



(<http://bernaerts.dyndns.org/linux/17...-voicemail-mp3>)

This guide will explain how to configure **Asterisk PBX** to send voicemail as email with messages as **mp3** attachment.

If, like me, you are using an Asterisk server, you may be using the voicemail by email functionality. It allows your Asterisk PABX to send your by email all the messages received in your voicemail.

Just imagine :

- someone is calling you on your landline
- he is leaving a message on your answering machine
- after few seconds, you are receiving a mail on your android or i-phone with the message as an attachment

You can call back immediatly or even forward the email for someone else to deal with that call !

One of the main drawback of that fonctionnality is that Asterisk can generate attachments using only the standard telephony codecs (wav49, gsm, wav). The wav format is widely recognised, but it is quite uncompressed. The wav49 or gsm formats are better in terms of compression, but sadly they are not recognised by all the devices able to deal with your emails.

So why not to use **mp3** format for the voicemail attachments ?

This is what this guide is all about. It will explain how to catch the emails sent by Asterisk and convert the audio attachment from a **wav** file to a **mp3** file.

NOTE: This guide is based on an Asterisk 1.6 version, running under Debian Squeeze.

As it uses only some very basic & standard linux tools, it should be compatible with any Asterisk installation.

It is based on the standard email structure generated by Asterisk.

If you have modify the Asterisk email format, you will need to adjust the script.

PRINCIPLE

The main principles are :

- Asterisk voicemail is configured to generate some wav attachments (uncompressed, best quality available)
- Instead of sending the mail to sendmail, Asterisk is configured to send the mail to one specific script
- This script will do all the job, and finally give back the mail to sendmail

The script will do some ordered tasks :

1. catch the stream,
2. analyse the email content,
3. separate each mime part,
4. pullout the audio attachment,
5. compress it to mp3,
6. regenerate the mail content,
7. send it to sendmail as expected

With this approach, you just need some specific configuration on Asterisk side.
From Asterisk side, the emails are just sent thru another emailer, that's it !

Unlike other methods, you don't need any patch for Asterisk.
You are still using a regular Asterisk, available from your distribution.
So, you can apply any standard update.

Any update to a new Asterisk version should remain fully compatible.

INSTALL THE PACKAGES

First thing to do is to install the packages needed by our script (unix2dos, dos2unix & lame).

```
aptitude install dos2unix lame
```

SETUP THE MAIN SCRIPT

Once these packages are installed, we can generate the script that will do all the job.

As Asterisk generates the mail content, including the wav attachment encoded in base64, the script main job will be to :

1. get the attachment from the email content
2. convert it to mp3
3. regenerate the mail content with new attachment
4. send the mail thru sendmail

This job will be done with some very basic linux tools like sed, awk, grep, base64, unix2dos, ...

One important thing is to respect the body structure of the mail content, including the **LF** and **CRLF** used to separate lines, depending on the different mime parts. In fact, in the mail body generated by Asterisk, all the lines are separated with LF, with the exception of the base64 coded audio attachment where they are separated with CRLF.

For the mp3 conversion, 2 compression options are available :

- CBR - Constant Bit Rate compression will create mp3 streams compatible with any device, but the size/quality ratio is not optimised.
- VBR - Variable Bit Rate is much more efficient in term of file size, but playback has been tested

very poor on some smartphones.

The script is providing both the compression parameters, you just need to adjust it to your need.

The detailed phases of the script are as follow :

1. Creation of a temporary directory, randomly generated, where we will generate all the temporary files
2. Dumping of the mail content sent by Asterisk in a file
3. Detection of the boundary pattern which separate the differents mime parts in the mail
4. With this pattern, cutting of the mail content into mime parts
5. Decoding of the WAV attachment which is base64 coded (you have to ignore parasite caracters in some configurations)
6. Compression into mp3 format
7. Coding of the mp3 audio file back to base64
8. Re-assembly of the full mail content, using the new mp3 attachment
9. Sending of the mail thru sendmail
10. Deletion of the temporary directory

This **sendmailmp3** script will be placed side to the sendmail command, under **/usr/sbin**.

`/usr/sbin/sendmailmp3:`

```
#!/bin/sh
# Asterisk voicemail attachment conversion script
# Revision history :
# 22/11/2010 - V1.0 - Creation by N. Bernaerts
# 07/02/2012 - V1.1 - Add handling of mails without attachment (thanks to Paul Thompson)
# 01/05/2012 - V1.2 - Use mktemp, pushd & popd
# 08/05/2012 - V1.3 - Change mp3 compression to CBR to solve some smartphone compatibility (thanks to Luca Mancino)
# 01/08/2012 - V1.4 - Add PATH definition to avoid any problem (thanks to Christopher Wolff)

# set PATH
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"

# save the current directory
pushd .

# create a temporary directory and cd to it
TMPDIR=$(mktemp -d)
cd $TMPDIR

# dump the stream to a temporary file
cat >> stream.org
```

```

# get the boundary
BOUNDARY=`grep "boundary=" stream.org | cut -d'"' -f 2`

# cut the file into parts
# stream.part - header before the boundary
# stream.part1 - header after the boundary
# stream.part2 - body of the message
# stream.part3 - attachment in base64 (WAV file)
# stream.part4 - footer of the message
awk '/'$BOUNDARY'/{i++}{print > "stream.part"i}' stream.org

# if mail is having no audio attachment (plain text)
PLAINTEXT=`cat stream.part1 | grep 'plain'`
if [ "$PLAINTEXT" != "" ]
then

    # prepare to send the original stream
    cat stream.org > stream.new

# else, if mail is having audio attachment
else

    # cut the attachment into parts
    # stream.part3.head - header of attachment
    # stream.part3.wav.base64 - wav file of attachment (encoded base64)
    sed '7,$d' stream.part3 > stream.part3.wav.head
    sed '1,6d' stream.part3 > stream.part3.wav.base64

    # convert the base64 file to a wav file
    dos2unix -o stream.part3.wav.base64
    base64 -di stream.part3.wav.base64 > stream.part3.wav

    # convert wav file to mp3 file
    # -b 24 is using CBR, giving better compatibility on smartphones (you can use -b 32 to increase quality)
    # -V 2 is using VBR, a good compromise between quality and size for voice audio files
    lame -m m -b 24 stream.part3.wav stream.part3.mp3

    # convert back mp3 to base64 file
    base64 stream.part3.mp3 > stream.part3.mp3.base64

    # generate the new mp3 attachment header
    # change Type: audio/x-wav to Type: audio/mpeg
    # change name="msg----.wav" to name="msg----.mp3"
    sed 's/x-wav/mpeg/g' stream.part3.wav.head | sed 's/\.wav/\.mp3/g' > stream.part3.mp3.head

    # generate first part of mail body, converting it to LF only
    mv stream.part stream.new
    cat stream.part1 >> stream.new
    cat stream.part2 >> stream.new
    cat stream.part3.mp3.head >> stream.new

```

```
dos2unix -o stream.new

# append base64 mp3 to mail body, keeping CRLF
unix2dos -o stream.part3.mp3.base64
cat stream.part3.mp3.base64 >> stream.new

# append end of mail body, converting it to LF only
echo "" >> stream.tmp
echo "" >> stream.tmp
cat stream.part4 >> stream.tmp
dos2unix -o stream.tmp
cat stream.tmp >> stream.new
```

fi

```
# send the mail thru sendmail
cat stream.new | sendmail -t
```

```
# go back to original directory
popd
```

```
# remove all temporary files and temporary directory
rm -Rf $TMPDIR
```

Be sure to make it executable.

NOTE: If you are using some other distributions than debian or ubuntu or some self compiled packages, you may have to specify the full path for the executable files (/usr/local/bin/lame for example).

VOICEMAIL CONFIGURATION

On the Asterisk server side, we just need to configure the voicemail extension to use or script to send the mails.

This is done thru the **voicemail.conf** file, where :

- In the **[general]** section, you need to setup some email specific parameters
- In the **[default]** section (I suppose you use the default context for your voicemail), you need to setup the email address to use for each voicemail box.

So, finally your **voicemail.conf** configuration file should include these parameters :

```
/etc/asterisk/voicemail.conf?
```

```
[general]
; Formats for writing voicemail. WAV is the best quality.
```

```
format=wav  
; Who the e-mail notification should appear to come from  
serveremail=yourpabxname@your.domain.com  
; the email should contain the voicemail as an attachment  
attach=yes  
; You override the default program to send e-mail to use the script  
mailcmd=/usr/sbin/sendmailmp3
```

```
[default]  
; here you declare your voicemail, the email address which will receive the email & the automatic deletion  
of the message after mail is sent  
yourvoicemailid => , yourname , your.email@address.com , , delete=yes
```

Once the file is saved, you just need to restart Asterisk for the configuration to be operational.

```
/etc/init.d/asterisk restart
```

That's it !

From now on, your Asterisk server is sending all the voicemails by email using mp3 attachments.

Hope it helps.