Contents

- 1 Usage
- 2 Comments
- 3 Samples
- 4 See Also

Usage

If you are trying to detect fax tones, using mod_spandsp is recommended as of Aug 2011.

Updated, preferred behavior:

```
<action application="set" data="execute_on_tone_detect=set my_tone=true"/>
<action application="set" data="tone_detect_hits=2"/>
<action application="set" data="execute_on_media=tone_detect mytone 820 w +30000"/>
<action application="bridge" data="sofia/gateway/foo/123456789"/>
```

This method will execute the application immediately. If you actually like to run the app async on the media stream specify this:

```
<action application="set" data="execute_on_tone_detect=set::mytone=true"/>
```

Older, still working behavior:

```
<action application="tone_detect" data="<key> <freqs> <flags> <timeout> <app> <data> <hits>">
```

field	description	examples
key	name of the tone (user defined)	• busy • fax
freqs	frequencies to "listen" for	• 425 • 480,620
flags	which direction to sniff the tones in (r = read, w = write)	r
timeout	window of time to listen for the tones in (0 = forever, +<milliseconds> = milliseconds relative to the time tone_detect is called, <time> = time in "epoch" (ie. seconds since January 1, 1970))	• +2000 (relative example resulting in 2 seconds) • 1227132612 (absolute example)

app [optional]	application to call when the tone detect conditions are met	transfer
appdata [optional]	arguments to the application specified above	'1000 XML default'
hits	number of hits needed before calling app/appdata (useful to avoid false positives)	3 means detect the tone 3 times in <timeout> before taking action

Notes:

- Once tone_detect is triggered, it won't be triggered again *unless* you have specified the "hits" argument in which case it will only get triggered once the "hits" count has been reached
- Only "key" and "frequencies" are required (the rest are optional)
- If you want to catch the DETECTED_TONE event with the channel-uuid use something like this: <action application="tone_detect" data="fax 1100 r +8000 **set fax=true** 2">

Comments

Some of the advanced knowledge about tone_detect is actually found in this nice little IRC exchange with anthm and MikeJ.

```
<MikeJ> <action application="tone_detect" data="test 350,440 r +100 hangup 'normal_clearing' 1"/>
<MikeJ> test is name, freqs, +100 is timeout I think..
<MikeJ> anthm?
<trixter_> test I assume is the name of the tone, what is r+100?  and is hangup normal_clearing t
<anthm> <key> <freqs> <flags> <timeout> <app> <data>
<anthm> flags r and w
<anthm> for which way you want to sniff
<trixter_> ok..
<anthm> everything after freqs is optional
<anthm> default is "r'
<MikeJ> timeout in secs ? and allways need the + ?
<anthm> with + is relative to now
<anthm> w/o is absolute
<MikeJ> and if you dont specify app.. you just get event, right/
<MikeJ> ?
<anthm> right
<anthm> once it's tripped it wont anymore
<anthm> but you can reset it with just key
<anthm> if you call one with only key name and it already exists it will reactivate it
<MikeJ> just app.. or api too?
<anthm> key is arbitrary and just what you get in the event
<anthm> app and api are identical but api needs uuid too as first arg and all the other are shift
<anthm> it won't let you decide that
<anthm> =D
<anthm> once you use it at all it adds 1
<anthm> and if you call it again it just adds more to the same 1
```

Samples

How to detect the fax tone and receive a fax with mod_spandsp.

```
    <extension name="5555555">
```

```
    <condition field="destination_number" expression="^(5555555)$"/>
      <action application="tone_detect" data="fax 1100 r +5000 transfer 'fax XML default' 1"/>
      <action application="answer"/>
      <action application="sleep" data="1000"/>
      <action application="ivr" data="main"/>
      <action application="hangup"/>
    </condition>
  </extension>


  <extension name="fax">
    <condition field="destination_number" expression="^fax$">
      <action application="answer" />
      <action application="playback" data="silence_stream://2000"/>
      <action application="rxfax" data="/usr/local/freeswitch/fax/${caller_id_number}-${strftim
      <action application="hangup"/>
    </condition>
  </extension>
```

How to detect the fax tone and receive a fax with socket2me.

```
  <extension name="5555555">
    <condition field="destination_number" expression="^(5555555)$"/>
      <action application="tone_detect" data="fax 1100 r +5000 transfer 'fax XML default' 1"/>
      <action application="answer"/>
      <action application="sleep" data="1000"/>
      <action application="ivr" data="main"/>
      <action application="hangup"/>
    </condition>
  </extension>


  <extension name="fax">
    <condition field="destination_number" expression="^fax$">
      <action application="answer"/>
      <action application="sleep" data="1000"/>
      <action application="set" data="fax_file_name=/usr/local/freeswitch/fax/${caller_id_numbe
      <!--"send" or "recv"-->
      <action application="set" data="fax_mode=recv"/>
      <action application="socket" data="127.0.0.1:8084 async"/>
      <action application="hangup"/>
    </condition>
  </extension>
```

How to detect an inband busy signal and send a real busy message:

```
    <action application="tone_detect" data="busy 480,620 w +15000 hangup 34"/>
```

Detect some SIT tones:

```
    <action application="tone_detect" data="sit-high-1 985.2 w +15000 hangup 41"/>
    <action application="tone_detect" data="sit-low-1 913.8 w +15000 hangup 41"/>
    <action application="tone_detect" data="sit-high-2 1428.5 w +15000 hangup 41"/>
    <action application="tone_detect" data="sit-low-2 1370.6 w +15000 hangup 41"/>
```

How to detect 3 consecutive busy tones any time during a call and hangup when detected:

```
<action application="tone_detect" data="busy 425 r 0 hangup 'normal_clearing' 3"/>
```

See Also

- Dialplan
- Misc._Dialplan_Tools_stop_tone_detect
- Variable_execute_on_tone_detect
- Variable_tone_detect_hits
- Spandsp call progress tone detector