mod_vmd

About

```
NOTE:  There is a new Advanced Voice Mail Detection (AVMD).  It is suggested that you use mod_avmd rather than
this (mod_vmd).  mod_vmd is deprecated and will be removed in a future version.
```

VMD stands for "Voice Mail Detection" and mod_vmd is designed to detect the beep sound at the end of voicemail or answering machine greetings. This is useful in cases where you wish to leave a recorded message on the recipient's message system but don't want to have the pregnant pause that is common when using wait_for_silence.

Usage

From mod_event_socket

To start the mod:

```
api vmd <uuid> start
```

To stop the mod:

```
api vmd <uuid> stop
```

To get mod_vmd events:

```
event plain CUSTOM vmd::beep
```

Lua Example

This is an example of a callback in Lua:

```
local human_detected = false;
local voicemail_detected = false;

function onInput(session, type, obj)
    if type == "dtmf" and obj['digit'] == '1' and human_detected == false then
        human_detected = true;
        return "break";
    end

    if type == "event" and voicemail_detected == false then
        voicemail_detected = true;
        return "break";
    end
end

session:setInputCallback("onInput");
session:execute("vmd","start");
```

Javascript example

This is an example of a callback in Javascript:

```
function onInput(s, type, obj, arg)
{
        try
        {
                if(type == "dtmf")
                {
                        console_log("info", "DTMF digit: "+s.name+" ["+obj.digit+"] len ["+obj.duration+"]\n");
                }
                else if(type == "event" && session.getVariable("vmd_detect") == "TRUE")
                {
                        console_log("info", "Voicemail Detected\n");
                }

        }
        catch(e)
        {
                console_log("err", e + "\n");
        }
        return true;
}

session.answer();
session.execute("vmd", "start");
while(session.ready())
{
        session.streamFile(argv[0], onInput);
}
```

From Dialplan

```
Note: all it does is set chan variable ${vmd_detect} to "TRUE" and nothing else, so don't rely on this for
acting quickly if a beep is detected.
  <!-- mod_vmd test extension (new mod)-->
  <extension name="vmdtest">
    <condition field="destination_number" expression="^(\d{10})$">
      <action application="answer"/>
      <action application="info"/>
      <action application="vmd"/>
      <action application="sleep" data="25000"/>
      <action application="info"/> <!-- Look for chan var "vmd_detect" here -->
      <action application="hangup"/>
    </condition>
  </extension>
```

Configuration

To compile it simply add this to 'modules.conf':

```
applications/mod_vmd
```

And do not forget to rebuild and install:

```
sudo make sure
sudo make install
```

To have FreeSWITCH load it on startup simply add this to '/usr/local/freeswitch/conf/autoload_configs/modules.conf.xml':

```
<load module="mod_vmd"/>
```

Notes

No voicemail detection can claim to be 100% accurate. The industry standard is 80% detection. This module if used properly should exceed the standard by a very wide margin.

If the signal quality is very poor the module may detect a single beep as many beeps and in very rare cases generate false positives. This is not usually an issue in detecting voicemail but keep it in mind when using the mod.

Related conversations

The following conversations took place shortly after the initial release of mod_vmd. They are not documentation per se but they do give some insight as to what the author (Eric Des Courtis) was thinking when he wrote the module as well as some thoughts from the FreeSWITCH developers.

```
<MikeJ> so.. it at least compiles on windows now..
<MikeJ> and I have super bitchy mode on
<MikeJ> so thats good
<ericdc> thats good
<MikeJ> anthm was suggesting making an application interface too
<MikeJ> probably makes sense to have one that just stays in the app till it gets the beep
<ericdc> it does have one
<ericdc> you can use it as an app
<MikeJ> why your right
<ericdc> lol I wrote it
<MikeJ> oh wait
<MikeJ> no.. I mean instead of an app that starts the media bug..
<ericdc> yes
<MikeJ> have a way to have it just block till beep
<ericdc> oh I see
<ericdc> you mean the app function would block?
<ericdc> it could be done I guess
<ericdc> without too many modifications
<MikeJ> yeah.. should be easy.. its a useful use case
<ericdc> so maybe a parameter to tell it to block
<ericdc> until a beep is detected
<ericdc> At the moment we have a lot of things until Dec 25th so I could add it before the new year?
<ericdc> possibly before that if we get ahead
<anthm> i just meant an app to turn the bug on and off
<anthm> so you could start the media bug in your dialplan
<anthm> or js
<anthm> or lua
<anthm> <action application="start_vm_detect"/>
<anthm> that returns instantly
<mercutioviz> I like that idea
<mercutioviz> quick, clean, efficient
<ericdc> It's already there
<ericdc> it has an app interface
<ericdc> you can also stop it
<ericdc> from the app interface
<ericdc> <action application="vmd" /> will start it I believe and the same command with the stop parameter will
stop it
<anthm> ok
<anthm> i only looked at the wiki page
<anthm> that showed api way of doing it
<mercutioviz> I'll test it both ways and make sure the wiki is all correct
* bkw_ (n=brian@freeswitch/developer/bkw) has joined #freeswitch-dev
* ChanServ gives channel operator status to bkw_
<ericdc> the lua example I believe is using the app
<ericdc> not the api call
<mercutioviz> yeah, it's just doing session execute "vmd"
<anthm> ok
<anthm> then it's perfect

<mercutioviz> i just want to make sure that the call from the dp actually works, then we'll iron out the
callback thingy
<ericdc> is it possible to catch callbacks with dialplans?
<mercutioviz> i don't think so? MikeJ?
<MikeJ> ericdc: the events you mean? ... no
<MikeJ> thats why the blocking app is interesting
<MikeJ> like how we do in wait for silence
<ericdc> okay then it's official it unsupported in the dial plan until I add it which is before the end of the
month
<MikeJ> unless I get board :D
```

Related

- mod_avmd