APRS IGate User's Guide
javAPRSSrvr 4.3

Table of Contents

Section 1 - Introduction

APRS IGate 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 APRS IGate:

- Gate messages to stations on RF based on the receiving station's distance computed by number of digipeater hops.
- Prevent gating messages to RF if the "from" station has been heard directly (not via an IGate) on RF or the "to" station has been heard directly on APRS-IS (also not via an IGate).
- Standard path for gating to RF. Special paths based on "from" and "to" callsigns.
- Support for standard directed queries.
- Standard third-party format for packets gated to RF.
- Special non-third-party packet formatting for packets gated to RF for "in-house" stations.
- Gate packets to RF based on originator callsign-SSID or object/item name (including prefix definitions).
- Block gating to RF for specific callsign-SSID's.
- Support for KISS TNCs (AX.25) and D-PRS (D-STAR®).

This application operates as a client to javAPRSSrvr. It requires javAPRSSrvr 4.0 to provide the network interface and do the packet parsing.

Section 2 - Program Requirements and Description

APRS IGate is designed to run on any OS and Java VM capable of running javAPRSSrvr 4.0.

APRS IGate is comprised of a number of classes which Java looks at as objects. APRS IGate classes are in APRSIGate.jar which must be specified in the ClassPath property in the IGate's properties file.

APRS IGate 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 directed queries.

APRS IGate 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 directed queries.

If a packet from the Internet is not a query, then it 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 recently been gated to RF.

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

APRS IGate will gate any packets to RF which are received from an APRS-IS client that has a callsign-SSID included in the directCalls property. These packets will be passed directly (not as third-party packets) to RF with the path changed to match the path in the directCalls property for that callsign. This makes it possible to have a local client connected to javAPRSSrvr which, in turn, appears to be on RF for RF users including remote IGates. This should be used sparingly as it has a potential for loops.

Section 3 - Configuration Properties

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.

Where TCPIP is referenced, it means the callsign-SSID set in NetworkID.
Where TCPXX is referenced, it means the callsign-SSID set in UnverifiedNetworkID.

javAPRSSrvr Properties

Clients=

This must include the file of this IGate's properties file.

IGate Properties

ClassPath=
(L) Must include APRSIGate.jar.

Class=
Must be set to net.ae5pl.aprsigate.APRSIGate.

PacketInterface=net.ae5pl.aprsigate.KISSTNCIntf
Packet interface class name. If this is class is not part of the distribution APRSIGate.jar, you must specify the class (or containing jar) in the ClassPath property.

IntfName=
This is the name of the serial interface. This must match the respective Serial Interface IntfName property.

StationCall=
This is the callsign-SSID for the IGate.
It must conform to AX.25 standards and it must be different from javAPRSSrvr's ServerCall (the server's callsign-SSID) and any other station's callsign-SSID visible to APRS-IS.

Symbol=I&
Table/overlay and symbol for posits.

IGatePath=
This is the comma separated path (destination call is not included) that will be used on RF.
Callsign-SSIDs must conform to AX.25 address specifications.

SatgateMode=false
If true, the IGate will be receive only and pass all packets heard on RF to the server except for packets that have a path but have not been digipeated..

RFTransmit=false
If true, the IGate will gate to RF.

DontSendPositsToRF=false
If true, the IGate will obey RFTransmit but not transmit posit and status to RF (primarily for non-APRS IGates).

(M)Upstream=false

FullFeed=false
Set this to true only if your server is on a limited feed from the upstream server (not on a full feed). If false, APRSIGate creates a filter server command for APRSFilter to provide a limited feed to APRSIGate.

(M)ReadOnly=false

(M)LocalOnly=false

(M)ServerCommand=
This will be set based on gate-to-RF settings.

(M)FixedCommand=false

(M)LoginCommands=

KISSTNCIntf Properties

TNCPort=0
This is the TNC port (leave at zero for single port TNCs) to use.

DPRSIntf Properties

DupeDelay=10
Width in seconds of the sliding window that prevents streaming of positions to APRS.

IGate Block Properties

noGateCalls=RFONLY;NOGATE

(L)These are "digis" which are not allowed to be gated to APRS-IS. TCPIP & TCPXX are always included.

noGateISCalls=TCPXX

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

DigiDontGate=

(L)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.

IGate Local Network Properties

MaxLocalDigis=0

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=

(L)"Callsigns" used with the UI flood algorithm.
This is set to callsigns used in the N-n flood algorithm. SR is an example under the new APRS paradigm. 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.

UITraceCalls=

(L)"Callsigns" used with the UI trace algorithm.
This is set to callsigns used in the N-n trace algorithm. WIDE is an example under the original APRS specification. WIDEn-n as used today is using the UI trace algorithm. Some areas still use WIDEn-n in flood mode.

NOIDDigis=

(L)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 Properties

DirectCallPaths=

(L)Stations whose packets will be gated directly (not as third-party) to RF. Each semicolon separated entry is a comma delimited path to be used on RF. The format of each entry is fromCall,path (note the call the path is based on is the first call in the comma delimited list). Do not include the destination call. The destination call is passed unchanged from the originating station.

SpecialPaths=

(L)Paths which will be used for the RF header when packets from the specified stations are gated as third-party packets to RF. Each semicolon separated entry is a comma delimited path to be used on RF. The format of each entry is fromCall,path (note the call the path is based on is the first call in the comma delimited list). Do not include the destination call. The destination call is passed unchanged from the originating station. A trailing asterisk (*) may be used with the fromCall to indicate a prefix. You may have prefixes of different lengths (ABC*,path;ABCD*,otherpath) and matching will occur on the longest matching prefix.

IGate Pass Properties

GateCalls=

(L)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 recently heard on RF. A trailing asterisk (*) may be used indicate a prefix.

GatePosits=

(L)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. A trailing asterisk (*) may be used indicate a prefix.

GateUnprotos=

(L)This specifies packets with the specified destination calls are to be gated to RF. Packets with the specified destination calls will be gated to RF if they are not heard on RF. A trailing asterisk (*) may be used indicate a prefix.

GateObjects=

(L)This specifies object or item names which are to be gated to RF. A trailing asterisk (*) may be used indicate a prefix. To pass individual object names, object names must be 9 characters. Names may include an asterisk by using ~ instead of *. To include a name with an embedded semicolon, use a list file and put each object name on a separate line.

GateIGates=

(L)This specifies callsign-SSID's of ports-of-entry that 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 q construct to determine if the packet was gated to APRS-IS by the specified station.

Section 4 - Installation Instructions

APRSIGate.jar must be placed in the wherever specified in the ClassPath property. Normally it will be in the same directory as javAPRSSrvr and ClassPath=APRSIGate.jar Add the properties file to Clients and start javAPRSSrvr.

Section 5 – XML Status Page Description

General XML

```xml
<clientrcv>
<time>
<connect utc="1340910174957"/>
<lastlinein utc="1340929554871"/>
</time>
<upstream>
false
</upstream>
<readonly>
false
</readonly>
<login>
<callssid verified="true">
AE5PL-10
</callssid>
<software version="4.0.0">
APRSIGate
</software>
</login>
<rcvdfrom bytes="262366" lines="3422" packets="3412">
<udp bytes="0" lines="0" packets="0"/>
</rcvdfrom>
<clientxmt>
<upstream>
false
</upstream>
<sentto bytes="36829" lines="338" packets="338">
<lastlinems>
1340929483893
</lastlinems>
<udp bytes="0" lines="0" packets="0"/>
</sentto>
<xmtqueue depth="0" depthms="0"/>
<adjuncts>
<filteraprsclient>
<filter>
<![CDATA[filter b/K5PRK-S/K5TIT-S/W5EJL/AE5PL-OG/AE5PL e/K5TIT-A/K5TIT-B/K5TIT-
C/K5PRK-A/K5PRK-B p/FWD u/APWL2K o/EL-*/IRLP*]]>
</filter>
</filteraprsclient>
</adjuncts>
</clientxmt>
<stations direct="4" local="46"/>
<messagesgatedtorf>
0
</messagesgatedtorf>
</clientrcv>
```

Detail XML

```xml
<clientrcv>
<time>
<connect utc="1340910174957"/>
<lastlinein utc="1340929694974"/>
</time>
<class name="APRSIGate">
<package name="net.ae5pl.aprsigate" revision="b01" title="APRS IGate" version="4.0.0"/>
</class>
<messages callssids="0" unacked="0"/>
<upstream>
false
</upstream>
<readonly>
false
</readonly>
<login>
<callssid verified="true">
AE5PL-10
</callssid>
<software version="4.0.0">
APRSIGate
</software>
</login>
<rcvdfrom bytes="262968" lines="3430" packets="3420">
<udp bytes="0" lines="0" packets="0"/>
</rcvdfrom>
<clientxmt>
<class name="IGateXmt">
<package name="net.ae5pl.aprsigate" revision="b01" title="APRS IGate" version="4.0.0"/>
</class>
<upstream>
false
</upstream>
<sentto bytes="36829" lines="338" packets="338">
<lastlinems>
1340929483893
</lastlinems>
<udp bytes="0" lines="0" packets="0"/>
</sentto>
<xmtqueue depth="0" depthms="0"/>
<adjuncts>
<filteraprsclient>
<filter>
<![CDATA[filter b/K5PRK-S/K5TIT-S/W5EJL/AE5PL-OG/AE5PL e/K5TIT-A/K5TIT-B/K5TIT-C/K5PRK-A/K5PRK-B p/FWD u/APWL2K o/EL-*/IRLP*]]>
</filter>
<class name="FilterClient">
<package name="net.ae5pl.aprsfilter" revision="b01" title="APRS Server Filter Adjunct" version="4.0.0"/>
```

```xml
</class>
<passfilters>
<filterset>
<buddy>
K5PRK-S
</buddy>
<buddy>
K5TIT-S
</buddy>
<buddy>
AE5PL-OG
</buddy>
<buddy>
W5EJL
</buddy>
<buddy>
AE5PL
</buddy>
<buddyprefix>
FWD
</buddyprefix>
<objectprefix>
<![CDATA[EL-]]>
</objectprefix>
<objectprefix>
<![CDATA[IRLP]]>
</objectprefix>
<entry>
K5TIT-A
</entry>
<entry>
K5TIT-B
</entry>
<entry>
K5TIT-C
</entry>
<entry>
K5PRK-A
</entry>
<entry>
K5PRK-B
</entry>
<unproto>
APWL2K
</unproto>
</filterset>
</passfilters>
<rejfilters>
<filterset/>
</rejfilters>
</filteraprsclient>
```

```xml
</adjuncts>
</clientxmt>
<stations direct="4" local="44"/>
<messagesgatedtorf>
0
</messagesgatedtorf>
<serialinterface>
AGW
</serialinterface>
</clientrcv>
```