Raspberry Asterisk Configuration

Part 4

Enhancements

Voicemail

A basic enhancement to one's phone system is the requirement to have a caller leave a message when a call is received and you are not available. Asterisk has the built-in capability to take a message and allow one to play it back at a later time.

Three files must be modified to set up voicemail service, voicemail.conf, sip.conf, and extensions.conf.

Voicemail.conf File

For each extension that is to have voicemail an entry must be made. Two basic options exist for receiving a voicemail, basic playback and email. Additional options exist that are beyond the basic scope of this discussion. If one wants the email option then the ability to send it must be also configured into the server. At this time the email option is not reviewed.

In the voicemail.conf file go to the end and create a new section called ***[default]*** .

Within the ***[default]*** section create a mailbox for each extension with the following format:

boxid => password,Username

The boxid is a number that identities the users mailbox. It does not have to be the user's extension for security purposes but is easiest on the user if it is kept the same.

The password is a security number given to the user. This number can be changed by the user when retrieving his / her voicemail.

The Username identifies the owner of the voicemail box. This can be used instead of the extension number to identify the mailbox in the sip.conf file.

For our basic example to extension 314 we might have:

314 => 2413,dennis

Note that the password has been modified from the normal numbers.

For basic voicemail, only the above need be added for each extension.

If one should wish to set up the email process, the format for the entry would be:

boxid => passwor,Username,name@mailaddress

Sip.conf File

Three additional lines are added to each extension in the sip.conf file that is allowed to access voicemail. The full configuration is:

```
[314]                      ; snom phone
type=friend                ; Friends place calls and receive calls
context=internal           ; Context for incoming calls from this user
secret=314
host=dynamic               ; This peer register with us
description = snom_phone
mailbox=314@default,2413        ; Mailbox(-es) for message waiting indicator
subscribemwi=yes           ; Only send notifications if this phone
                           ; subscribes for mailbox notification
vmexten=222                ; dialplan extension to reach mailbox
                           ; sets the Message-Account in the MWI notify message
                           ; defaults to global vmexten which defaults to "asterisk"
```

The three new lines are mailbox, subscribemwi, and vmexten.

The mailbox line authenticates the mailbox number and password for the message waiting indicator.
314@default specifies the extension number.
2413 specifies the password specified in the voicemail.conf file.

The subscribemwi line specifies that if the phone supports the message waiting indicator (mwi) through the appropriate protocol, then it is to be utilized, assuming that the phone has a mwi lamp.

The vmexten line is only required if the phone supports the discovery for a voicemail button on the phone from the Asterisk server, a feature that is not detailed for this discussion. If the phone does not support a message waiting button this line is not required.

Extensions.conf File

To each extension we will add 3 additional lines after modifying the first:

exten => 314,1,Dial(SIP/314,20)

exten => 314,2,Voicemail(314<,option>)
exten => 314,3,Playback(vm-goodbye)
exten => 314,4,Hangup

Note that the first line has been modified with the ",20". This provides a delay of 20 seconds, or three rings. Standard ring cadence is 2 seconds on and 4 seconds off, thus allowing a short period after the third ring. (One may modify this time if desired.)

The second line specifies that the call is to be forwarded to the Voicemail box and that an option exists. The general format for the line is:

exten => extension,2,Voicemail(Boxid<,options>)

The boxid is commonly the extension number but can be another number if desired for security purposes. Note that the boxid is a separate number, not the extension. Recommend that it be kept the same to limit confusion when a user wishes to retrieve messages.

These options include:

blank	Give standard full voicemail response.
s	Skip all voicemail responses and give a tone.
u	Give unavailable message and please leave message
b	Give busy message and please leave message
su	Give unavailable message only
bu	Give busy message only

After the caller has left a message, the system will play the goodbye message and then hang up (lines 3 and 4). In general the options are not required allowing the standard messages to be given.

Retrieving Voicemail

Having voicemail service is no good if one is not able to retrieve it. One must create two additional extensions in the ***extensions.conf*** file to allow one to collect his / her voicemail, one for dialing directly from one's phone and a second for when dialing from someone else's phone.

From own phone

Need to create an extension to dial. By default of the installation extension 8500 already exists but can change this if desired. The standard number is for demo purposes and we want to add more features.
Create a new extension, 222 in the extensions.conf file under [internal]:

exten => 222,1,VoicemailMain(${CALLERID(num)})

VoicemailMain   Main menu for voicemail administration

${CALLERID(option)}) Takes callerid for specified option

option num Maps to boxid in the voicemail.conf file since using the same number as the extension

option name Takes name / title for the phone being used

option all Takes both number and name

For name & all need to set up appropriate info in boxid

For this discussion, name and all are not used.

s${CALLERID(option)}) Skips password check (not recommended)

From Different Extension

Need to create a different extension which would ask for the mailbox number and password.

Create a new extension, 223 in the extensions.conf file under [internal]:

exten => 223,1,VoicemailMain()

With this requirement the user must dial in his / her extension in order to collect the voicemail. The password must also be entered.

Voicemail Options

Several options exist after one has retrieve one's voicemail.

Options include:

- 2: Change Folders
  Allows one to switch between different voicemail folders
- 3: Advanced Options:
  - 5: Leave message
  - *: Return to main menu
- 0: Mailbox options:
  - 1: Record unavailable message
  - 2: Record busy message
  - 3: Record name
  - 4: Record temporary greeting
  - 5: Change password
  - *: Return to main menu

Automated Attendant

Two options are available for setting up an automatic answering and redirecting a call, Auto Attendant (AA) and Interactive Voice Recognition (IVR). IVR is a lot more complex options whereas Auto Attendant provides for a basic answering of a call and then through a basic response from the user redirects the call. The first requirement to set up an Auto Attendant (AA) system is to define what you wish to accomplish. For a simple system that can be supported by the Raspberry Pi running Asterisk we propose two scenarios, both basically the same just different ends.

The first scenario is for use in your home where an incoming call is directed to an AA extension and that then asks the caller who he (she) wishes to talk to. For the second scenario the caller is asked which department they wish to talk to. The only difference is in the delivered message. When setting up the message options Asterisk includes a large number of standard voice responses, for those that are different you will have to record the response yourself, which will be discussed.

For our simple example we will have a small business system with six departments where we will set up the following extension requirements:

Business			**Home**		
Person	Extension	Dial	Person	Extension	Dial
Sales	314	1	John	314	1
Manufacturing	315	2	Diane	315	2
Management	316	3	Robert	316	3
Attendant	100	0	Nancy	317	4
Voicemail	223	9	Everyone		5
			Voicemail	223	0

We need to set up an AA extension where an incoming call, from CallCentric in our example since we have previously configured our system for that. The extension will be 301. Requirements exist for configuring both the ***sip.conf*** and ***extensions.conf*** files.

AA Step 1

This initial step is used to confirm the basic concept of having a message played to the caller. After this step additional modifications will be made to actually provide messages that one wants. The two files that need to be modified are the sip.conf and extensions.conf.

Configuring the sip.conf file

The sip.conf file has already been set up but we need to validate its configuration in the [general] section.

```
Udpbindaddr = ::                  ; Support for both IPv4 and IPv6
context = from-callcentric        ; Previously set up to support CallCentric trunking
host=dynamic
type=friend
```

Save the file and exit.

Configuring the extensions.conf file

The extensions.conf file will require a new section to forwarding the call, this being extension 300 for our example. This example is to only demonstrate the ability of Asterisk to answer a call and give a simple message and then hang up.

To gain a basic understanding a temporary answering configuration is setup in the [internal] section (at the end of the file), add the following entry:

[internal]
exten => 300,1,Answer()
exten => 300,2,SayDigits(1234)
exten => 300,3,Hangup()

Save the file and exit.

Quick Test

Now start the Asterisk CLI by issuing the command ***asterisk –r*** . Then issue the command to reload the files:

CLI> ***core reload***

You should now be able to dial extension 300 and obtain an answer with the dictation of the numbers.

After you see that it works, comment out the three lines as a new entry will be made.

Recording Announcements [1] [2]

In order to have an appropriate message for AA one must record it. To do this the best procedure is to set up a special dedicated extension to record messages from one's phone. You can use an external source but using the phone is more than adequate.

1 http://www.voip-info.org/wiki/view/Asterisk+cmd+Record
2 http://www.voip-info.org/wiki/view/Asterisk+sound+files

The general format of the ***record*** function is:

Record(filename.format[,silence[,maxduration [,options]]]_

where

filename	name of the file that you wish to create
format	filename extension and format that it is recorded in
silence	the maximum amount of silence that may occur during the recording
maxduration	the maximum duration of a recording
options	options for the recording
a	Append to an existing file
n	Do not answer but record anyway
q	quiet (do not play a beep tone)
s	skip recording if line not yet answered
t	use alternate '*****' terminator DTMF key instead of default '**#**' key
x	ignore all termination DTMF keys, record until hangup
k	keep recorded file upon hangup
y	terminate recording if **any** DTMF digit is received

From the Asterisk CLI you can issue the command ***core show file formats*** to see the various formats that are available. It is recommended that the recording be either in gsm or ulaw (mu-law) format.

By default the recording will be saved in the directory ***/usr/share/asterisk/sounds*** as ***Announce*** (note documentation points to a different directory).

There is one issue that will prevent the file from being saved – the sounds directory is owned by root, the administrator. To allow the recordings to be saved the administrator must modify the permissions to the directory. Use the following sequence to validate and change the permissions:

sudo ls –l /usr/share/asterisk/

Note that the entry for the ***sounds*** directory reads:

drwxr-xr-x 3 root root 4096 Aug 8 18:15 sounds

The date at the end will vary and that the number following the last x may be different.

Now issue the command:

sudo chmod 666 /usr/share/asterisk/sounds

Repeat the above command to display the long listing, you should now see:

drwxrwxrwx 3 root root 4096 Aug 8 18:19 sounds

The directory may now be written to by any user, including the user asterisk.

From the Asterisk CLI reload the configuration files with the ***core reload*** command.

Test the announcement recording by dialing extension 200 and after the tone record some announcement. Change to the /usr/share/asterisk/sounds directory (documentation specifies /var/lib/asterisk/sounds but in testing it was found that it was in the /usr... directory). You should now see the file Announce.gsm.

To the extensions.conf file create the following extension with attributes: [3]

```
[internal]
exten => 200,1,Answer
exten => 200,2,Playback(vm-intro)
exten => 200,3,Wait(1)
exten => 200,4,Record(Announce.gsm)
exten => 200,5,Wait(2)
exten => 200,6,Playback(Announce)
exten => 200,7,Hangup()
```

The above extension will record and then play back your recording. Multiple extensions are available but ***gsm*** is the preferred format.

Documentation specifies that line 6 for the playback one can use the format:

Playback(${RECORDED_FILE})

can be used but it was found that through experimentation that it did not and the same filename (without extension) had to be used, as in the example above.

What is required in this process is that for a properly recorded announcement, the file in the /usr/share/asterisk/sounds must be immediately renamed in order to allow one to record another message. Not a difficult situation with several terminal windows open. After the file is validated to what you want, rename the file with a name that is appropriate to the announcement. As an example, say we have made a recording for the Sales department. Rename it as:

mv Announce.gsm Sales.gsm

You can also use the ***Festival*** utility which is a speech synthesizer to give message, only requiring that the Festival utility be installed (can use apt-get install festival).

AA Step 3 – Setting up options menu [4]

Now that the basic concept of answering the call has been verified and that one can record messages, the answering options needs to be changed in the extensions.conf file.

For example say we rename the recorded files as:

1st Announce.gsm →	Opening.gsm	Opening 1
2nd Announce.gsm →	Extensions.gsm	Opening 2

---

3 http://www.asteriskdocs.org/en/3rd_Edition/asterisk-book-html-chunk/Autoattendant_id287976.html

4 http://www.asteriskdocs.org/en/3rd_Edition/asterisk-book-html-chunk/Autoattendant_id272753.html

3rd Announce.gsm	→	TransferSales.gsm	Dial 1
4th Announce.gsm	→	TransferManufacturing.gsm	Dial 2
5th Announce.gsm	→	TransferManagement.gsm	Dial 3
6th Announce.gsm	→	TransferOperator.gsm	Dial 0
7th Announce.gsm		TransferVM	Dial 9

The previous entry to extensions.conf for extension 300 will be modified to allow for an example of how the basic system works. Re-write the extension 300 like the following:

[internal] [5]
exten => 300,1,Answer()
exten => 300,2,Set(TIMEOUT(digit)=3)
This sets the inter-digit timer
exten => 300,3,Background(Opening & Extensions)
These are in the default /usr/share/asterisk/sounds directory
*Note no spaces around the '**&**'*
exten => 300,4,WaitExten(5)
This allows multiple digits to be entered

At this point the caller has dialed the desired number and the code is now set up to proceed to the desired extension. Add the following extensions:

exten => 1,1,Playback(TransferSales)
exten => 1,2,Dial(SIP/314)
Jump to extension 314 priority 1 line

exten => 2,1,Playback(TransferManufacturing)
exten => 2,2,Dial(SIP/315)

exten => 3,1,Playback(TransferManagement)
exten => 3,2,Dial(SIP/316)

exten => 0,1,Playback(TransferOperator)
exten => 0,2,Dial(SIP/301,1)

exten => 9,1,Playback(TransferVM)
exten => 9,2,Dial(SIP/223)

exten => i,1,Playback(TransferOperator)
exten => I,2,Dial(SIP/301,1)
Dial Operator if invalid digit(s) dialed

5 https://wiki.asterisk.org/wiki/display/AST/Handling+Special+Extensions

exten => t,1,Playback(TransferOperator)
exten => t,2,Dial(SIP/301,1)

Dial Operator if digit timeout occurs

The '***I***' extension is for when someone dials a non existent extension / option.

The '***t***' extension is for when no number is dialed and the wait times out.

Note that the filename does not include an extension. Asterisk knows how to interpret the file type (it is actually embedded into the file data).

The ***Background*** command plays a message and listens for the user to use the keypad to input a number (extension). The ***Playback*** command only plays a message.

After the above has been done save the changes to the extensions.conf file and do a ***core reload*** from the asterisk CLR.

Now reload your extensions.conf file by typing the below command into the asterisk cli.

dialplan reload or
core reload

Now dial into your AA again and try it out by pressing 1, 2, and 3.