

Installing (X)Net on Linux and Raspberry Pi Distros 12-9-2022

This is a “quickstart” walk-through for installing (X)Net v1.39 (aka "linuxnet") on Linux Mint and the Raspberry Pi, configured to use one or more Direwolf sound card TNCs. It will likely also work on Ubuntu and other Linux distros. The Raspberry Pi installation has a couple different procedures, noted in the text. It’s a consolidation using docs and info provided by Brian N2KGC, Lee K5DAT, Brian N1URO (sk), online sources, and the English version of the (X)Net manual linked to in this document.

It installs the minimum needed to have a functional (X)Net node connected to neighboring (X)Net, Flexnet and NetRom nodes via RF. It also includes configuration to link to a BPQ NetRom node if you to have BPQ running on the same machine as (X)Net. Internet node-linking info will be added to this doc at a later date.

There may be redundant or unnecessary configuration settings, but this is what worked for me. Details like auto-starting (X)Net, keep-alive crontab scripts, etc. will be added to this doc later. Feel free to email corrections, suggestions, or additions to kp3ft@yahoo.com .

For both Linux and Raspberry:

1. First download the [linuxnet.zip](https://www.qsl.net/s/swlkp3//Packet/(X)Net/linuxnet.zip) file into your `/home/USER/` folder or whichever folder of choice (mine is `/home/kp3ft/`) from here:

[https://www.qsl.net/s/swlkp3//Packet/\(X\)Net/linuxnet.zip](https://www.qsl.net/s/swlkp3//Packet/(X)Net/linuxnet.zip)

2. Right-click the file and click “Extract Here”. It unzips as a new folder called "linuxnet" containing several files.

*** For Raspberrys, you also need to download the [xnet_arm7.zip](https://www.qsl.net/swlkp3/Package/(X)Net/) file for Raspberry Pi2, Pi3, and Pi4, or the [xnet_arm6.zip](https://www.qsl.net/swlkp3/Package/(X)Net/) file for Raspberry Pi1 and Zero from here:

[https://www.qsl.net/swlkp3/Package/\(X\)Net/](https://www.qsl.net/swlkp3/Package/(X)Net/)

Unzip them in the same linuxnet folder mentioned above.

3. Now open the linuxnet folder; there are some text files in that folder you’ll need to edit in a text editor. (Note: Some configurations of (X)Net installations use a file called IP.NET, but for this basic setup it’s not used.) These filenames must be kept in uppercase letters. The first two files are for port, routing, and beacon configuration: AUTOEXEC.NET and AUTOBOOT.NET. Any text after a “#” is just a comment line, not part of the config. To disable a line completely, place a “#” at the beginning of the line. The same # applies to the Direwolf config file edits mentioned later in this document.

4. Text-edit the AUTOEXEC.NET file, and change the "my call KP3FT-4", "my tcall KP4DOG", and "my alias CHENGO" to your own config. The "my call" is your actual node call, "my tcall" is your user-call, and "my alias" is your node alias. Next, edit or disable the beacon lines. Only the port number, time, and text need to be changed. In the example below, "1800" is the time interval in seconds, "2" is my UHF radio port 2, and anything after "text" is the actual transmitted text:

```
BEACON A 1800 2 ID text CHENGO:KP4DOG in South New Berlin, NY
```

5. Close and save the file. Now open and edit the AUTOBOOT.NET file. The top three lines are ports 0, 1, and 2. The first line is for linking to a BPQ node running on the same machine as (X)Net, so disable it if you aren't linking to BPQ. (If you are linking to BPQ, at the bottom of this document is the procedure.)

The next two lines are virtual comports for two different instances of Direwolf.

****IMPORTANT**** For Linux, *not* Raspberry:

```
attach sdev0 kiss 1 1 19200 /dev/tty0 #virtual COM1 port for Direwolf 1200bd VHF
attach sdev1 kiss 2 1 19200 /dev/tty1 #virtual COM2 port for Direwolf 1200bd UHF
```

If using only one Direwolf instance, edit the first line and disable the second line.

Only three items need to be changed for different comports, which is the digit right after "sdev", the first digit after "kiss", and the digit after "tty". For example, if you want (X)Net to use a third instance of Direwolf that is configured for com3, the line would be: attach sdev2 kiss 3 1 19200 /dev/tty2.

****IMPORTANT**** For Raspberrys, those lines need to be edited a little more. The difference is subtle and easily missed. The Linux "tty" is changed to "ptyq" for Raspberrys:

```
attach sdev0 kiss 1 1 19200 /dev/ptyq0 #virtual COM1 port for Direwolf 1200bd VHF
attach sdev1 kiss 2 1 19200 /dev/ptyq1 #virtual COM2 port for Direwolf 1200bd UHF
```

Next, below the port attachment lines is the configuration for each port. Disable or edit for your own setup. My UHF port example:

```
#70cm RF backbone port (comment line)
po 2 baud 2400 (port 1 at 2400-baud)*
po 2 acon 1 (auto-connect to Netrom nodes. Use 1 after "acon" to enable, 0 to disable)
po 2 dup 0 (duplex operation. Default 0 is disabled, 1 is enabled)
po 2 win 4 (window size)
po 2 quality 192 (node quality)
```

```
po 2 name UHF backbone (name of port that shows in "port" list)
route fl ad 2 w2rgi-2 (route Flexnet on port 2 to neighboring node W2RGI-2)
route bc ad 2 w2rgi-2 (route Netrom on port 2 to neighboring node W2RGI-2)
```

*According to documentation, the baudrate should be the same as the actual radio baudrate the TNC is using, to avoid overflow. Example: for a VHF user-port on Port 1 with Direwolf configured for 1200-baud, use "po 1 baud 1200". For an HF packet port on Port 3 with Direwolf configured for 300-baud, use "po 3 baud 300".

6. Next, open and edit the C.TXT, D.TXT, and INFO.INF files to your preferences. The C.TXT file is the connect-text users will see when they connect on any port. D.TXT is the disconnect-text. INFO.INF is info about your system that users will see if they send the "info" or "i" command. You can also create port-specific connect-text and disconnect-text files instead. For example, if you only want c-text and d-text on a VHF user-port on Port1, create C1.TXT and D1.TXT files. Port2 would be C2.TXT and D2.TXT, and so on. The HELP.TXT file can probably be left alone, or edit to your preference. Its contents are sent when a connected user sends "h" or "help". Close and save each file when finished editing.

7. Next, edit your Direwolf's "direwolf.conf" file. (If Direwolf isn't installed yet, stop here and install the latest stable version using this walk-through: [https://www.qsl.net/swlqp3/Packet/\(X\)Net/DW_Install.doc](https://www.qsl.net/swlqp3/Packet/(X)Net/DW_Install.doc))

Scroll down to the "KISS 8001" line, and change 8001 to 0 to disable it. (Don't add a "#" to the line instead to disable it. Normally, a "#" disables a line, but in that case for whatever reason, Direwolf will still keep a KISS port open on port 8001).

****IMPORTANT****

For [Linux](#), create a new line called: SERIALKISS /dev/ptyp0 19200

For [Raspberrys](#), create a new line called: SERIALKISS /dev/ttyq0 19200

which will be the virtual com1 that you have (X)Net configured for. More details on configuring Direwolf's direwolf.conf config file are in the aforementioned DW_Install doc. Close and save the file.

If you will be running multiple Direwolf instances with (X)Net, each instance will need their own ".conf" file named whatever you choose. For example, besides the default "direwolf.conf", I have "backbone.conf" for running a second Direwolf instance. In each ".conf" file, a different "ptyp#" (or "ttyq#" for Raspberrys) port number needs to be set for (X)Net, e.g.:

SERIALKISS /dev/ptyp1 19200 for virtual COM2 port

SERIALKISS /dev/ptyp2 19200 for virtual COM3 port , and so on.

***(Don't forget to change Direwolf's AGW port numbers in the other .conf files to something other than the default 8000. (X)Net doesn't use the AGW ports, but there will be a conflict between the Direwolf instances if they are all configured with the same AGW 8000 port. The AGW ports are handy for connecting with a packet terminal client such as UZ7HO's EasyTerm client, to use for testing or just a completely separate TNC/Terminal system with a different callsign, etc.)*

8. Now we need to enable Linux to use the “ptyp” or “ttyq” ports.

***** IMPORTANT***** This next step is for Linux, *not* Raspberry. For Raspberrys, skip down to step #11 [For Raspberry](#) section.

In a Terminal, run:

```
sudo nano /etc/default/grub
```

Change the line from:

```
GRUB_CMDLINE_LINUX=""
```

to:

```
GRUB_CMDLINE_LINUX="pty.legacy_count=10"
```

9. Close and save it, then run:

```
sudo update-grub
```

10. **IMPORTANT:** reboot the computer for the changes to take effect. After reboot, linuxnet and Direwolf are now configured to talk to each other.

11. [For Raspberry](#) :

Raspberry's don't use Grub, so its *ttyq* ports need to be set manually for each Direwolf instance. To do this, you first need to install “socat” on the Raspberry:

```
sudo apt install socat
```

After socat is installed, run:

```
sudo socat -d -d -ly PTY,link=/dev/ttyq0 PTY,link=/dev/ptyq0 &
```

and leave it running.

This enables Direwolf to use the *ttyq* port you configured it for earlier. A second instance of Direwolf configured to use *ttyq1* would need another Terminal entry:

```
sudo socat -d -d -ly PTY,link=/dev/ttyq1 PTY,link=/dev/ptyq1 &
```

and so on for each additional Direwolf instance and pty port.

IMPORTANT: That command needs to be run first before starting Direwolf on the Raspberry.

12. Installation is finished. Now you can run and test (X)Net and Direwolf. Direwolf needs to be started first, with root permission. In a Terminal start Direwolf with:

```
sudo direwolf -qd
```

13. In another Terminal, cd into your linuxnet folder and start (X)Net, also as root:

For Linux, run:

```
sudo ./linuxnet
```

For Raspberry Pi2, Pi3, and Pi4 units, run:

```
sudo ./xnet_arm7
```

For the Raspberry Pi1, Zero, and Pico, run:

```
sudo ./xnet_arm6
```

*** If the `xnet_arm6` or `xnet_arm7` executables don't run, you may have to make them executable by running in Terminal:

```
chmod +x xnet_arm6
```

or

```
chmod +x xnet_arm7
```

depending on which model Raspberry you have.

14. (X)Net should now be running in the Terminal, looking something like this:

```
Terminal - prdogs@prdogs-Latitude-2100: ~/linuxnet
File Edit View Terminal Tabs Help
prdogs@prdogs-Latitude-2100:~/linuxnet$ sudo ./linuxnet

***** HighSpeedTNC3 *****
* (X)NET Router Firmware *
* DAMA/TURBOXHOST Version 1.39 *
* (C) '93 - '07 by Jimmy, DL1GJI *
* 500 channel 8192 KByte RAM *
*****

Modem 0: 115200 Baud
Modem 1: 1200 Baud
Modem 2: 1200 Baud

Node: CHENG0:KP4D0G
```

When you start (X)Net, you'll need to wait around 10 minutes for (X)Net to start sending/receiving (X)Net and Flexnet node broadcasts and link to the neighbor node(s) you configured for. After that, your node and destination lists should be populated (provided the neighbor node(s) are configured on their end to link to you). The axUDP link to BPQ takes a shorter time to establish, around a minute or two.

15. Hit the ESC key. Then at the * prompt, enter:

`c <your my call>` (in my case it's: `c kp4dog`)

You should now have a "=>" prompt. From there you can use it as a normal packet terminal. At this point you can immediately test the connection to your configured neighbor (X)Net or Flexnet node, by sending "`c callsign`", in my case: `c w2rgi-2`. To disconnect from that node, send "`q`" or "`quit`".

If configured to link to a BPQ node on your LAN, wait a couple minutes for (X)Net and BPQ to "see" each other, then you can test the BPQ link by connecting to the BPQ node, in my case : `c kp3ft-7`.

You can also go into SYSOP mode by entering "`sys`" at the * prompt, which allows you to add or make changes such as PACLEN, RETRIES, etc. Detailed documentation on SYSOP and other commands, configuration, etc. can be found in the English translation of the (X)Net manual here:

[https://www.qsl.net/swlkp3/Packet/\(X\)Net/xneten.doc](https://www.qsl.net/swlkp3/Packet/(X)Net/xneten.doc)

To quit SYSOP mode, enter “q” or “quit”.

If you want to terminate the running (X)Net completely, enter “quit” again (sending just “q” doesn’t work in this case).

A couple things of note: Direwolf *must* be started before linuxnet, and both Direwolf and linuxnet *must* be started with sudo privileges. Also, if you close linuxnet and then restart it, it won't reconnect to Direwolf. You need to close Direwolf and linuxnet, then restart them with Direwolf being first. If running BPQ linked to (X)Net, BPQ doesn’t need to be started in any particular order, or restarted at any point.

If you edit AUTOEXEC.NET and AUTOBOOT.NET, be sure to close the running linuxnet first. You can edit and save the C.TXT, D.TXT, INFO.INF, and HELP.TXT files without closing linuxnet, and the changes also have immediate effect.

BPQ-to-(X)Net link config:

Configures BPQ and (X)Net to link to each other if running on the same machine address 127.0.0.1.

1. For BPQ and (X)Net to link via UDP, udp ports 10093 and 10094 are used. First shut down BPQ and (X)Net if they are running. Open BPQ’s “bpq32.cfg” file and scroll down to its AX/IP port settings. (My entire BPQ AX/IP port config is below at the end of this document for reference.) Make sure you have a line “UDP 10093” enabled. Then add a line to point to your (X)Net node callsign using port 10094, for example mine is:

```
MAP KP4DOG 127.0.0.1 UDP 10094 B
```

Close and save when finished.

2. Now, in your (X)Net’s AUTOBOOT.NET file, set up an axUDP line to link to BPQ. For example, mine:

```
attach ip0 axUDP 0 1 110094 d10093 127.0.0.1
```

Further down that file, create or edit a port section for the BPQ link relevant to your setup. Mine is:

```
#BPQ port (comment line)  
po 0 dup 0  
po 0 baud 115200
```

```
po 0 win 7
po 0 name kp3ft-7 BPQ (name that shows in the ports list)
po 0 qual 203
route bc ad 0 kp3ft-7 (route broadcast on port 0 to your BPQ node callsign)
route bc ad 0 NODES (route broadcast node-list on port 0)
```

BPQ AX/IP port for reference:

```
PORT
PORTNUM=6
ID=AX/IP
DRIVER=BPQAXIP
QUALITY=203
MINQUAL=168
FRACK=5000
RESPTIME=1000
RETRIES=5
MAXFRAME=7
PACLEN=255
DIGIFLAG=0
CONFIG
MHEARD
UDP 10093
BROADCAST NODES
  MAP KP4DOG 127.0.0.1 UDP 10094 B
ENDPORT
```