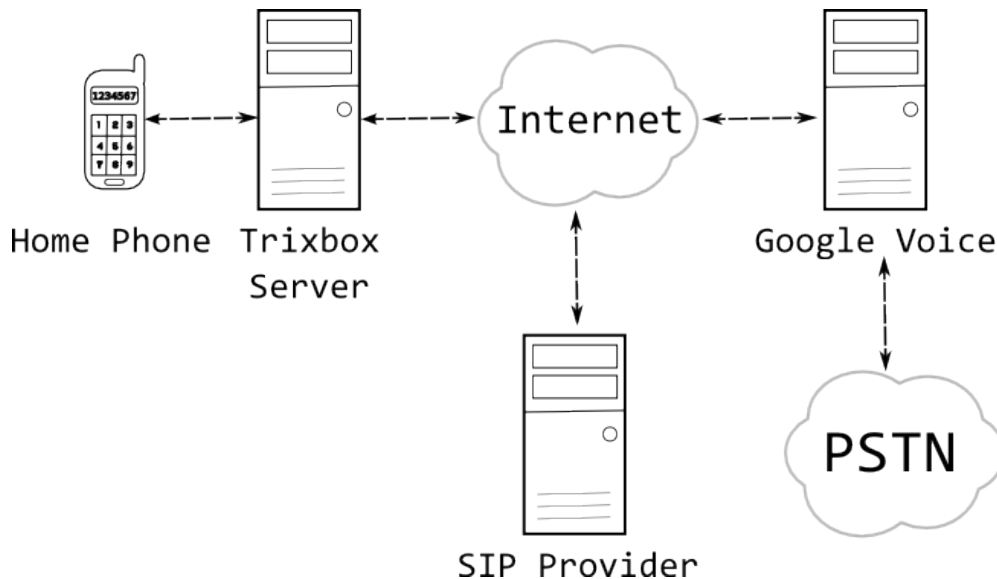


How to scrap your landline and switch to Google Voice

Trixbox is a freely available linux distribution that is pre-configured to run telephony software such as Asterisk (the best free PBX software on the planet).

Finally, after days of fighting with config files, I was able to get Trixbox, a free SIP provider, and Google Voice working together a home phone system! I could go on for hours about how Google Voice (hereafter referred to as GV) and Asterisk will revolutionize telecommunications, but I'm going to assume that if you are reading this tutorial then you are probably already in agreement with me. I will say upfront that this tutorial is not for the faint of heart. I expect you to have a basic working knowledge of UNIX-based operating systems, networking, and VoIP technology. In case your understanding of internet telephony is a little fuzzy, I have included a simple diagram below that shows how the various components of our setup will be connected.



This article is very similar to others out there such as the massive one from [nerdvittles](#). They were very helpful to me, but not quite what I needed. Most existing tutorials cover setup with a DID (http://en.wikipedia.org/wiki/Direct_inward_dialing) from IPKall or SIPGate, but I have chosen to go with IP Communications. IPKall does not have any sort of CallerID, so I initially set up a DID with SIPGate, but I experienced a 20-second delay on average between the time I called my GV number and the time my phone started ringing so I switched providers.

That said, let's get started. First you will need a server that you can use for running Asterisk and Trixbox. An older machine with a Pentium III or above

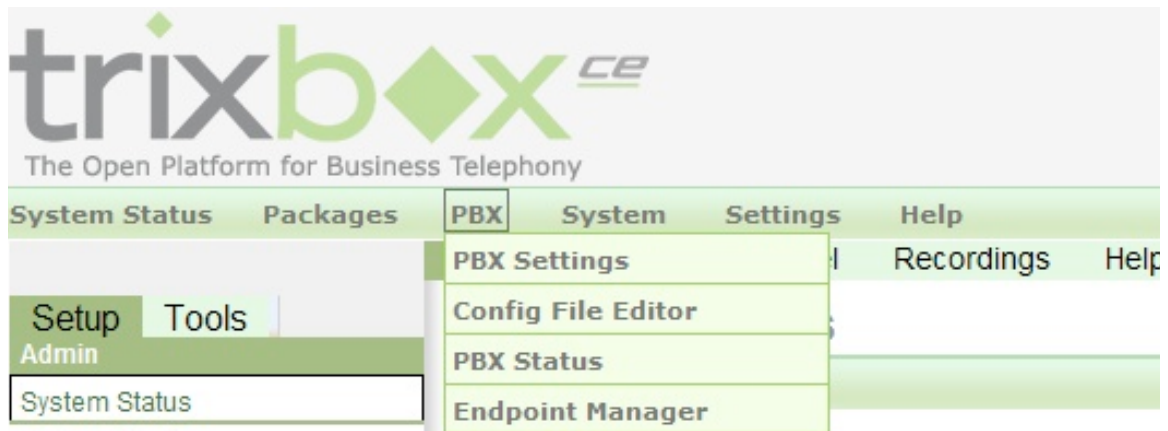
Although a cheap system will work fine, it may not handle multiple calls, call waiting, and voicemail as nicely as a better system. If you have the option to upgrade any of the components, upgrade the RAM first.

processor will work fine (or you can set up a virtual machine if you have a decent desktop that can double as a server). Second, you will need an account with IP Communications. IP Communications will assign you a free phone number from which you can receive calls. Your calls will be answered by the Trixbox server, and, if all goes well, your phone will start ringing. You can sign up by going to <http://www.ipcomms.net/product-freedid.html>.

To make setting up your server or Virtual Machine easier, we are going to use Trixbox. The installation CD may be downloaded here: <http://www.trixbox.org/downloads>. I am going to assume that if you are attempting this, then you already know how to burn a CD and install it on to your server. If you feel stuck at any point in reading this tutorial, call up the local computer guru and he'll be happy to set it up for you if you bribe him with some food.

After you have a working installation of Trixbox, you will need to install a little Python package called pygooglevoice. This is one of the magic pieces to our puzzle that allows us to make outgoing phone calls using our Google Voice number. To install it, log in via the console of your Trixbox machine and type `easy_install pygooglevoice`.

At this point, we have everything we need installed on the server. Now comes the tricky part— configuring it. Configuration will be done from the Trixbox web interface. You can connect to your Trixbox installation from your web browser by going to <http://your-server-ip-address/maint> (replacing your-server-ip-address with the IP address of your server). Note that the default username is `maint` and the password is `password`. Our first configuration will take place in the Config File Editor. A screen clipping is shown below to help you find the editor:



The trixbox administration page

Once you are in the Config File Editor, select `extensions_custom.conf` and paste the following lines in at the end of the file:

```
[gv-inbound]
```

```

exten => s,1,Set(DID_EXTEN=${SIP_HEADER(To):5})
exten => s,n,Set(DID_EXTEN=${CUT(DID_EXTEN,@,1)})
exten => s,n,GotoIf($[${CALLERID(num)} = <gv_number>]:normalcall)
exten => s,n,NoCDR()
exten => s,n,Bridge(${DB_DELETE(gv_dialout/channel)})
exten => s,n(normalcall),Goto(from-trunk,123456,1)
;exten => s,n(normalcall),Goto(from-trunk,${DID_EXTEN},1)

[gv-outbound]
exten => _X.,1,Wait(1)
exten => _X.,n,Set(ACCTNAME=<gv_email>)
exten => _X.,n,Set(ACCTPASS=<gv_password>)
exten => _X.,n,Set(RINGBACK=<sip_ringback_number>)
exten => _X.,n,System(gvoice -e ${ACCTNAME} -p ${ACCTPASS} call ${EXTEN}
${RINGBACK})
;exten => _X.,n,Set(DB(gv_dialout/channel)=${CHANNEL})
;exten => _X.,n,Wait(10)
;exten => _X.,n,Hangup
exten => _X.,n,Set(DB(gv_dialout/channel)=${CHANNEL})
exten => _X.,n,Set(GLOBAL(BRIDGE)=YES)
exten => _X.,n,Wait(10) ; Changed this from 5 to 10 to allow more time
while waiting for incoming gv call
exten => _X.,n,Set(GLOBAL(BRIDGE)=NO) ; RESET global variable BRIDGE in
the event of timeout
exten => _X.,n,Hangup
exten => _X.,h,Set(GLOBAL(BRIDGE)=NO) ; RESET global variable BRIDGE in
the event of hangup

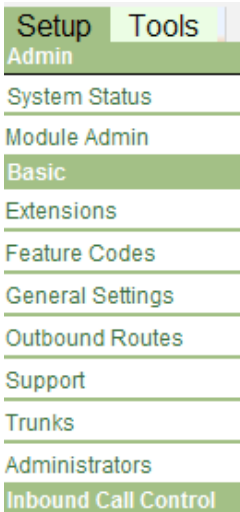
```

After you have pasted this, you will need to modify a few things in the file as indicated by the emphasized placeholders.

Placeholder	Description
gv_number	Your 10-digit Google Voice number.
gv_email	The e-mail address that you used to register your Google Voice account.
gv_password	Your Google Voice password.
sip_ringback_number	Your DID number from IP Communications.

The rest of our configuration will be in the PBX Settings. These can be accessed by hovering over the PBX entry in the top navigation bar and clicking on “PBX Settings.”

Before we continue setting up the rest of the system, we will create a virtual extension that will receive incoming calls. This can be connected to by an IP phone, SIP client software, or an Analog Telephone Adapter in the future.



The left menu

From the left side of the Trixbox admin page, click the “Extensions” option. When asked which type of device you want, select a Generic SIP Device. The only required fields on this page are “User Extension,” “Secret,” and “Display Name.” Fill in “1234” for the extension, and your name for the display name. Finally, click submit at the bottom of the page.

On the left side of the Trixbox administration page, you will see a number of options. The one we are interested in for this step is the “Trunks” option which can be found under the “Basic” heading. From this trunk administration page, you must click the “Add Trunk” button and select “Add Custom Trunk.” On the following screen, enter the following:

1. In the “Dial Rules” text box: 864+NXXXXXX (replacing 864 with the area code you want the system to assume for 7-digit dialing)
2. In the “Custom Dial String” field: local/\$OUTNUM\$@gv-outbound

After you do this, you may click “Submit Changes” to save your work. Next, we are going to add an outbound route. You create outbound routes in similar manner to the way we created the custom trunk. Select “Outbound Routes” from the “Basic” heading of the Trixbox admin interface, click “Add Route,” and enter the following information on the form:

1. Route Name: Outbound
2. Dial Patterns (replacing 864 with your preferred area code):
1864|NXXXXXX
864|NXXXXXX
NXXXXXX
NXXNXXXXXX
011. (this one is only necessary if you wish to make international calls from this phone)
3. Trunk Sequence: local/\$OUTNUM\$@gv-outbound

Click “Submit Changes” when you are finished. Back to the left hand navigation bar, we will make a visit to the “Tools” tab (at the top of the menu) and select “Custom Destinations.” Enter the following information into the form:

1. Custom Destination: gv-inbound,s,1
2. Description: GV-Inbound

Once you have entered this information, click “Submit Changes.” In the navigation menu on the left side, click back to the “Setup” tab. Next, we are going to create an inbound route by selecting the appropriate option from the menu (it is under the “Inbound Call Control” heading). Enter the following information into the form:

1. Description: Inbound
2. DID Number: The number you received when you registered with IP Communications
3. Under the “Set Destination” heading at the bottom of the page: Custom Destinations: GV-Inbound

Again, click the button that says “Submit” at the bottom of the page. Click “Add Incoming Route” (upper right-hand corner of the page) to add another route with the following information:

1. Description: Non-GV-Outbound-Calls
2. DID Number: 123456
3. Under the “Set Destination” heading at the bottom of the page: Extensions: Your Name <1234>

At this point, you can click the “Submit” button for the last time and then click the orange “Apply Configuration Changes Changes” button at the top of the page and click the continue button.



Congratulations! You should now have a working Trixbox setup! The only thing left to do is hook up a phone so you can use your new phone service. There are three options for making calls with your new system described below. You will need to consult the documentation for the product you choose to find out how to connect it to Trixbox/Asterisk. This will be done via the SIP protocol.

Option	Description	Price
Softphone	A program that runs on your computer. I recommend X-Lite (cross-platform) or Telephone (Mac Only).	Most are free
IP Phone	A physical phone that connects to Trixbox via network (wifi or ethernet).	\$100+
ATA (Analog Telephone Adapter)	A device that acts as a bridge between Trixbox (connecting via ethernet) and a standard analog phone. This allows you to keep your existing phone system.	About \$50