USING DIRE WOLF WITH OUTPOST PM

Dire Wolf and Outpost PM

John Huggins, KX4O, asked me to experiment with the prototype VAPN packet node. At first it was just if I could hear the beacon which, being 5 miles away, I did. Then he wanted me to test the Bulletin Board System (BBS) functions using the Terminal Node Controller (TNC) built into my Kenwood TS-2000 using the standard TNC command set. All worked as expected. Then John asked me work with and test the VAPN BBS with Outpost.

Outpost Packet Message Manager

Outpost PMM is PC based software client that allows you to work with an amateur radio BBS much like you would with your own email application. It "front-ends" the interaction with a BBS using a GUI interface and allows the receiving and sending of BBS messages without the end user needing to know the primitive traditional BBS command set. It allows the user to compose messages and to store sent and received messages in various folders on the local PC.

Outpost's primary function is to be part of an ECOM response system. It has features to support that purpose with built in message forms and reports. It also allows the use of tactical call signs in emergency response situations. However this doesn't mean it cannot be used for other purposes such as VAPN's uses.

I was successful in getting Outpost working with the VAPN BBS with my TS-2000. I will explain more on setting up Outpost later in this article.

Dire Wolf

John's goal was to get away from having a hardware TNC and to using a computer soundcard and software. This is not a new idea, George Rossopolos, SV2AGW, packet engine AGWPE has been around been around since the late 1990's. It has been wide use since then and there are many programs that have been written using this application, Outpost being one of them. John was interested in using Dire Wolf as the soundcard application because of it superior performance.

I had never heard of Dire Wolf before John asked me. Dire Wolf is an acronym for "**D**ecoded **I**nformation from **R**adio **E**missions for **W**indows **O**r **L**inux **F**ans". It was developed by John Lagner, WB2OSZ. As the name implies it can run on PC's and even a Raspberry Pi. It can handle data rates from 300 to 38,400 bps depending on the protocol. It can also accommodate up to six sound ports. It can decode more than 1000 error-free frames using WA8LMF TNC Test CD.

Dire Wolf was written primarily for APRS as a simple standalone APRS decoder, digipeater, APRStt gateway or Internet Gateway (IGate). But it can act as a virtual TNC for many other applications. It can interface with applications by the AGW network Protocol, KISS network protocol or KISS serial port. Recently Dire Wolf was enhanced to support connected mode, APRS is unnumbered information – no specific destination, where in connected mode is interacting with a specific destination.

Putting It All Together

I should describe my set up here at my QTH. I am using a Dell Latitude laptop running Windows 7 Ultimate, a Tigertronics SignaLink USB sound card interface and a Kenwood TS-2000 multi-band transceiver.

So forward I went. John thought at first using Dire Wolf via its virtual serial port. In order to try to connect an application serial port you need to use a virtual serial null modem connection. This was accomplished using freeware software called “com0com”. It creates a virtual null modem where the application uses one of the virtual com ports for input and using the other virtual com port as the input to Dire Wolf.

I configured Outpost to use the virtual null modem com port input and pointed Dire Wolf to output side the null virtual com port as input. I then set up Dire Wolf to use my SignaLink sound card for transmit and receive.

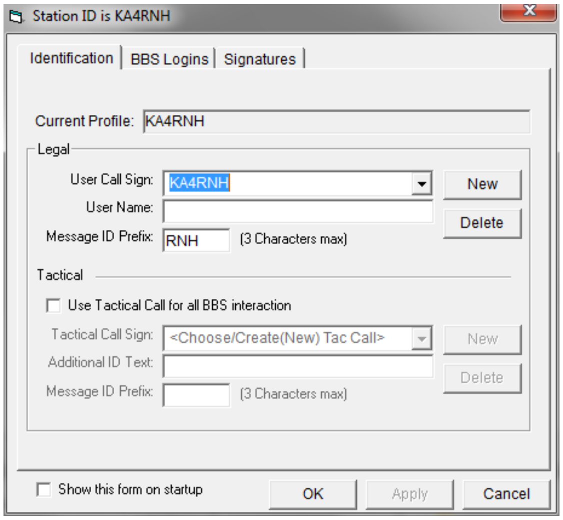
So I then tried to get it all to work. I checked and double checked everything to no avail. I could not get Outpost to communicate with Dire Wolf. I looked and I couldn’t figure out what was wrong. Finally it hit me what the problem was. Dire Wolf’s serial input is a KISS TNC. In serial mode Outpost is looking for a hardware TNC. The very nature of the way it works with a serial port, “reading” command prompts and “responding” with commands, it would not work with a KISS TNC because by its very nature a KISS TNC offloads all of those process to software.

So how can I make these two applications work together? Outpost can interface with AGWPE. Dire Wolf accepts the AWGPE network protocol. So I put the two together and that was it, I successfully was able to have Outpost interact with the VAPN BBS using the Dire Wolf TNC.

Setting It All Up – Outpost

All of the configuration settings for Outpost are accessed via the setup menu.

The first item to set up was my station ID.



Station ID is KA4RNH

Current Profile: KA4RNH

User Call Sign: KA4RNH

User Name:

Message ID Prefix: RNH (3 Characters max)

☐ Use Tactical Call for all BBS interaction

Tactical Call Sign: <Choose/Create(New) Tac Call>

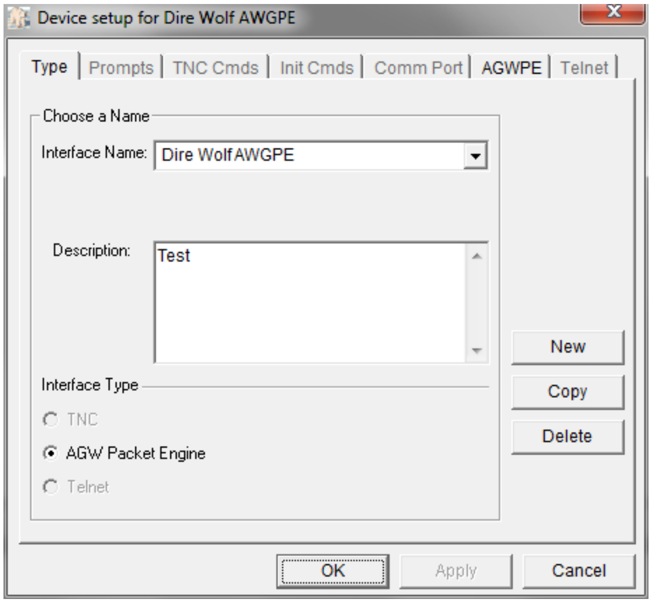
Additional ID Text:

Message ID Prefix: (3 Characters max)

☐ Show this form on startup

Since I was just testing there was no need for any other identification but as you can see many other options are available. At this time there was no need for using passwords.

The next item to set up is the Interface (yes I know this is backwards in order in the setup menu). The first tab is type.



Device setup for Dire Wolf AWGPE

Choose a Name

Interface Name: Dire Wolf AWGPE

Description: Test

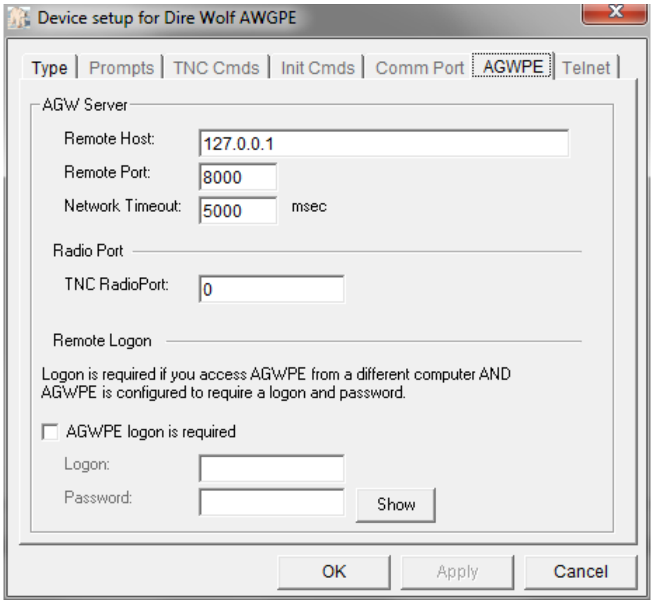
Interface Type

○ TNC

● AGW Packet Engine

○ Telnet

Select New, Type in a name describing the interface, optionally a description and select the "AGW Packet Engine" and click "Apply". This will enable the AGWPE tab. Select this tab.



Device setup for Dire Wolf AWGPE

AGW Server

Remote Host: 127.0.0.1

Remote Port: 8000

Network Timeout: 5000 msec

Radio Port

TNC RadioPort: 0

Remote Logon

Logon is required if you access AGWPE from a different computer AND AGWPE is configured to require a logon and password.

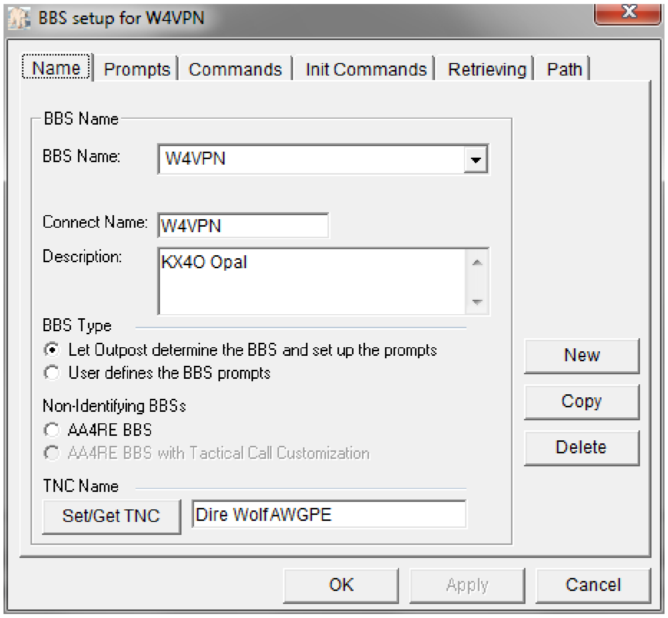
□ AGWPE logon is required

Logon:

Password:

For testing purposes no changes were necessary as Dire Wolf's default AGW network interface defaults to these parameters. However note the TNC radio port. Since Dire Wolf is capable of multiple ports this value may be different in a multi-port configuration.

That is all that needs to be done for the interface settings. The next thing is the BBS setup parameters.



BBS setup for W4VPN

BBS Name

BBS Name: W4VPN

Connect Name: W4VPN

Description: KX4O Opal

BBS Type

● Let Outpost determine the BBS and set up the prompts

○ User defines the BBS prompts

Non-Identifying BBSs

○ AA4RE BBS

○ AA4RE BBS with Tactical Call Customization

TNC Name

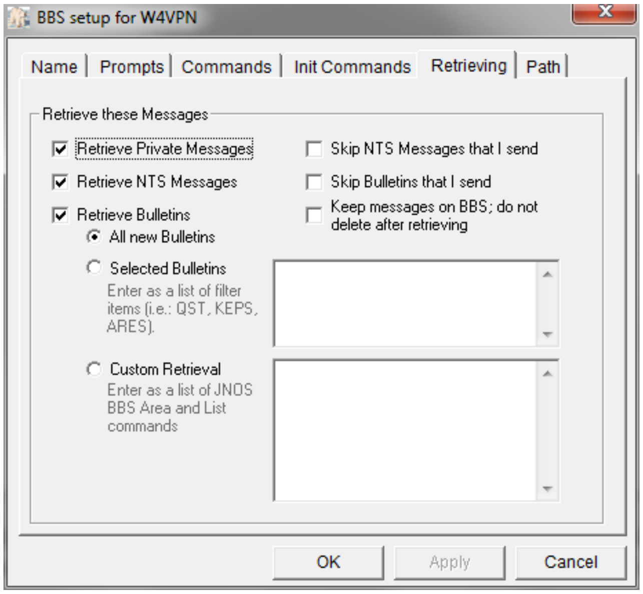
Set/Get TNC: Dire Wolf AWGPE

Select New. Give this BBS setting a name; in this case I named it W4VPN. The connect name should match the name of the BBS you are connecting to. You can put in a description. I let “BBS Type” and “Non-identifying BBSs” default.

Under Set/Get TNC you select the interface we just created above.

The next three tabs, “Prompts”, “Commands” and “Init Commands” I let default.

Under “Retrieve Commands” tab



BBS setup for W4VPN

Retrieve these Messages

☑ Retrieve Private Messages

☑ Retrieve NTS Messages

☑ Retrieve Bulletins

● All new Bulletins

○ Selected Bulletins
Enter as a list of filter items (i.e.: QST, KEPS, ARES).

○ Custom Retrieval
Enter as a list of JNOS BBS Area and List commands

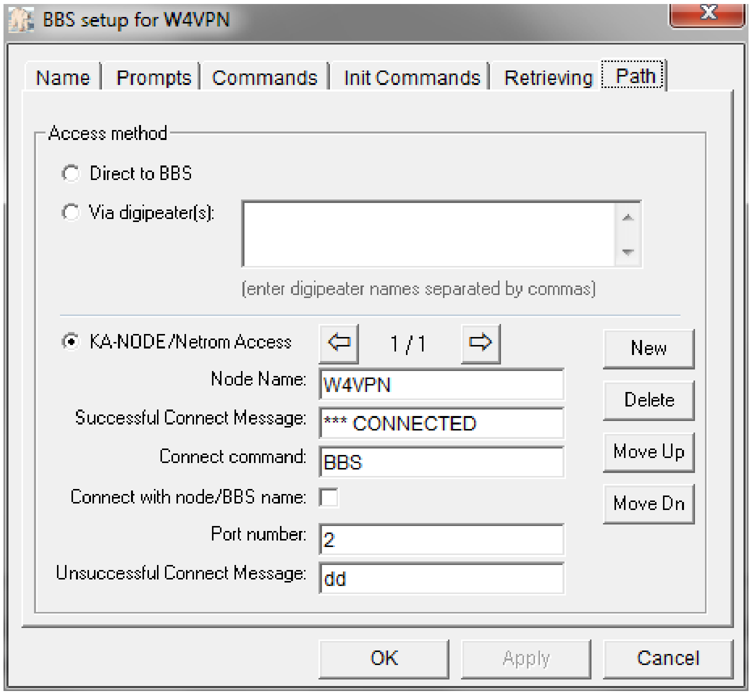
☐ Skip NTS Messages that I send

☐ Skip Bulletins that I send

☐ Keep messages on BBS; do not delete after retrieving

I checked “Retrieve Private Messages”, “Retrieve NTS Messages” and “Retrieve Bulletins” although for initial testing I probably only needed to check “Retrieve Private Messages”.

The next tab “Path” is critical for the VAPN BBS.


BBS setup for W4VPN
Name | Prompts | Commands | Init Commands | Retrieving | Path
OK
Apply
Cancel
New
Delete
Move Up
Move Dn

Access method

○ Direct to BBS

○ Via digipeater(s):

(enter digipeater names separated by commas)

● KA-NODE/Netrom Access 1 / 1

Node Name: W4VPN

Successful Connect Message: *** CONNECTED

Connect command: BBS

Connect with node/BBS name: □

Port number: 2

Unsuccessful Connect Message: dd

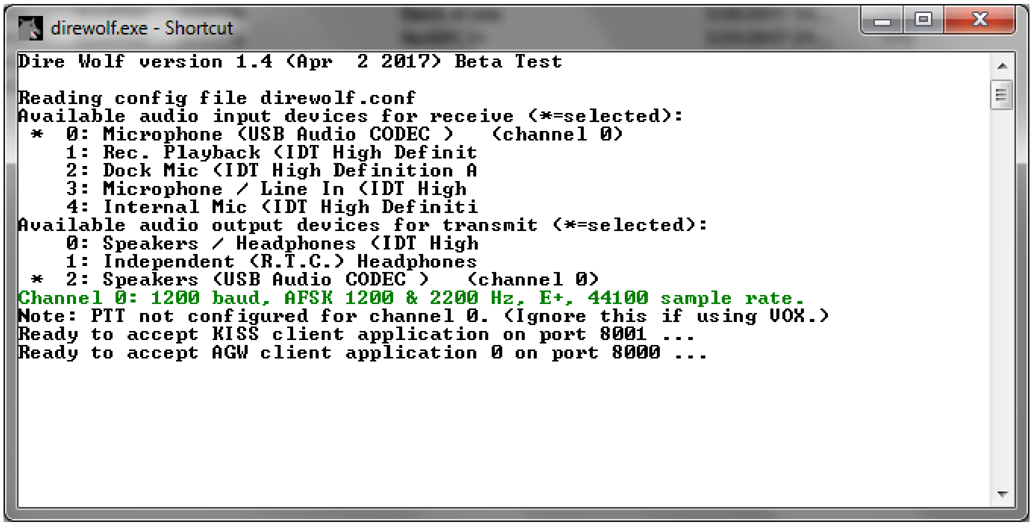
In order for me to connect to VAPN the parameters must be correct using the KA-Node settings. So select the KA-Node radio button. Then these parameters were set exactly to these values:
Node Name: “W4VPN”
Successful Connect Message: “*** Connected” = exactly what the BBS sends after the station connects with the node.
Connect command: “BBS” = VAPN does not default into BBS mode. This is the command to access the BBS of the node.

I let the other parameters default.

That is all that I needed to configure for Outpost.

Setting It All Up – Dire Wolf

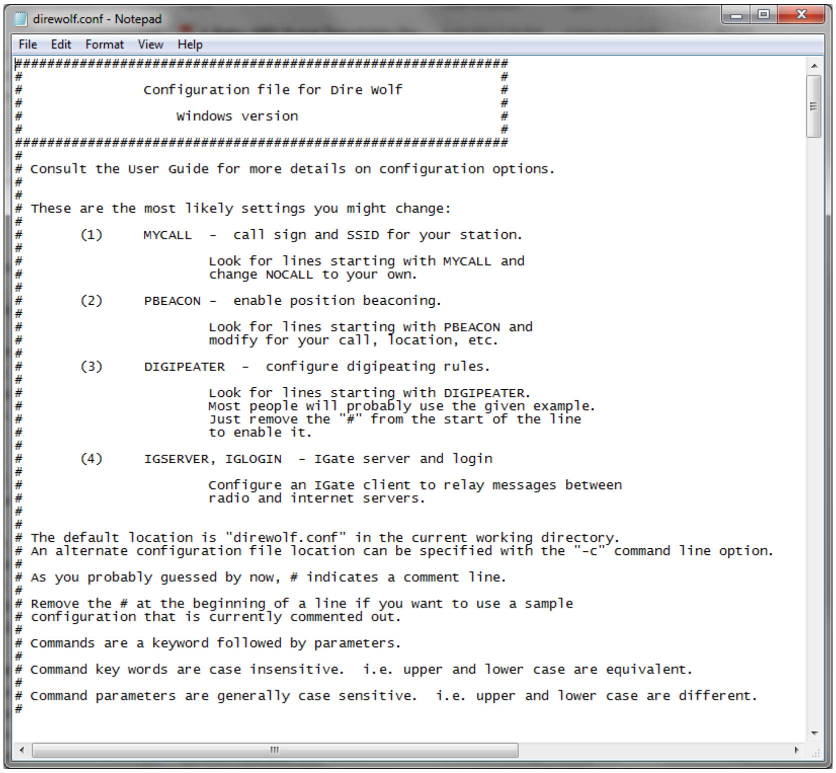
Dire Wolf does not have a UI. It does have a status screen that displays the configuration and color coded status messages.



```
Dire Wolf version 1.4 (Apr  2 2017) Beta Test

Reading config file direwolf.conf
Available audio input devices for receive (*=selected):
 *  0: Microphone (USB Audio CODEC )   (channel 0)
    1: Rec. Playback (IDT High Definit
    2: Dock Mic (IDT High Definition A
    3: Microphone / Line In (IDT High
    4: Internal Mic (IDT High Definiti
Available audio output devices for transmit (*=selected):
    0: Speakers / Headphones (IDT High
    1: Independent (R.T.C.) Headphones
 *  2: Speakers (USB Audio CODEC )   (channel 0)
Channel 0: 1200 baud, AFSK 1200 & 2200 Hz, E+, 44100 sample rate.
Note: PTT not configured for channel 0. (Ignore this if using VOX.)
Ready to accept KISS client application on port 8001 ...
Ready to accept AGW client application 0 on port 8000 ...
```

The entire configuration is done via Dire Wolf's direwolf.conf file. It is a simple text file that is well documented that you use to configure all of Dire Wolf's settings. A more in-depth explanation and details of all the configuration parameters are in the surprisingly detailed, and so far kept up to date, User Manual that is a separate download from the same directory that you download the setup file from.

Here is the beginning of the file:

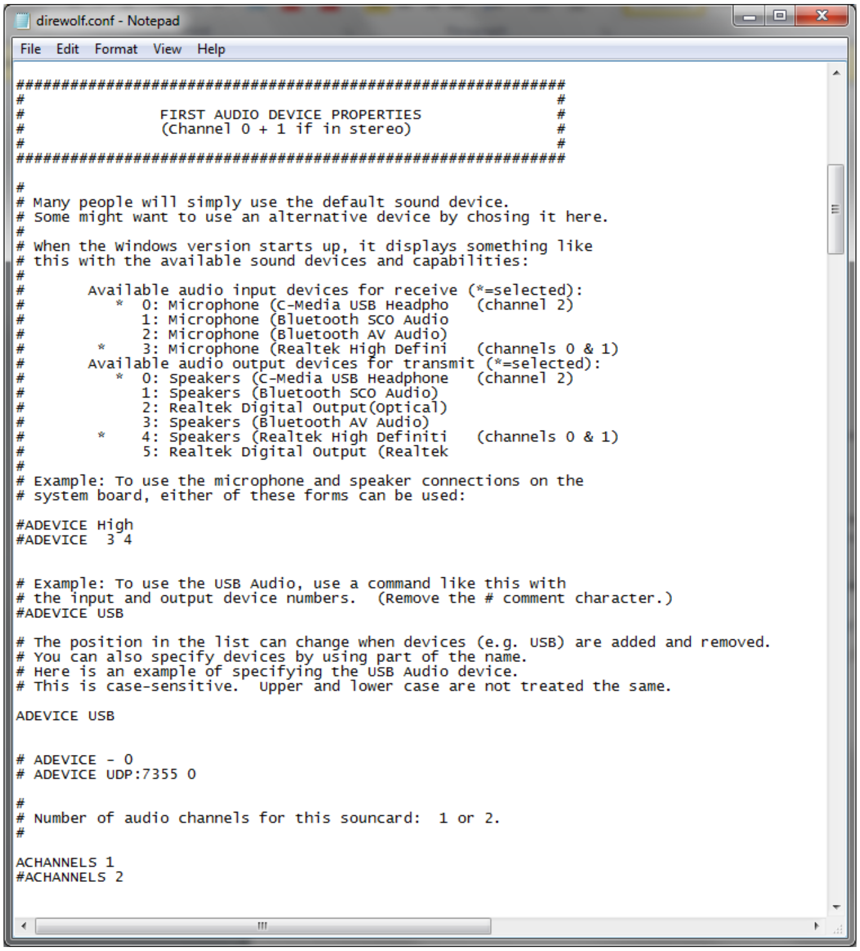


```
#############################################################
#                                                           #
#               Configuration file for Dire Wolf            #
#                                                           #
#                   Windows version                         #
#                                                           #
#############################################################
#
# Consult the User Guide for more details on configuration options.
#
#
# These are the most likely settings you might change:
#
#       (1)     MYCALL  -  call sign and SSID for your station.
#
#                       Look for lines starting with MYCALL and
#                       change NOCALL to your own.
#
#       (2)     PBEACON -  enable position beaconing.
#
#                       Look for lines starting with PBEACON and
#                       modify for your call, location, etc.
#
#       (3)     DIGIPEATER  -  configure digipeating rules.
#
#                       Look for lines starting with DIGIPEATER.
#                       Most people will probably use the given example.
#                       Just remove the "#" from the start of the line
#                       to enable it.
#
#       (4)     IGSERVER, IGLOGIN  - IGate server and login
#
#                       Configure an IGate client to relay messages between
#                       radio and internet servers.
#
#
# The default location is "direwolf.conf" in the current working directory.
# An alternate configuration file location can be specified with the "-c" command line option.
#
# As you probably guessed by now, # indicates a comment line.
#
# Remove the # at the beginning of a line if you want to use a sample
# configuration that is currently commented out.
#
# Commands are a keyword followed by parameters.
#
# Command key words are case insensitive.  i.e. upper and lower case are equivalent.
#
# Command parameters are generally case sensitive.  i.e. upper and lower case are different.
#
```

Keeping in mind what I wanted to accomplish: Use Dire Wolf's sound card TNC, using my SignaLink USB sound card interface, to interact with VAPN's BBS using Outpost. As stated above, I needed to use Dire Wolf's AGWPE network interface protocol.

Notice that there are a lot of options given as examples and you simply uncomment (remove the '#') the parameters you want and comment out (put in the '#') the ones you don't.

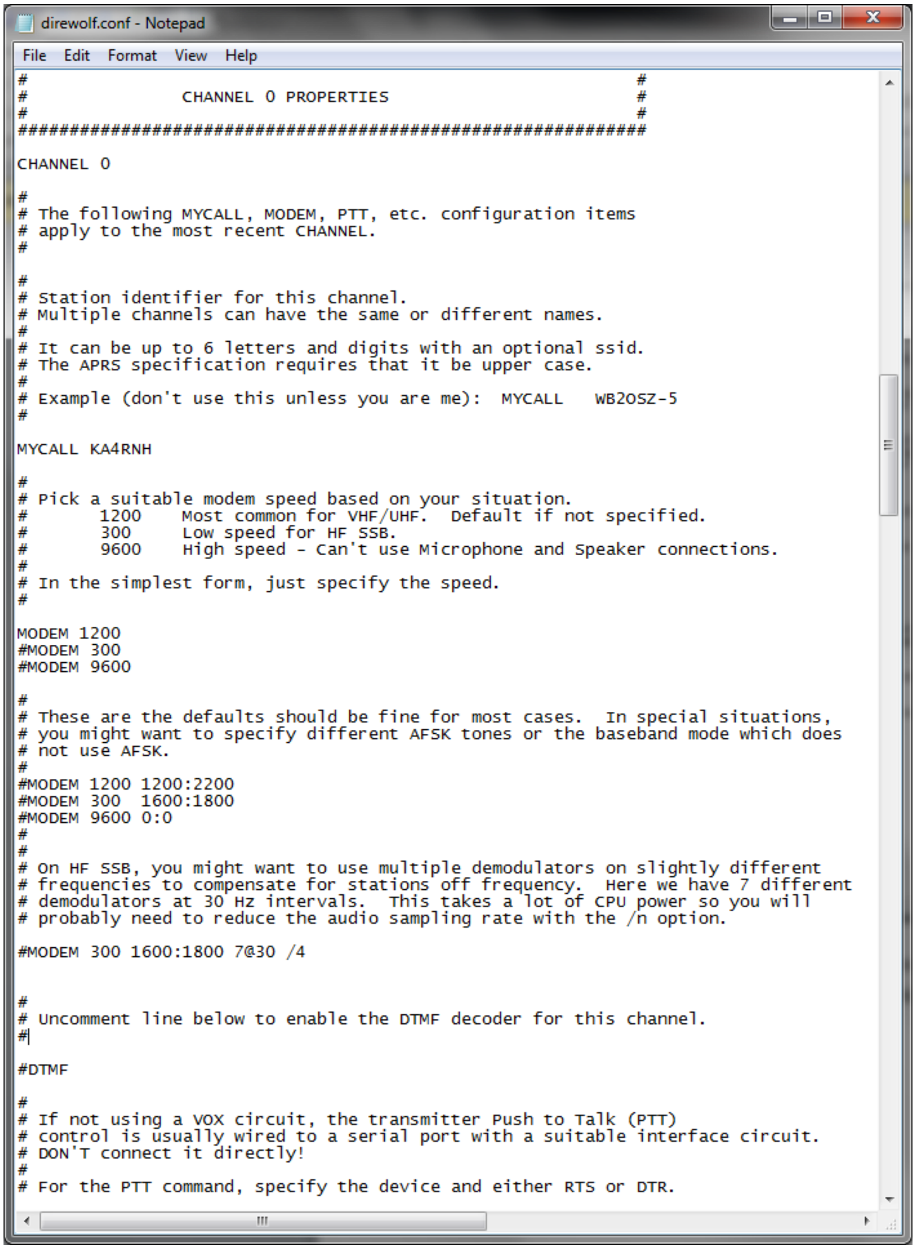
First thing you need to do is set the device (my SignaLink) and how it is to be used. I am only using one device and one channel (sound channels – if you are using a stereo device L and R could be two different channels, each independently handling their own packets).



```
#############################################################
#                                                           #
#               FIRST AUDIO DEVICE PROPERTIES               #
#               (Channel 0 + 1 if in stereo)                #
#                                                           #
#############################################################

#
# Many people will simply use the default sound device.
# Some might want to use an alternative device by chosing it here.
#
# When the Windows version starts up, it displays something like
# this with the available sound devices and capabilities:
#
#       Available audio input devices for receive (*=selected):
#          *  0: Microphone (C-Media USB Headpho   (channel 2)
#             1: Microphone (Bluetooth SCO Audio
#             2: Microphone (Bluetooth AV Audio)
#        *    3: Microphone (Realtek High Defini   (channels 0 & 1)
#       Available audio output devices for transmit (*=selected):
#          *  0: Speakers (C-Media USB Headphone   (channel 2)
#             1: Speakers (Bluetooth SCO Audio)
#             2: Realtek Digital Output(Optical)
#             3: Speakers (Bluetooth AV Audio)
#        *    4: Speakers (Realtek High Definiti   (channels 0 & 1)
#             5: Realtek Digital Output (Realtek
#
# Example: To use the microphone and speaker connections on the
# system board, either of these forms can be used:

#ADEVICE High
#ADEVICE  3 4


# Example: To use the USB Audio, use a command like this with
# the input and output device numbers.  (Remove the # comment character.)
#ADEVICE USB

# The position in the list can change when devices (e.g. USB) are added and removed.
# You can also specify devices by using part of the name.
# Here is an example of specifying the USB Audio device.
# This is case-sensitive.  Upper and lower case are not treated the same.

ADEVICE USB


# ADEVICE - 0
# ADEVICE UDP:7355 0

#
# Number of audio channels for this souncard:  1 or 2.
#

ACHANNELS 1
#ACHANNELS 2
```

Since I am using my SignaLink I commented out the other "ADEVICE" settings and uncommented the "ADEVICE USB" line. I then verified that "ACHANNELS" was set to one "1".

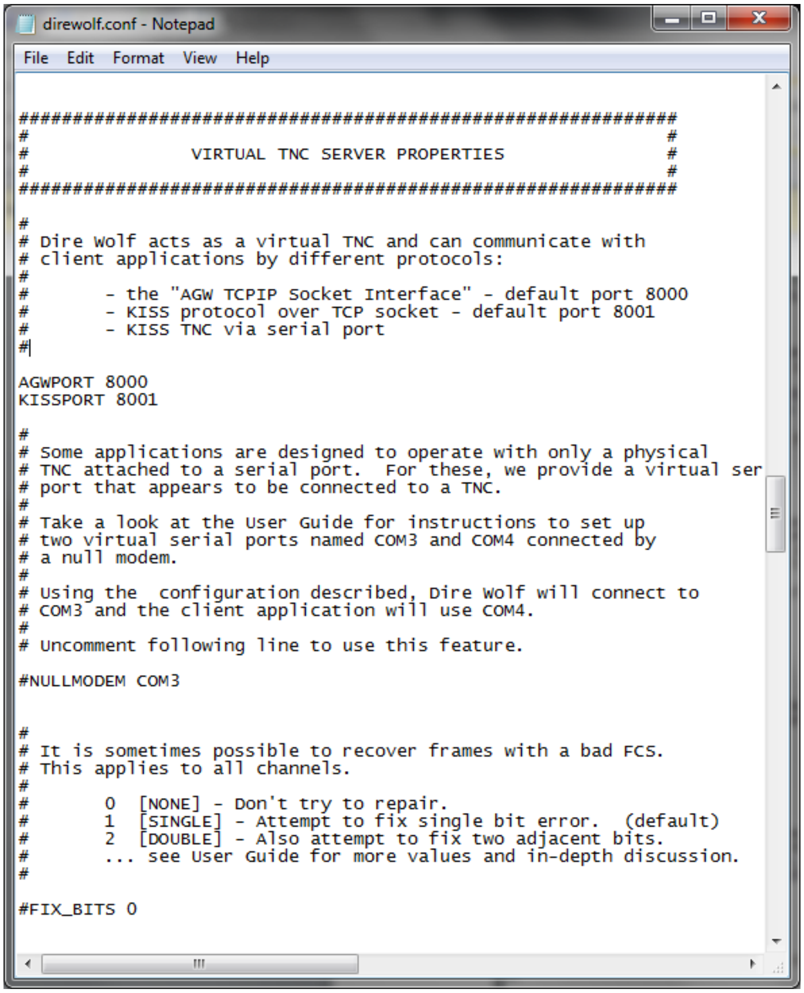
The next thing to configure is each specific channel properties. Each channel is separate with its own call, modem speed, DTMF and PPT properties.



```
#                                                              #
#               CHANNEL 0 PROPERTIES                           #
#                                                              #
################################################################

CHANNEL 0

#
# The following MYCALL, MODEM, PTT, etc. configuration items
# apply to the most recent CHANNEL.
#

#
# Station identifier for this channel.
# Multiple channels can have the same or different names.
#
# It can be up to 6 letters and digits with an optional ssid.
# The APRS specification requires that it be upper case.
#
# Example (don't use this unless you are me):  MYCALL   WB2OSZ-5
#

MYCALL KA4RNH

#
# Pick a suitable modem speed based on your situation.
#       1200    Most common for VHF/UHF.  Default if not specified.
#       300     Low speed for HF SSB.
#       9600    High speed - Can't use Microphone and Speaker connections.
#
# In the simplest form, just specify the speed.
#

MODEM 1200
#MODEM 300
#MODEM 9600

#
# These are the defaults should be fine for most cases.  In special situations,
# you might want to specify different AFSK tones or the baseband mode which does
# not use AFSK.
#
#MODEM 1200 1200:2200
#MODEM 300  1600:1800
#MODEM 9600 0:0
#
#
# On HF SSB, you might want to use multiple demodulators on slightly different
# frequencies to compensate for stations off frequency.  Here we have 7 different
# demodulators at 30 Hz intervals.  This takes a lot of CPU power so you will
# probably need to reduce the audio sampling rate with the /n option.

#MODEM 300 1600:1800 7@30 /4


#
# Uncomment line below to enable the DTMF decoder for this channel.
#

#DTMF

#
# If not using a VOX circuit, the transmitter Push to Talk (PTT)
# control is usually wired to a serial port with a suitable interface circuit.
# DON'T connect it directly!
#
# For the PTT command, specify the device and either RTS or DTR.
```

Since I am only using one channel. I set the call sign to my own call sign. Then I uncommented out the "MODEM 1200" line. As you can see by the other examples you modify the behavior of the modem parameters in a variety of ways. There is no need for a DTMF decoder. Not show are the PTT parameters, because I was using VOX, however those can be set for each channel too.

The final settings to verify are the Virtual TNC Settings.



```
#############################################################
#                                                           #
#               VIRTUAL TNC SERVER PROPERTIES               #
#                                                           #
#############################################################

#
# Dire Wolf acts as a virtual TNC and can communicate with
# client applications by different protocols:
#
#       - the "AGW TCPIP Socket Interface" - default port 8000
#       - KISS protocol over TCP socket - default port 8001
#       - KISS TNC via serial port
#

AGWPORT 8000
KISSPORT 8001

#
# Some applications are designed to operate with only a physical
# TNC attached to a serial port.  For these, we provide a virtual ser
# port that appears to be connected to a TNC.
#
# Take a look at the User Guide for instructions to set up
# two virtual serial ports named COM3 and COM4 connected by
# a null modem.
#
# Using the  configuration described, Dire Wolf will connect to
# COM3 and the client application will use COM4.
#
# Uncomment following line to use this feature.

#NULLMODEM COM3


#
# It is sometimes possible to recover frames with a bad FCS.
# This applies to all channels.
#
#       0  [NONE] - Don't try to repair.
#       1  [SINGLE] - Attempt to fix single bit error.  (default)
#       2  [DOUBLE] - Also attempt to fix two adjacent bits.
#       ... see User Guide for more values and in-depth discussion.
#

#FIX_BITS 0
```

As you can see this is where the AGWPORT is set and the KISS network port are set. I left them to default. **Note the AGWPE port number is the same port number as specified in the Outpost Interface->AGWPE tab.** Notice also that this is where I uncommented the “NULLMODEM COM3” and set the value for the virtual null modem when I was trying to get Outpost to work via a COM port. I commented this line out when I switched to using the AGWPE network port.

There are many other Dire Wolf settings that follow including beaconing, digipeating and a bunch of APRS settings. This is a testament to how versatile and powerful Dire Wolf is as a virtual TNC.

After putting all of the pieces together everything worked as I had hoped. This is the simplest of scenarios. But hopefully you can see how VAPN can build on these tools to achieve to goals of the organization.