer is more than adequate for your needs. Simply download the x86 or x64 freeswitch.msi from http://files.freeswitch.org/windows/installer/ and run the installer. It is extremely simple to do. More information about the binaries can be found online at http://wiki.freeswitch.org/wiki/Installation\_for\_Windows#Precompiled\_Binaries.

With the new features present in Microsoft's Visual Studio 2010, it is now highly advisable that users should use this development environment instead of Visual Studio 2008. Please do note that the recommendation also applies to the Express Editions. Some of the exciting new modules added in FreeSWITCH since 1.0.6 may not be present in the Visual Studio 2008 project files as the contributors of the project mainly focused on developing applications with Visual Studio 2010.

At the time of writing, please do not attempt to import the Visual Studio2010 solution file in a Visual Studio 2012 build environment as FreeSWITCH currently does not build correctly with it.

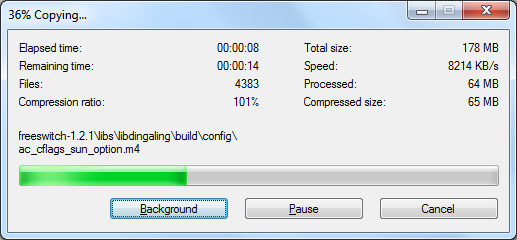
## Building the solution with MSVC/MSVCEE

There are several small steps to take prior to building with MSVCEE. They are as follows:

1. Create a new folder and copy the bz2 file into it. In our example, we'll use

C:\FreeSWITCH\freeswitch-1.2.1.tar.bz2.

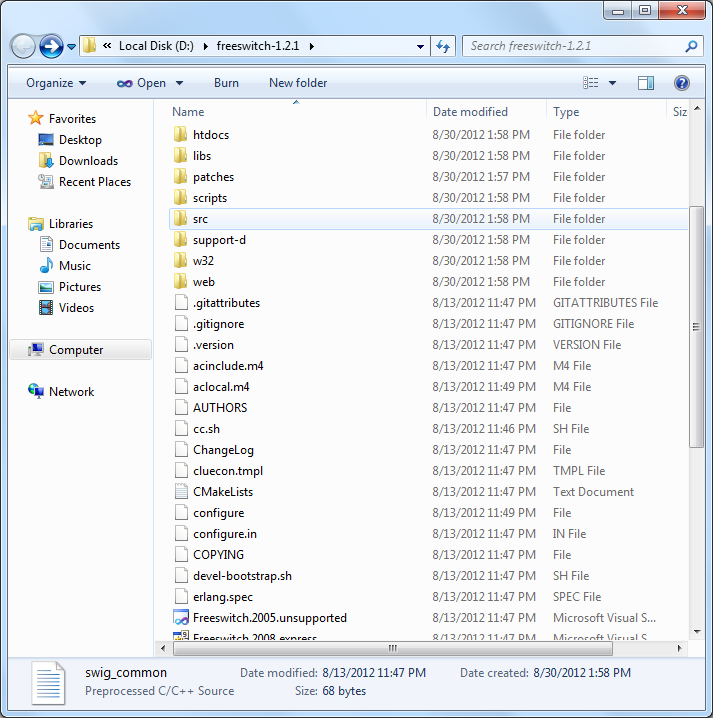
1. Right-click freeswitch-1.2.1.tar.bz2 and extract the files with your decompression utility. You will now have a new file named freeswitch-1.2.1.tar.
2. Right-click freeswitch-1.2.1.tar and extract files. This process will take a few moments. You will then see a window similar to the following screenshot:



Insert 1004\_02\_01.png

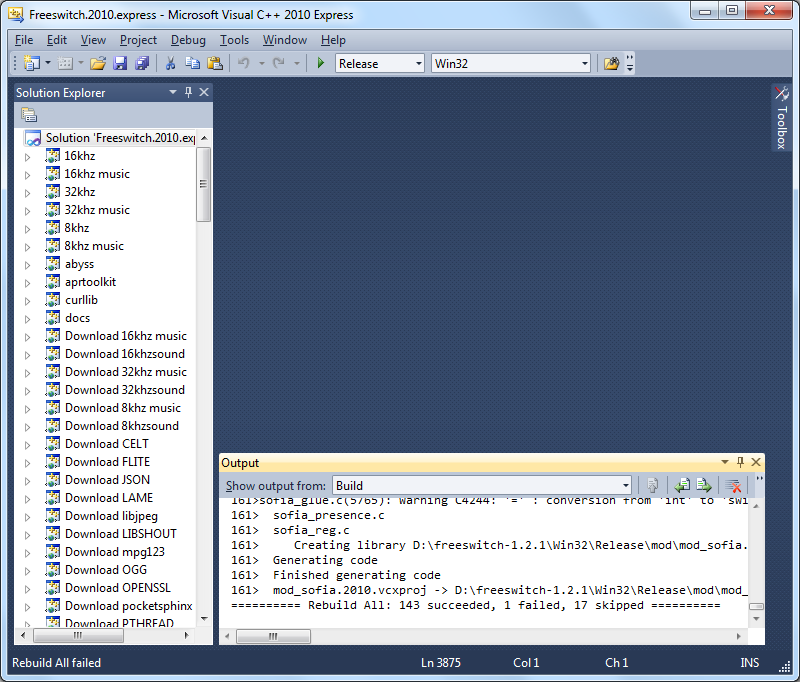
Note: WinRAR decompresses both the .gz and .tar files in a single step.

1. After extraction you will have a new sub-folder named after the latest version of FreeSWITCH. In our example, we now have a sub-folder named freeswitch-1.2.1. Double-click on the folder to see the complete FreeSWITCH source tree. It will be similar to the screen in the following screenshot:



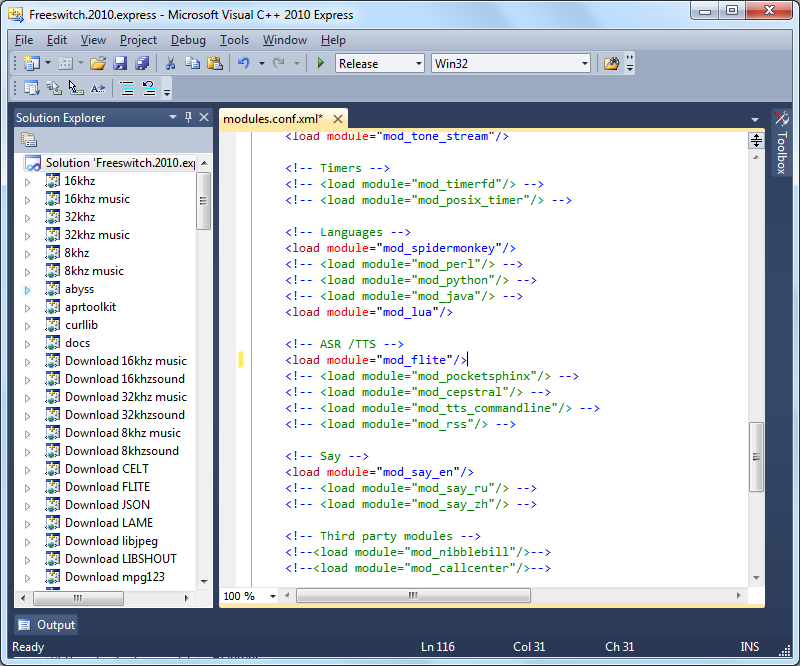
Insert 1004\_02\_02.png

1. While there are many files, the only ones we care about right now are the two solution files. For MSVC, the file is named Freeswitch.2010.sln and for MSVCEE, it is named Freeswitch.2010.express.sln. Double-click the appropriate solution file for your edition of MSVC. The screenshots in this example will show MS Visual C++ Express Edition. However, the Professional, and Ultimate editions will be very similar.
2. After the solution file loads, click the drop-down box (located on the toolbar) and change from “Debug” to “Release”, then click Build > Build Solution or press F7. If you are using the Visual Studio 2010 IDE, then click Build > Build Solution or enter the CTRL+SHIFT+B key sequences. The solution will start building. Numerous messages will appear in the Output window. When the solution has finished building, you will see a message at the bottom of the Output window as in the following screenshot:



Insert 1004\_02\_03.png

1. Note: The MSVC/EE solution files will automatically perform several steps that are usually done manually in a Linux/Unix installation. These include downloading of all the sound and music files, and building optional modules like Flite (text-to-speech) and PocketSphinx (speech recognition). However, these optional modules still need to be enabled in modules.conf.xml if you wish to have them automatically loaded when FreeSWITCH starts. (See the next screenshot.)
2. Go back to the Windows Explorer. You will see that the build process has created a new folder named Release. This is the FreeSWITCH installation directory. The last step before launching FreeSWITCH is to edit the modules.conf.xml file in order to enable mod\_flite to be loaded by default when FreeSWITCH is started. We will be using the mod\_flite text-to-speech (TTS) engine in several examples throughout this book.
3. Double-click the conf folder, then double-click on the autoload\_configs folder. Open modules.conf.xml in an editor. In our example, we'll use MSVCEE to edit the file as seen in the following screenshot:



Insert 1004\_02\_04.png

1. Locate the following line near the end of the file:

<!-- <load module="mod\_flite"/> -->

Remove the <!-- and --> tags so that it looks like the following:

<load module="mod\_flite"/>

1. Save the file and exit the editor. You are now ready to launch FreeSWITCH for the first time.

# Starting FreeSWITCH

Once you have compiled FreeSWITCH, it is time to launch the application.

Linux/Unix/OS X: run /usr/local/freeswitch/bin/freeswitch.

Windows: run freeswitchconsole.exe from the Release directory.

The system will start loading, and numerous messages will display on the screen. Console messages are color-coded for readability. Do not worry about all of the messages right now, just make sure that your system starts up and you get to the FreeSWITCH console, which we call the command-line interface (CLI). The CLI prompt looks like the following:

freeswitch@localhost>

Let's issue a few commands to verify that the system is operational. First, issue the version command to verify the version of FreeSWITCH that we have installed. You’ll see something similar to this:

FreeSWITCH Version 1.2.1

Next, issue the status command which displays a few statistics about your system. You’ll see output similar to this:

freeswitch@localhost> status

UP 0 years, 0 days, 0 hours, 0 minutes, 16 seconds, 808 milliseconds, 260 microseconds

FreeSWITCH is ready

0 session(s) since startup

0 session(s) 0/30

1000 session(s) max

min idle cpu 0.00/100.00

Current Stack Size/Max 240K/8192K

These are just a few of the many commands you will learn about in FreeSWITCH. For a complete list of commands, simply type help and press Enter. Lastly, shut down FreeSWITCH with this command: fsctl shutdown. The system will display numerous messages as it shuts down, and will return you to the system command prompt. (If you launched freeswitchconsole.exe from the Windows explorer then the FreeSWITCH window will simply close.)

# Running FreeSWITCH in the background

In most cases, you will want FreeSWITCH to run in the background. In a Unix/Linux environment this is frequently called running as a daemon. In Windows this is called running as a service.

To launch FreeSWITCH as a daemon in Unix/Linux:

#>/usr/local/freeswitch/bin/freeswitch –nc

The various Linux and Unix distributions take different approaches to automatically running a daemon at system start up. Several init script examples are available on the FreeSWITCH wiki: wiki.freeswitch.org/wiki/Freeswitch\_init. Consult the system administration documentation for your specific distribution for instructions on how to configure the init script to launch FreeSWITCH at system start up.

Windows requires just a few steps to have FreeSWITCH run as a service. They are as follows:

1. Open a Windows command-line session (Click on Start | Run , type cmd , and then click the OK button).
2. Change the directory into your FreeSWITCH installation directory, as follows:

cd FreeSWITCH\freeswitch-1.2.1\Release

1. Run freeswitchconsole.exe with the –install argument, as follows:

freeswitch –install FreeSWITCH

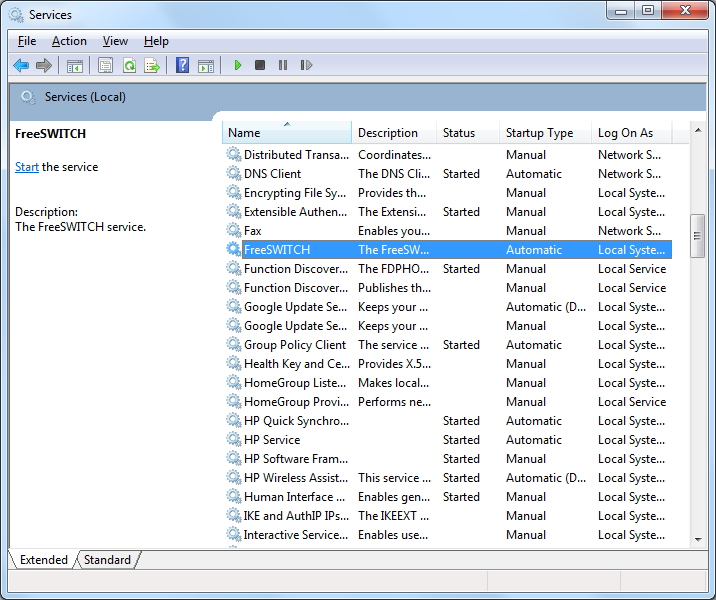
1. The last step is to configure the service itself.

If you are using Windows XP or Server 2003, open the services tool, click on Start | Control Panel | Administrative Tools | Services.

Otherwise, if you are using later versions of Windows, simply type in "Services" in the Start Menu's search text box. Select the Services icon in the results bar.

Alternatively, you can also bring up the Services MMC Console by entering Windows Key + R and type in services.msc, and then click on OK.

FreeSWITCH should now appear in the list of services:



Insert 1004\_02\_05.png

1. Right-click FreeSWITCH and click start. The service will take a moment to start up.
2. Confirm that the service is running by using the fs\_cli.exe utility found in the Release folder:

fs\_cli.exe

1. You will see a welcome screen and a command prompt. Issue the status command, to confirm that the system is running.
2. Type /exit to close the fs\_cli.exe program.

You now have FreeSWITCH running as a service in Windows.

The fs\_cli utility is discussed in greater detail in Chapter 10, Controlling FreeSWITCH Externally.

# Summary

In this chapter, we accomplished a number of objectives. They are as follows:

* Downloaded and installed FreeSWITCH
* Customized the installation by modifying the modules.conf to compile the mod\_flite TTS module (Linux/Unix/Mac OS X only)
* Customized the FreeSWITCH configuration by modifying modules.conf.xml to automatically load mod\_flite when FreeSWITCH is launched
* Launched FreeSWITCH and issued several commands to confirm its operational status
* Launched FreeSWITCH as a daemon (Linux/Unix) or as a service (Windows)

In the following chapter, we will put our new installation into action as we explore the demonstration configuration of FreeSWITCH.