BTinternet pppd mini−HOWTO

Matt Wright

Matt Wright Consulting

matt@consultmatt.co.uk

Greg Ferguson – Converted the mini−HOWTO from HTML to Docbook 3.1 (SGML).

2002−03−26

Revision History

Revision v0.29	2002−03−26	Revised by: mww
Added a small bit of first−hand tech support about cross−frequency.		
Revision v0.28	2002−01−17	Revised by: mww
New information about PPPoATM involving kernels and distibutions.		
Revision v0.27	2001−12−20	Revised by: mww
Minor technical problems highlighted by Robert Smith.		
Revision v0.26	2001−11−21	Revised by: mww
Added a point about the Kernel HOWTO.		
Revision v0.25	2001−11−17	Revised by: mww
Added a troubleshooting answer about "_mmx_memcpy". Other minor updates as well.		
Revision v0.24	2001−11−09	Revised by: mww
Technical detail with the chatscript timeout (and found a spelling mistake or two!). Thanks again to Bill Staehle.		
Revision v0.23	2001−11−07	Revised by: mww
Changed the Chatscript dialing method, thanks go to TonyC from btinternet.linux newsgroup.		
Revision v0.22	2001−11−06	Revised by: mww
Changed a couple more little botches. Thanks again go to Bill Staehle.		
Revision v0.21	2001−11−03	Revised by: mww
Changed discrepancies reported by Bill Staehle.		
Revision v0.20	2001−11−01	Revised by: mww
Added Alcatel Speedtouch Information.		
Revision v0.19	2001−10−31	Revised by: mww
Initial public release.		

This document describes how to setup a modem pppd link to Btinternet in the UK.

Table of Contents

1. Introduction

This HOWTO exists because a mate of mine needed to easily set up an Internet connection to BTinternet and here is a quick and concise way to get a running PPPd to Btinternet. This HOWTO also briefly covers Howto setup basic IP Masquerading to allow connection sharing.

1.1. Copyright and License

Send feedback to *matt@consultmatt.co.uk*.

1.2. Mailing Lists

Please note: There is a mailing list available for this howto. If you wish to join please email `<majordomo@consultmatt.co.uk>` with the body:

```
subscribe bti-howto
```

This will then be followed up by the mailing list bot and subsequent emails should arrive confirming your request. The list is intended to be for annoucing changes to this HOWTO and also to dicuss the finer points of getting pppd to work on BTinternet.

1.3. About the author

My name is Matt Wright. I'm 16 year−old student in Blackburn, Lancashire. I'm a freelance Linux consultant. I am the proud owner of a Duron 950Mhz (all I could easily afford) with 256MB SDRAM, Voodoo 4 Video Card, ATI All−in−Wonder Pro Video Card. I also have a 266Mhz Cyrix that runs my USB ADSL connection, of which if you are reading this from http://www.consultmatt.co.uk you will be using.

You can find me at www.consultmatt.co.uk. Or at matt@consultmatt.co.uk.

1.4. Acknowledgements

A major thanks of gratitude to the guys at Linuxdoc in particular Greg Ferguson and David Merrill for help with getting this into trusty SGML. Also to Rob Smith the guy behind me writing this :) (As he couldn't sort his own Linux box out :P). A great vote of thanks to Bill Staehle for his input in ironing out some major flaws. Also a thank you goes to TonyC at btinternet.linux and the rest of that newsgroup for originally getting me on my feet with BTi. Finally and most importantly my thanks goto my Lord and Saviour Jesus Christ, without whom I'd be lost.

2. Requirements

The following is needed to use this HOWTO:

- A working modem, external or internal hardware modems are the easiest to get working, see Modem−HOWTO. If you have a WinModem then see Winmodems and Linux HOWTO
- An account with Btinternet, this HOWTO covers BTi Surftime and Anytime
- Some basic understanding of the Linux OS and its layout

---

3. Surftime

3.1. PPPd Setup

Right, to start off you will need a working installation of PPPd, I know from experience that Mandrake 8.0 that sometime is doesn't install pppd so check its installed first. Once it is it usually resides in `/usr/sbin/pppd` check that any users you want to access it have setuid access.

Note: Setuid allows pppd to run as−if root allowing non−superusers to run pppd without less problems. To do this use the command **chmod 4750 /usr/sbin/pppd**. Beware: Some distributions have a watchdog program that will change the pppd permissions back to normal. I have not tested this but on RedHat derived systems with Linuxconf installed then removing the `/usr/lib/linuxconf/redhat/perm/ppp` should stop this happening.

3.2. Chatscript (Dialup)

The Chatscript is a text file, usually residing in `/etc/ppp` that contains the commands passed to the modem to make it dial to BTi. If you want to compare this to DUN in Windows then this is the phone number and init commands.

Below is the chatscript used to dial BTi, I've now opted for putting the \T metacharacter in the script. The means when we run **chat**, inside the pppd command, we can use a `−T` parameter and supply the needed telephone number.

```
"" "ATZ"

# The next two lines should be left commented out until
# the script works.

# "OK" "ATL0"
# "OK" "ATM0"

SAY "Dialing modem...\n"
"OK" "ATDT \T"
ABORT BUSY
ABORT "NO CARRIER"
TIMEOUT 60
CONNECT \c
```

This script will dial BTi providing you put in the \T so we can added the telephone number later. I suggest saving the chatscript as something like `/etc/ppp/chatscript` Once you've saved it we can move onto setting up BTi's CHAP authentication.

3.3. Authentication

With PPP dialup's the most widely used authentication method, apparently, is PAP (Password Authentication Protocol). However, just to be a pain in the backside BTi use CHAP (Challenge Handshake Authentication Protocol). To be fair CHAP is a more secure authentication method than PAP but it's still a pain. Now PPPd does support this but not through nice easy to use Linuxconf dialogs.

In the `/etc/ppp` directory should be a `chap−secrets` that looks roughly like this when first installed:

```
# Secrets for authentication using CHAP
# client    server    secret    IP addresses
```

We can safely ignore the IP Address column but the others me must worry about. So fill in the gaps like this:

```
# Secrets for authentication using CHAP
# client    server    secret    IP addresses
"bloggs@btinternet.com"  *   "mypasswordhere"
```

These two extra lines will try and authenticate ANY outgoing PPP connection that responds using CHAP using your BTi details. It goes without saying that you replace *bloggs@btinternet.com* and *mypasswordhere* with your details. Also, the second line is not strictly needed but I've found it helps sometimes.

Note: If you're not using BTi at this point you can usually get away with having identical `chap−secrets` and `pap−secrets`. If you the same kind of pattern as above it should work.

I suggest that you do a "**chmod 600 /etc/ppp/*secrets**". It was pointed out to me that it stops pppd shouting about security.

3.4. Setting your global options

In your `/etc/ppp` directory there is a file called `options`. If your open it with your favourite text editor you should see:

```
lock
```

And thats it, now we need to add some more settings in here so make it look like:

```
lock
usepeerdns
defaultroute
noipdefault
noauth
asyncmap 0
crtscts
modem
115200
```

Don't worry what they do, if you're interested look at the man page for pppd. As a passing note if you would like PPPd to redial dropped connections then add *persist* to the end of your options file.

3.5. Testing your link

Now you've created your Chatscript there are only two steps left till you should be on the Internet. First is getting a working link and second is creating a couple of easy dialup scripts to help you.

To test your link try this command:

pppd ttyS0 connect '/usr/sbin/chat −v −TPHONE_NUM_HERE −f /etc/ppp/chatscript' updetach debug name bloggs@btinternet.com

Now obviosuly if your testing at Daytiem rate add the Daytiem number where instead of `PHONE_NUM_HERE` and similar with Surftime.

This will tell PPPd to dial the modem on ttyS0 (COM1) using the chatscript. The `updetach` tells PPPd to only fork away to the commandline after the connection is established and the `debug` will show all authentication commands onscreen (providing updetach it set) while it trys to connect.

If that connected then move onto the next section and we can make a set of scripts to allow easy dialing.

3.6. Dialing scripts

Here I'll just set out a set of scripts that will let your dial BTi from a command line easily. The following should be self−explanatory, the italic filename is the filename to put the data into and the proceeding text the file contents:

/etc/ppp/peers/bt−surf

```
ttyS0 connect '/usr/sbin/chat −v −TSURFTIME_NUMBER_HERE −f /etc/ppp/chatscript'
updetach name bloggs@btinternet.com
```

/etc/ppp/peers/bt−day

```
ttyS0 connect '/usr/sbin/chat −v −TDAYTIME_NUMBER HERE −f /etc/ppp/chatscript'
updetach name bloggs@btinternet.com
```

Change any of the filenames to suit what you called them and the username. Do the same with this:

/usr/bin/internet

```
#!/bin/bash
# a script to dial BTi
case "$1" in
daytime)
   /usr/sbin/pppd call bt−day
   ;;
surftime)
   /usr/sbin/pppd call bt−surf
   ;;
off)
   killall pppd
```

```
    ;;
  esac
```

Once you've done that run this command at the command prompt:

chmod a+x /usr/bin/internet

Now you've done, simply type **internet daytime/surftime/off** and it will connect you to the internet.

4. BTi Anytime

I'm sorry, for all you Anytime customers its really hard! NOT! Read all the instructions for Surftime above. They all apply in most places, the only things to watch out for is you obviously don't need any of the Surftime/daytime distinctions. So only have one peers file (using the correct phoen number) and one dialup option in your `/usr/bin/internet` file. (eg. dial and off)

---

5. BTOpenworld Home 500 (Alcatel Speedtouch USB)

5.1. About this section

There is rather a large amount of credit due here, to Chris Jones for writing the Alcatel Speedtouch USB ASDL Modem mini−HOWTO that is now part of the DSL HOWTO. This helped me a great deal when trying to get my Speedtouch to work.

5.2. Warning

At home I use Linux Mandrake, although the version of the kernel I had was patched with the ATM kernel source I did end up patching a different kernel source to get it working. Please, please inspect your kernel source to see if you have the PPPoATM source patched against your kernel. To do this go into you kernel source directory, usually `/usr/src/linux` and do a **make menuconfig**. In the Network Device Support section check for:

- Network device support−>PPP over ATM

If it does exist then make sure it is present it you current kernel and you can skim−read the "Patching you kernel" section to make sure you have the correc toptions compiled in and then carry on.

This was just a minor warning as I orginially had a kernel patched with PPPoATM and without realising managed to trash my kernel tree by trying to force the patch onto it.

5.3. Distribution Specific Information

I have had reports about different distros that have varying PPPoATM support already builtin to the kernel:

- Mandrake 8.0: The default 2.4.3 kernel already has the PPPoATM patch applied, I cannot remember if it compiled in but ignore steps refering the "Patching the Kernel" below. PS: You still need to configure the kernel.
- Mandrake 8.1: Mandrake 8.1 supports the Speedtouch automatically, I have no had first hand experience but from what I can gather you simply have to download the Alcatel binaries and then use DrakNet to sort it out.
- Debian: I have had reports that the standard Debian installtion does not include libpam, this must be installed for the PPPoATM plugin modules to work.

NOTE: From roughly kernel 2.4.16 the PPPoATM patch is included in Linus' main source tree. Therefore you can miss out patching the kernel and just configure it.

5.4. Requirements

To get your Speedtouch USB working in Linux you have a fairly heavyweight task ahead of you, but hey, if I could do it so can you! This is what you'll need to get it working:

- You must have the kernel source installed and know the procedure for installing and compiling a new kernel. If this is a problem then read the Kernel HOWTO.
- You must be running one of the following Kernels: 2.3.39, 2.4.0−test4, 2.4.1−pre7, 2.4.7, 2.4.8−pre5. This is because the PPPoATM patch for the kernel exists patched against specific kernels, some may work with similar kernel versions but I cannot vouch for that
- You, obviously, need a USB controller of some description with at least one free plug. It also must be Linux compatible, nowadays this is most USB controllers that are UHCI/OHCI based. If you don't have one your local supplier would probably have a PCI USB Controller.
- A heap−load of confidence with meddling with your config. eg: kernel recompiling, program installation...

5.5. Software Downloads

To get the Speedtouch working under Linux you will need some software and kernel patches found below:

- The kernel patch for your kernel. They can be found at http://www.kernel.org/pub/linux/kernel/people/axboe/PPPoATM/. Please note not all the kernels have patches.
- The latest SpeedTouch driver from http://sourceforge.net/project/showfiles.php?group_id=3581
- The latest SARlib library from http://sourceforge.net/project/showfiles.php?group_id=22221
- The Alcatel speed management software. You can get it from http://www.alcatel.com/consumer/dsl/dvrreg_lx.htm. I can't distribute this because of Alcatel's licensing scheme so get it from them.
- Some description of PPPoATM aware PPPd binary:
    - ♦ Red Hat 7 RPM (glibc 2.2): http://sourceforge.net/project/showfiles.php?group_id=23818
    - ♦ Debian (.deb): http://sourceforge.net/project/showfiles.php?group_id=23818
    - ♦ Tarball: http://sourceforge.net/project/showfiles.php?group_id=23818
- The Linux Hotplug software from http://linux−hotplug.sourceforge.net. Get it installed as per their instructions. It seemed simple enough so I won't cover it here

---

5.6. Patching your kernel

Once you have the PPPoATM kernel patch (this assumes you use the patch against kernel 2.4.7) you need to make sure you have a working 2.4.7 kernel tree, next unzip the PPPoATM patch by doing:

NOTE: From rougly kernel 2.4.16 (I haven't tested to see hwo far back it goes) the PPPoATM patch is included in Linus' main kernel tree, therefore you may skip the patching below and resume ready to configure the kernel.

gzip −d pppoatm−2.zip

Next we will need to test−patch the kernel using the following commands:

patch −p1 −s −E −−dry−run < /point/to/pppoatm−2

If that ran without failure then patch the kernel by removing the `--dry-run` as such:

patch −p1 −s −E < /point/to/pppoatm−2

That should have patched the kernel good−and−proper so we can go ahead and configure it, make sure the following options are selected along with your personal build options:

- Code maturity levels−>Prompt for development and/or incomplete code/drivers
- Networking options−>Asynchronous Transfer Mode (ATM)
- Network device support−>PPP (point−to−point protocol) support
- Network device support−>PPP support for async serial ports
- Network device support−>PPP Deflate compression
- Network device support−>PPP BSD−Compress compression
- Network device support−>PPP over ATM
- USB support−>Support for USB
- USB support−>Preliminary USB device filesystem

  You have to make a choice here, if your USB controller is UHCI based then select:

  USB support−>UHCI (Intel, PIIX4, VIA, ...) support

  Alternatively choose:

  USB Support−>OHCI (Compaq, iMacs, OPTi, SiS, ALi, ...) support

You could select any of these as modules or compiled−in but as I followed Chris Jones' HOWTO I compiled the all but UHCI/OHCI as compiled−in code. Save the kernel config and compile the kernel and modules as you normally do.

5.7. Kernel Drivers and Software

Now that the kernel will support using PPPoATM we can start compiling the bits to run the modem. Well start with the Kernel mode driver; first decompress the SARlib sources to a build directory. (Personally I build all my non−kernel sources in `~/sources`) and do a **make** on it. There is no need to do a **make install** with this library.

Next return to your source root and decompress your Speedtouch drivers (from Sourceforge not Alcatel!), go in there and do a **make**, and then a **make install**.

Note: If you get an "Error 1" then check the `Makefile` for a line starting `SARLIB:=` and check it points to the right directory, the one where you just compiled SARlib.

Next install the Hotplug software and make sure it works. Once you've done that decompress Alcatel's binary management software and do a **make** on that. Then do a **make install**, the clever bit here is their installation registers the Speedtouch kernel driver and their binary to be run when the USB device is "hotplugged" (or coldplugged) into the system. *Kiss goodbye to the hours of trying to writing modules loading scripts that always fail.*

Next install the new PPPoATM aware PPPd program, I had no luck getting it compile from source on my machine so I used the RPM. Sorry you're on your own there!

5.8. PPPd Configuration

Warning: The action will remove all the default settings for any previous PPPd connection. (Not that you want them now you've got shiny new ADSL ;P)

In the `/etc/ppp` directory there is a file called `options` in that file put the following:

```
lock
defaultroute
noipdefault
noauth
passive
asyncmap 0
name bloggs@hg5.btinternet.com
user bloggs@hg5.btinternet.com
plugin /usr/lib/pppd/plugins/pppoatm.so
0.38
```

Make sure of the following things:

- That you replace both name ***and*** user variables with your username.
- That you get the correct VPI/VCI ATM pair, these are in the windows information software, from BTi my VPI:0 and VCI:38 so I use `0.38`. Make sure you get this right or it ***will not work***.

Next in your `/etc/ppp/chap−secrets` put:

```
# Secrets for authentication using CHAP
# client    server    secret    IP addresses
"bloggs@hg5.btinternet.com"  *   "mypasswordhere"
```

5.9. Testing your link

Next thing to do is get the link up and test it. If you've got a nice distro like Mandrake you should find it will auto−init your USB drivers and filesystem, if not then the following can be used:

/sbin/modprobe usb−uhci

mount none /proc/bus/usb −tusbdevfs

Next load the Speedtouch driver by doing: **/sbin/modprobe speedtch**. Next use the speedmgmt program to get the modem init'ed: **/usr/sbin/mgmt**. After a while of the lights flashing on the modem the main console (and/or the syslog) should report:

```
Speedmgmt[2074]: Modem initialised at 576 kbit/s downstream and 288 kbit/s upstream
```

Once the modem has been init'ed now make sure the PPPoATM module is loaded by doing:

modprobe pppoatm

Now start the PPP link by typing **pppd** You should see something similar to this:

```
Oct 28 14:01:25 ds9 pppd: PPPoATM plugin_init
Oct 28 14:01:25 ds9 pppd: PPPoATM setdevname_pppoatm
Oct 28 14:01:25 ds9 pppd: PPPoATM setdevname_pppoatm - SUCCESS
Oct 28 14:01:26 ds9 pppd: Using interface ppp0
Oct 28 14:01:26 ds9 pppd: Connect: ppp0 <--> 0.38
Oct 28 14:01:28 ds9 pppd: local IP address 255.255.255.255
Oct 28 14:01:28 ds9 pppd: remote IP address 255.255.255.255
Oct 28 14:01:28 ds9 pppd: primary DNS address 213.120.62.100
```

Once that's done you're in luck, now just configure the pppd to autodial at startup (beyond the scope of this HOWTO, sorry!). Hopefully HotPlug will get your device up and going before pppd needs it! :).

6. Simple IP Masquerading

To allow sharing of your internet conenction you can eitehr use a proxy server or IP Masquerading. I'll cover IP Masquerding as it's simple to set up on the other client machines. I use a 2.4.x generation kernel and in effect I use IPTables. If you use a 2.2.x or 2.0.x kernel then you need the IP Masquerading HOWTO.

This part of the HOWTO assumes that your Netfilter software is modularised, if it isnt then no big deal, either ignore the `modprobe` lines or recompile your kernel.

Simple now, just run the commands:

modprobe iptable_nat

iptables –t nat –F POSTROUTING

iptables –t nat –A POSTROUTING –o ppp0 –s 10.0.0.0/16 –j MASQUERADE

echo 1 > /proc/sys/net/ipv4/ip_forward

Note: The space between 1 and > is vital. It seems not to activate the IP Forwarding if the space is not there.

Change the `ppp0` and/or the `10.0.0.0/16` for your relevant network settings and put that file either before the `case "$1" in` in the `internet` file or somewhere in your startup.

7. Troubleshooting

7.1. Help! Lynx gives unable to access document

Problem: Lynx is unable to access document or ping gives "No route to host" but the link is stable!

Solution: Have a look at the output of **route –n** you should see a refenence specifing a gateway of either your Remote PPP IP and using the adapter/dev of ppp0. If not here is how to fix it. In your `/etc/ppp/ip-up` file above `exit 0` on the last line type:

```
route del default
route add default gw $4 $1
```

Save the file and redial. The problem should be solved.

7.2. My kernel < 2.4.2 crashes using the Alcatel driver

Problem: Your kernel below 2.4.2 crashes using the Alcatel driver.

Solution: You're probably using a kernel compiled with SMP support either don't use SMP or try here: http://sourceforge.net/project/showfiles.php?group_id=23818

7.3. PPPd crashes when it loads the PPPoATM module

Problem: When PPPd attempts to access the ATM device it crashes.

Solution: The kernel was probably compiled on a different type of processor. I tried to get the kernel compiled quicker by compiling the kernel on my 950Mhz Duron, even with the processor type to suit the target 266Mhz Cyrix the ATM module crashes. So dont take short cuts and compile on the target processor.

7.4. insmod speedtch.o gives: "Unresolved symbol: _mmx_memcpy"

Problem: When you have compiled the Speedtouch Kernel driver (speedtch.o) trying to insmod/modprobe returns: "Unresolved symbol: _mmx_memcpy" or similar.

Solution: Right, I haven't got a concrete answer here but I couldn't find it anywhere on the internet so here goes. I found this error come up when I recompiled my kernel once. It did it the first time but I forgot how to fix it. A combination of things fixed it for me:

- Configured the kernel module support to ignore kernel version checking
- Recompiled SARlib (this is what ultimately fixed it) before recompiling speedtch.o
- Recompiled speedtch.o and reinstalled. Et voila!

As I said I can't be sure about this one but it worked for me.

7.5. There are strange noises on the telephone line when I pick up the phone.

Problem: When you pick up the phone there are strange noises present, more over the use of the phone line causes the internet to disconnect.

Solution: You seem to have something which I thing is called cross−frequency. The ADSL modem works by sending data down the unused frequencies of you copper phone line. If the hardware is set up so the ADSL and phone ranges overlap the two can receive each other's data. Ring BTO Customer Support and they should be able to help.

8. Further Reading

For further reading on related topics see this list:

* The PPP−HOWTO
* The IP Masquerading HOWTO
* The DSL HOWTO
* The Modem−HOWTO.
* The Winmodems and Linux HOWTO