# mod_local_stream

## About

This module stream files from a directory and multiple channels connected to the same stream will hear the same (looped) file playback .. similar to a shoutcast stream. Useful for Music-on-hold type scenarios.

## Configuration

This comes from the sample local_stream.conf.xml file. It demonstrates what all is possible in a given local stream. Note that music on hold is really just a local stream for the directory where the default music files are stored:

```
<directory name="default" path="$${sounds_dir}/music/8000">
    <param name="rate" value="8000"/>
    <param name="shuffle" value="true"/>
    <param name="channels" value="1"/>
    <param name="interval" value="20"/>
    <param name="timer-name" value="soft"/>
    <!-- list of short files to break in with every so often -->
    <!-- <param name="chime-list" value="file1.wav,file2.wav"/> -->
    <!-- frequency of break-in (seconds) -->
    <!-- <param name="chime-freq" value="30"/> -->
    <!-- limit to how many seconds the file will play -->
    <!-- <param name="chime-max" value="500"/> -->
</directory>
```

### Directory Attributes

- **name** - The name of the stream, used in the dialplan, i.e. local_stream://<name>
- **path** - Path to the directory containing the sound files for this stream

### Directory Parameters

- **rate** - the sampling rate (in Hertz) of the sound files
- **shuffle** - when set to true will randomize the order in which the sound files are played
- **channels** - number of channels in the sound files (usually "1" but stereo would be "2")
- **interval** - number of milliseconds of silence in between sound files
- **timer-name** - which timer to use (do not change unless you know what you are doing)
- **chime-list** - comma-separated list of "break-in" sound files to play
- **chime-freq** - number of seconds in between the start of each "break-in"
- **chime-max** - max number of "break-in" attempts

### Using a playlist

#### moh.loc

You can stream a playlist by placing a file call 'moh.loc' in the directory path folder. For example, if your stream is configured like:

```
<directory name="dynamic_playlist" path="/var/lib/moh/playlist1">
    <param name="channels" value="1"/>
    <param name="interval" value="20"/>
    <param name="rate" value="8000"/>
    <param name="shuffle" value="false"/>
    <param name="timer-name" value="soft"/>
</directory>
```

You can place a file called 'moh.loc' in '/var/lib/moh/playlist1' which contains a list of file paths like so:

```
/var/lib/music/song1.wav
/var/lib/music/song2.wav
/var/lib/music/song3.wav
```

This configuration will play the files listed in 'moh.loc'

### Using shoutcast

It is possible to play a shoutcast stream in local_stream.

For details on how to set it up please see mod_shout#Using_shoutcast_stream_for_MOH

## How do I call it?

### Dialplan

Call it from the dialplan as follows:

```
<extension name="Music">
  <condition field="destination_number" expression="^787326$">
    <action application="playback" data="local_stream://foo"/>
  </condition>
</extension>
```

To set for legs/bridges etc:

```
<extension name="Music">
  <condition field="destination_number" expression="^787326$">
    <action application="set" data="hold_music=local_stream://foo"/>
    <action application="bridge" data="sofia/default/$1@1.2.3.4"/>
  </condition>
</extension>
```

## Execution Parameters

There is a "timeout=" parameter expressed in milliseconds that limits the duration of the local_stream played. This is useful when multiple instances of local_stream is needed in a single statement.

### Timeout Example

In this example, an incoming call rings local extensions, if no one answers, cell phones are bridged into the call. It would be helpful to let the caller know the progress of the call. As soon as the bridge command starts we want the caller to hear:

1. Custom greeting to caller based on caller ID. 3 seconds.
2. Music for 20 seconds <--- If we use US Ring, some callers are confused by a voice followed by ringing, so music is better.
3. Custom 2nd greeting telling caller to please wait while call is forwarded. 3 seconds. Cell phones start ringing due to leg_delay_start=23 in bridge
4. Music for 20 seconds

If the call times out: please leave a message, vm, emailed, etc.....

If a single bridge was used to ring extenstions, then cell phones, we would need a single early media stream that would play steps 1 through 4:

```
<action application="set" data="ringback=file_string:///foo1!{timeout=20000}local_stream://moh!
file_string:///foo2!local_stream://moh"/>
```

Without the timeout, the first moh would play until the originate_timeout occurred and the caller would never hear foo2 or anything else remaining in the ringback.

# API

## local_stream reload

```
local_stream reload <stream_name>
```

Safely reloads a local_stream configuration using the local_stream.conf.xml

Please note, the reload is not instantaneous. In order to not drop calls, the reload will wait until a streaming file is done playing and the stream is not in use. In the case that the streaming file is done playing and the stream is in use, a ***partial reload*** will take place. Once the stream is not in use and a file is done playing, the ***full reload*** will take place.

- On a ***partial reload***, the following parameters are reloaded from local_stream.conf.xml:
    - shuffle
    - chime-freq
    - chime-max
    - chime-list
    - interval
- On a ***full reload***, the ALL parameters are reloaded from local_stream.conf.xml

## local_stream stop

```
local_stream stop <stream_name>
```

Stops and unloads a local_stream.

In order to not drop calls the 'stop' action will be queued and fire once the local_stream is not in use.

## local_stream start

```
local_stream start <stream_name>
```

Instantly starts a new local_stream. The stream must be configured in local_stream.conf.xml

## local_stream show

```
local_stream show [stream_name] [as xml]
```

Shows status of local_streams. If stream_name is provided, it will display details about the stream.

## local_stream hup

```
local_stream hup <stream_name>
```

The API changed in 1.6, earlier in 1.4 it was:

hup_local_stream => local_stream hup <stream>
reload_local_stream => local_stream reload <stream>
stop_local_stream => local_stream stop <stream>
start_local_stream => local_stream start <stream>
show_local_stream => local_stream show <stream> [as xml]

# FAQ

## What file formats can be in directory?

Q: Do the files need to be of a certain format (eg, wav) and/or a certain sample rate / bit width?
A. It can handle any format, bitrate, and bit width supported by sndfile (See: FreeSwitch Dependencies), but optimal performance will be achieved by resampling to PCM encapsulated in a WAV container at the native bitrate and bit width of the channel. For example, 8kHz, 16kHz, and 32kHz.

## What's this I keep seeing about a leaking stream handle?

Q: I keep seeing an error like this: "Leaking stream handle!" - what should I do?

A: You should not panic. This is more of an "oh by the way" warning than a serious error. Anthony had this to say on a ML thread on the subject:

A: A normal file handle waits for you to read, the local_stream one has a buffer feeding it from a master stream, so when you don't read from it, you lose the audio because the music is playing on without you even when you pause a playlist .

In other words don't sweat it.