

javAPRSIGate User's Guide

2.3b05

javAPRSIGate is Copyright (c) 2006 - 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
javAPRSSrvr Parameters	3
IGateAdjunct=	3
IGate General Parameters	4
IGateCall=.....	4
TNCModule=.....	4
TNCIFieldMax=256	4
noGateISCalls=TCPXX.....	4
IGateISCalls=TCPIP.....	4
IGateRecentTime=30.....	4
IGateBackups=.....	4
IGateDigiDontGate=.....	4
IGateGateToRF=false.....	4
IGateExcludeCalls=.....	4
IGateMaxHops=1.....	5
UIFloodCalls=	5
NOIDDigis=.....	5
IGate Path Parameters.....	6
IGateVia=.....	6
IGateCallSpclPaths=.....	6
IGatePrefixSpclPaths=.....	6
IGate Pass Parameters.....	7
IGatePassCalls=	7
IGatePassPrefixes=	7
IGatePassCallPosits=	7
IGatePassPrefixPosits=.....	7
IGatePassUnprotos=	7
IGatePassUnprotoPrefixes=.....	7
IGatePassObjects=	7
IGateObjectPrefixes=.....	7
IGatePassGates=	7
IGate Beacon Parameters.....	8
IGateLat=	8
IGateLon=	8
IGateSymbol=I&	8
IGatePositCmt=	8
IGateStatus=.....	8
IGatePositInterval=20.....	8
IGateStatusInterval=60.....	8
Section 4 - Recommended Configurations	9
Section 5 - Installation Instructions	10
Section 6 - TNC Interfaces	11
Section 7 - Status Page.....	12

Section 1 - Introduction

javAPRSIGate was written to provide the amateur radio APRS community a simple, but effective IGate. It does not have a GUI as it is an adjunct to javAPRSSrvr. It does, however, provide most features desired in an IGate:

- Gate messages to stations based on the receiving station's distance in number of digi hops.
- Prevent gating based on the "from" station being heard directly (not via an IGate) on RF or the "to" station being seen as an IGate or directly on APRS-IS.
- Standard path for gating. Special paths based on "from" and "to" callsigns.
- Support for standard queries.
- Standard 3rd party format to RF.
- In addition to message gating, gate based on callsign-SSID, callsign prefix, object/item name, or object/item name prefix.
- Block gating for specific callsign-SSID's.
- Support for multiple TNC interface types. TNCInterface may be used by any developer to create other interface types.

This application operates as an IGate adjunct to javAPRSSrvr. It requires javAPRSSrvr to provide the network interface and do the packet parsing.

Section 2 - Program Requirements and Description

javAPRSIGate is designed to run on any OS with any recent Java Virtual Machine. It has been successfully tested with the Microsoft JVM for Windows and with the Sun 1.1.8 JDK for Windows.

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

javAPRSIGate monitors the RF receive to determine recently-heard, local-RF (within maxHops distance), and direct-heard status of stations. It also checks to see if the station is an IGate and marks it appropriately. Once this is done, the packets are checked for queries, general or directed.

javAPRSIGate monitors the Internet feed to determine if stations on the recently-heard list are also local to the Internet (have TCPIP* or TCPXX* in the header). Once this is done, the packets are checked for queries, general or directed.

If a packet from the Internet is not a query, then it is checked to see if the from station was recently heard on RF. If so, it is dropped. Otherwise, it is checked to see if it is on one of the pass lists, a message to a non-Internet connected RF station, or a posit from a station for whom a message has been gated.

Note that while javAPRSIGate is normally used for APRS-IS gating (TCPIP in the header), any gate identifier can be used. For instance, if you are gating from an isolated LAN (not connected to the Internet), you might use MYLAN for a gate header instead.

javAPRSIGate will gate any packets to RF which are received from a locally connected APRS client that has the same callsign-SSID as IGateCall. These packets will be passed directly (not as third-party packets) to RF with the path changed to match IGatePath. This makes it possible to have a single GUI client connected to javAPRSSrvr which, in turn, is directly connected to the TNC and have that client's packets sent to RF directly.

Section 3 - Configuration Parameters

The configuration parameters reside in a configuration file which, by default, is called javaprssrvr.cfg. You can use any text file if you pass the name into javAPRSSrvr 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.

javAPRSSrvr Parameters

IGateAdjunct=

This must be set to IGateAdjunct.

IGate General Parameters

IGateCall=

This is the callsign-SSID for the IGate.

It must conform to AX.25 standards and it must be different from javAPRSSrvr's userCall (the server's callsign-SSID).

TNCModule=

This is the class name for the TNC interface.

The TNC module must implement the TNCInterface. For instance, to use the KipSS interface, TNCModule=KipSSInterface

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.

noGateISCalls=TCPXX

(R)These are "digis" which are not allowed to be gated to RF.

IGateISCalls=TCPIP

(R)These are "digis" which are considered to be local to APRS-IS.

IGateRecentTime=30

(R)Number of minutes that IGate considers recent.

This is used by all "heard" tables.

IGateBackups=

(R)(List parameter)This defines IGates which will be monitored for activity by javAPRSIGate.

If any station in this list is not heard for IGateRecentTime minutes, javAPRSIGate will begin to gate to RF.

IGateDigiDontGate=

(R)(List Parameter)This specifies callsign-SSID's of digipeaters which preclude gating to RF.

Stations heard directly via the specified digipeater(s) are not considered "local". If the station is heard via another route as well, standard maximum hop processing will occur for the station.

IGateGateToRF=false

(R)This specifies whether data is sent to the TNC interface.

IGateExcludeCalls=

(R)(List Parameter)This specifies callsign-SSID's which are not to be gated to RF.

Packets from the specified callsigns or messages to the specified callsigns will not be gated to RF.

IGateMaxHops=1

(R)This specifies the maximum digi hops away a station can be to be considered local. javAPRSIGate does compensate for UIFlood digis, including digis with NOID turned off.

UIFloodCalls=

(R)(List parameter)"Callsigns" used with the UI flood algorithm.

This is set to callsigns used in the N-n flood algorithm. WIDE is an example under the original APRS specification. WIDEn-n as used today is not a UIFloodCalls because it is using the UI trace algorithm. Some areas still use WIDEn-n in flood mode.

NOIDDigis=

(R) (List parameter)Specifies digipeaters that insert/replace their callsign on a UI flood algorithm.

Use this to specify digipeaters that insert their callsign before a flood callsign like WIDEn-n. Also, if your area commonly uses WIDE1-1,WIDE2-1 for a path with WIDEn-n being used as a flood algorithm, set this to WIDE1 to consider WIDE1*,WIDE2-1 as a single hop. Also, using the same logic, if a nearby digipeater is used as the first hop by base stations (e.g. W5MRC-15), add that repeater to this parameter.

IGate Path Parameters

IGateVia=

(R)This is the comma delimited path to be used on RF.
Do not include the unproto. The unproto is set to APJI20.

IGateCallSpclPaths=

(R)(List Parameter)This is the comma delimited path to be used on RF. Paths are separated by semicolons. The format is fromCall,path (note the call the path is based on is the first call in the comma delimited list).
Do not include the unproto. The unproto is set to APJI20.

IGatePrefixSpclPaths=

(R)(List Parameter)This is the comma delimited path to be used on RF. Paths are separated by semicolons. The format is prefix,path (note the prefix of the call the path is based on is the first call in the comma delimited list).
Do not include the unproto. The unproto is set to APJI20.

IGate Pass Parameters

IGatePassCalls=

(R)(List Parameter)This specifies callsign-SSID's which are to be gated to RF. Packets from the specified callsigns will be gated to RF if they are not heard on RF.

IGatePassPrefixes=

(R)(List Parameter)This specifies callsign-SSID prefixes which are to be gated to RF. Packets from the specified callsign prefixes will be gated to RF if they are not heard on RF.

IGatePassCallPosits=

(R)(List Parameter)This specifies callsign-SSID's which are to be gated to RF. Only position packets will be gated to RF unless messaging is occurring with an RF station.

IGatePassPrefixPosits=

(R)(List Parameter)This specifies callsign-SSID prefixes which are to be gated to RF. Only position packets will be gated to RF unless messaging is occurring with an RF station.

IGatePassUnprotos=

(R)(List Parameter)This specifies packets with the specified unproto which are to be gated to RF. Packets with the specified unprotos will be gated to RF if they are not heard on RF. IGateCallSpclPaths and IGatePrefixSpclPaths use the unproto for determining if the packet should be gated using a special path.

IGatePassUnprotoPrefixes=

(R)(List Parameter)This specifies unproto prefixes which are to be gated to RF. Packets with the specified unproto prefixes will be gated to RF if they are not heard on RF. IGateCallSpclPaths and IGatePrefixSpclPaths use the unproto for determining if the packet should be gated using a special path.

IGatePassObjects=

(R)(List Parameter)This specifies object or item names which are to be gated to RF.

IGateObjectPrefixes=

(R)(List Parameter)This specifies object or item name prefixes which are to be gated to RF.

IGatePassGates=

(R)(List Parameter)This specifies callsign-SSID's of IGates whose gated packets are to be gated to RF. Packets gated to APRS-IS by the specified callsigns will be gated to RF. This uses the callsign after the qAR, qAr, or qAo construct to determine if the packet was gated to APRS-IS by the specified station. IGateCallSpclPaths and IGatePrefixSpclPaths use the IGate call for determining if the packet should be gated using a special path.

IGate Beacon Parameters

IGateLat=

(R)This specifies the latitude of the IGate.
The latitude is specified in decimal degrees, south is negative.

IGateLon=

(R)This specifies the longitude of the IGate.
The longitude is specified in decimal degrees, west is negative.

IGateSymbol=I&

(R)This specifies the APRS symbol to use in the posit.
I& is the preferred symbol for an IGate.

IGatePositCmt=

(R)This specifies text to follow the posit data.
You can put PHG values, altitude, etc. here.

IGateStatus=

(R)This specifies the status text for the status beacon.
Do not include the > data type.

IGatePositInterval=20

(R)This specifies the position beacon rate in minutes.

IGateStatusInterval=60

(R)This specifies the status beacon rate in minutes.

Section 4 - Recommended Configurations

I recommend using default settings except where necessary for proper IGate operation.

If you want to disable all IGate operation, set IGateAdjunct=

For normal IGate operation, set IGateCall, IGateVia, IGateLat, IGateLon, and IGateGateToRF.

Section 5 - Installation Instructions

IGateAdjunct is contained in all of the combined jar and exe files. Simply add IGateAdjunct=IGateAdjunct to activate it.

Section 6 - TNC Interfaces

The source for the TNC interfaces is provided for reference by other developers. All TNC interfaces must implement the TNCInterface 2.0 to work with javAPRSIGate 2.0. Currently, there are interfaces for KipSS (Windows, APRS+SA), AGWPE (Windows), Linux AX.25 support, and direct serial interface to KISS TNC's ('ix' operating systems).

Section 7 – Status Page

IGate Callsign	AE5PL-9	Callsign used by the IGate
Status	Gating to RF	Gating to RF turned on and TNC reporting connected

Packets Gated to Server	14,221	Packets passed from TNC to server
Packets Gated to RF	447	Packets gated to TNC
Messages Gated to RF	293	Message packets gated to TNC

Maximum Digi Hops for Local Stations	1	IGateMaxHops
History Time for Station Lists (minutes)	30	IGateRecentTime

Recently Heard Stations	69	Number of stations heard within IGateRecentTime
Local RF Stations	55	Number of stations heard within IGateRecentTime within IGateMaxHops + 1
Directly Heard Stations	10	Number of stations heard within IGateRecentTime Heard direct (no digi)

Bytes Sent to RF	52,885	Total bytes sent to TNC
Bytes Received from RF	1,168,568	Total bytes received from TNC