

## Overview

This program is designed to monitor the JS8Call application for various received messages, examine the messages, then pass on the relevant information to mapping application(s), so the contact can be automatically plotted.

## Prerequisites

Python 2.7 or higher - Available for Windows, Mac, Linux, Raspberry Pi OS

## Mapping Applications Supported

N1MM+ with County Mapper - Available for Windows.

GridTracker - Available for Windows, Mac, Linux, Raspberry Pi OS

GeoChron Digital Displays - OS Independent

Yet Another APRS Client (YAAC) - Available for Windows, Mac, Linux, Raspberry Pi OS

Any mapping application that can import ADIF files.

## Upgrading From a Previous Version

To upgrade your previous installation, start by locating the folder on your computer where you installed JS8Call Monitor.

Rename the config file to config.old

Unzip the contents of the js8call\_monitor.zip file to a suitable place on your computer.

Copy the js8call\_monitor.py and config files into your installation folder, replacing the old files.

Using your favorite text editor, edit the config file for your setup. You can reference the config.old file for your previous settings, and refer to the section below on any new settings.

If you are upgrading from version 0.16 or older of JS8Call Monitor you will need to replace the contents of the pywsjtx folder by following the instructions below under new installation. The pywsjtx python module has been updated in release 0.17 to include additional functionality to support GridTracker live events.

## New Installation

Unzip the contents of the js8call\_monitor.zip file to a suitable place on your computer.

Copy the entire pywsjtx folder and its' contents to the 'Lib' directory of your Python installation.

Windows: On Windows, the Lib directory is typically named "Lib" and is located beneath the directory where Python is installed. Example c:\program files\python27\Lib

Linux and Raspberry PI: On these systems the exact name of the Lib directory will depend on the version of Python installed on your system. For example;

```
/usr/lib/python2.7  
/usr/lib/python3.8
```

To copy the pywsjtx folder on Pi or Linux use the following command, substituting the correct name for your version of Python. You can find out what versions of Python are on your system by doing an "ls /usr/lib" and looking for directories named pythonX.X.

```
"sudo cp -r pywsjtx /usr/lib/python3.8"
```

If your system has several versions of Python installed, repeat the command for each version of Python.

Using your favorite text editor, edit the config file for your setup.

- The **STATION** section contains various default settings that are sent to the loggers to identify your station in log entries. If you plan on mapping only data related to your station, the OPERATOR field needs to be set to your FCC callsign.
- The **JS8Call** section contains the details of how the JS8Call application will connect to JS8Call Monitor.
- The **AUTH** section contains the authorization details for clients that connect to JS8Call Monitor. This includes any automation scripts.

**enabled** if set to true, specifies that clients that connect to this instance of JS8Call Monitor must authenticate before they can execute any commands.

**token** contains the authentication token (password).

- The **N1MM** section contains the details of how to connect to the N1MM+ logger. If not using N1MM+ set the enabled flag to false.

- The **GRIDTRACKER** section contains the details of how to connect to GridTracker. If not using GridTracker set the enabled flag to false. If running both N1MM+ and GridTracker on the same computer, choose a unique port for GridTracker.
- The **GEOSERVER** section contains the details of how to connect to the GeoServer application for sending messages and spot reports to a GeoChron digital display. If not using GeoServer set the enabled\_msg and enabled\_spot flags to false. This feature requires the installation of an additional Python script. Further information on setting up this feature is covered in a separate instruction.
- The **YAAC** section contains the details of the file, to log spot reports for the Yet Another APRS Client (YAAC) program. If not using YAAC set the enabled flag to false. Set the logfile name to something suitable. If you do not include the full path in the filename, the YAAC csv file will be placed in the same folder where you installed JS8Call Monitor. This file is in the same format produced by the JSMapper program, and is typically forwarded by command stations at the end of an exercise to National.

Note: YAAC mapping is not done in real time. You will need to import the YAAC csv file into YAAC through the File > Load > APRS Packets dialog in YAAC.

- The **ADIF** section contains the details of the file, to log spot reports in ADIF format. If not using ADIF set the enabled flag to false. Set the logfile name to something suitable. If you do not include the full path in the filename, the ADIF file will be placed in the same folder where you installed JS8Call Monitor.

Note: ADIF mapping is not done in real time. You will need to import the ADIF file into your compatible mapping program. This file is in the same format produced for the GridTracker integration, and provides an alternate method for forwarding spot reports by command stations at the end of an exercise to National.

- The **SMTP** section contains various settings for connecting to an email server, to allow for the sending of email alerts and texts.

**enabled** if set to true, specifies this feature should be turned on. If enabled, when INFO and STATUS messages are received, a notification will be sent to the specified mailto address.

**host** should be set to the IP address or FQDN of your email server.

**port** should be set to the port of your email server that accepts SMTP connections. This is typically port 25.

**user** and **pass** should be completed with your account credentials if your email server requires authentication. If your email server does not require authentication, leave these fields blank.

**mailfrom** should be set to the email address that will show as the email address the notification originated from.

**mailto** should be set to the email address that the notification should be sent to. If you would like the notification sent as a text message, then enter the sms address for your mobile phone. For most carriers, this address typically takes the form of <cell phone number>@<mobile providers sms to email gateway>.com

- The **GREEN**, **YELLOW**, **RED**, **UNKNOWN**, **NOPIR**, and **NOINFO** sections are used to search for tags in the INFO messages. These messages are sent in response to an INFO? request in JS8Call. If a match is found, the 'map' parameter specifies the ADIF tag and value that should be sent to the logger to indicate a match. The 'color' parameter specifies the color the spot should be set to in the GeoServer integration. The 'aprs' parameter specifies the symbol that should be associated with the spot report in YAAC. If the INFO message is blank (no data) then the corresponding tag is updated from the NOINFO section. If the INFO message is present but no matches are found, then the corresponding tag is updated from the NOPIR section.
- The **STATUS** section contains the values sent to the various loggers when a STATUS message is received.
- The **GRIDS** section contains various setting that determine what is mapped.

**frominfo** if set to true, specifies the program should look for grid squares in INFO messages. These INFO messages are configured by each station in their JS8Call configuration and sent in response to an INFO? request. The program is expecting the INFO line be formatted in a specific way, namely the grid square be at the beginning of the INFO line, and be followed by a semi-colon, with no spaces  
I.E. EM73;

**gridlength** specifies the maximum characters sent in the GRIDSQUARES tag to the loggers. For example, a value of 4 means the resolution of GRIDSQUARES is limited to the first 4 characters of the GRIDSQUARES received from JS8Call (I.E. EM73).

**map\_all** if set to true, will map SNR reports received for all stations. If set to false it will limit mapping to SNR reports received for the callsign specified in the OPERATOR field contained in the STATION section. This affects map\_snr messages (see below).

**map\_cq** if set to true, will map messages received containing "CQ".

**map\_heartbeat** if set to true, will map messages containing '@HB' or 'HEARTBEAT'.

**map\_info** if set to true, will map INFO messages.

**map\_status** if set to true, will map STATUS messages.

**map\_log** if set to true, will cause data from manually logged QSOs in JS8Call to be sent to the loggers. This is independent of the N1MM and N3FJP logger settings in JS8Call configuration. If you will be using the LogIt Windows client application, this parameter should be set to true.

**map\_qso** if set to true, will map all other messages that do not fit within one of the other categories of messages.

**map\_snr** if set to true, will map messages containing "SNR", indicating the message contains a signal report.

If map\_all is set to true, then signal reports for all stations heard will be mapped.

If map\_all is set to false, then only signal reports where the callsign contained in the OPERATOR field in the STATION section, mentioned above, will be mapped.

**auto\_close** if set to true, will cause JSC8CALL Monitor to shut down and exit when it receives a shutdown event from JS8Call. If using JS8Call Monitor with multiple instances of JS8Call, this parameter should be set to false.

- The **DATABASES** section contains the settings for various offline and online resources for resolving missing grid squares in received messages. Grid squares are required for the mapping applications to be able to plot points on the map for the received messages. Note: If the grid square was resolved using one of the following methods, then the following symbol will be added to the beginning of the comments field: [!]

**location** should be set to the full path where the offline database files are located. The path should end with a slash I.E. c:\ham0\. If this field is left blank, then JS8Call Monitor will look for the database files in the same directory where it is installed.

Offline lookups are used to resolve missing grid squares, by searching database files located on the same computer as JS8Call Monitor. This is useful for when internet connectivity is not available. The database files are prepared by using a separate utility program. Further information on the utility is covered in a separate instruction.

**FccData** if set to true, will search a database extracted from FCC callsign data.

**HamCallCD** if set to true, will search a database extracted from HamCall CD data.

**RacCD** if set to true, will search a database extracted from Radio Amateur Callbook CD data.

**LocalDB** if set to true, will search the local database. Further information on this feature is covered in a separate instruction.

## JS8Call Monitor

**LocalDB\_learn** if set to true, will save the results from successful online grid square lookups to a local database, so that subsequent lookups for the same callsign, can be resolved offline.

**Collect\_Rejects** if set to true, will capture the callsigns that were unable to be resolved to a grid square location to a separate file for further processing.

Online lookups are used to resolve missing grid squares by searching various online databases. To use these services, internet connectivity is required from the computer where you are running the JS8Call Monitor program.

**Callook** if set to true will attempt lookup of missing grid squares via the callook.info website.

**HamCallOnline** if set to true will attempt lookup of missing grid squares via the hamcall.net website. HCusername and HCpassword contain you login credentials.

- The **DEBUG** section has various flags for debugging the program. See the section on debugging for an explanation of these parameters.

Save your changes.

Setup JS8Call and GridTracker (more below).

### Running the program

Run the program from Python. This can be done by any of the following methods;

- a) from the OS command line "python js8call\_monitor.py"
- b) by right clicking in Windows and select Open with Python

When you first start the program, if it complains about missing modules, you can get the missing ones by running "python -m pip install <module>" from an OS command prompt.

Note: If using Python 3 substitute "python3" for "python" in the above commands.

Start the GridTracker program.

Start the N1MM+ logger (optional).

Start the County Mapper program (optional).

As JS8Call messages are received and processed, the map(s) should be automatically updated with the Grid Squares, based on your settings in the config file.

## JS8Call Setup

The screenshot shows the 'Settings' window for JS8Call, with the 'Reporting' tab selected. The 'API' section is expanded, showing configuration options for UDP and TCP servers, and two logging options (N3FJP and N1MM). The 'Logging' section at the top has a text field for 'Operator Callsign' and a list box for 'Never send spotting reports from these callsigns'. The 'API' section includes a checkbox for 'Allow setting station information', and fields for 'UDP Server Hostname' (127.0.0.1), 'UDP Server Port' (2217), 'TCP Server Hostname' (127.0.0.1), 'TCP Server Port' (2442), and 'TCP Max Connections' (1). Checkboxes for 'Enable UDP Server API' and 'Accept UDP Requests' are checked. The 'N3FJP Logger' and 'N1MM Logger' sections each have a checkbox to 'Enable sending logged contacts' and fields for 'Server' and 'Server Port'.

Section	Option	Value	Status
Logging	Operator Callsign (if different than Station Callsign):		
	Never send spotting reports from these callsigns (comma separated):		
API	Allow setting station information (Grid, Info, Status, etc) from the APIs		<input type="checkbox"/>
	UDP Server Hostname:	127.0.0.1	<input checked="" type="checkbox"/> Enable UDP Server API
	UDP Server Port:	2217	<input checked="" type="checkbox"/> Accept UDP Requests
	TCP Server Hostname:	127.0.0.1	<input type="checkbox"/> Enable TCP Server API
	TCP Server Port:	2442	<input type="checkbox"/> Accept TCP Requests
	TCP Max Connections:	1	
N3FJP Logger	Enable sending logged contacts to N3FJP logging software		<input type="checkbox"/>
	N3FJP Server:	127.0.0.1	
	N3FJP Server Port:	1100	
N1MM Logger	Enable sending logged contacts to N1MM logging software		<input type="checkbox"/>
	N1MM Server:	127.0.0.1	
	N1MM Server Port:	2237	

In JS8Call under Settings > Reporting in the API section, set the UDP Server Hostname and Port information. If JS8Call and JS8Call Monitor will be running on the same computer you can use 127.0.0.1 for the hostname. If they will be running on different computers, then the hostname needs to be set to the IP address of where you are running the JS8Call Monitor program. The port can be any value that is not in use. The default is 2217.

The settings in the [JS8Call] section of the config file need to match what you set here.

Check the Enable UDP server API and Accept UDP requests boxes. You do not need to setup the TCP server information, or check the box for Allow Setting Station Information.

Optional: Configure the N3FJP Logger and/or N1MM Logger section to send logged contacts from JS8Call to your favorite logger directly. These two sections are used by the JS8Call program for contact logging and are not used by the JS8Call Monitor program.

Click OK to save your changes.

## N1MM+ Setup

N1MM+ is open source software, and can be found at: <https://n1mmwp.hamdocs.com>. If you haven't already, download and run the setup to install N1MM+

N1MM+ is used as a passthrough to County Mapper, and only requires the minimal configuration as described below.

The screenshot shows the 'Configurer' window with the 'WSJT/JTDX Setup' tab selected. The window has a blue title bar and a standard Windows-style interface. At the top, there are tabs for 'Digital Modes', 'Other', 'Winkey', 'Mode Control', 'Antennas', 'Score Reporting', 'Broadcast Data', 'Audio', and 'WSJT/JTDX Setup'. Below the tabs, a message states: 'N1MM+ Logger needs to be restarted for changes made below to take effect.' The main content area is divided into several sections: 'WSJT-X and JTDX UDP Settings' with sub-sections for 'Radio #1 Settings' and 'Radio #2 Settings'; 'JTDX / Others TCP Settings' with sub-sections for 'Radio #1 Settings' and 'Radio #2 Settings'; 'Path to WSJT/JTDX' with fields for 'WSJT/JTDX Path Used for S01V,S02V mode and Radio1 in S02R' and 'WSJT/JTDX Path Used for S02R Radio 2'; and 'Auto Load the WSJT Decode List Window when WSJT-X/JTDX Loads.' with checkboxes for 'Radio #1' and 'Radio #2'. At the bottom are 'OK', 'Cancel', and 'Help' buttons.

**Configurer**

Digital Modes Other Winkey Mode Control Antennas Score Reporting Broadcast Data Audio **WSJT/JTDX Setup**

N1MM+ Logger needs to be restarted for changes made below to take effect.

**WSJT-X and JTDX UDP Settings**

WSJT and JTDX UDP connection settings. IP Address and port must match each programs settings. This allows UDP message communications to take place, usually done on port 2237. Logging from other programs can also take place, usually done on port 2333. (Radio #1 Default: 2237)

Radio #1 Settings			Radio #2 Settings		
Enable	IP Address	UDP Port	Enable	IP Address	UDP Port
<input checked="" type="checkbox"/> Enable	127.0.0.1	2237	<input type="checkbox"/> Enable	127.0.0.1	2239

**JTDX / Others TCP Settings**

Sets the IP Address and port that an external program can connect to N1MM+ via TCP Port for logging purposes. The Default port for JTDX is 52001. (Radio #1 Default: 52001 - Radio #2 Default: 52006)

Radio #1 Settings			Radio #2 Settings		
Enable	IP Address	TCP Port	Enable	IP Address	TCP Port
<input checked="" type="checkbox"/> Enable	127.0.0.1	52001	<input type="checkbox"/> Enable	127.0.0.1	52006

**Path to WSJT/JTDX**

WSJT/JTDX Path Used for S01V,S02V mode and Radio1 in S02R.

Not Set

WSJT/JTDX Path Used for S02R Radio 2

**Command Line Params**

Not Set

**Command Line Params**

Not Set

**Auto Load the WSJT Decode List Window when WSJT-X/JTDX Loads.**

Radio #1 ☐ Enable Radio #2 ☐ Enable

Under Config > Configure Ports, Mode Control, Winkey, Etc > WSJT/JTDX setup > under UDP settings, check the enable box under the Radio #1 settings. If N1MM+ and JS8Call Monitor will be running on the same computer you can use 127.0.0.1 for the IP address. If they will be running on different computers, then the IP address needs to be set to the IP address of where you are running N1MM+. The port can be any value that is not in use. The default is 2237.

The settings in the [N1MM] section of the config file need to match what you set here.



## JS8Call Monitor

The screenshot shows the 'N1MM+ Configurer' window with the 'Broadcast Data' tab selected. The window has a blue title bar and a standard Windows-style interface. The 'Broadcast Data' tab is highlighted in orange. Below the tabs, there is a text box with instructions: 'Select the type of data you wish to broadcast, and the the IP Address(es) and port(s) for the receiver(s) of the data. Use 127.0.0.1 for the local machine. Use 12060 as the port unless the receiving application requires a different port. 255 in the low order octet will broadcast to your current subnet.' Below this, there is a table with two columns: 'Type of data' and 'IP Addr:Port IP Addr:Port...'. The table contains several rows with checkboxes and text input fields. The 'Application Info' and 'Contacts' checkboxes are checked. The 'IP Addr:Port' for 'Application Info' is '10.10.10.32:12060'. The 'IP Addr:Port' for 'Contacts' is '10.10.10.32:12060'. The 'All Computers' checkbox is unchecked. The 'Spots' checkbox is unchecked. The 'Rotor' checkbox is unchecked. The 'Score' checkbox is unchecked. The 'External Callsign Lookup' checkbox is unchecked. At the bottom of the window, there are three buttons: 'OK', 'Cancel', and 'Help'.

Type of data	IP Addr:Port IP Addr:Port...
<input checked="" type="checkbox"/> Application Info	10.10.10.32:12060
<input type="checkbox"/> Radio	127.0.0.1:12060
<input checked="" type="checkbox"/> Contacts <input type="checkbox"/> All Computers	10.10.10.32:12060
<input type="checkbox"/> Spots	127.0.0.1:12060
<input type="checkbox"/> Rotor	127.0.0.1:12040
<input type="checkbox"/> Score	127.0.0.1:12060
<input type="checkbox"/> External Callsign Lookup	127.0.0.1:12060

To setup for County Mapper you will also need to configure the Broadcast Data tab as well. Check the Application Info and Contacts check boxes. If N1MM+ and County Mapper will be running on the same computer you can use 127.0.0.1 for the IP address. If they will be running on different computers, then the IP address needs to be set to the IP address of where you are running County Mapper. The port can be any value that is not in use. The default is 12060.

The settings in County Mapper need to match what you set here.

Click OK to save your changes.

test.s3db

**New log For: ARRL 222 MHZ Up Distance Contest**

Log Type

Start Date

Use Up/Down cursor keys to see long description above.

Total number of contacts in database 1898

Contest

**Category**

Operator

Band

Power

Mode

Overlay

Station

Assisted

Transmitter

Sent Exchange  Omit RST. E.g. CQ/WW: 05 SS: A 56 EMA

Operators

Soapbox

Note - the program does not validate categories. Check the contest rules for valid categories.

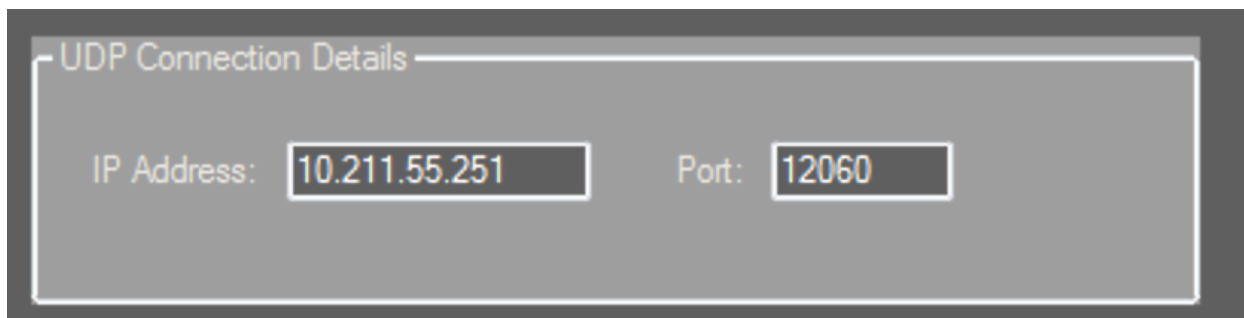
State for Log Type QSOPARTY

Time Category

To enable N1MM+ to feed data to County Mapper, you will need to setup a contest. In either a current or new database file, create a new log in database. For FTX exercises I have been setting ARRL-222 as the log type and taking the rest of the settings as default.

## County-Hunting Mapper Setup

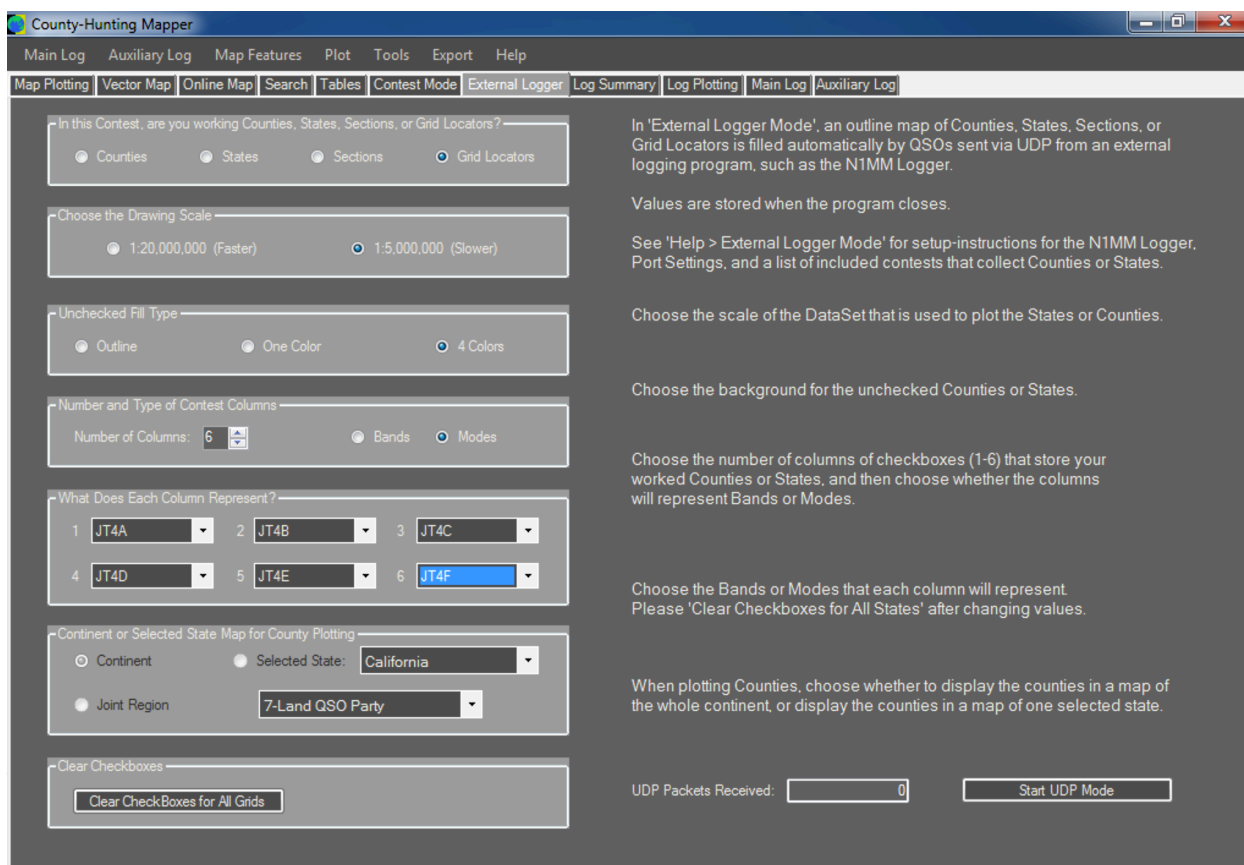
County-Hunting Mapper is commercial software. A standard license is around \$15.00 and offers a free 30 day trial. The software can be found at: <https://www.mapability.com/ei8ic/chm/index.php>. If you haven't already, download and run the setup to install County-Hunter Mapper.



Under help, select Options. Fill out the user Details tab. If N1MM and County Mapper will be running on the same computer you can use 127.0.0.1 for the IP address. If they will be running on different computers, then the IP address needs to be set to the IP address of where you are running County Mapper. The port can be any value that is not in use. The default is 12060.

The settings in N1MM need to match what you set here.

Save your changes.



## JS8Call Monitor

Click on the External Logger tab. For FTX exercises these are the settings I use;

Grid Locators

Scale 1:5,000,000

4 Colors

Modes and 6 Columns

Columns Represent

1-JT4A

2-JT4B

3-JT4C

4-JT4D

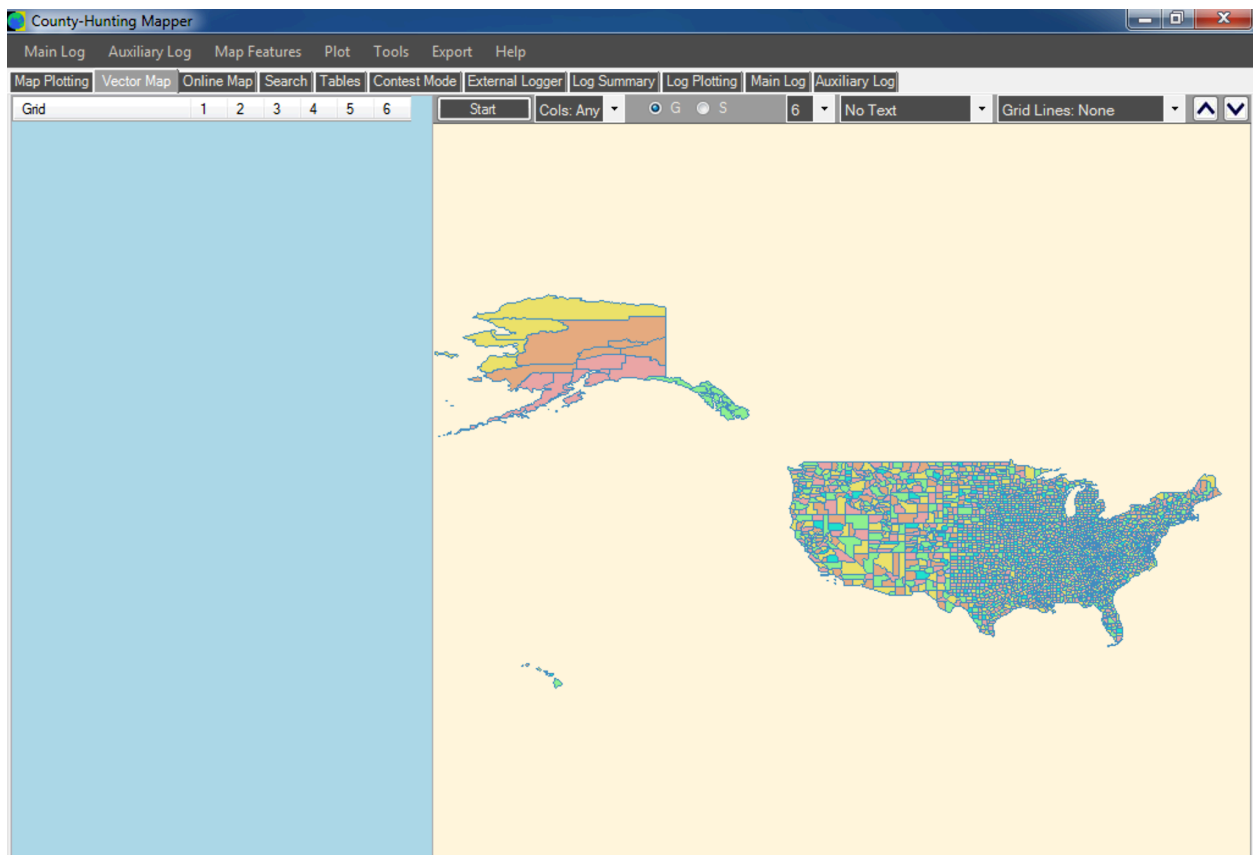
5-JT4E

6-JT4F

Optional: Click Clear checkboxes for all grids

Click Start UDP Mode.

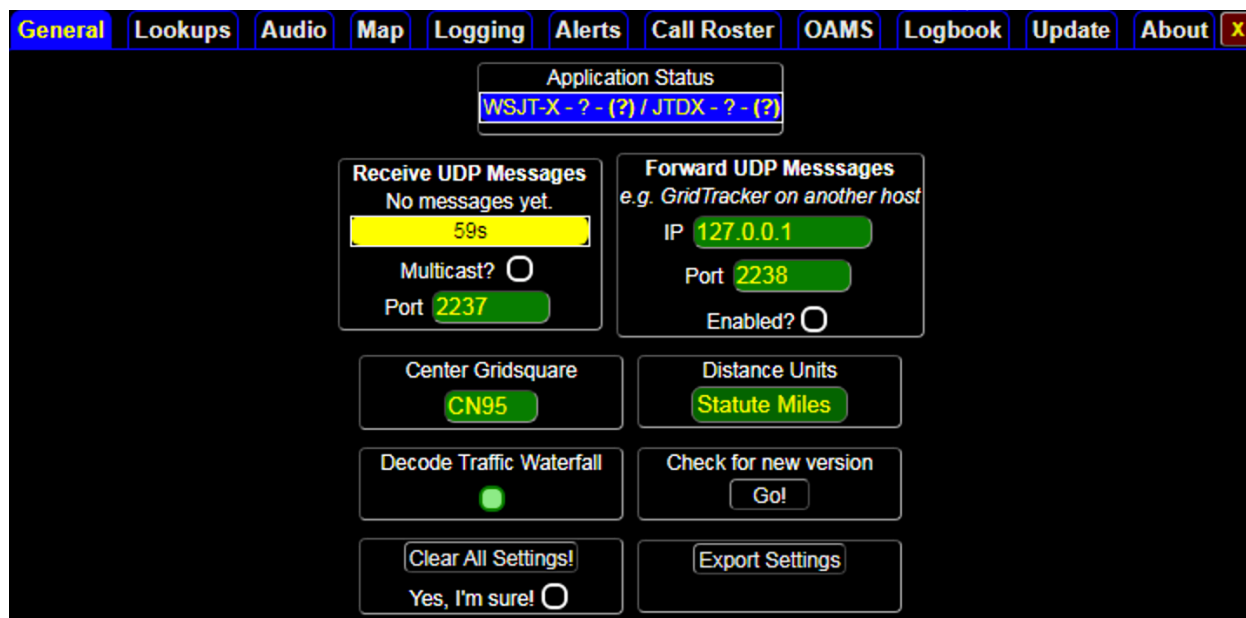
County Mapper should now switch to the Vector Map tab.



On the vector map select "G" instead of "S" in the tool bar and click Start

## GridTracker Setup

GridTracker is shareware software and can be found at: <https://gridtracker.org>  
If you haven't already, download and run the setup to install GridTracker.



Click on the gear icon for setup. In the General tab set the port in the receive UDP Messages box. The port can be any value that is not in use. The default is 2237.

Click the red X in the corner to save your settings.

The settings in the [GRIDTRACKER] section of the config file need to match what you set here.

A note about GridTracker. It was developed against the rich messaging interfaces of WSJT-X. JS8Call on the other hand provides very limited data in its' API interface. As a result, some of the more advanced features of GridTracker may not work as expected. I.E GridTracker will not display anything until after the first JS8Call activity that generates a status update occurs.

## How It Works

This explanation assumes the default configuration.

JS8Call Monitor, monitors the messages received from JS8Call via its' API interface. When it gets a message, it extracts what data it can from the message. If an INFO message is found, it is scanned for the keywords contained in the config file. If a keyword is located, the associated ADIF tag substitution is performed. JS8Call Monitor then "logs" the contact to the associated logger, N1MM and/or GridTracker by sending a set of ADIF fields containing data extracted from the JS8Call message.

In the case of GridTracker, this logged contact is then displayed/logged.

In the case of N1MM, the contact is logged, then "broadcast" to County Mapper, where the grid is displayed, and the appropriate column is checked based on the mode tag.

So to bring this together, consider the following;

From the transmitting stations JS8Call config;

INFO = EM73;PIR1=G

From the receiving stations JS8Call Monitor config;

```
[GREEN]
search = PIR1=G
map = MODE:JT4A
color = #00f900
aprs = ['\','W']
```

When a JS8Call message containing an INFO line is received, it is scanned for matches against the 'search' parameters in the config file.

If the JS8Call message contained an INFO line with the tag "PIR1=G", it would match the [GREEN] section in the config file. The 'map' parameter is then read for the ADIF tag substitution instructions. The result would be, the MODE tag would be set to JT4A.

JS8Call Monitor would then send the contact to N1MM, where it is logged and broadcast to County Mapper, who in turn displays the grid square EM73 and puts a check in column 1. Column 1 in this example would represent all grids reporting a (G)reen status for PIR1.

Hopefully this is enough information to get you started for the next FTX exercise.

## Debugging

In the config file there are three parameters in the DEBUG section.

**consolelevel** is set to a value between 0 and 10, and is used to determine the verbosity of the messages written to the console window running the program.

**logfilelevel** is set to a value between 0 and 10, and is used to determine the verbosity of the messages written to the logfile.

**logfile** specifies the path and file name of the logfile where the debugging information is written.

The settings for the levels is described below.

0 - Logging is disabled

1 - LOG\_SMDR, station message detail records. A short transaction summary designed for display in the console window, or consumption by your favorite log monitoring tool.

2 - LOG\_SEVERE, severe errors that require program exit.

3 - LOG\_ERROR, error messages that can't be recovered from but the program can continue to run.

4 - LOG\_WARNING, recoverable problem that you should be notified about (e.g., invalid value in a configuration file, so you fell back to the default).

5 - LOG\_INFO, informational messages.

6 - LOG\_ENTRY, log entry and exit to all functions.

7 - LOG\_PARM, log entry and exit to all functions with parameters passed and values returned (including global effects if any).

8 - LOG\_DEBUG, general debugging messages, basically useful information that can be output on a single line.

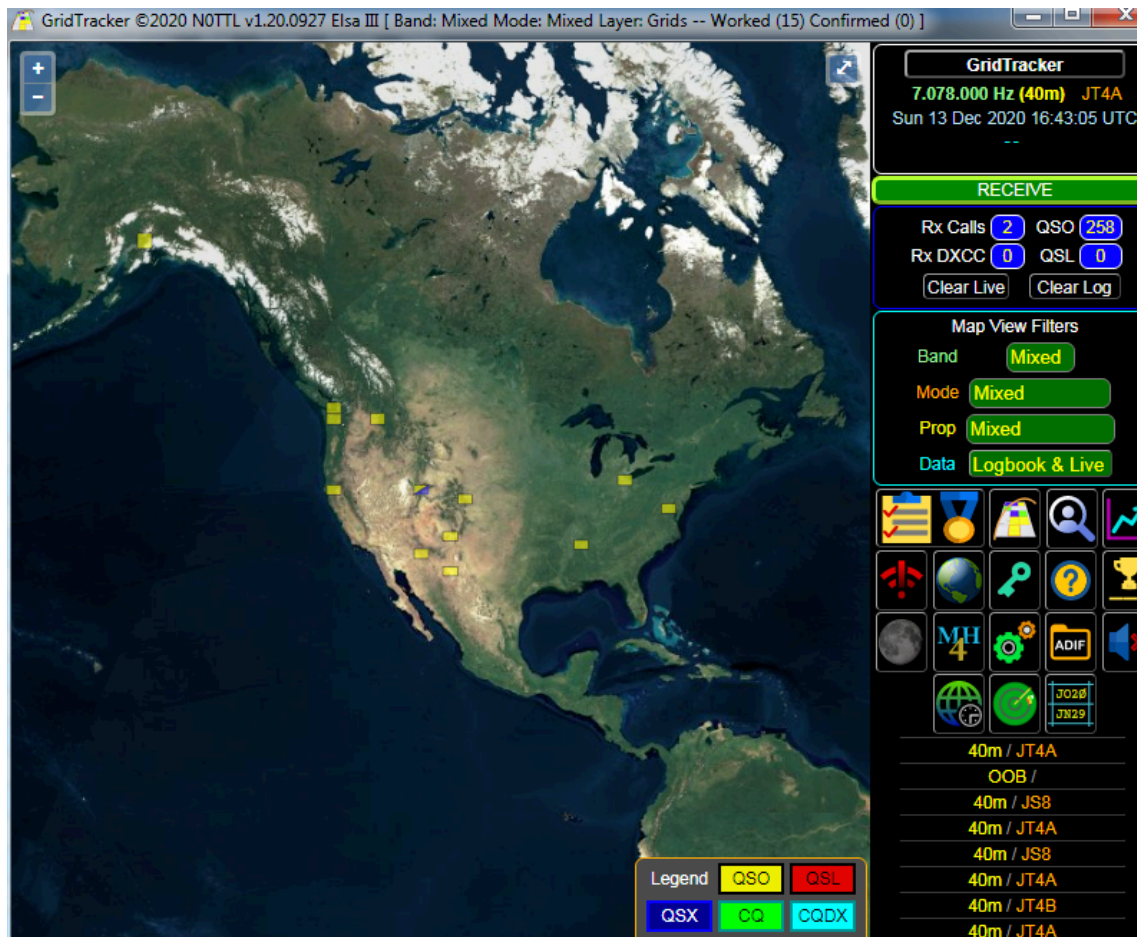
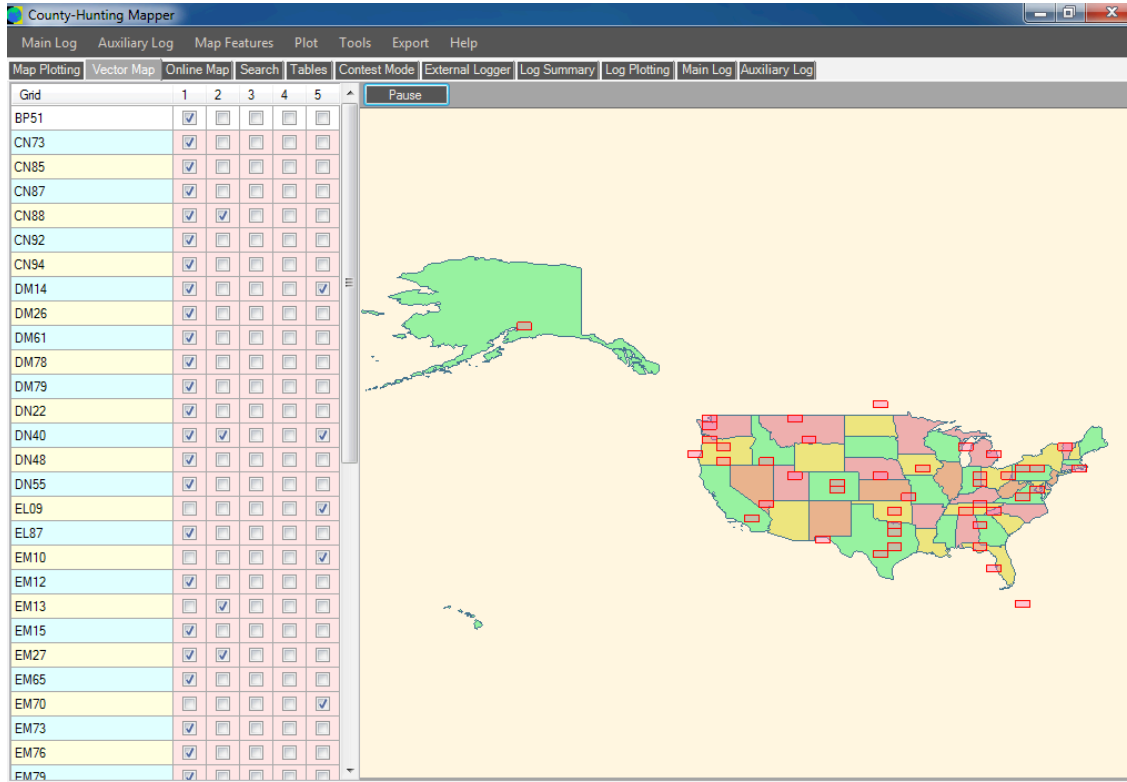
9 - LOG\_HIDEBUG, far more detailed debugging messages such as hex dumps of buffers.

10 - EXPERIMENTAL, used to implement experimental functions.

Each level also logs messages in 'lower' levels.

# JS8Call Monitor

## Sample Map Screen Shots





## Attribution

This program was inspired by the wsjtx\_to\_n3fjp adapter written by Dave Slotter and is available here: [https://github.com/dslotter/wsjtx\\_to\\_n3fjp](https://github.com/dslotter/wsjtx_to_n3fjp)

This program uses the py-wsjtx Python library written by Brian Moran, to interpret and generate UDP packets to communicate with GridTracker, and is licensed under the MIT License. A copy of the license is included in the library folder. The version in this distribution has been modified for compatibility with Python 2.7, and additional functions have been added.

The original library can be found here: <https://github.com/bmo/py-wsjtx>.

## License

This program is licensed under the GNU GENERAL PUBLIC LICENSE, a copy of which is included in this distribution.

## Change Log

Version 0.21 (November 2024)

- Fixed issue with manual LOG QSO not updating YAAC and GeoServer icons properly.
- Minor bug fixes and performance improvements.
- Updated the documentation.

Version 0.20 (November 2024)

- Added support for client authentication.
- Added support for GeoServer authentication.
- Minor bug fixes and performance improvements.
- Updated the documentation.

Version 0.19 (November 2024)

- Added support for all commands to JS8Call from scripts via JS8Call Monitor.
- Added support for APRIS commands.
- Updated sample scripts.

Version 0.18 (July 2024)

- Minor bug fixes and performance improvements.
- Updated the documentation.

Version 0.17 (June 2024)

- Added support for GridTracker live displays.
- Added support for GridTracker CQ message color coding.
- Added support for GridTracker decode messages.
- Added support for decode messages to pywsjtx python module.
- Added support for sending commands to JS8Call from scripts via JS8Call Monitor.

## JS8Call Monitor

- Added command queue and queue management.
- Added stubs for all command message types.
- Added RADIO in memory database.
- Changed PING event to capture IP and Port info for each RADIO.
- Changed get\_host routine to resolve IP and Port info for RADIO.
- Minor bug fixes and performance improvements.

### Version 0.16 (May 2024)

- Minor bug fixes and performance improvements.

### Version 0.15 (April 2024)

- Added support for the JS8Call Monitor LogIt app.
- Minor bug fixes and performance improvements.

### Version 0.14 (June 2022)

- Added support for mapping STATUS messages
- Added support for logging to ADIF database files (.adi)
- Changed internal PIR 'status' references to 'pir' to avoid confusion with STATUS messages.
- Fixed issue where callsigns with slash (/) modifiers would not properly resolve gridsquare lookups.
- Fixed an issue where manual logging would cause JS8Call Monitor to exit.
- Minor bug fixes and performance improvements.

### Version 0.13 (May 2022)

- Added support for YAAC logger.
- Fixed issue where debug logfile was caching some entries.

### Version 0.12 (April 2022)

- Added additional error handling for logging routines.
- Improved INFO message detection.
- Resolved some Python3 compatibility issues.
- Minor bug fixes and performance improvements.

### Version 0.11 (April 2022)

- Improved Python version detection and compatibility.

### Version 0.10 (April 2022)

- Fixed some Python3 compatibility issues.

### Version 0.9 (March 2022)

- Added option to disable JS8Call Monitor shutdown on JS8Call exit.
- Added option to map all QSOs.
- Changed date and time stamps to all be in UTC.
- Changed HEARTBEATS handling so it is now consistent for all heartbeat types.

## JS8Call Monitor

- Refactored the grids section of the config file for consistency.
- Minor bug fixes and performance improvements.

### Version 0.8 (March 2022)

- Added support for multiple running instances of JS8Call.
- Refactored how INFO messages are handled in the RX.DIRECTED event.

### Version 0.7 (March 2022)

- Added date and time stamps to log and console displays.
- Added support for email and text alerts.
- Added support for grid square lookups via FCC data (off grid).
- Added support for grid square lookups via Radio Amateur Callbook CD (off grid).
- Added support for capturing failed grid square lookups to a rejects file.
- Added GeoServer integration to support posting of messages and spots to GeoChron digital displays.
- Changed flag to limit length of grid squares reported to the loggers, to numeric value.
- Changed how the FREQ tag is passed to N1MM+, so subsequent ADIF exports from N1MM+ now work as expected.
- Improved data capture of manually logged QSO messages.
- Refactored the grid square lookup routines to make them modular.
- Refactored the database section of the config file.
- Refactored the log levels to accommodate a new level SMDR.
- Refactored the debug logging to produce machine readable log files.
- Minor bug fixes and performance improvements.

### Version 0.6 (March 2021)

- Added flag to limit length of grid squares reported to the loggers.
- Added stubs for all receive message types.
- Added support for processing manually logged QSO messages.
- Added support for grid square lookups via hamcall.net (online).
- Added support for grid square lookups via local database.
- Added support for self learning of callsign lookups.
- Minor bug fixes and performance improvements.

### Version 0.5 (January 2021)

- Added the ability to map CQ messages.
- Added the ability to map Heartbeats.
- Added the ability to map Signal Reports.
- Added the ability to map all, or only messages related to the configured station.
- Added support for grid square lookups via HamCall CD (off grid).
- Added support for grid square lookups via Callook.info (online).
- Removed the RX\_Messages logic (no longer needed).
- INFO message detection now works consistently.
- Updated the documentation.

### Version 0.4 (January 2021)

- Moved the JS8Call listener into its own thread and implemented message queuing.

## JS8Call Monitor

- Resolved issue with missing some JS8Call messages, especially during pileups.

### Version 0.3 (January 2021)

- Improved debug logging messages.

### Version 0.2 (December 2020)

- Added the ability to send debug log to files and/or console.
- Added logic to capture fragmented RX\_Messages from JS8Call, stitch them together, and scan them for INFO messages.
- Added logic to capture the Grid square from the INFO messages.
- Refactored how debug logging was done.
- Renamed application from js8call\_to\_n1mm to JS8Call Monitor.

### Version 0.1 (December 2020)

- Original version.