

Design News Gadget Freak

“AutoNag” reminder system

Background:

This is an automated paging system for playing back pre-recorded announcements at specified times. It has the side benefit of being a full voice over IP (VOIP) telephone switch as well. It is built on all open-source software.

A fairly old PC is acceptable for the server, I've had success with machines as slow as 266 MHz, though I recommend 300 MHz or more. A minimum of 128 MB of RAM is required, but 256 will provide better performance. Be certain that you do not have any important data on this machine, as **the installation process will erase the hard drive**. An excellent and inexpensive machine is the Everex gPC. It uses very little power and is available for under \$200 at the time of this writing (look for the version with the Linux OS, not the Windows version, which costs more).

An external audio amplifier is recommended, located near the PC. Wires to paging speakers may then be run to multiple locations. Ensure that you don't put too many speakers in parallel, or you will overload the audio amplifier. Allied stocks 70V speaker isolation transformers that can be used in conjunction with paging amplifiers for best multi-speaker performance. For a small home installation, I just use two speakers, but also distribute the paging messages through the VOIP speaker phones in the house.

Server OS Installation:

Visit <http://www.pbxinaflash.net/downloads/> and download the ISO image from one of the mirror sites. Burn the ISO to a CD – this will be the system installation disc. This is a bootable (as long as your BIOS supports it) CD. Insert it into the computer you plan to use as the server, again noting that **the installation process will erase the hard drive!** Don't use this disc on a machine that still contains any important data. You will need an internet connection during the installation process, but the network card may be removed after the setup process is complete. You will also need a sound card (or built-in audio) that's supported by Linux (specifically, CentOS 5) or you will have to use a SIP-capable speakerphone for the sound output.

As of this writing, the current version is 1.3, and unless you're using a supercharged machine, you'll need the 32 bit version. If you'd like to know about more of the capabilities of the software, look around <http://www.pbxinaflash.net/> where you'll find a support forum, scripts site, and much more. If you prefer not to use a stand-alone machine and would like to run this as a virtual system within Windows or the Mac OS, there are VMWare and other virtualization options available as well, but this guide will focus on using a dedicated PC.

Boot the computer with the CD you burned from the ISO image, again remembering that your **hard drive will be erased**. At the *boot:* prompt, type `text` and press `<enter>`. This will start a text mode installation (I suggest not using the graphical installation mode in case your video card is too old and not supported.)

You will be presented with a series of prompts. Select the US (default) keyboard layout by pressing `<enter>`, leave the partitioning type at the default by pressing `<tab> <tab> <enter>`. At the

“review partitioning” screen, select NO by pressing `<tab> <enter>`. Select your timezone, as the reminder system will use the system clock. Choose a city in your timezone from the list, then tab to the OK button and press `<enter>`. The next step is critical – you are prompted to enter a root password for the system. Enter it twice, as prompted, once in each box, and `<tab> <enter>` to continue the installation process. At this point, you can take a long break while the system is set up. At the end of the process, you are prompted to remove “any media used during the installation process” (the CD in the CD-ROM drive). Take out the CD and press `<enter>` to reboot.

After rebooting, the system will attempt to connect to the pbxinaflash website to download the remaining software. Choose option A, which will download the most recent package. Once the download begins, take an even longer break while the software installation completes.

Server Configuration:

Eventually, after a reboot or two, you will see a `pbx login:` prompt. Log in with the username `root` and the password you entered earlier. Now you will see a `root@pbx:~` prompt.

Now is a good time to test your sound card to ensure it's compatible with CentOS and therefore pbxinaflash. Type `rec text.wav <enter>` to start an audio recording with your sound card. If you receive an error, your card may need to be configured or it may not be compatible. You may be able to use Google to find a solution if you have this problem. If you don't get an error, press `<control> c` to cancel the recording.

It's time to edit some configuration files to enable paging through the sound card. First, switch user accounts so you're operating as the asterisk user by typing `su - asterisk <enter>` (there is a hyphen in between `su` and `asterisk`, which tells the system you want the environment loaded when you switch to the asterisk user).

Edit the `modules.conf` file by typing `nano -w /etc/asterisk/modules.conf <enter>`, which will open a text editor. Use the keyboard arrow keys to change this line near the bottom of the file from:

```
noload => chan_oss.so
```

to

```
load => chan_oss.so
```

Type `<control> x` to exit, type `y` to save the file, and press `<enter>` to confirm the save.

Now create a new oss configuration file by typing `nano -w /etc/asterisk/oss.conf <enter>` to start the editor. The file will be empty, type in the following exactly:

```
[general]
autoanswer=yes
context=from-internal
overridecontext=yes
extension=s
language=en
playbackonly=yes
```

Again, type `<control> x` to exit, `y` to save the file, and `<enter>` to confirm the save.

Now edit your `extensions_custom.conf` by typing `nano -w`

`/etc/asterisk/extensions_custom.conf` <enter> and then scroll down until you find the `[from-internal-custom]` section, and insert the following lines at the beginning of the section (right after `[from-internal-custom]`):

```
exten => 199,1,Dial(console/dsp,20,A(beep))
```

```
exten => 199,2,Hangup()
```

Now go to the very bottom of the file and add the following:

```
[custom-announcement]
```

```
exten => _X.,1,Answer
```

```
exten => _X.,n,Wait(3)
```

```
exten => _X.,n,Playback(beep)
```

```
exten => _X.,n,Playback(custom/announcement-${EXTEN})
```

```
exten => _X.,n,Wait(1)
```

```
exten => _X.,n,Hangup
```

And as before, save the file and exit the text editor. Once you are back at the `[asterisk@pbx:~]` prompt, type `exit` <enter> to return to the `root@pbx:~` prompt. Type `asterisk -rx 'restart now'` <enter>, which will reload the Asterisk dialplan and the configuration changes we made.

Now it's time to upload the announcement sound files. Record the announcement audio on any computer on the same network with a microphone and save each of them as a `.wav` file, saved PCM Encoded, 16 bits, at 8000 Hz sample rate. Use a web browser on this computer to connect to the new paging server by typing `http://<ip-address-of-server>/admin` into the location bar of the browser. To find the IP address, type `ifconfig` <enter> at the command prompt (look for "inet addr:"). The default username and password are both `admin`, which can be changed if desired (type `help-pbx` at the command prompt for details).

When the page opens, select the link on the left hand side for "System Recordings." From this page you can upload the announcement sound files. Browse to the sound file, then click "Upload." After the upload completes, the page will reload. Now enter a name for the recording in the "Name this recording" box. Name each file with a unique name, and use names with no spaces (substitute hypens or underscores, and save all names in lowercase letters for simplicity). I encode the time of day in the name of the announcement to make it easier to change later.

For example, I'll upload `announcement-0800.wav` to my server for an 8:00 am announcement. Click the "Save" button after uploading each recording, and it will be added to the list on the right side of the window. Note that the file naming convention must match what we put in the `extensions_custom.conf` above.

The final step is to build cron jobs to generate call files on the server. A call file is saved in the `/var/spool/asterisk/outgoing` folder, and automatically creates a phone call event at the date and time (modification date and time) of the call file itself. We use the `touch` command to change the modification dates into the future for each call file, as it's generated. Asterisk places the calls (and plays the announcements) at the right time. The cron job runs once a day, just after midnight, and generates all the call files for the day. You can do this differently, if desired, and you could have a cron job just create a single call file at the time it's needed. If you do so, remember that the call file must be created elsewhere, then moved into the "outgoing" folder, rather than created in place. Otherwise the file may not be complete when Asterisk tries to access it.

Switch back to the asterisk user account by typing `su - asterisk <enter>` to get to the `[asterisk@pbx:~]` prompt. Create a folder for the cron jobs by typing `mkdir cron.daily<enter>` and change to the directory with `cd cron.daily<enter>`. Create the script by typing `nano -w announcements.sh<enter>` and copy and paste the following, substituting the file names and times as appropriate for your announcements.

```
#!/bin/bash
makecallfile ()
{
cat <<EOF > /var/lib/asterisk/tempcallfile
Channel: Local/199@from-internal
MaxRetries: 3
RetryTime: 60
waittime: 60
callerid: "AutoNag" <5551234>
Context: custom-announcement
Extension: $1
Priority: 1
EOF
touch -t `date +%Y%m%d$1.17` /var/lib/asterisk/tempcallfile
mv /var/lib/asterisk/tempcallfile
/var/spool/asterisk/outgoing/announce$1.call
}
makecallfile "0800"
makecallfile "0805"
makecallfile "0810"
#and so on...
```

Save the file with the usual `<control>x` then `y <enter>`.

Make this script executable by typing `chmod +x announcements.sh<enter>`.

Set up the cron job to run by typing `env EDITOR='nano -w' crontab -e <enter>` and add the following line to the file for weekday-only announcements (the 1-5 in the syntax is for Monday through Friday, change it to * for 7-day-a-week announcements). This will run at 4 minutes past midnight.

```
4 0 * * 1-5 run-parts /var/lib/asterisk/cron.daily >/dev/null 2>&1
```

As always, `<control>x, y, <enter>` will save the file. This concludes the setup of the system. You can test individual announcements by creating a text file with the following (substituting the announcement file name as appropriate, then copying the file to the `/var/spool/asterisk/outgoing` folder.

```
Channel: Local/199@from-internal
MaxRetries: 3
RetryTime: 60
waittime: 60
```

callerid: "AutoNag" <5551212>
Context: custom-announcement
Extension: 0800
Priority: 1

The modification time will be in the past, so Asterisk should make the announcement immediately. If it doesn't work, examine the `/var/log/asterisk/full` logfile for clues to the problem. Perhaps the file names aren't correct, or there is a typo in one of the files you modified.

Parting Thoughts:

This project can be complex, particularly if you don't use it for any other functions. I encourage you to explore the world of VOIP telephony. Start out at <http://www.pbxinaflash.com/> and <http://www.nerdvittles.com/> (neither of which I'm associated with, I'm just a fan). The forums are great and the tutorials are highly instructive.

With a few well-chosen VOIP phone providers, you can cut your phone bill down quite a bit, too.

I started out setting up an Asterisk-based phone system at work for my day job, but ended up with an even more elaborate system at home. I've added to the system, it's also an OpenVPN server for remote access to my network when I'm away from home, a "squid" web proxy to speed up my web surfing, it runs dansguardian to filter web content for the whole network, and it monitors my home network with nagios and other tools. Overall, pbxinaflash is a great mechanism to learn Linux and other great open-source tools.

If your BIOS supports it, you may want to set up the machine to reboot automatically after a power failure.

