Contents

- 1 Introduction
- 2 Formats and File Names
- 3 Dialplan Examples
- 4 Script to convert a sound file to specific formats to avoid transcoding

Introduction

The native file module is designed to make it easy to play sound files where no transcoding is necessary. The default FreeSWITCH sound files are in wav format. Generally, these require transcoding when being played to callers. However, if a native format sound file is available then FreeSWITCH can use it.

This module is useful to play files that are encoded as ulaw, alaw, etc.

Formats and File Names

File names need to have the proper extensions. Mod_native_file uses the IANA naming conventions. Please be sure to use those extensions. (Case sensitive?)

Dialplan Examples

When specifying a file to play do not add the file name's extension. FreeSWITCH will pick the correct file.

```
<action application="playback" data="/tmp/hello"/>
```

If you have several files in /tmp then FreeSWITCH will select the appropriate one for the codec in use on the call leg.

```
 /tmp/hello.wav
 /tmp/hello.GSM
 /tmp/hello.PCMU
 /tmp/hello.PCMA
 /tmp/hello.G722
 /tmp/hello.G729
 ...etc.
```

NOTE: the file type for the current codec must be present, otherwise you'll receive an error.

(More to come...)

Script to convert a sound file to specific formats to avoid transcoding

- I've created a script to convert WAV files to various Mod_native_file formats using sox and Asterisk. It currently supports GSM, PCMU, PCMA, G729, and G723.
- I've also used it to convert the default FreeSWITCH prompts into the file formats above. Files available here.
- Use the included fs_encode program in the bin folder.

```
fs_encode -l mod_speex file.wav file.SPEEX

Usage: ./fs_encode [options] input output

The output must end in the format, e.g., myfile.SPEEX
                 -l module[,module]      Load additional modules (comma-separated)
                 -f format               fmtp to pass to the codec
                 -p ptime                ptime to use while encoding
                 -r rate                 sampling rate
                 -b bitrate              codec bitrate (if supported)
                 -v                      verbose
```