

javAPRSCQSrvr User's Guide

1.0b01

javAPRSCQSrvr is Copyright (c) 2007 - 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 | 3 |
| IGateCall= | 3 |
| IGateStatusInterval=20 | 3 |
| CQ Server Parameters | 3 |
| CQMinSendTime=25 | 3 |
| CQMaxIdleTime=12 | 3 |
| Section 4 - Recommended Configurations | 4 |
| Section 5 - Installation Instructions | 5 |
| Section 6 – Operator Guide | 6 |
| To send a message to a group (and register with that group): | 6 |
| To unregister from a group: | 6 |
| Section 7 – Status Page | 7 |

Section 1 - Introduction

javAPRSCQSrvr was written to provide the amateur radio APRS community a simple means to call CQ to a known subset of APRS users.

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

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

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

javAPRSCQSRvr 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 (CQSRVR is used as IGateCall for these examples):

```
STN1>APRS::CQSRVR :CQ GROUPABC Hi everyone!{00
```

Will cause the following to be sent by javAPRSCQSRvr (STN2 already a member of the group):

```
CQSRVR>APJIC1,TCPIP*::STN1 :ack00
CQSRVR>APJIC1,TCPIP*::STN1 :1 messages sent to GROUPABC
STN1>APJIC1,qAR,CQSRVR::STN2 :Hi everyone!
```

Note that CQSRVR only sends unnumbered messages. It appears to STN2 that "Hi everyone!" came directly from STN1 which allows them to respond directly without relying on CQSRVR.

Once a CQ message is received by the javAPRSCQSRvr, that station will remain registered with that group for 12 hours after their last CQ message. A group will remain active as long as there is at least one message within the last 12 hours. STN1 can unregister from GROUPABC by sending the following message:

```
STN1>APRS::CQSRVR :U GROUPABC{01
```

CQSRVR will respond with an ack and a message indicating that they have been removed from GROUPABC.

The commands (CQ and U) and the group names are case insensitive and are converted to upper case by javAPRSCQSRvr.

Section 3 - Configuration Parameters

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

javAPRSSvr Parameters

IGateAdjunct=

This must be set to javAPRSCQ.

IGate General Parameters

IGateCall=

This is the callsign-SSID for javAPRSCQSvr.

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

IGateStatusInterval=20

(R)This specifies the status beacon rate in minutes

CQ Server Parameters

CQMinSendTime=25

(R)Minimum time between messages from one station to one group (minutes).

CQMaxIdleTime=12

(R)Maximum time in hours that a group or station may remain inactive before it is deleted.

Section 4 - Recommended Configurations

I recommend that all settings be left to default. CQSRVR is a reserved IGateCall so use any other valid, non-duplicate callsign-ssid.

Section 5 - Installation Instructions

You must include javAPRSCQsrvr.jar in your classpath if using Java. If using .NET, place javAPRSCQsrvr.dll in your javAPRSSvc directory.

Section 6 – Operator Guide

This section describes how users interact with the server.

To send a message to a group (and register with that group):

Send an APRS message to IGateCall with “CQ groupname message” as the message text.

```
AE5PL-10>APRS::CQSRVR :cq scouts Hi everyone!
```

The group name must be a single word and is case-insensitive (is converted to upper case by javAPRSCQSRvr).

To unregister from a group:

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

```
AE5PL-10>APRS::CQSRVR :u scouts
```

Section 7 – Status Page

| IGate Adjunct | |
|----------------------|----------------------------------------|
| javAPRSCQsrvr 1.0b01 | Copyright © 2007 - Peter Loveall AE5PL |
| CQ Server Callsign | IGateCall |
| | |
| Messages Received | Numbered messages received |
| Messages Sent | Unnumbered messages sent to stations |
| Valid Commands | Number of CQ and U messages received |
| Invalid Commands | Number of non-CQ and non-U messages |
| Active Groups | Number of groups |