

Whols User's Guide

javAPRSSrvr 4.3.2b06

WholsGate is Copyright © 2019 - 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 Properties	3
javAPRSSvr Properties	4
Clients=.....	4
WholsGate General Properties.....	5
Class=.....	5
ClassPath=	5
StationCall=	5
(M)Upstream=false	5
(M)FullFeed=false	5
(M)ReadOnly=false	5
(M)LocalOnly=false	5
(M)ServerCommand=	5
(M)FixedCommand=false	5
(M>LoginCommands=	5
(M)MessageHoldTime=-1	5
(M)LastHeardTime=-1	5
StatusText=	Error! Bookmark not defined.
LookupProviderClass=net.ae5pl.whoisgate.jFindunet	5
LookupProvider Interface:	5
HamInfo Class:.....	6
QRZcom Properties	8
QRZUser=	8
QRZPwd=	8
APRS Message Text Properties	9
MsgInvalidCmd=%1\$s is an invalid command!	9
MsgSvcNotAvail=QRZ service not available!	9
MsgNotFound=%1\$s was not found!	9
MsgErrorResp=Error looking up %1\$s.....	9
Section 4 - Recommended Configurations	10
Section 5 - Installation Instructions	11
Section 6 – Operator Guide	12
Do a One Line Query	12
Do a Multi-Line Query	12
Section 7 – XML Status Page	13
General XML	13
Detail XML	14

Section 1 - Introduction

WholsGate is written to provide the amateur radio APRS community a simple means to lookup callsigns on QRZ.com or other callsign lookup provider.

WholsGate extends `net.ae5pl.aprssrvr.ClientRcv`. It requires `javAPRSSrvr` to provide the network interface and do the packet parsing.

Section 2 - Program Requirements and Description

WholsGate is designed to run on any OS and Java VM supported by javAPRSSrvr.

WholsGate is comprised of a number of classes which Java looks at as objects. The main class is `net.ae5pl.whoisgate.WholGate`. This class is called at startup, sets parameters, and begins execution.

WholsGate 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), WholsGate will query the LookupProvider for the detail information on the requested callsign using XML. LookupProviders supplied with WholsGate are QRZcom and jFindunet. If the sysop does not have a data subscription to QRZ.com, they can access the QRZ.com database via jFindu.net using the jFindunet LookupProvider. The sysop can also define their own LookupProvider to populate a HamInfo object.

WholsGate supports two commands: "whois" which returns a single line and "full" which returns up to three lines of name and address information from the HamInfo object.

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.

javAPRSSrvr Properties

Clients=

(L)This must include the WholsGate properties file.

WholsGate General Properties

Class=

This must be set to net.ae5pl.whoisgate.WholsGate.

ClassPath=

(L)This must include WholsGate.jar.

StationCall=

This is the callsign-SSID for WholsGate.

It must conform to APRS-IS standards, be unique, and it must be different from serverCall (the server's callsign-SSID).

Do not use WHO-IS or WHO-15 which are reserved.

(M)Upstream=false

(M)FullFeed=false

(M)ReadOnly=false

(M)LocalOnly=false

(M)ServerCommand=

(M)FixedCommand=false

(M>LoginCommands=

(M)MessageHoldTime=-1

(M)LastHeardTime=-1

LookupProviderClass=net.ae5pl.whoisgate.jFindunet

This defines the LookupProvider to use. The other LookupProvider supplied with WholsGate is net.ae5pl.whoisgate.QRZcom.

The following is the java interface which must be implemented in any LookupProviderClass and **the LookupProvider must have a constructor of the format classname(ciProperties props) to initial LookupProvider specific properties.**

LookupProvider Interface:

```
package net.ae5pl.whoisgate;
```

```
import org.w3c.dom.*;
```

```
/**
```

```
*
```

```
* @author Pete AE5PL
```

```
*/
```

```
public interface LookupProvider
```

```
{
```

```
/**
```

```
* Retrieve info from provider.
```

```
*
```



```

* @param callsign Callsign to look up
* @return HamInfo obtained from provider
*/
public HamInfo parseHamInfo(String callsign) throws Exception;

/**
 * This generates an Element for the statusdoc
 * @param statusdoc
 * @param detailed
 * @return New Element for status document
 */
public Element getStatus(Document statusdoc, boolean detailed);
}

```

HamInfo Class:

```

package net.ae5pl.whoisgate;

/**
 *
 * @author Pete AE5PL
 */
public class HamInfo
{
    public String Callsign = null;
    public String Name = null;
    public String Addr1 = null;
    public String Addr2 = null;
    public String State = null;
    public String Zip = null;
    public String Country = null;
    public String PrevCallsign = null;
    public String LicenseClass = null;

    public HamInfo()
    {
    }

    @Override
    public String toString()
    {
        StringBuilder sb = new StringBuilder();
        if (Callsign != null)
            sb.append(Callsign).append("\r\n");
        if (LicenseClass != null)
            sb.append(LicenseClass).append("\r\n");
        if (Name != null)
            sb.append(Name).append("\r\n");
        if (Addr1 != null)
            sb.append(Addr1).append("\r\n");
        if (Addr2 != null)
            sb.append(Addr2).append("\r\n");
        if (State != null)
            sb.append(State).append("\r\n");
        if (Zip != null)
            sb.append(Zip).append("\r\n");
        if (Country != null)
            sb.append(Country).append("\r\n");
        if (PrevCallsign != null)

```

```
        sb.append(PrevCallsign).append("\r\n");  
    return sb.toString();  
}  
}
```

QRZcom Properties

QRZUser=

This is the user name (callsign) for your QRZ subscription.

QRZPwd=

This is the password for your QRZ subscription.

Caution: This password is in plain text in the properties file.

QRZEncPwd=

This is the password for your QRZ subscription. This is the password encrypted using the encpwd application.

APRS Message Text Properties

Messages, where indicated, use the `java.util.Formatter` method for inserting a parameter. In all of these cases, there is, at most, only 1 parameter (%1) formatted as a string (\$s).

MsgInvalidCmd=%1\$s is an invalid command!

This is sent to the requesting station when the command (%1 is a placeholder) in the APRS message is invalid.

MsgSvcNotAvail=QRZ service not available!

This is sent to the requesting station when there is an error connecting to QRZ.

MsgNotFound=%1\$s was not found!

This is sent to the requesting station when the callsign (%1 is a placeholder for the callsign that was looked up) is not found in the lookup provider database.

MsgErrorResp=Error looking up %1\$s

This is sent to the requesting station when there is an error looking up the callsign(%1 is a placeholder for the callsign that was looked up).

Section 4 - Recommended Configurations

Use only the settings necessary to properly communicate with your lookup provider.

Section 5 - Installation Instructions

You must include WholsGate.jar in the ClassPath property in your WholsGate properties file.

Set Clients= to point to your WholsGate properties file.

JAVA 9+ Note:

WholsGate requires javax.activation and javax.xml.bind to be installed. These are part of the java.ee module which will be removed from the Java SE distributions as of Java 11. Therefore, it is recommended that you download the latest release of JAXP from <https://github.com/javaee/activation> and the latest release of JAXB from <https://javaee.github.io/jaxb-v2/>. Then add the following entries to your Java startup (before javAPRSSrvr.jar call):

```
--add-modules java.activation,java.xml.bind --upgrade-module-path javax.activation.jar;javax.xml.bind.jar
```

Also add ;jaxb-core.jar;jaxb-impl.jar to the ClassPath property in your WholsGate properties file.

Section 6 – Operator Guide

This section describes how users interact with the server.

Do a One Line Query

Send a numbered APRS message to StationCall with the callsign as the only word in the message text.

AE5PL-10>APRS::WHO-IS :ae5pl{00

or

AE5PL-10>APRS::WHO-IS :w ae5pl{00

Do a Multi-Line Query

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

AE5PL-10>APRS::WHO-IS :f ae5pl{00

Section 7 – XML Status Page

General XML

```
<clientrcv>
<time>
<connect utc="1341494412278"/>
<lastlinein utc="1341506650885"/>
</time>
<upstream>
false
</upstream>
<readonly>
false
</readonly>
<login>
<callssid verified="true">
WHO-IS
</callssid>
<software version="4.0.0">
WholsGate
</software>
</login>
<rcvdfrom bytes="304" lines="4" packets="4">
<udp bytes="0" lines="0" packets="0"/>
</rcvdfrom>
<clientxmt>
<upstream>
false
</upstream>
<sentto bytes="434" lines="8" packets="8">
<lastlinems>
1341506651448
</lastlinems>
<udp bytes="0" lines="0" packets="0"/>
</sentto>
<xmtqueue depth="0" depthms="0"/>
</clientxmt>
<whois>
<invalidcommands>
2
</invalidcommands>
<lookups found="2" fullfound="0" invalidresponses="0" notavailable="0" notfound="0">
<qrzcom/>
</lookups>
</whois>
</clientrcv>
```


Detail XML

```
<clientrcv>
<time>
<connect utc="1341494412278"/>
<lastlinein utc="1341506650885"/>
</time>
<class name="WholsGate">
<package name="net.ae5pl.whoisgate" revision="b01" title="QRZ.com Lookup Client" version="4.0.0"/>
</class>
<messages callssids="3" unacked="0"/>
<upstream>
false
</upstream>
<readonly>
false
</readonly>
<login>
<callssid verified="true">
WHO-IS
</callssid>
<software version="4.0.0">
WholsGate
</software>
</login>
<rcvdfrom bytes="304" lines="4" packets="4">
<udp bytes="0" lines="0" packets="0"/>
</rcvdfrom>
<clientxmt>
<class name="WholsXmt">
<package name="net.ae5pl.whoisgate" revision="b01" title="QRZ.com Lookup Client" version="4.0.0"/>
</class>
<upstream>
false
</upstream>
<sentto bytes="434" lines="8" packets="8">
<lastlinems>
1341506651448
</lastlinems>
<udp bytes="0" lines="0" packets="0"/>
</sentto>
<xmtqueue depth="0" depthms="0"/>
</clientxmt>
<whois>
<invalidcommands>
2
</invalidcommands>
<lookups found="2" fullfound="0" invalidresponses="0" notavailable="0" notfound="0">
<qrzcom server="subscription"/>
</lookups>
</whois>
</clientrcv>
```