



JS8Spotter

Search Term Traffic Scanner
And Other Utilities for Use With JS8Call

License: (c) 2026 Joe Lyman. Permission is granted to use and adapt the contents of this document as needed, with the exception of any personally identifiable private information such as license identifiers, names, organization names or membership, email addresses, and phone numbers, which may not be redistributed. You may remove this license notice on adapted versions or extracts only if all personally identifiable private information, as described above, is removed as well. This is not an official document of any formal organization.

Table of Contents

Overview.....	4
Prerequisites.....	4
Windows users.....	4
Linux Users.....	4
Upgrading From a Previous Version.....	5
Software Settings.....	6
In JS8Call:.....	6
In JS8Spotter:.....	7
Miscellaneous JS8Spotter Settings:.....	7
QuickStart.....	8
Software Features.....	9
1. Profile-Based Activity Search.....	9
Creating, Editing, or Removing a Profile.....	10
Background Scan.....	10
Search Terms.....	10
Adding and Removing Search Terms.....	10
Importing and Exporting Search Terms.....	11
Activity Matching.....	11
Quick Filter.....	11
Status Bar.....	11
2. Expect Subsystem.....	12
The Expect Window.....	12
Adding an Expect Statement.....	13
Edit, Remove, or View an Expect Statement.....	13
Importing and Exporting Expect Statements.....	14
Triggering Expect Responses.....	14
Sending an Expect Response Manually.....	15
Sending an Expect Response Automatically.....	15
Expect Operating Tips.....	16
Pausing the Expect System.....	16
Expect Blocklist.....	16
3. Multiple Choice / Micro Forms.....	17
Creating Form Templates.....	17
Filling Out and Posting Forms.....	19
Transmitting a Form Over the Air.....	20
Form Responses.....	21
Response Gateway.....	21

JS8Spotter de KF7MIX, Companion Application for JS8Call de KN4CRD

MCForms Datecode.....	22
MCForms Prompts.....	22
4. Additional Features.....	24
APRS Commands.....	24
Simple Offline Maps.....	24
Visualize Waterfall.....	25
CommStat Compatible Features.....	25
Simple Database Search.....	25
Database Trim.....	26
Getting Additional Help.....	26

Overview

JS8Spotter is an API (application programming interface) based companion application for JS8Call. It has been tested with JS8Call (and -improved) versions 2.2.0 through 2.5.2. It communicates with your running JS8Call instance over TCP (transmission control protocol; on your local machine or optionally over a network).

JS8Spotter is not a standalone application. It requires a running instance of JS8Call to operate.

JS8Spotter provides several useful features in a single Python application (with supporting files). Information gathered by *JS8Spotter* is stored in an SQLite3 database, and can be accessed through freely available third-party software.

The primary function of *JS8Spotter* is to **apply automatic keyword searches to incoming JS8Call traffic, so that operators can more easily store and track traffic that interests them**. Additional features include expanded auto-responders, micro forms, APRS commands, as well as map and waterfall visualizations.

Prerequisites

In order to use *JS8Spotter*, you must have a fully functional JS8Call setup. If you're unfamiliar with JS8Call, you will need to start there first. There are **no** radio, sound card, or other hardware configuration requirements related to *JS8Spotter*; all of those concerns are handled in JS8Call.

It is highly recommended that operators looking to use *JS8Spotter* be very comfortable with JS8Call. Learning JS8Call first will help operators understand the need for and value of *JS8Spotter*. Configuration issues having to do with the operation of JS8Call are outside the scope of this document.

Windows users

If you're using the binary build, everything should be setup for you to run the program using this manual. **The Visual C++ Redist package may be required if it's not on your system already**. If you're using the source version of *JS8Spotter*, you may need to scan through the Linux Users section below to get hints about configuring Python3 and the libraries.

Download the latest Windows build from <https://kf7mix.com/js8spotter.html> and save the .ZIP file where you would like to unpack it. **The Visual C++ Runtimes are also on this page**. Locate the file in File Explorer, right-click and Extract All. A new folder should be created for *JS8Spotter*. Enter the folder and double-click the application icon to start the program.

Linux Users

You will require a functional Python3 configuration with Tkinter and Pillow ImageTk libraries installed. On most distros, this is a simple matter of installing the needed packages using pip or your package manager (apt, etc.). On an apt based system (where Python3 is already installed), for example, the following command should suffice to setup a basic environment:

```
"sudo apt install python3-tk python3-pil python3-pil.imagetk"
```

Upgrading From a Previous Version

If you're upgrading from a previous version of *JS8Spotter*, please note the following:

- You must run *JS8Spotter* from its folder, with all supporting files present
- You may have more than one copy (including different versions) of *JS8Spotter* on a single system
- The js8spotter.db database file contains all of your user settings, profiles, and gathered activity
- If you have custom WAV notification sounds, or custom forms, you will need to copy those manually into the new installation after the database is updated

To upgrade, you'll need to complete the following actions:

- Always make a backup of your previous version js8spotter.db file
- Download and unpack the latest version of *JS8Spotter*; Do not overwrite your existing copy
- Some version changes will require a database update. If needed:
 - Download and unpack the latest version of the *JS8Spotter* DBTool
 - Place your old js8spotter.db file in the DBTool folder.
 - Read the included .txt instructions, then run the DBtool (either with the exe or .py file)
 - After your database file is updated, test the newest version to make sure it works
- After everything is working, feel free to replace your old version or update your links

Because *JS8Spotter* runs in-place, with no additional files installed anywhere on your system, you can download, upgrade, and test without losing your working copy. Please ensure that the new version works for you before doing anything with your existing copy! There is no reason to lose your working version while upgrading. Follow the steps above carefully to ensure you can upgrade, and reach out to me if there are issues.

ALWAYS BACKUP YOUR JS8SPOTTER.DB FILE
AND ANY CUSTOMIZED FORM OR WAV FILES
BEFORE TRYING A NEW VERSION

Software Settings

Once you have a functional JS8Call setup (and python3 environment for source users), you'll need to verify that your settings are correct so that your JS8Call API is accessible in *JS8Spotter*. Sane default settings are included, but it is up to you to make sure that everything is set and matching in both applications.

In JS8Call:

- In JS8Call, load the settings by pressing "F2", or by using the "File>Settings" menu item
- Visit the "Reporting" tab
- Make sure the "Allow setting station information..." check box is enabled
- Make sure the "Enable TCP Server API" check box is enabled
- Make sure the "Accept TCP Requests" check box is enabled
- Take note of the "TCP Server Hostname" and "TCP Server Port" values

The screenshot shows the 'Reporting' tab of the JS8Spotter settings window. The 'API' section is highlighted with a red box, containing the following settings:

- ☒ Allow setting station information (Grid, Info, Status, etc) from the APIs
- UDP Server Hostname: 127.0.0.1
- UDP Server Port: 2242
- TCP Server Hostname: 127.0.0.1
- TCP Server Port: 2442
- TCP Max Connections: 1

On the right side of the 'API' section, there are four checkboxes, with the last two highlighted by a red box:

- ☒ Enable UDP Server API
- ☒ Accept UDP Requests
- ☒ Enable TCP Server API
- ☒ Accept TCP Requests

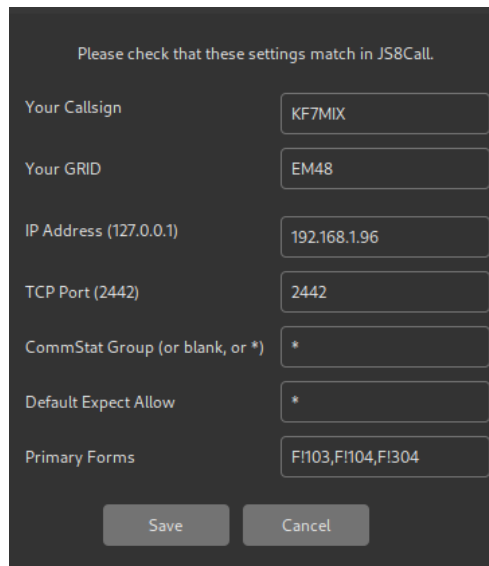
Other visible settings include:

- Logging: Operator Callsign (if different than Station Callsign):
- Network Services:
 - ☒ Enable spotting to reporting networks (JS8NET, PSKReporter, etc)
 - ☒ Enable spotting @APRSIS messages to the APRS-IS network
 - APRS Server: rotate.aprs2.net
 - APRS Server Port: 14580
 - Never send spotting reports from these callsigns (comma separated):
- N3FJP Logger

Buttons at the bottom: Cancel, OK.

In JS8Spotter:

- Ensure JS8Call is running before starting *JS8Spotter*; restart *JS8Spotter* whenever you restart JS8Call
- In *JS8Spotter*, load the settings by using the "File>Settings" menu item
- Enter your Callsign and GRID (4 character)
- Ensure that the "IP Address" and "TCP Port" in *JS8Spotter* match the settings in JS8Call
- If you use the CommStat features, enter your group name, otherwise leave blank
- Set your default Expect Allow targets, or leave at the default of *
- Set your default Primary Forms, or leave as provided



Please check that these settings match in JS8Call.

Your Callsign	KF7MIX
Your GRID	EM48
IP Address (127.0.0.1)	192.168.1.96
TCP Port (2442)	2442
CommStat Group (or blank, or *)	*
Default Expect Allow	*
Primary Forms	F!103,F!104,F!304

Save Cancel

Miscellaneous JS8Spotter Settings:

The following optional settings in *JS8Spotter* will help you control how you wish to view and operate the software.

- "View>Hide Heartbeats" will let you hide heartbeat traffic from the "Matched Activity" pane. You may also Hide/Show RX.SPOT, RX.ACTIVITY and RX.DIRECTED items
- "View>Dark Theme" will let you toggle between a light and dark software theme
- "View>Highlight Range" will temporarily highlight new activity in the "Search Terms" pane
- "Tools>Expect – Pause Replies" menu item will disable all Expect auto-replies
- "Tools>Disable Sounds" will temporarily disable all profile notification sounds
- **If you plan to use Expect or APRS commands to transmit from *JS8Spotter*, the Autoreply setting in JS8Call must be enabled**

QuickStart

This section is primarily for those that prefer hands-on learning. Each of the primary features of *JS8Spotter* are described in very high-level detail below. There is enough here to get you started. For additional information, see each of the corresponding sections in this manual.

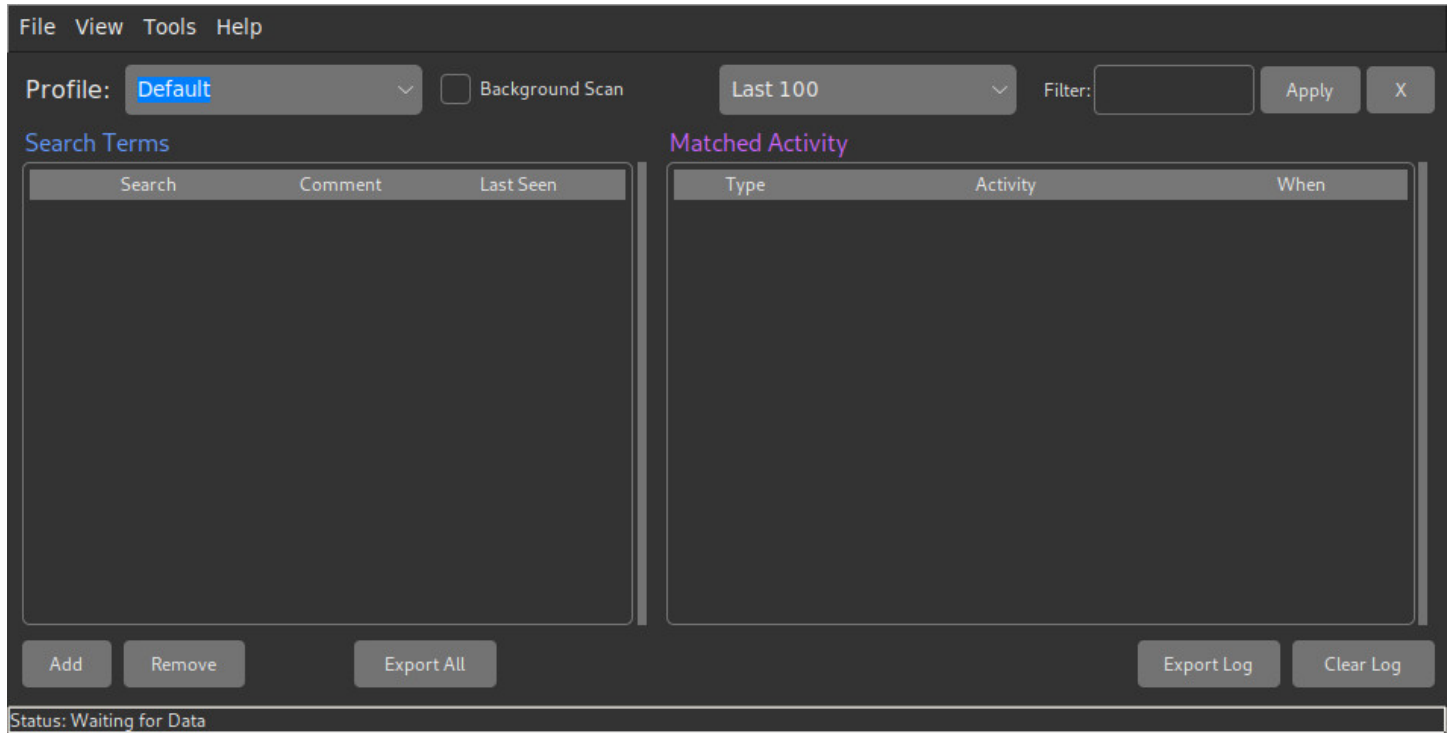
- **Profile-Based Activity Search:** Primary function of *JS8Spotter*. Create profiles (e.g. "Contacts", "Chess", "Weather"). Enable "Background Scan" and they'll scan even when not open. In each profile, create search terms to scan for (e.g. "KF7MIX", "HURRICANE"). The same search terms may appear in multiple profiles. As traffic comes in through the JS8Call API, *JS8Spotter* will look for matches and save matched traffic in the database.
- **Expect Subsystem:** This is a user-defined auto-responder. Your station will "expect" some text, and "respond" with text you provide. It will only "reply" to stations you allow, and it will only auto-reply the number of times you permit. Auto TX may be set for an expect statement, up to twelve times daily.
- **MCForms:** Micro Forms aka Multiple Choice Forms. Look in the forms folder for example forms. Create your own forms if desired, and publish them on kf7mix.com or on your own site. Fill in forms and post them to the Expect Subsystem. Your station will automatically retrieve forms seen on the air and store them.
- **APRS Commands:** *JS8Spotter* includes shortcuts for sending APRS commands through JS8Call. Simply fill in the boxes and send.
- **Simple Offline Maps:** Maps show the approximate location of stations that are reporting their GRID. Helps with understanding propagation. North America, Europe, Australia, and Indonesia are included, but you may create your own maps for anywhere in the world.
- **Visualize Waterfall:** A static visualization of the JS8Call waterfall with signal location and size represented for a period of time from 5min to 7 days. A good way to see which parts of the waterfall are not as crowded.

Software Features

Each of *JS8Spotter*'s four main features are covered in detail below.

1. Profile-Based Activity Search

This is the main function of *JS8Spotter*. When you first open the application, the main window display has two panes: On the left, the "Search Terms" pane; and on the right, the "Matched Activity" pane.



You may wish to create multiple profiles, to organize the search terms and traffic you wish to track. For example, if you wish to track your contacts/friends as well as weather events, you might have profiles and search terms like these fictional ones:

Profile: Contacts	Profile: Weather
Search Terms: KF7MIX ABC1DE @JS8CHESS JOHN SMITH	Search Terms: WX HURRICANE STORM TRACK FLOOD WATCH

Users may create as many profiles and search terms as they like. The same search terms may appear in multiple profiles, at the user's discretion. Profiles may be sorted using the "File>Sort Profiles" menu option.

Creating, Editing, or Removing a Profile

To create a profile, you simply use the "File>New Profile" menu item. Enter the name of your profile in the box, and create it. To edit a profile name, use the "File>Edit Profile" menu item, edit the name and save it. You may set a notification WAV sound if desired. All listed WAV files are located in the *JS8Spotter* folder.

To remove a profile, first make sure you have that profile selected by using the "File>Switch Profile" menu item, or by using the select box in the main window. Next, use the "File>Remove Profile" menu item. Removing a profile will remove all the associated search terms and traffic.

Background Scan

In the main window, next to the profile select box, is a check box for "Background Scan".

- When this box is ☐ **unchecked**, search terms in this profile will only be scanned when this profile is actively selected and displayed in the main window
- When this box is ☒ **checked**, search terms in this profile will be scanned in the background, even if you are currently viewing a different profile.

Search Terms

Search terms may be single or multiple words, using any standard ASCII characters you wish to use. All search terms will be converted to capital letters.

Search terms are matched exactly only. If you wish to match narrowly, use exact terms such as a call sign or a full word (e.g. "KF7MIX", "THUNDER STORM", "YELLOW CARS"). If you wish to match broadly, use partial terms (e.g. "KF7", "STORM", "CAR"). You may use a space at the end of your search term, if you want it to be less broad (e.g. "CAR " to avoid matching "CARROTS").

Use multiple variations of search terms and/or search concepts if needed (e.g. "WX", "WEATHER").

The Search Terms pane may be sorted by clicking on the column headers. Sorting will reset when new data is received.

Adding and Removing Search Terms

To add a search term, click the "Add" button beneath the left "Search Terms" pane. In the space provided, type or paste your search terms, one search term per line, then click "Add Batch". If desired, you may press <tab> after each search term and include a comment, which will be displayed in the main Search Terms pane.

To remove search terms from a profile, use one of the two following methods. You will be prompted before the search terms are removed:

1. Highlight the search term and click the "Remove" button beneath the left "Search Terms" pane.
2. Use <Shift> and/or <Ctrl> to click and highlight multiple search terms, then click the "Remove" button beneath the left "Search Terms" pane.

Importing and Exporting Search Terms

Importing is equivalent to adding, but remember that what you import may be typed or pasted from another source. For example, if multiple users of a group wish to have a profile that uses the same terms (i.e. a JS8Games group, an emcomm group, or a SKYWARN group), they could share those search terms via email or a text file on a thumb drive. Each user could "Import" the same set of data into *JS8Spotter*.

To export search terms, select the profile you wish to export from, and click the "Export All" button beneath the left "Search Terms" pane in the main window. You may copy or save the search terms for later use.

The Search Terms Add and Export All features both operate with blocks of text, with one search term per line, plus the tab-delineated optional comments.

Activity Matching

As traffic comes in to JS8Call, it is provided to *JS8Spotter* through the API over TCP. *JS8Spotter* examines the incoming traffic and attempts to match on search terms in the active profile and in any profiles that have Background Scan enabled. When a match is found, the activity is saved in the database.

The right "Matched Activity" pane in the main shows matches for the selected profile's search terms. You may choose to view the "Last 100" items irrespective of date, or you may select a time range that you wish to view, irrespective of count.

To view details on matched activity, double-click an entry in the "Matched Activity" pane. Right-click on a highlighted entry to copy the report contents to your copy/paste buffer.

If you wish to see additional saved activity, locate the associated search term in the left "Search Terms" pane, and double-click it. This will bring up all matched activity for that search term in the selected profile. Double-clicking or right-clicking entries in this view will bring up activity reports or copy them to your copy/paste buffer respectively.

Traffic need not be directed at your station to be recorded. *JS8Spotter* will keep track of any traffic containing the search terms you provide.

As times, JS8Call provides the same traffic to the API multiple times and/or in multiple ways. This may result in duplicate matches in *JS8Spotter*.

Quick Filter

Above the "Matched Activity" pane is a quick filter box. By entering text in this box, you can filter the results below to contain only activity matching the text. Press <Enter> to immediately filter, or press the Apply button; or, simply wait for new activity to load and be filtered automatically. To clear the filter, you may click the "X" button, or simply erase the text or press <Esc> to clear the box completely.

Status Bar

At the bottom of the main window is the Status Bar, which normally shows, "Waiting for Data". Watch this area for messages about automated actions (Expect transmissions, AutoTX, forms, etc.). To clear a message, visit the corresponding section of the program, or simply click the Status Bar.

2. Expect Subsystem

The "Expect" subsystem provides a method of expanding JS8Call's user-configurable auto-response ability. The built-in user field commands in JS8Call are as follows:

- **"INFO?"** - This provides station information that you've saved in JS8Call's settings. People usually use this line for rig info, but it may be used for anything.
- **"STATUS?"** - This provides status information that you've saved in JS8Call's settings. People sometimes use this to relay idle time or JS8Call program version information.
- **"GRID?"** - This provides your grid as saved in the JS8Call settings.

As all JS8Call users should be aware, if you send any one of these commands in the blind to any other station with autoreply enabled, that station will immediately respond with the stored text. Expect works the same way, but with more flexibility.

With Expect, you can create your own commands and responses (collectively called "statements"), and control the Callsigns and/or groups that are allowed to trigger them. You also have a log of when they were accessed, and the ability to disable them after a fixed number of E? transmissions if desired.

All Expect statements are triggered using the "E? " prefix. See "Triggering an Expect Response" below.

You should be aware that Expect does not include any statements by default. This means that other stations have no way of knowing that you are running Expect, or what commands are available, unless you choose to expose that information using an Expect statement. Expect relies on operators working together in groups in order to make the feature useful.

The Expect Window

The Expect window provides the interface to add, edit, remove, import/export, and view Expect statement.

Expect	Response	Allowed Calls/Groups	Sent To	Count	Created
TEST	TESTING	*,@HERC	AUTO TESTING DRIFT ON 27.12, 7/00	7/00	2025-08-04
FI104	FI104 100 ST[MO] GR[EM48AT] #HCWA	*	EM48AT 100 ST[MO] GR[EM48AT] #HCWA	6/00	2025-08-04
FI700	FI700 13526 YES, AUTO DRIFT #G3S1	*,@4SQRP		0/00	2025-07-30
FI105	FI105 132H11 CS[KF7MIX] FN[JOE] ST[MO] < *			0/00	2025-06-12
MORE	GRP, MCF, QTH, EMAIL, WX, CMD5	*,@SITREP	INFO	1/00	2025-04-10
CMD5	RIG, ANT, STATUS, COND, URL, MORE	*,@SITREP	INFO	64/00	2025-04-10
FI102	FI102 541311 #AMTR	*	INFO	2/00	2025-04-07
ANT	EFHW-8010	*	INFO	17/00	2025-04-07
QTH	FULTON, MO EM48AT	*	INFO	2/00	2025-04-07
WX	NTR	*	INFO	2/00	2025-04-07
GRP	@HERC,@4SQRP,@NTXNVIS,@SITREP,@N	*	INFO	14/00	2025-04-07
STATUS	NTR EM48AT	*,@HERC,@SITREP		0/00	2025-04-07
COND	G	*,@SITREP		0/00	2025-04-07
INFO	"E? CMD5" ZTERSE.COM/JS8	*,@HERC,@SITREP		0/00	2025-04-05
FI304	FI304 11111111 #DBNQ	*,@HERC		0/00	2025-04-02
FI103	FI103 N114 #CLKW	*		0/00	2025-03-12
MCF	FORMS STORED: 100, 102, 103, 104, 300	*	INFO	35/00	2025-03-08
NTX	GREETINGS TO THE NORTH TEXAS CREW!	*,@NTXNVIX		2/00	2025-02-14
FI100	FI100 7234442137 OP JOE #BMLL PNH	*	INFO	4/00	2025-02-13

Import Export Send Now Add Edit Delete

Adding an Expect Statement

To add an Expect statement, open the Expect system and click the "Add" button at the bottom-right.

- **"Text to Expect"** – Single word, 6 characters or less; A-Z, 0-9, and ! allowed, no spaces. Do not include the "E?" prefix here, it is always assumed
- **"Text to Respond With"** – Transmitted when an allowed station or group sends the "Text to Expect"
- **"Allowed Callsigns/Groups"** – Auto-populates from settings. List of callsigns or groups that are allowed to trigger this response. It can be the "*" character for anyone individual station, or a list of comma-separated specific callsigns and/or groups. (e.g. "*" or "*,@WX" or "KF7MIX" or "KF7MIX,AB1CD,EF2GH" or "KF7MIX,@SKYWARN")
- **"Max Replies"** – The number of times your station is allowed to auto-respond for this expect; Enter 0 to temporarily disable the responses, or enter a number, 99=∞
- **"TX Speed"** – in JS8Call-improved v2.4.0 the API for setting speed was fixed. If you're using that version or newer, you can determine the TX speed of your Expect replies. Otherwise, this is ignored.
- **"Auto TX Schedule"** – Up to twelve times (HH:MM format), comma-separated; JS8Spotter will attempt to transmit this Expect statement at the entered times daily. (e.g. "09:32,14:27,22:45")
- **"Auto TX Target"** – The target for the auto TX system; should be a single callsign or @GROUP name

Edit, Remove, or View an Expect Statement

To edit an Expect statement, highlight the row and click "Edit", or double-click the row. Make your changes and click "Save" or "Cancel". To remove an Expect statement, highlight the row and hit the "Delete" button, or the <Delete> key on your keyboard. To view information about an expect statement, including a report on the Callsigns and groups it has been transmitted to, simply edit the row and view the information on the right.

Importing and Exporting Expect Statements

Import and export are simple functions with tab-delimited data. It is recommended that you view the export data to understand the import format. Imported expect statements will overwrite existing statements with the same "text to expect".

Triggering Expect Responses

For Expect to work, the following needs to happen:

- Multiple operators **should coordinate and practice** to understand this system. There are no default Expect statements. **Conventions may be adopted**, and some are recommended in the tips section below
- You must **DISABLE/UNCHECK** the "Allow sending standard messages without callsign" option in JS8Call settings "General>Behavior" tab, or "E?" requests will not function
- Autoreply **must be enabled** in JS8Call in order for JS8Spotter to cause JS8Call to transmit.
- Expect statements are triggered with the "E? " prefix. Note the space. The format is: "E? <Space><Expect Text>"
- You must have no station selected in your JS8Call Call Activity Pane, TX must be enabled, and your outgoing message box must be empty.

As an example, say that my station, KF7MIX, is within range (verified with "SNR?"), and has stored the following Expect statement data:

Expect:	Respond:	Allowed:	Max:
CMDS	WX, RIG, ANT, MCF	*	99

To trigger a response to this Expect statement, a station would have to send the following text directed to my station, such as in:

"KF7MIX: E? CMDS"

If station AB1CD sent the above command, it would be received as "AB1CD: KF7MIX E? CMDS", and my station would automatically reply with "KF7MIX: AB1CD WX, RIG, ANT, MCF"

As another example, say that the station AB1CD is within range and has stored the following Expect statement data:

Expect:	Respond:	Allowed:	Max:
WX	75F SUNNY	*,@SKYWARN	99

Imagine also that both AB1CD and myself, KF7MIX, participate in the @SKYWARN group. If I send:

"@SKYWARN E? WX"

Then AB1CD (and any other @SKYWARN participants in range) would receive "KF7MIX: @SKYWARN E? WX" and their station (or multiple stations at once, if others are also setup) would automatically respond with "AB1CD: @SKYWARN 75F SUNNY"

As one final example, imagine three fictional friends: EF2GH, IJ3KL, and MN4OP. They like checking in on one another, and they each store a "STAT" statement with a response just for their friends. Below would be the Expect statement data that EF2GH has stored:

Expect:	Respond:	Allowed:	Max:
STAT	ALL CLEAR	IJ3KL,MN4OP	99

If station IJ3KL sent "IJ3KL: EF2GH E? STAT", then the response would come back as "EF2GH: IJ3KL ALL CLEAR". Likewise, MN4OP could trigger the response. But if AB1CD saw them communicating, and one day tried "AB1CD: IJ3KL E? STAT", there would be no response, as his station is not in the allowed list.

Sending an Expect Response Manually

You can send Expect responses manually by first right-clicking a row to load it, then clicking the "Send Now" button at the bottom of the Expect window. Fill in the target station or group (which do **not** need to be in the allowed list for manual TX), and optional check box for sending the response as a MSG (which will prompt an ACK from the receiving station.)

Manual Expect responses are primarily intended for transmitting filled out MCForms, which will be covered later. However, you may transmit any Expect response manually, at any time.

After manually transmitting an Expect response (or copying for manual TX), the target group or station will be saved for the next manual transmission.

Manual transmissions are not limited by the Max Replies setting, and are not recorded in the Expect log.

Sending an Expect Response Automatically

You can attempt an Expect auto transmission, up to twelve times daily. Add valid data for the Auto TX items in your Expect entry, and make sure that JS8Call and *JS8Spotter* are running during the times you specify. The message pane in JS8Call must be empty to allow for API transmit commands.

Since the JS8Call API provides no way to check if JS8Call is busy, or to queue a message to transmit, or to verify if a message was transmitted, JS8Spotter cannot verify that your Auto TX messages are actually sent, but it will attempt them.

Manual transmissions attempts are not limited by the Max Replies setting, but are recorded in the Expect entry log.

Expect Operating Tips

To maintain flexibility and honor varying perspectives on privacy, Expect only does what you tell it to do. It does not expose new universal remotely-activated commands. It will not reveal a list of available commands unless you set that up manually. For this reason, it is valuable to keep some potential practices in mind:

- Expect is primarily intended for groups of operators. These groups should operate off a common set of procedures, such as a published SOP. At the very least, these groups should agree on how to configure their Expect systems so that they know what commands to send to trigger responses.
- It is good practice and highly recommended to setup a "CMDS" Expect statement. This statement should be accessible to "*" and should respond with a list of Expect commands that you want the general public to be aware of. It could be short or long, and it doesn't have to list all of your commands, just the public ones. Examples of "public" commands might include:
 - WX – store and report your daily weather
 - RIG – store information about your rig here, and use "INFO?" to direct them to "E? CMDS"
 - ANT – store information on your antenna system specifically, instead of cramming it into "INFO?"
 - URL – store a website URL that other operators can visit
 - GRP – store a list of groups you participate in
 - MCF – store a list of any forms you have available for retrieval
- You may include the text "MSG " at the beginning of your responses, if you want to transmit those responses as JS8Call messages. This will prompt an ACK from the receiving station
- You may send E? requests through a relay, but response is delayed for 120seconds to allow for the JS8Call built-in ACK replies and other multiple communications associated with a relay. Simply wait for the station to send the E? response after all the JS8Call chatter is completed

Pausing the Expect System

You may temporarily pause the Expect system using the "Tools>Pause Expect System" menu item. None of your Expect statements will be lost, but your station will not auto-respond until you un-pause the system. This setting is retained even if you exit the program, but a notification is provided at different points to remind you that Expect has been disabled.

Expect Blocklist

You may prevent unlicensed or undesired stations from triggering Expect responses by adding them to the blocklist. Access the comma-separated list in the Tools menu.

3. Multiple Choice / Micro Forms

JS8Call is a slow mode, tailored specifically to weak signals. Normal forms, as seen in programs such as Winlink and Flmsg, are open format, free-form input intensive, and large (by JS8 standards). The MCForms (which stands for both Micro Forms and Multiple Choice Forms) features of *JS8Spotter* bring forms to JS8Call in a way that is respectful of the mode and of the operators using the mode. MCForms aims to:

- Allow operators to craft complex multiple-choice forms that convey a great deal of information
- Limit transmitted characters by modeling forms around choices rather than free-form information
- Provide for limited additional free-form information and comments
- Model possibilities through included forms
- Provide a standard convention for form identification and categorization
- Provide a resource for publicly publishing forms, so that transmission meaning is never obscure
- Include an internet gateway for groups who want to submit gathered forms to a central location

Creating Form Templates

The plain-text (ASCII) format for MCForms templates is designed to be quickly human readable.

MCForms Template Format Requirements	
First Line	<p>The first line of the form text file must contain:</p> <p><Title text><Pipe><Form ID> (e.g. "Test Form F!000)</p> <ul style="list-style-type: none"> • The <Title text> may be anything but should be short and to the point • <Form ID> convention is the letters "F!" followed by three digits as per the following standard: <ul style="list-style-type: none"> ◦ 000 — 099 Preliminary. For proposed or developed forms ◦ 100 — 199 Informational, Informal, Non-Emergency ◦ 200 — 299 Reserved for future allocation ◦ 300 — 399 Emergency & Emergency Exercise, Situation Reports ◦ 400 — 499 Reserved for future allocation ◦ 500 — 599 Surveys of general conditions ◦ 600 — 699 Reserved for future allocation ◦ 700 — 799 Group and Net forms for specific groups ◦ 800 — 899 Reserved for future allocation ◦ 900 — 999 Misc. forms and overflow • For custom forms, you may use F! with any number/letter combo if you wish to operate outside the standard convention

Question Lines	<p>Questions are opened by a line in this format:</p> <p><Questionmark><Space><Text> (e.g. "? What is your favorite color?")</p> <ul style="list-style-type: none"> • Start the line with a question mark and space, which won't be displayed • A Question Line must open a question before Answer Lines start • Question Lines need not end with a "?", often, a ":" is more appropriate.
Answer Lines	<p>Answers are added to an open question in this format:</p> <p><At Symbol><Character><Space><Text> (e.g. "@1 Red" or "@R Red")</p> <ul style="list-style-type: none"> • The answer <Character> you choose can be a number or letter • You don't have to have any special order to your answer <Character>s • The <Character>s will be transmitted over the air • The <Text> can be anything you choose • Answer Lines are associated with the Question Line that precedes them
Prompt Lines	<p>Prompt for user input in this format: <[><two A-Z0-9 chars><]><space><prompt></p> <p>Please see the section below on MCForms Prompts for more information.</p>
Headers	<p>Header lines are in this format:</p> <p><Up to three exclamation marks for level 1,2,3><Space><Text></p>
Instructions	<p>Instruction lines are in this formation: <Period><Space><text></p>
Comments	<p>Start comment lines with "#". The first line in the file must not be a comment!</p>
Please Note	<p>It is recommended that you use a basic text editor (such as notepad++ for Windows) and use only ASCII characters in your form files (no UTF-8, icons, emojis, etc).</p>

Creating multiple choice forms requires flexible thinking. Many forms that would otherwise require the operator to type out a brief response, may be broken down into one or more multiple choice answers that require less dependence on operator training by creating response boundaries. Take this theoretical example:

<p>Standard Form Question:</p> <p>1. Status of utilities at your QTH:</p> <p>Free-form Answer TXed:</p> <p>1. "My power is on, but my gas is shut off. We've got city water, but it's only working part of the day. The internet is down but my cell phone is working some of the time, I can text and that's about it. There are radio and TV stations working too, I think they're all up." (269 characters. JS8Call will use dictionary encoding to reduce the transmission time, but allowing unlimited words in a form is not wise for this mode.)</p>	<p>MCForms Questions (see form F!301):</p> <ol style="list-style-type: none"> 1. Operator reporting for: 2. Phone landlines functioning? 3. Cell phone functioning? 4. Local AM/FM/TV stations broadcasting? 5. Internet functioning? 6. Public water works functioning? 7. Commercial power functioning? 8. Natural gas supply functioning? 9. NOAA Weather Radio functioning? <p>MCForms Answers TXed:</p> <p>F!301 <i>143132111</i> #ABCD (21 characters; to decode the italic responses to questions 1-9, see the image in the next section, or view form F!300 in the software)</p>
---	---

The above example doesn't show the answers available to the operator, but you can see them in form F!301 included with *JS8Spotter*. You may not be able to replace every free form question with multiple choice questions. However, it is possible to convey a great deal of information cleanly and uniformly by breaking questions up.

Filling Out and Posting Forms

In the *JS8Spotter* main window, using the "Tools>MCForms – Forms" menu item, you may access a list of all the forms that are located in the "forms" subfolder in your *JS8Spotter* installation directory. A series of forms have been included by default. You may add to these by creating new text files for the appropriate format, in this folder.

By selecting a form from the sub-menu of "Tools>MCForms – Forms", you will be presented with a window like this one:

To fill out the form, select the best answer to each of the questions presented. Add a comment at the bottom as needed, keeping it as brief as possible.

To post a form, click "Post Form to Expect". If you have previously posted this same Form ID, it will be overwritten (you will be notified of this fact each time you post). You will be asked if you wish to transmit the form immediately using the Send Now functionality of the Expect system.

You may load a previously posted form from the Expect subsystem by clicking "Load Posted Expect Form." This will allow you to edit your answers, and update a posted form.

If you wish to remove a posted form, simply delete the Expect statement.

Transmitting a Form Over the Air

Forms are posted to the Expect subsystem, and are automatically transmitted upon request. The format for transmitting forms over the air can be easily viewed after posting a form to Expect, and is:

F!<three characters><space><form response><comment and or space(s)><datecode>

Transmission may be preceded by "MSG" if desired, but it must be added manually. To transmit a form immediately, post the form then visit Expect, load the form into the edit fields, and click the "Send Now" button. To store a form, or relay, use the "Send Now" screen to copy the form text, then transmit or store it as usual. The <datecode> will be explained later in this section.

Form Responses

JS8Spotter will gather all forms that it sees transmitted over the air. These responses are available by using the "Forms>MCForms – Responses" menu item in the main window. You may manually import form responses.

View Form Responses: View All Form Types All Time No Filter

From	To	Form #	Form Responses	Message	Timestamp	GW	Received
W4DQK	W4DQK	FE01	FE01		12/17 11:42	✓	2025-01-14 23:37:43
K0DQK	W4DQK	FE01	FE01		1/1 18:12	✓	2025-01-11 22:00:13
K0DQK	K0DQK	FE01	FE01		12/23 8:08		2025-01-08 01:55:42
K0DQK	K0DQK	FE01	FE01	THIS HPS ST STATION, SLOWING POWER-DN	1/5 10:50		2025-01-05 16:58:58
K0DQK	N0DQK	FE01	FE01	SHACK ROOM	1/1 12:52		2025-01-01 19:53:28
K0DQK	N0DQK	FE01	FE01	SHACK ROOM	1/1 12:52		2025-01-01 19:45:43
K0REL	W4DQK	FE01	FE01	SEPH	12/11 20:38		2024-12-29 01:40:28
K0REL	W4DQK	FE01	FE01	SEPH	12/11 20:28		2024-12-29 01:32:57
K0DQK	K0DQK	FE01	FE01		12/19 17:56		2024-12-20 03:01:42
K0DQK	W4DQK	FE01	FE01	K0DQK OFF-LN	12/19 12:32		2024-12-20 00:58:28
K0DQK	N0DQK	FE01	FE01	IND-ON	12/18 14:14		2024-12-18 20:19:28
K0DQK	K0DQK	FE01	FE01		12/17 23:54		2024-12-18 04:56:58
K0DQK	K0DQK	FE01	FE01		12/17 23:42		2024-12-18 04:44:42
K0REL	W4DQK	FE01	FE01	N0REL SEPH	12/17 21:18		2024-12-18 02:23:28
K0REL	W4DQK	FE01	FE01	N0REL SEPH	12/16 21:04		2024-12-17 02:06:58

Import Export All Gateway: Save GW Forward

In the Responses window, you may:

- Filter by Form ID, time received, and/or duplicates
- View information at a glance, including whether the form was submitted to a Gateway ("GW")
- Manually Import/Add form responses to the list in the format:
 - FROM<space>TO<space>FORM#<space>RESPS<space>MSG<space>TIMESIG
- Use the "Export All" button to save responses in a spreadsheet-type format
- Double-click a response to view/save a formatted human-readable version, select and right-click for GUI
- Highlight one or more responses and hit "Delete" to remove them
- Forward a response to another station or group

Response Gateway

In this context, the Gateway is a user-provided web-based application which receives specific HTTP/HTTPS requests and processes the included data, storing and/or performing other functions with that data. A very basic sample gateway is available for testing (with source) on kf7mix.com.

The Gateway is an optional feature intended for groups of operators who wish to leverage the internet to increase the speed at which the group may communicate forms, or integrate with other systems.

In the Response window you have the option of setting a Gateway URL or leaving the field blank. If you provide a properly formatted URL, forms received by *JS8Spotter* will be submitted to that URL. For example:

Sample/Test Gateway URL: https://kf7mix.com/mcfgateway_submit.php

Example Posted Request:

https://kf7mix.com/mcfgateway_submit.php?fromcall=AB1CD&tocall=KF7MIX&typeid=F!301&responses=143132111&msgtxt=HELLO×ig=#CJIB

Groups may choose to keep their Gateway URL private, and include additional key/value pairs in the Gateway setting (by adding "?name=value" to the Gateway URL) to provide basic authentication. For example:

Fictional Gateway URL with added key/value pair:

https://ab1cd.com/spotter_gateway.php?operatorcode=719015221

Example Posted Request:

https://ab1cd.com/spotter_gateway.php?operatorcode=719015221&fromcall=AB1CD&tocall=KF7MIX&typeid=F!301&responses=143132111&msgtxt=HELLO×ig=#CJIB

MCForms Datecode

The datecode at the end of an MCForm transmission provides an approximate low-resolution timestamp in a human-readable format with as few characters as possible over the air. It starts with "#" and the 4 characters following are interpreted as:

First Character	Month, represented as a single letter: A-L = 1-12
Second Character	Day, represented as a single letter or number: A-Z = 1-26, 0-4 = 27-31
Third Character	Hour, represented as a single letter: A-X = 0-23
Fourth Character	Minute, represented as a single letter, 2min resolution: A-Z, 0-3 (A=0, B=2, C=4, etc.)

If you wish to manually encode or decode a Datecode, you may access the datecode tool in the "Tools>Datecode Tool" menu.

MCForms Prompts

An MCForm Prompt allows you to programmatically fill the comments section with specific data defined by the form. A prompt provides a text entry box in the JS8Spotter application when a form is filled out, and the resultant user-supplied text is copied into the comments section along with an identifier.

The identifier is a two-character string, made up of A-Z or 0-9. A valid prompt would be: "[GR] Please Enter Your Grid".

Please note the following if using this feature in your forms:

- You cannot have a form with no multiple-choice questions and only prompts, this will cause errors in form storage, retrieval, and processing
- Prompts should only be used when/if multiple-choice is simply not an optional
- Any time you include a prompt, the person filling out the form can answer with as many characters as they wish to use, so proper operating procedures and etiquette will be vital
- If individual operators and groups are creating custom forms, every effort should be made to minimize prompt usage and maximize multiple choice questions, to save transmit time and keep the bands friendly
- Users filling out an MCForm with prompts should be careful not to modify the generated prompt details in the comments, if/when they add additional information at the end of the comment

4. Additional Features

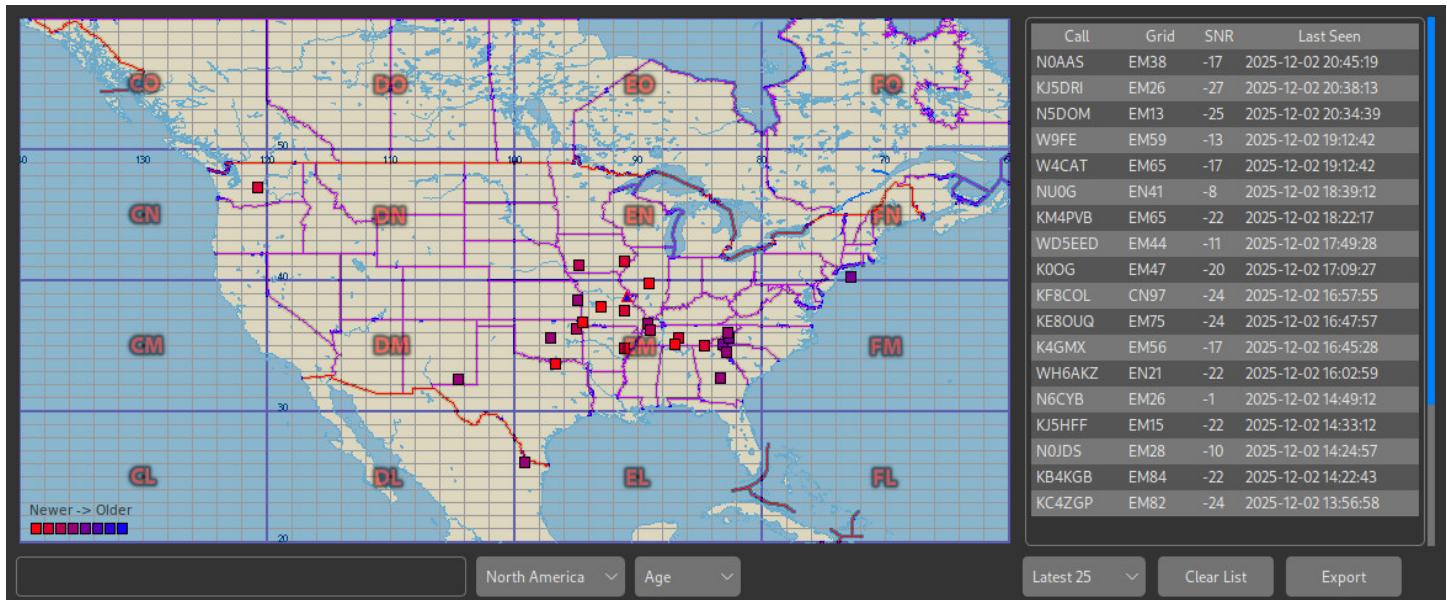
APRS Commands

JS8Call supports APRS-IS integration natively. APRS commands transmitted in the proper format may be received by JS8Call stations and forwarded through the internet to the APRS system. JS8Call and *JS8Spotter* provide mechanisms for emails, SMS, WhatsApp, grid reports, and POTA Gateway.

JS8Spotter makes entering APRS commands easier by providing simple fields to fill out. These are found in the "Tools" menu of the main window. You may use these to generate the APRS statements, which may be copied or transmitted directly.

Simple Offline Maps

JS8Call transmits maidenhead grid coordinates regularly. Because these grids are already included in many transmissions, it makes sense to provide a way to visualize them to better understand propagation.

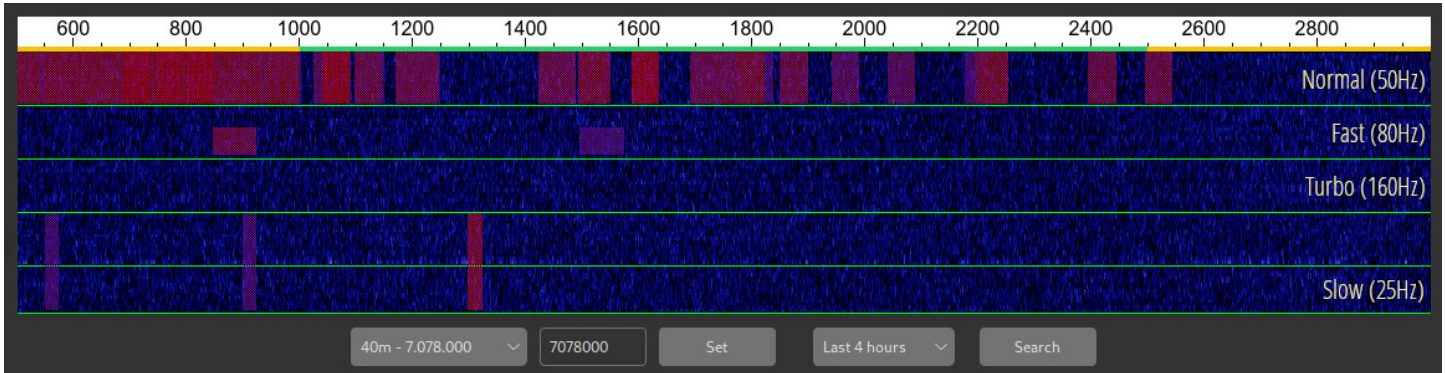


The map feature is located in the "Forms" menu of the main window.

- North America, Europe, Australia, Northern and Southern Africa, and Indonesia are included
- You may filter results by age or by SNR, and show the last n results
- Click a quadrant to zoom in to that quadrant. Click again to zoom out
- Make sure your own grid location is set in the settings, as described in this manual
- Double-click a line to view details on that line; Select a line and right-click or hit <Delete> to remove
- Export or Clear all saved grid reports with the corresponding buttons
- Only the most recent grid report of any given callsign will be shown
- Locations are very approximate (4-letter grid)

Visualize Waterfall

For operators who like to leave their station running unattended, it can be difficult to select a portion of the waterfall that isn't busy. This feature will help you visualize JS8Call traffic over time on a simulated static waterfall.



- You may select a default band and frequency, or you may type in a custom frequency in Hz and click "Set". The frequency you enter should be your dial frequency
- Only bands/frequencies you're actively using in JS8Call will gather data for this visualization
 - For example, if you only ever monitor 7.078, then you will never see data for any other band or frequency. If you manually switch JS8Call to use 7.101, and you gather data there, then you may enter 7101000 to view the simulated waterfall for that frequency
- The most recent signals are red, the oldest signals are gray
- Signals are separated by speed: normal at the top, fast next, turbo, then slow at the bottom
- The heartbeat portions of the band will always be the busiest
- Use "Search" to filter/search by callsign(s)

CommStat Compatible Features

JS8Spotter includes some basic CommStatOne compatibility features, including:

- The ability to capture, view, forward, and export received CommStat Status Reports (statreps)
- The ability to fill out and transmit a CommStat formatted statrep
- The ability to fill out and transmit a CommStat compatible net checkin

These features are meant to provide basic compatibility only, and are in no way designed to replace the CommStatOne software package.

Simple Database Search

A very basic database search tool has been included in the Tools menu. This feature will allow you to quickly search through specific database tables based on text you provide. Users who do not wish to install free database software such as SQLiteStudio or DB Browser for SQLite may use this basic search instead.

Database Trim

Running spotter continuously over long periods of time, especially with large profiles, may cause your database file to become quite large. If you notice a slowdown in JS8Spotter on your system, you may use the database trim feature to reduce the size of your database file. The trim feature will simply remove all records which are older than 6 months. It is recommended that you make a backup of your database file before trimming.

Getting Additional Help

Additional help, insights, and training videos are available online:

<https://kf7mix.com/js8spotter.html>