

# **javax.comm Interface User's Guide**

## **javAPRSSrvr 4.3**

javaxcommIntf is Copyright © 2019 - Pete Loveall AE5PL [pete@ae5pl.net](mailto: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.

# Table of Contents

Section 1 - Introduction .....	1
Section 2 - Program Requirements and Description .....	2
Section 3 - Configuration Parameters .....	3
javAPRSSrvr Properties .....	4
SerialPorts= .....	4
javaxcommIntf General Properties .....	5
ClassPath= .....	5
Class=.....	5
LibraryPath= .....	5
IntfName= .....	5
InitFile= .....	5
RFSpeed=1200.....	5
SharedTransmit=true .....	5
KISSMode=false .....	5
TCPPorts= .....	5
Serial Port Properties .....	6
SerialPortName= .....	6
SerialSettings=9600,8,1,n.....	6
SerialFlowCtrl=none .....	6
Section 4 - Recommended Configurations .....	7
Section 5 - Installation Instructions .....	8

## Section 1 - Introduction

javaxcommIntf is written to provide a universal interface between Java applications and a serial port using the javax.comm serial package available from the Oracle and others.

javaxcommIntf extends net.ae5pl.serialintf.SerialIntf allowing a single serial port to be shared by multiple clients.

## Section 2 - Program Requirements and Description

javaxcommIntf is designed to run on any Java VM supported by javAPRSSrvr and any OS supported by the javax.comm package you are using.

javaxcommIntf is comprised of a number of classes which Java looks at as objects. The main class is net.ae5pl.javaxcommintf.javaxcommIntf. This class is called at startup, sets parameters, and begins execution of the different support threads.

javaxcommIntf extends net.ae5pl.serialintf.SerialIntf to provide full, bidirectional communication with the serial port. All IGate and TNC logic is handled at in other classes which leaves the javaxcommIntf to concentrate on sending and receiving serial data.

## Section 3 - Configuration Parameters

The configuration properties reside in properties files for each client adjunct, server adjunct, and port. The main properties file is called javaprssrvr.properties by default. You can use any text file for the main properties file if you pass the name into javAPRSSrvr as a command line parameter.

The property names are not case sensitive but the values can be. Defaults are shown below.

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

**List parameters (L)** may be defined on the property line or may be defined in a text file with the suffix .lst. 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 in the .lst file and the file name is the property value. Do not put blank lines in the file. For instance, this could be a definition for ListProperty (example only):

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

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

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

with ListProperty=hubs.lst

Properties preceded by a (M) are unchangeable and should not be included in your properties files. They are included in the descriptions below to indicate what common properties are available vs. those that have been forcibly overridden.

## ***javAPRSSrvr Properties***

### **SerialPorts=**

(L)This must include the javaxcommIntf properties file.

## ***javaxcommIntf General Properties***

### **ClassPath=**

(L) Must include javaxcommIntf.jar.

### **Class=**

Must be set to net.ae5pl.javaxcommintf.javaxcommIntf.

### **LibraryPath=**

(L) Must include the path to the javax.comm native library(ies).

### **IntfName=**

This is the name of the serial interface. This must match the respective client IntfName property.

### **InitFile=**

The contents of this file are sent to the serial interface as-is at start-up.

### **RFSpeed=1200**

This is used to pace output to the serial interface.

### **SharedTransmit=true**

Sets whether clients will see each other's packets.

### **KISSMode=false**

If true, a number of optimizations are implemented at the serial interface to enhance KISS TNC support.

### **TCPPorts=**

(L)Either IP:port or port list. Each defined port opens a listener which provides bidirectional access to the serial interface.



## ***Serial Port Properties***

### **SerialPortName=**

This is the “file” name of the serial port.

### **SerialSettings=9600,8,1,n**

This are the basic settings for the serial port.

The settings are speed(bps), data bits(5, 6, 7, **8**), stop bits (**1**, 1.5, 2), and parity:

**n = None**

e = Even

o = Odd

s = Space

m = Mark

### **SerialFlowCtrl=none**

This specifies the flow control for the serial port.

Valid values are:

**none (no flow control) (recommended)**

hardware (hardware flow control RTS/CTS)

xonxoff (software flow control Ctrl-S/Ctrl-Q)

## **Section 4 - Recommended Configurations**

Set the port to the settings required by your “modem” or TNC.

## Section 5 - Installation Instructions

javaxcommIntf is javaxcommIntf.jar. Set the ClassPath property in your javaxcommIntf properties file to javaxcommIntf.jar and the javax.comm jar file from Oracle or other vendor. You will need to load the javax.comm library file(s) in a directory and point the LibraryPath property in your javaxcommIntf properties file to that directory.