

Introduction

0. About

In this Introduction we provide a brief overview of FreeSWITCH in laymen's terms. We will then introduce all the key concepts in FreeSWITCH, and guide you on how to navigate the documentation. By following this introduction, you should be able to setup a basic deployment of FreeSWITCH in no time. Should you decide to dig deeper, and get familiar with more advanced features, we provide the links where you can find that too. At the end we will also provide a walk-through of the [default configuration](#), so that you can get a hands-on tutorial, and most commonly used features.

1. What is FreeSWITCH?

FreeSWITCH is a Software Defined Telecom Stack enabling the digital transformation from proprietary telecom switches to a versatile software implementation that runs on any commodity hardware. From a Raspberry Pi to a multi-core server, FreeSWITCH can unlock the telecommunications potential of any device. Combined with our hosted cloud platform, SignalWire, FreeSWITCH can interconnect with the outside world and scale to any size. (Visit <https://signalwire.com> for more info)

FreeSWITCH can handle voice, video and text communications from an IP Network (VoIP) and the PSTN (i.e., regular landlines). FreeSWITCH supports all popular VoIP protocols as well as interfacing with PRIs. For a full listing of supported protocols, see the [Endpoints](#) page.

Some common capacities, that FreeSWITCH is used for, include

- [PBX](#) (Office phone system)
- [Class 5 Softswitch](#) (Telephone carrier)
- Application server such as Voicemail, conferencing, IVR
- [Softphone](#)

This list is by no means comprehensive, FreeSWITCH is extremely flexible, and can be used in any way you can imagine.

The [default configuration](#) showcases a full-featured [PBX](#) with many applications.

2. Where can I run FreeSWITCH?

At its core, FreeSWITCH is a library which can be [embedded](#) in your application on any device. However, more commonly, it is built to run as a [background process](#) (daemon in UNIX or Linux systems, service on Windows platforms). When you run FreeSWITCH as a daemon, you can use the [CLI](#) to interact with FreeSWITCH.

FreeSWITCH can run on many Platforms, including Linux, Mac OS X, BSD, Solaris and even Windows.



If you don't want to operate your own server, the corporate sponsor of FreeSWITCH, [SignalWire](#), provides cloud-hosted FreeSWITCH services from dedicated FreeSWITCH servers to auto-scaling cloud-hosted services.

Although FreeSWITCH can run on many Linux distributions, such as Debian, Ubuntu, CentOS, Fedora and RHEL, our preferred distribution is Debian, because we have worked out all dependencies on Debian so you can have a smooth installation and startup. Debian is also what the FreeSWITCH developers use, therefore have the most experience with. If you run on other distributions, you might have a hard time getting all dependencies right.



See [Release Notes](#) and [Installation](#) instructions for the latest supported platforms.

Hardware requirements depend on how you will use FreeSWITCH. FreeSWITCH can run on hardware as small as a Raspberry Pi, and can scale up to powerful data center servers with dozens of CPU cores. FreeSWITCH is capable of handling thousands of concurrent phone calls, depending on what hardware you have and which applications you are using.



For more information about hardware requirements see [Release Notes](#) and [Performance Testing and Configurations](#).

3. Architecture

When designing FreeSWITCH the goal was to have the following properties:

- **Extensibility** - Allowing new features to be added easily.

- **Flexibility** - Users should be able to pick and choose which functionalities are enabled, while also allowing them to replace parts of the system with different implementations
- **Scalability** - FreeSWITCH can be run on systems as small as an embedded [softphone](#) or as big as a cluster of full-blown carrier switches.
- **Stability** - Problems with one feature should not bring down the whole system.

To accomplish these goals FreeSWITCH is built using a **modular system**:

- There is a small **core** that provides the base system used by all modules, however most of the functionalities are not implemented in the core itself,
- but rather in **standalone modules** that are not dependent on each other.

When you install FreeSWITCH with the [default configuration](#), the modules that are required for most common scenarios are enabled. We will see later in the [configuration](#) section how to enable/disable modules.

The [modules](#) are grouped by the type of functionality they provide. We will now explore the types of modules, and what functionality is provided by each.

3.1 Module Types

| Type | Description |
|------------------------------|--|
| Endpoint | Endpoint modules support types of communications devices such as VoIP , PSTN (i.e., regular landline), Skype, Google Talk etc. Endpoint modules are one of the most important modules in FreeSWITCH. |
| Application | This is where all the action is happening! There are hundreds of application modules included in the default setup a few examples are playing a file, joining a conference, send a call to voicemail, play an IVR menu. Many of the common applications are provided by the dptools module. |
| Dialplan | A Dialplan module is responsible for routing calls, based on information such as Caller ID, Destination Number and more. The default dialplan module is the XML Dialplan . We will cover this in detail in the Dialplan section. |
| Directory | Provides authentication and configurations for users that can register with FreeSWITCH. The most common directory module is the XML Directory . |
| Codecs | Codecs are used to encode and compress audio for streaming. |
| File Formats | Most of the common audio formats are supported by the <code>mod_dptools:playback</code> . See the list of supported formats there. <div style="background-color: #ffcc00; padding: 2px; display: inline-block; margin-bottom: 5px;">TODO</div> From 3. Supported file types section in <code>mod_dptools:playback</code> : " <i>FreeSWITCH has an abstraction layer for file formats.</i> " (See TODO there also.) |
| Loggers | Record log messages. Some of the included loggers are console and log file . <code>xml_cdr</code> is another commonly used logger to output call detail records. |
| Languages | Support scripting languages which can be run as part of the dialplan. The most popular language is Lua . Javascript and a few others are also supported. |

There a few more types of modules, but these are the common ones. For a listing of common module see the [Modules](#) page. If you require some functionality that's not provided by one of the existing modules, you can [write your own module](#). Modules can be written in many popular programming languages. See the [developer documentation](#) for more information.

API

Many modules also have API commands that can be issued from the command line, script or sent from a remote computer via the event socket. What the API commands can do are up to each individual module, but some common functionalities include returning status information (such as how many listeners are in a conference) or control the currently running application (such as pausing a file being played) There are hundreds of APIs available from the different modules. In the command line you can type `show api` to see all APIs based on the modules that are loaded.

There are also API commands provided by the core, these are in the [Commands](#) module.

Navigating the documentation

This table provides a roadmap you can use to go through the documentation in a structured fashion.

| Page | Description |
|------------------------------------|---|
| Installation | How to download and install FreeSWITCH |
| Configuration | Overview of the structure of FreeSWITCH Configuration files. |
| Running FreeSWITCH | How to run FreeSWITCH in the console or as a daemon |
| Call Legs | Explains an important concept required to better understand Endpoints and Dialplans |

| | |
|--|---|
| Endpoints | Explains the endpoint concept in general, as well as a more detailed overview of the common endpoints |
| Introduction to Dialplan | Dialplan is one of the most complex parts of FreeSWITCH, this page distills it |
| Directory | How to setup users and devices |
| Logging & CDR | Setup Logging and Call Records |
| Databases | How to configure FreeSWITCH to store channel state in a database |
| Default Configuration | Walkthrough of the default configuration |
| Session Scripts | How to create custom functionality by writing scripts that run in the dialplan, mod_lua , mod_python , mod_perl , mod_v8 (for Google V8 JavaScript), mod_java |
| Event Socket | Allows an outside program to interact with FreeSWITCH over a network connection. The CLI uses the Event Socket to allow you to run commands and view the output of FreeSWITCH. |

Support

- There is an active community on our mailing lists which can be found at <http://lists.freeswitch.org/mailman/listinfo/freeswitch-users>. (Learn how to ask [smart questions](#) before posting!)
- The FreeSWITCH developers also hang out in our [Slack room](#).
- For larger implementations and for complex support issues we offer commercial support plans for more information please [email](#) us or call 877.422.2583.
- If you need a new feature, you can try putting up a bounty on the [bounty](#) page.
- There are community sites in several languages.
 - Russian - [freeswitch-ru](#) google group & Jabber conference freeswitch@conference.jabber.ru ()
 - Chinese - [FreeSWITCH-CN](#) & [wiki](#)

Licensing

FreeSWITCH itself is licensed under the [MPL 1.1](#) (Mozilla Public License), however some individual modules might be using other licenses.

Additional Resources

- Join the [weekly conference](#) to keep up-to-date with the latest developments in the FreeSWITCH world.
- The annual [ClueCon](#) takes place in Chicago, Register today!
- [FAQ](#)
- Buy our books

Contributing

- Donate, click on the donate link on our [website](#)
- Join the [documentation team](#).
- [Report bugs](#)
- [Develop Software](#) - Are you a developer? Help us develop FreeSWITCH!