

**javAPRSQRZ User's Guide**  
**3.0b01**

javAPRSQRZ is Copyright (c) 2007 - 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 Parameters .....	3
javAPRSSrvr Parameters .....	3
IGateAdjunct= .....	3
IGate General Parameters .....	3
IGateCall= .....	3
IGateStatusInterval=20 .....	3
QRZ.com Parameters.....	4
QRZUser= .....	4
QRZPwd= .....	4
AltQRZURI= .....	4
MaxAttempts=4.....	4
APRS Message Text Parameters.....	4
MsgInvalidReq=Invalid Request! .....	4
MsgInvalidCmd=% is an invalid command! .....	4
MsgSvcNotAvail=QRZ service not available!.....	4
MsgInvalidCall=% is an invalid callsign!.....	4
MsgNotFound=% was not found!.....	4
MsgErrorResp=Error looking up %! .....	4
Section 4 - Recommended Configurations .....	5
Section 5 - Installation Instructions .....	6
Section 6 – Operator Guide .....	7
Do a One Line Query .....	7
Do a Multi-Line Query .....	7
Section 7 – Status Page .....	8

## Section 1 - Introduction

javAPRSQRZ was written to provide the amateur radio APRS community a simple means to lookup callsigns on QRZ.com.

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

## Section 2 - Program Requirements and Description

javAPRSQRZ is designed to run on any OS with any recent Java Virtual Machine 1.1.4 or higher.

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

javAPRSQRZ monitors the Internet feed to determine if a station has sent it a message. If it receives a message, it responds depending on the message content.

When a message is received from APRS-IS which is properly formatted for processing as a lookup (see section 6), javAPRSQRZ will query [online.qrz.com](http://online.qrz.com) for the detail information on the requested callsign using XML. If the sysop does not have a subscription to [online.qrz.com](http://online.qrz.com), then javAPRSQRZ will default to accessing it via jFindu.net. The sysop can also define their own XML service as long as it is compliant with <http://online.qrz.com/specifications.html>.

javAPRSQRZ supports two commands: "whois" which returns a single line and "full" which returns all of the name and address information available on QRZ.

## 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 javAPRSQRZ.

#### ***IGate General Parameters***

#### **IGateCall=**

This is the callsign-SSID for javAPRSQRZ.

It should conform to AX.25 standards, be unique, and it must be different from javAPRSSrvr's userCall (the server's callsign-SSID). Do not use WHO-IS or WHO-15 which are in use.

#### **IGateStatusInterval=20**

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

This is needed to ensure reception of messages destined for the IGate. Do not modify unless absolutely necessary.

## ***QRZ.com Parameters***

### **QRZUser=**

This is the user name (callsign) for your QRZ subscription.  
Omit this parameter if you do not have a QRZ subscription.

### **QRZPwd=**

This is the password for your QRZ subscription.  
Omit this parameter if you do not have a QRZ subscription.

### **AltQRZURI=**

This can be set to a URL of an XML provider similar to QRZ.com subscription service.  
This URL will be used if QRZUser is blank or omitted. It will be appended with the callsign received in the message from the client.

### **MaxAttempts=4**

(R) This specifies how many tries (at one minute intervals) javAPRSQRZ will use to get an APRS message through to the recipient.

## ***APRS Message Text Parameters***

### **MsgInvalidReq=Invalid Request!**

(R) This is returned when the APRS message contains an invalid request.

### **MsgInvalidCmd=% is an invalid command!**

(R) This is returned when the % (% is a placeholder) command in the APRS message is invalid.

### **MsgSvcNotAvail=QRZ service not available!**

(R) This is returned when there is an error connecting to QRZ.

### **MsgInvalidCall=% is an invalid callsign!**

(R) This is returned when % is an invalid callsign (% is a placeholder for the callsign that was looked up).

### **MsgNotFound=% was not found!**

(R) This is returned when % (% is a placeholder for the callsign that was looked up) is not found.

### **MsgErrorResp=Error looking up %!**

(R) This is returned when there is an error looking up % (% is a placeholder for the callsign that was looked up).

## **Section 4 - Recommended Configurations**

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

## Section 5 - Installation Instructions

Add javAPRSQRZ.jar to the Java classpath. If using .NET javAPRSSvc, add javAPRSQRZ.dll to the javAPRSSvc directory. In both cases, set IGateAdjunct=javAPRSQRZ in javaprssvr.cfg.

## Section 6 – Operator Guide

This section describes how users interact with the server.

### ***Do a One Line Query***

Send an APRS message to IGateCall with the callsign as the only word in the message text.

```
AE5PL-10>APRS::WHO-IS :ae5pl
```

or

```
AE5PL-10>APRS::WHO-IS :w ae5pl
```

### ***Do a Multi-Line Query***

Send an APRS message to IGateCall with the letter F (case-insensitive) followed by a space and then the callsign.

```
AE5PL-10>APRS::WHO-IS :f ae5pl
```

## Section 7 – Status Page

IGate Adjunct	
<b>javAPRSQRZ 3.0b01</b>	<b>Copyright © 2007 - Peter Loveall AE5PL</b>
QRZ Lookup Server Callsign	IGateCall
Messages Received	Total messages to IGateCall received
Valid Commands	Valid "w" or "f" commands
Invalid Commands	Improper message format
Messages Sent	Number of total messages sent
Total APRS Messages Delivered	Number of messages acked
Total APRS Messages Not Delivered	Number of messages sent but not acked
Total Failed Lookups - Service Not Avail	Number of times QRZ.com not available
Total Failed Lookups - Invalid Callsign	Number of times callsign invalid
Total Failed Lookups - Callsign Not Found	Number of callsigns not found
Total Failed Lookups - Invalid Response	Number of times unknown resp from QRZ
Total Successful Lookups - W Command	Number of successful 1 line lookups
Total Successful Lookups - F Command	Number of successful 3 line lookups