Contents

- 1 History
- 2 Fundamentals
- 3 Threaded Model (Parallel Operations)
- 4 Stable Protected Core
- 5 Dynamic Modules
- 6 Detailed Event and Logger System
- 7 Hooks into Everything
- 8 Lua Example
- 9 C Version
- 10 How It All Fits Together
- 11 Real-Time
- 12 FreeTDM
- 13 Future Plans
- 14 Credit
- 15 Learn More

History

- 1999 (Start of Asterisk and OpenH323)
- 2003 (anthm and bkw Join Asterisk)
- 2005 (First ClueCon) (Asterisk 1.2 Released)
- 2006 (Start of FreeSWITCH / OpenPBX)
- 2007 (OpenZAP Introduced)
- 2008 (FreeSWITCH 1.0 ?Phoenix? Released)
- 2010 (FreeTDM Supercedes OpenZAP)

Fundamentals

- Threaded Model (Parallel Operations)
- Stable Protected Core
- Dynamic Modules to Extend Functionality
- Detailed Event and Logger System
- Hooks into Everything

Threaded Model (Parallel Operations)

- Every connection has its own thread.
- Sub-systems run in background threads.
- Event handlers push events into threads.

Stable Protected Core

- Critical data structures are opaque.
- Complicated code is all in one place.
- Complicated routines get maximum reuse.
- Higher level code cannot be misused.
- With each new level of API the power increases as the complexity decreases.

Dynamic Modules

- FSAPI interface allows the addition of commands that can be accessed remotely.
- Application interface allows creation of custom IVRs and call routing apps.
- XML interface allows real-time dynamic access for user directory and configuration.
- Endpoint interface connects IM and phones.
- Formats, Automated Speech Recognition (ASR) & Text-to-Speech (TTS), and Say & Phrase Macros extend input and output capabilities.

Detailed Event and Logger System

- Events are fired whenever something significant happens.
- Detailed logging makes debugging easier.
- External applications can bind to events and track the status of the entire system.

Hooks into Everything

- Learn to think 4th dimensionally!
- Embedded languages simplify the API even more.

- Same principle as the CGI / Web server.

Lua Example

```
-- answer the call
session:answer();

-- play a file
session:streamFile("hello.wav");

-- hangup
session:hangup();
```

C Version

```
SWITCH_STANDARD_APP(demo_function)
{
    switch_channel_t *channel;

    channel = switch_core_session_get_channel(session);

    switch_assert(channel);

    switch_channel_answer(channel);
    switch_core_session_stream_file(session, "hello.wav");
    switch_channel_hangup(channel, SWITCH_CAUSE_NORMAL_CLEARING);

}
```

How It All Fits Together

- Low level C code
- Core API
- High level API (Module and IVR API)
- Inter-module communication with events and API calls
- Support for many programming languages

Real-Time

- The XML registry (dialplan, user directory, configuration, phrase macros)
- Binding to the sections (mod_xml_curl)
- Poll based info from FSAPI (XML-RPC)
- Push based info from events (mod_event_socket)

FreeTDM

FreeTDM supports

- multiple telephony boards such as Digium and Sangoma.
- PRI, BRI, SS7, MFC-R2 and Analog signaling.

Future Plans

- ~~Provide G.729 Licensing.~~
    - G.729 licenses are now available for Linux systems.
    - The Windows version is in testing.
- Add OPAL support for improved H.323.
- ~~Stabilize and branch FreeSWITCH 1.0 ?Phoenix?.~~
- ~~Complete and release FreeSWITCH 1.2 ?Dragon?.~~

Credit

- Mike Jerris (Building, Maintaining, Improving)
- Brian West (Q/A, Community, Features)
- Everyone who has reported a bug or submitted a patch
- All the testers and community members
- Fellow open source projects
- Everyone who supports Open Source VoIP

Learn More

You can learn more from the Wiki and the Documentation

The best way to discuss FreeSWITCH and get support is through IRC and the freeswitch-users mailing list.