

# **APRS Filter User's Guide**

## **APRSFilter 4.3.3**

APRS Filter is Copyright © 2022 - 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.

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 Properties .....	3
javAPRSSrvr Properties .....	4
ServerAdjuncts= .....	4
FilterAPRS.Properties= .....	4
Filter Properties .....	5
MaxAreaFilterDegrees=90.0 .....	5
MaxRangeSizekm=9999 .....	5
useSoftReference=true .....	5
Client Properties .....	6
ServerAdjuncts=false .....	6
ServerCommand= .....	6
FixedCommand=false .....	6
LoginCommands= .....	6
Section 4 - Installation Instructions .....	7
Section 5 – Filter Commands .....	8
Section 6 – XML Status Page Description .....	10
General XML .....	10
Detail XML .....	11

## Section 1 - Introduction

APRS Filter was written to provide the amateur radio APRS community a basic server-side filtering capability. The original concept and implementation at javAPRSFilter was done as a server adjunct to the prior version of javAPRSSrvr. javAPRSFilter was written by Roger Bille SM5NRK to provide a method for attaching remote clients to a feed that is carrying less than a full APRS-IS feed. This was and is necessary as most clients are not designed to accept and process a full APRS-IS feed.

## Section 2 - Program Requirements and Description

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

APRS Filter is comprised of a number of classes which Java looks at as objects.

APRS Filter uses the PacketServerAdjInterface to monitor all packets to maintain a 24 hour position reference table, IGates, and non-posit wx stations. It also creates a client per that interface for every ServerAdjunct enabled client to process outbound packets. APRS Filter does NOT affect inbound (from the client) packets.

APRS Filter is controlled via the ServerCommand, FixedCommand, and LoginCommands in the client properties. If FixedCommand=false, the client can also set the filter via login, comment, or APRS message to SERVER.

## 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*

### **ServerAdjuncts=**

**(Deprecated)** This must include APRSFilter.jar.

### **FilterAPRS.Properties=**

This can point to a properties file with the filter properties (compatible with pre-4.3.0b04 configurations).

## ***Filter Properties***

If these are in the javaprssvr.properties file, they must carry the FilterAPRS. prefix (e.g. FilterAPRS.MaxAreaFilterDegrees).

### **MaxAreaFilterDegrees=90.0**

This is the maximum top-bottom and right-left distance in degrees. In other words, if top-bottom > MaxAreaFilterDegrees or right-left > MaxAreaFilterDegrees, the filter is ignored.

### **MaxRangeSizekm=9999**

This is the maximum range for all range filters including friend and me filters. The absolute maximum is 99999 based on parsing.

### **useSoftReference=true**

A SoftReference is held available by the garbage collector longer than a WeakReference. If you want faster dismissal of cached filters, set this to false.



## ***Client Properties***

### **ServerAdjuncts=false**

Set to true to enable APRS Filter on this client.

### **ServerCommand=**

This sets server adjunct command(s) for the client (may be overridden by client).

### **FixedCommand=false**

If true, the ServerCommand is forced. If false, enables LoginCommands.

### **LoginCommands=**

This is the file name containing station-specific server commands. This file is a properties file with the callsign-SSID as the property name and the server adjunct command(s) as the property value.

## Section 4 - Installation Instructions

APRSFilter.jar must be placed in the ServerAdjuncts property in the javaprssvr.properties file.. Normally it will be in the same directory as javAPRSSvr and ServerAdjuncts=APRSFilter.jar.

## Section 5 – Filter Commands

The following are the filter commands supported. Note these commands are additive (m/50 t/t means packets from stations within 50 km of my station AND all telemetry packets).

You can prevent the filter from passing certain packets by prefixing the filter parameter with a hyphen (-). This tells the filter to approve any packets that match the include filters **except** those that match the exclude filters. Standard port functionality such as messaging for IGates is not affected.

For instance, to get all stations within 200 km of me except stations with the prefix of CW, I would use:  
filter m/200 -p/CW

The filter command may be set as part of the login line, as an APRS message to SERVER, or as a separate comment line (#filter r/33/-97/200). The preferred method is to set the command as part of the login which is supported by most current APRS software.

Below are the available filters (asterisk (\*) wild cards are only allowed to indicate a prefix; highlighted => new with 4.0):

Parameter	Filter Type	Description
r/lat/lon/dist	Range filter	Pass posits and objects within dist km from lat/lon. lat and lon are signed <b>decimal</b> degrees, i.e. negative for West/South and positive for East/North. Up to 9 range filters can be defined at the same time to allow better coverage. Messages addressed to stations within the range are also passed.
p/aa/bb/cc...	Prefix filter	Pass traffic with fromCall that start with aa or bb or cc...
b/call1/call2...	Budlist filter	Pass all traffic from exact call: call1, call2, ... (* wild card allowed)
o/obj1/obj2...	Object filter	Pass all objects and items with the exact name of obj1, obj2, ... (* wild card allowed) (spaces not allowed) (  => / and ~ => *)
<b>os/obj1/obj2...</b>	<b>Strict Object filter</b>	<b>Pass all objects and items with the exact name of obj1, obj2, ... (* wild card allowed) (  =&gt; / and ~ =&gt; *) Objects are always 9 characters and Items are 3 to 9 characters. There can only be one o filter and that filter must be at the end of the line.</b>
t/poimqstunw t/poimqstuw/call/km	Type filter	Pass all traffic based on packet type. One or more types can be defined at the same time, t/otq is a valid definition.  p = Position packets o = Objects i = Items m = Message q = Query s = Status t = Telemetry u = User-defined n = NWS Weather & Weather Objects w = Weather  Note: The weather type filter also passes positions packets for positionless weather packets.  The second format allows putting a radius limit around "call" (station callsign-SSID or object name) for the requested station types.
s/pri/alt/over	Symbol filter	pri = symbols in primary table (  => /) alt = symbols in alternate table (  => /) over = overlay character (case sensitive)  For example: s/-> This will pass all House and Car symbols (primary table) s//# This will pass all Digi with or without overlay s//#/T This will pass all Digi with overlay of capital T

d/digi1/digi2...	Digipeater filter	The digipeater filter will pass all packets that have been digipeated by a particular station(s) (the station's call is in the path). This filter allows the * wildcard.
a/latN/lonW/latS/lonE	Area filter	The area filter works the same as rang filter but the filter is defined as a box of coordinates. The coordinates can also be seen as upper left coordinate and lower right. Lat/lon are <b>decimal</b> degrees. South and west are negative. Up to 9 area filters can be defined at the same time.
e/call1/call1/...	Entry station filter	This filter passes all packets with the specified callsign-SSID(s) immediately following the q construct. This allows filtering based on receiving IGate, etc. Supports * wildcard.
g/call1/call1/...	Group message filter	This filter passes all message packets with the specified callsign-SSID(s) as the addressee of the message. Supports * wildcard.
u/unproto1/unproto2/...	Unproto filter	This filter passes all packets with the specified destination callsign-SSID(s) (also known as the To call or unproto call). Supports * wildcard.
q/con/ana	q Construct filter	<p>q = q Construct command  con = list of q Construct to pass (case sensitive)  ana = analysis based on q Construct.</p> <p>l = Pass positions from IGATES identified by qAr or qAR.</p> <p>For example:  q/C Pass all traffic with qAC  q/rR Pass all traffic with qAr or qAR  q//l Pass all position packets from IGATES identified in other packets by qAr or qAR</p>
m/dist	My Range filter	This is the same as the range filter except that the center is defined as the last known position of the logged in client.
f/call/dist	Friend Range filter	This is the same as the range filter except that the center is defined as the last known position of call. Up to 9 friend filters can be defined at the same time.

## Section 6 – XML Status Page Description

### **General XML**

```
<filteraprs>
<igates msholdtime="3600000" size="1"/>
<nonpositweatherstations msholdtime="3600000" size="9"/>
<stationswithposits msholdtime="3600000" size="69"/>
</filteraprs>

<clientrcv>
...
<clientxmt>
...
<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>
...
</clientrcv>
```

## Detail XML

```
<filteraprs>
<class name="FilterAPRS">
<package name="net.ae5pl.aprsfilter" revision="b01" title="APRS Server Filter Adjunct"
version="4.0.0"/>
</class>
<igates msholdtime="3600000" size="1"/>
<nonpositweatherstations msholdtime="3600000" size="9"/>
<stationswithposit msholdtime="3600000" size="69"/>
</filteraprs>
<clientrcv>
...
<clientxmt>
...
<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"/>
</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>
</adjuncts>
...
</clientxmt>
...
</clientrcv>
```