

ax25Interface User's Guide

3.0b02

ax25Interface is Copyright (c) 2008 - Pete Loveall AE5PL pete@ae5pl.net

Use of the software is acceptance of the agreement to not hold the author or anyone associated with the software liable for any damages that might occur from its use.

APRS is a trademark of Bob Bruninga

Other trademarks included in the following text are recognized as belonging to the respective trademark holders.

Table of Contents

Section 1 - Introduction	1
Section 2 - Program Requirements and Description	2
Section 3 - Configuration Parameters	3
javAPRSIGate Parameters	4
TNCModule=.....	4
TNCIFieldMax=256.....	4
TNC General Parameters	4
TNCspeed=1200	4
TNCMaxVias=7.....	4
TNCPortInit=	4
TNCPortInitWait=true.....	4
ax25 General Parameters	4
ax25TNCPortName=1	4
Section 4 - Recommended Configurations	5
Section 5 - Installation Instructions	6
Section 6 – Status Page	7

Section 1 - Introduction

ax25Interface was written to provide an interface between javAPRSIGate and the Linux kernel ax25 socket support.

The ax25Interface source code is published for use by other developers to work with the javAPRSIGate TNCInterface.

ax25Interface makes use of javax25 written by Øyvind Hanssen and maintained on sourceforge.net.

Section 2 - Program Requirements and Description

ax25Interface is designed to run on Linux with any recent Java Virtual Machine.

ax25Interface is comprised of a number of classes which Java looks at as objects. The main class is ax25Interface. This class is called at startup, sets parameters, and begins execution of the different support threads.

ax25Interface works in conjunction with the javAPRSGate TNCInterface to provide full, bidirectional communication with ax25. All IGate logic is handled at in javAPRSGate which leaves the ax25Interface to concentrate on formatting and sending packets to the TNC log ports.

This manual assumes that the user is familiar with ax25 support in Linux and has properly set up ax25ports and run the associated configuration programs.

Section 3 - Configuration Parameters

The configuration parameters reside in a configuration file which, by default, is called javaprssvr.cfg. You can use any text file if you pass the name into javAPRSSvr as a command line parameter.

The parameters are CASE SENSITIVE. Defaults are shown below.

NOTE: UNLESS YOU REQUIRE A SETTING OTHER THAN THE DEFAULT, DO NOT INCLUDE ANY PARAMETERS WITH DEFAULT SETTINGS.

List parameters may be defined on the line or may be defined in a text file. If defined on the line, each entry is separated by a semicolon. If defined in a file, each entry is put on a separate line. Do not put blank lines in the file. The file must have the extension .lst For instance, this would be the definition for hubs where you want to connect to first.aprs.net and second.aprs.net port 1313:

```
hubs=first.aprs.net:1313;second.aprs.net:1313
```

Or you could have the following 2 lines in hubs.lst:

```
first.aprs.net:1313  
second.aprs.net:1313
```

You would then put the following line in your configuration file:

```
hubs=hubs.lst
```

(R) at the beginning of the parameter description means that the parameter can be changed on-the-fly from the console with either the S or R commands.

javAPRSIGate Parameters

TNCModule=

This must be set to ax25Interface.
Set to TNCModule=ax25Interface

TNCFieldMax=256

This sets the maximum information field length for packets gated to RF.
Modify this only if there is an absolute requirement. Some TNC's "break" if they send or receive packets with more than 256 octets in the information field.

TNC General Parameters

TNCSpeed=1200

This sets pacing for the TNC interface.
Set this to the RF speed of the TNC. This reduces the possibility of TNC overruns in KISS mode.

TNCMaxVias=7

This sets the maximum number of digipeaters in the path.
Modify this only if there is an absolute requirement. Some TNC's "break" if they send or receive packets with more than 7 digipeaters in the path.

TNCPortInit=

This defines a command line to be run before opening the TNC port.
This may be a single command or a batch/script file. In Linux, be sure to prefix a script file with ./ so the file is executed.
Also be sure to have execute rights for the desired script and programs.

TNCPortInitWait=true

This determines whether the interface should wait for the TNCPortInit command to complete before continuing.
Set this to false if you are running a program which will remain operational after TNC initialization.

ax25 General Parameters

ax25TNCPortName=1

(R)This is the port name for the TNC as defined in ax25ports.

Section 4 - Recommended Configurations

I recommend using matching the settings to your installation of ax25 sockets.

Section 5 - Installation Instructions

ax25Interface is included in all of the combined jar and exe files. Simply add `TNCModule=ax25Interface` to activate it.

Unzip `libjavax25.zip` into its own directory. If you are running Linux on a Intel 386 or above machine and running the Sun 1.4.2 or higher JVM, copy `libjavax25.so` to the `lib/i386` directory found under the JRE directory.

If you need to make a new `.so` file (requires the Sun JDK), change to the native subdirectory. Edit Makefile and add the following line to the top of the file:

```
INCLUDES = -I /usr/java/j2sdk1.4.2_04/include -I /usr/java/j2sdk1.4.2_04/include/linux
```

Run "make". Copy the resulting `.so` file to your Java's library directory (`/usr/java/j2sdk1.4.2_04/jre/lib/i386` for Sun's 1.4.2_04 SDK).

Start `javAPRSSrvr` by entering at the command line in your `javAPRSSrvr` directory:

```
/usr/java/j2sdk1.4.2_04/bin/java -server -jar javAPRSSrvr.jar
```

(replace `/usr/java/j2sdk1.4.2_04` with the directory of the Java SDK you are running)

Section 6 – Status Page

javAX25 0.2	Øyvind Hanssen LA7ECA	javAX25 version and author
TNC Port Name		1 Name found in /etc/ax25ports