

# HOWTO build and configure chan\_mobile on Trixbox

Contributed by Kevan

Installing and configuring chan\_mobile allows bluetooth mobile/cellphones and headsets to work as trunks and extensions on your Trixbox.

This howto contains my work and the work of others from the Trixbox forums.

Here are the steps for installing chan\_mobile with Trixbox 2.3 Beta:

1. Let's add a few components:

```
[trixbox1.local ~]# yum -y install bluez-utils bluez-libs bluez-libs-devel bluez-hcidump automake autoconf subversion gcc gcc-c++
```

2. Using the Package manager install the ncurses libraries, while you are at it you should also install WebMin:

3. Type the following commands:

```
[trixbox1.local ~]# cd /usr/src
[trixbox1.local ~]# wget http://downloads.digium.com/pub/asterisk/old-releases/asterisk-1.4.13.tar.gz
[trixbox1.local ~]# tar xvfz asterisk-1.4.13.tar.gz
[trixbox1.local ~]# cd asterisk-1.4.13
[trixbox1.local ~]# ./configure
[trixbox1.local ~]# make
```

4. Type the following commands:

```
[trixbox1.local ~]# svn co http://svn.digium.com/svn/asterisk-addons/trunk asterisk-addons-trunk
[trixbox1.local ~]# cd asterisk-addons-trunk
[trixbox1.local ~]# svn update -r 421
[trixbox1.local ~]# cat chan_mobile.c | sed -e 's/ast_debug(1,/ast_log(LOG_DEBUG,/' | sed -e 's/, config_flags);/);/' >
chan_mobile_1.4.c
[trixbox1.local ~]# mv chan_mobile_1.4.c chan_mobile.c
[trixbox1.local ~]# ./configure --with-asterisk=/usr/src/asterisk-1.4.13
[trixbox1.local ~]# make clean
[trixbox1.local ~]# make menuselect
```

Press 1, Select chan\_mobile (deselect other stuff) and then Type "x" to exit the menu

NOTE: I just left the default selections... and hit an "x" without deselecting items...

```
[trixbox1.local ~]# make
[trixbox1.local ~]# make install
[trixbox1.local ~]# cp /usr/src/asterisk-addons-trunk/chan_mobile.so /usr/lib/asterisk/modules
```

5. Load chan\_mobile.so  
(asterisk -r (and then...) module load chan\_mobile.so)

6. cp /usr/src/asterisk-addons-trunk/configs/mobile.conf.sample /etc/asterisk/mobile.conf

7. nano /etc/bluetooth/hcid.conf and replace the contents with:

```
options {
  autoinit yes;
  security auto;
  pairing multi;
  passkey "1234";
}# Default settings for HCI devices
device {
  name "Asterisk PBX";
  class 0x000100;
  iscan enable; pscan enable;
  lm accept;
  lp rswitch,hold,sniff,park;
}
```

## 8. /etc/init.d/bluetooth restart

You'll probably be told that your system couldn't stop bluetooth (because it wasn't running) and then it'll restart.

## 9. Now let's make sure everything is running that should be:

```
/etc/init.d/bluetooth status
```

You should see messages that look like this:

```
hcid (pid somenumber) is running ...
sdpd (pid somenumber) is running ...
```

If you're alerted that some other application isn't running, we don't care.

## 10. Now let's be sure the system has found your Bluetooth adapter:

```
hcitool dev
```

You should see something like this:

Devices:

```
hci0 00:16:38:39:44:88
```

## 11. Copy the hardware address from the output of the following command for the next step. (eg. xx:xx:xx:xx:xx:xx). Now edit /etc/asterisk/mobile.conf:

```
nano /etc/asterisk/mobile.conf
```

In the first [adapter] stanza where `&id=blue`; replace the mac address with the one you copied down.

## 12. Configuring Linux Bluetooth Software to Start Automatically. You don't want to have to manually start up your Linux Bluetooth software each time you reboot your server. The easiest way to automatically start it is to use WebMin which comes preinstalled with our TrixBox builds. While logged in as root, make sure WebMin is running:

```
/etc/webmin/start
```

Then point your web browser to port 10000 on your Asterisk server:

```
https://192.168.0.108:10000 (use your IP address)
```

When prompted for username and password, type root and the password you assigned the root user on your system. From the Main Screen of WebMin, click the System button and then the Bootup and Shutdown link. Find bluetooth in the list of applications and click on it. The Action Details screen should show that bluetooth is running. Now click the Yes button beside "Start at Boot Time?" and then click the Save button to reconfigure your server. That wasn't hard, was it? Go back to the command prompt on your Asterisk server and stop WebMin by issuing the following command:

```
[trixbox1.local ~]# /etc/webmin/stop
```

No need to waste processing cycles for a tool we're not using.

## 13. Now let's make your trixbox discoverable:

```
[trixbox1.local ~]# dbus-send --system --type=method_call --print-reply --dest=org.bluez /org/bluez/hci0 org.bluez.Adapter.SetMode string:discoverable
```

## 14. Grab your cell phone, turn Bluetooth on and search for devices. You should find your PBX as Asterisk PBX. Pair with a pin of 1234.

## 15. Make your phone discoverable via Bluetooth.

16. Let's make sure we can see your phone or headset. Use the command:

```
[trixbox1.local ~]# hcitool scan
```

You should see something like this:

```
[trixbox1.local ~]# hcitool scan
Scanning ...
00:1B:63:E4:2F:05 Dan's iPhone
[trixbox1.local ~]#
```

17. If you see your phone above all is well and we can now search for your bluetooth devices using the CLI command 'mobile search'. Be patient with this command as it will take 8 - 10 seconds to do the discovery.

```
[trixbox1.local ~]# asterisk &ndash;r
trixbox1*CLI> mobile search
```

18. This will return something like the following:

```
Address Name Usable Type Port
00:1B:63:E4:2F:05 Dan's iPhone Yes Phone 8
00:1C:CC:1C:E1:0C BlackBerry 8320 Yes Phone 3
00:0B:9E:11:74:A5 Hello II Plus Yes Headset 1
```

19. This is a list of all bluetooth devices seen and whether or not they are usable with chan\_mobile.

- The Address field contains the 'bd address' of the device. This is like an ethernet mac address.
- The Name field is whatever is configured into the device as its name.
- The Usable field tells you whether or not the device supports the Bluetooth Handsfree Profile or Headset profile.
- The Type field tells you whether the device is usable as a Phone line (FXO) or a headset (FXS).
- The Port field is the number to put in the configuration file.

20. Choose which device(s) you want to use and edit /etc/asterisk/mobile.conf

```
[trixbox1.local ~]# nano /etc/asterisk/mobile.conf
```

For a phone create a stanza like the one below and paste in the MAC address on the address line and also make sure you use the port that was displayed during the mobile search. The 'id' of the device [bitinbrackets] can be anything you like, just make the unique.

```
[my_phone]
address=xx:xx:xx:xx:xx:xx ; the address of the phone
port=x ; the port number (from mobile search)
context=from-mobile ; dialplan context for incoming calls
adapter=blue ; adapter to use
```

Now let's configure your headset. If your are configuring a headset be sure to include the type=headset line:

```
[my_headset]
address=xx:xx:xx:xx:xx:xx ; the address of the headset
port=x ; the port number (from mobile search)
type=headset ; tells asterisk it is a headset
adapter=blue ; adapter to use
```

21. Having done this, unload chan\_mobile.so and load it again

(asterisk -r (and then...) module unload chan\_mobile.so (and then...) module load chan\_mobile.so  
If asterisk bombs out after load/unload you need to go back in with asterisk -r to execute the above  
You should see something like this...

```
trixbox1*CLI> module load chan_mobile.so
```

```
== Parsing '/etc/asterisk/mobile.conf': Found
== Registered application 'MobileStatus'
== Registered application 'MobileSendSMS'
== Registered channel type 'Mobile' (Bluetooth Mobile Device Channel Driver)
```

22. The CLI command 'mobile show devices' can be used at any time to show the status of configured devices, and whether or not the device is capable of sending / receiving SMS via bluetooth:

```
trixbox1*CLI> mobile show devices
ID Address Connected State SMS
my_phone 00:12:56:90:6E:00 Yes Free Yes
my_headset 00:0B:9E:11:74:A5 Yes Free No
```

```
trixbox1*CLI>
```

23. All being well Asterisk will now try and establish a connection to each configured device.

If it can't it will retry after 'interval' seconds, infinitely. This means that as your cell phone comes into range and goes out of range, Asterisk will automatically connect and disconnect from it. You don't need to worry about it.

24. As each phone is connected you will see a message on the Asterisk console:

```
trixbox1*CLI>
Loaded chan_mobile.so => (Bluetooth Mobile Device Channel Driver)
-- Bluetooth Device my_phone has connected.
-- Bluetooth Device my_headset has connected.
```

25. If someone calls your cell phone now, Asterisk will handle the call and it will be sent to the context you specified, or the default context. Hopefully this means some SIP phone somewhere will ring, pick it up and take the call.

26. Important: Watch what your cell phone is doing the first few times. Asterisk won't make random calls but if chan\_mobile fails to hang up for some reason and you get a huge bill from your telco, don't blame me.