How Ubuntu’s broken bluetooth support came to be

by Benny Bottema - Sunday, August 08, 2010

http://www.bennybottema.com/2010/08/08/how-ubuntus-broken-bluetooth-support-came-to-be/

Checking in on 2016: yep, still the same.

Pairing up your Bluetooth device was relatively well documented, by the community anyway. It was mostly done using the bluez package from bluez.com. Ubuntu town and in fact all of Linux land was generally at peace and the trendy folk at the BlueTooth Bar were being all trendy-like.

Then it happened. Ubuntu went up one release: **Jaunty came to be and subsequently Bluetooth support died**. Even now the Internet is teeming with Bluetooth bug reports, problems and forums are filled with questions such as ‘my keyboard is not being paired’. In fact, Debian seemed to be hit as well as some other distributions that leveled up.

What had happened?

Contents

Bluez changed is what happened

The developers of bluez, once the nice Bluetooth package everyone could live with, decided it was time for a major overhaul: they merged the packages *bluez-libs* and *bluez-utils* and then some. And with ‘and then some’ meaning completely changing the way Bluetooth devices are configured and thereby breaking backwards compatibility.

All of this shouldn’t really matter, except for one little detail: **The Ubuntu dev team thought it was a**

good idea to drop the old Bluetooth support and go for the new version. They included bluez 4.0 and everybody broke in tears. That’s all there is to this little story, Bluetooth has been neglected since 2008 when Jaunty came about and **things have not improved**.

Why is that?

Bluez is a bunch of cowboy coders is why

Actually, I know nothing of the team except that it seemingly has about one active developer. Bluez arguably is an awesome team: they have these coders, stomping away on probably wireless keyboards to create utterly thorough Bluetooth support for Linux. They’re awesome right? Perhaps not.

Bluez doesn’t document anything. At all. It’s a widely acknowledged fact (Gentoo too?) that they didn’t document the legacy Bluetooth support and they didn’t document the new support. If you read the bluez 4.0 release notes you still don’t know anything and you won’t either. Nobody will, because there is no documentation! Oh, they’ll completely overhaul Bluetooth support but they’ll never tell how it works now. That’s for you to find out. I’ve even glanced over the source code only to find out that they don’t like to comment their code either.

> Although normally I couldn’t care less about some random piece of software, but if you include a significant framework in your operating system, there has got to be *some* kind of set of standards that includes documentation, right?
> Definition of a Cowboy Coder

That’s why everyone is still dealing with the same problems it did in 2008: **the old** community documentation **is still there, but it isn’t getting updated**. That page is just one of hundreds ofcourse. Here’s the typical research process one goes through figuring out how to get Bluetooth working (notice how he’s constantly confronted with conflicting guides and ultimately didn’t find a solution). The problem again is that nobody *can* update those pages. If you are feeling particularly masochistic, try reading this thread (especially the last post where it says ‘SOLVED’).

Let’s discuss what must be one of the most encountered problem: ‘hcid.conf is missing!’. Old documentation will tell you how to configure your Bluetooth device in Ubuntu so that it connects when your pc boots. That is to say, how you need to edit a */etc/bluetooth/hcid.conf* configuration file. Well, that file was removed by the bluez team in bluez 4.0. Reread the bluez 4.0 release notes:

> With this new major version a lot of things have been changed:
> – The main daemon is now called bluetoothd (instead of hcid)
> – The main configuration file is /etc/bluetooth/main.conf and follows INI-style syntax

And indeed, scanning the source files reveal that the hcid.conf is no longer used, but other configuration files instead. It appears to me that hcid.conf has become main.conf, except that scanning the code doesn’t confirm this. Instead, main.conf and input.conf are only used to read the options already provided. Changing the names of configuration files ofcourse isn’t the only thing that changed, since just using main.conf or whatever doesn’t seem to magically do the trick. I still can’t figure out how to enumerate devices in the new configuration files, it’s not like you can just say, *‘Well, instead of hcid.conf just use blah.conf’*. I did get a long way configuring my keyboard in rfcomm.conf, but since my keyboard doesn’t seem to support SMP (Session Multiplex Protocol, I got errors reporting this to me), I assume it is used mainly for pairing phones.

Even now, everyone still uses the deprecated but still supported hidd, primitive but effective bluetooth adapter tool. You can install it using the command *sudo apt-get install bluez-compat* and then use it the way I used it to connect my keyboard. You just can’t use it to reconnect devices with, from reboot, wake up, or the device coming out of suspension. The hcid.conf was used for this, but now it is dead.

About hidd being deprecated anyway

If hidd has become deprecated, I wonder what replaced it. Searching the internet revealed *nothing*. Until I found this deeply hidden gem. In this Debian bug report, this guy ‘Phil’ mentions how a guy on the bluez mailing list mentioned that a wiki mentioned something about hidd being replaced with a more generic so called ‘”input service” over dbus‘ approach? Woh, have I just found a vague piece of information about how things changed? Let’s see this Wiki! Oh wait! IT DOESN’T EXIST! Sigh.

> I had set up my bluetooth keyboard based on the instructions in /usr/share/doc/bluez-utils/README.Debian, i.e. starting hidd –server by enabling it in /etc/default/bluetooth, running hcitool scan, and then hidd –conect . This method is now deprecated; instead we should use something called the “input service” over dbus. See the following page, which contains examples of dbus-send commands to do this: http://wiki.bluez.org/wiki/HOWTO/InputDevices

Ofcourse I just *had* to know what was on that Wiki, so I turned to the ever so awesome Wayback Machine. Here’s what I found. Read it yourself. Hint: you won’t get anywhere, but here’s what I get from it. It seems the new approach is a more generic ultra obfuscated command that accepts an instruction class or identifier such as ‘org.bluez.input.Manager.RemoveDevice’. No manual and certainly nothing about configuring your connections for auto-connect during boot.

I don’t know why they took down the wiki, but from the looks of it, most of it was about the API anyway, useful for developers, not end-users using the command-line.

Also, there’s an interesting discussion on bluez 4.0 in Arch Linux I found, the first one I’ve encountered where the participants actually try to get the new ‘dbus’ way working. Nothing we can use in Ubuntu though. I’ve tried to get bluez talking to me using dbus, but I couldn’t get it to do anything but spout a range of errors.

Finally, a guide I found on using Bluetooth with bluez 3 and 4 not related to a particular Linux distribution. Unfortunately, the dbus keeps barfing up on everything I throw at it, so no dice for Ubuntu I guess.

Current state of Ubuntu’s Bluetooth support

Technically the title of this blog post is wrong: **Ubuntu’s Bluetooth support is probably better now than it ever was, it’s just that nobody knows how to use it**. To be fair, it’s not an entirely lost cause; people still get their devices to work properly after many people have tried to fix some variation of a given problem and people figured out how to connect a mouse or a phone using the new Bluetooth support for example, often with obscure workarounds. But a lot of people, including me, still can’t get a Bluetooth keyboard to connect and pair during boot.

Also there is a solution if you happen to use some desktop, using the Bluetooth Applet. It handles everything for you and reportedly it generally works. It’s no use to me however as I’m only running a server, but I might check out the applet’s source code to see how it works, hopefully using bluez’ command line API.

As of yet, there has been no answer from the dev team on the number one search result for ‘missing hcid.conf’ bug report (2009): Bug 365779. Technically it’s not a bug, but the fact that ~~it hasn’t been closed, deferred to another bug or had a dev on it since May 2009~~ it expired after 6 years of neglection indicates it either has no priority or nobody had the courage to fix it. Notice though there are literally hundreds of bluez related bug reports for numrous distro’s.

Also a search on the original writers of the last hcid.conf man page didn’t reveal a comment on how things changed and how things now work.

So what needs to be done to get some clarity on the matter? If you know, please shout out, because yes, I’m currently still looking to fix my keyboard under Ubuntu 10.04.

Meanwhile, try downgrading the bluez package to its previous version (note: I couldn’t get the bluetooth package to work with the older bluez packages in Ubuntu 10.04, good luck and have fun).

PDF generated by Kalin's PDF Creation Station