**WSJT-X User’s Guide**

**Version 1.0**

**Joe Taylor, K1JT**

**Copyright © 2013**

**April 29, 2013**

Contents

[Introduction 3](#_Toc355004474)

[System Requirements 4](#_Toc355004475)

[Installation and Setup 4](#_Toc355004476)

[Operating Hints 8](#_Toc355004477)

[Menus 9](#_Toc355004478)

[Keyboard Shortcuts 11](#_Toc355004479)

[Special Mouse Commands 12](#_Toc355004480)

[Acknowledgments 12](#_Toc355004481)

[Appendix A: Installed and Generated Files 13](#_Toc355004482)

[Appendix B: The JT9 Protocol 15](#_Toc355004483)

[Appendix C: Source Code 15](#_Toc355004484)

# Introduction

[*WSJT-X*](http://physics.princeton.edu/pulsar/K1JT/wsjt.html)is a computer program designed to facilitate basic amateur radio communication using very weak signals. The first four letters in the program name stand for “Weak Signal communication by K1JT”, and the “*-X*” suffix indicates that *WSJT-X* started as an experimental branch of program *WSJT*.

Version 1.0 of *WSJT-X* offers a new mode called JT9, designed for use on the LF, MF, and HF bands. JT9 shares many characteristics with the modes JT65 and JT4 made popular in *WSJT*. All three are designed for making minimal QSOs under extreme weak-signal conditions. They use nearly identical message structure and source encoding; JT65 is used for EME on the VHF/UHF bands and for worldwide QRP communication at HF, while JT4 is used mainly on the microwave bands. In contrast, JT9 is optimized for HF and lower frequencies. JT9 is about 2 dB more sensitive than JT65A while using less than 10% of the bandwidth. World-wide QSOs are possible with power levels around 1 W and compromise antennas, and several dozen JT9 signals fit easily into a 1 kHz slice of spectrum.

All WSJT modes use timed sequences of alternating transmission and reception periods. JT9 offers five choices for the sequence durations: submodes JT9-1, JT9-2, JT9-5, JT9-10, and JT9-30 use 1, 2, 5, 10, and 30 minutes, respectively. A minimal QSO with JT9-1 takes 4 to 6 minutes: 2 or 3 transmissions by each station, one transmitting in odd minutes and the other even. The remaining JT9 submodes take proportionally longer, so for obvious reasons JT9-1 is the preferred submode under most circumstances. The sub-modes with longer transmissions trade reduced throughput for smaller bandwidth and increased sensitivity. The slowest sub-mode, JT9-30, has total bandwidth 0.4 Hz and operates at signal-to-noise ratios as low as –40 dB measured in the standard 2.5 kHz reference bandwidth. It requires very stable oscillators in both transmitter and receiver. JT9-1 is always the recommended submode unless you really need the additional sensitivity of a slower mode.

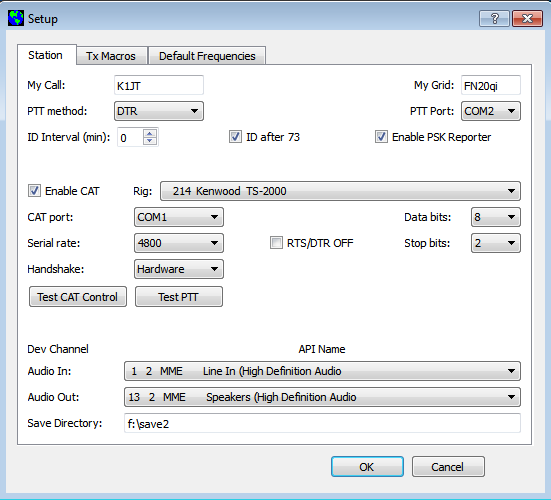
Plans for future program development call for *WSJT-X* and *WSJT* to merge together: *WSJT-X* will gradually acquire the modes JT65, JT4, FSK441, and ISCAT that are now supported in *WSJT*. The entire WSJT-related effort is an open-source project. If you have programming skills or would like to contribute to the project in other ways, please make your interests known to K1JT and the rest of the development team. The project’s source-code repository can be found at <http://developer.berlios.de/projects/wsjt/>.

# System Requirements

* SSB transceiver and antenna
* Computer running Windows XP or later. (*WSJT-X* also runs under Linux and OS X, but we do not yet provide click-to-install packages for these operating systems. See Appendix C.)
* 1.5 GHz or faster CPU and 512 MB of available RAM
* Monitor with at least 800 x 600 resolution (more is better)
* Sound card supported by your operating system
* Computer-to-radio interface using a serial port to key your PTT line, or CAT control.
* Audio connections between transceiver and sound card
* A means for synchronizing your computer clock to UTC. I recommend *Meinberg NTP*, see <http://www.satsignal.eu/ntp/setup.html> for installation instructions.

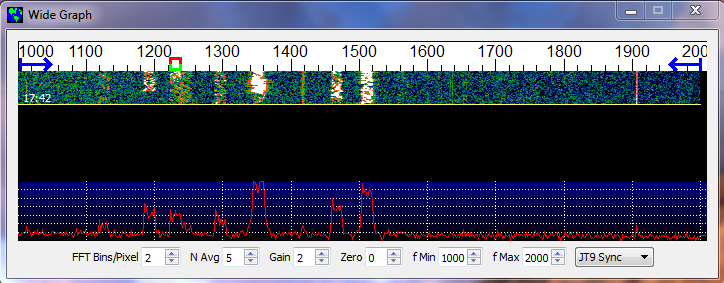
# Installation and Setup

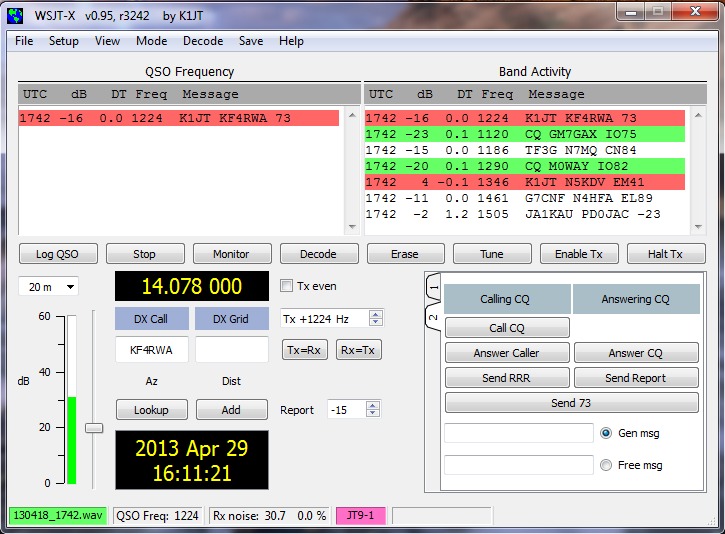
1. *WSJT-X* can be downloaded from the WSJT Home Page at <http://www.physics.princeton.edu/pulsar/K1JT/>. Click on the WSJT link at the left margin and then on the appropriate download link for *WSJT-X*. Especially during phases of active program development, you should also consult the project’s ChangeLog at <http://physics.princeton.edu/pulsar/K1JT/wsjtx_changelog.txt>.
2. Under Windows, execute the downloaded file and follow the installation instructions. Install *WSJT-X* into its own directory (the suggested default is C:\WSJTX) rather than the conventional C:\Program Files\WSJTX. All files relating to *WSJT-X* will be stored in this directory and its subdirectories. Nothing is written to the Windows registry, so you can completely uninstall *WSJT-X* simply by removing the installation directory and all its contents.
3. Start *WSJT-X* and Select **Configuration** from its **Setup** menu. Enter your callsign and Maidenhead grid locator as shown in the screen shot on the next page, and set the remaining parameters as required for your station. Try clicking the **Test CAT Control** and **Test PTT** buttons to see that you have established the desired control of station functions. Then click **OK** to dismiss the **Configuration** window.



1. If necessary, click the **Stop** button on the main window to halt any data acquisition. Select submode **JT9-1** from the **Mode** menu and **Normal** from the **Decode** menu. On the Wide Graph window select **JT9 Sync** (rather than **Current** or **Cumulative**) for data display. Select **Tab 2** (just below the **Erase** button on the main window) to choose the alternative set of controls for generating and selecting messages to be transmitted. Then select **File | Open**, navigate to directory …\Save\Samples under your WSJT-X installation directory, and open the example file 130426\_0027.wav. You should see something like the screen shots on the next page.

Notice the green, red, and blue markers on the waterfall frequency scale. Decoding in JT9 mode takes place at the end of a receive sequence and is organized in two stages. The first decodes happen at the selected Rx frequency, indicated by the green marker. These decoding results appear in the both the left (“QSO Frequency”) and right (“Band Activity”) text windows. The decoder then finds and decodes all JT9 signals between the blue markers at frequencies **f Min** and **f Max.** The normal wideband decoding range is 1000 – 2000 Hz, but you can move the limits using the **f Min** and **f Max** spinner controls. The red marker indicates the Tx frequency.





1. Note that at least eight JT9 signals are present in the example file; all but one of them are decodable. At the time this file was recorded KF4RWA was finishing a QSO with K1JT. Since the green marker is placed at his frequency, 1224 Hz, his message “K1JT KF4RWA 73” appears in both decoded text windows. The “Band Activity” window shows this one as well as all the other decodes at nearby frequencies: GM7GAX and M0WAY are calling CQ, N7MQ is calling TF3G, N4HFA is calling G7CNF, PD0JAC is sending a signal report to JA1KAU, and N5KDV is calling K1JT as a tailender, on a different Tx frequency. The CQ lines are highlighted in green, and lines containing the callsign entered on the **Setup | Configuration** screen as “My Call”, in this case K1JT, are highlighted in red. (For this step and the next one, you may want to pretend you are K1JT by entering that call temporarily as “My Call”. Your results should then be identical to those shown here.)
2. To gain some feeling for the controls you will use when making QSOs, try clicking with the mouse on the decoded text lines and on the waterfall spectral display. You should be able to confirm the following behavior:
   1. Double-click on either of the decoded lines highlighted in green. This action copies callsign and locator of a station calling CQ to the “DX Call” and “DX grid” entry fields. It also generates suitable messages for a minimal QSO and checks (or clears) the **Tx even** box so that you will transmit in the proper minutes. Rx and Tx frequency markers will be moved to the CQing station’s frequency, and the **Gen Msg** (“generated message”) radio button at bottom right of the main window will be selected. If you had selected “Double-click on call sets Tx Enable” on the **Setup** menu, **Enable Tx** would also be activated, and you would start to transmit automatically, at the appropriate time.
   2. Double-click on the decoded line with the message “K1JT N5KDV EM41”, highlighted in red. Results will be similar to (a), except the Tx frequency (red marker) is not moved. (Such messages are usually in response to your own CQ, or response to a tail-ender. You probably want your Tx frequency to stay where it was.) By holding down the Ctrl key when double-clicking on the decoded line, you can cause both Tx and Rx frequencies to be moved.
   3. Now double-click on the message from KF4RWA in either window. He is sending “73” to K1JT, signifying that the QSO is over. Most likely you want to call CQ again, so the message “CQ K1JT FN20” is automatically generated and selected.
   4. Clicking on the waterfall moves the Rx frequency (green marker) to the selected frequency. CTRL-click on waterfall moves both Rx and Tx frequencies.
   5. Double-click on the waterfall moves the Rx frequency and causes a narrow-band decode there (at the new QSO frequency). Decoded text appears in the left window only.
   6. CTRL-double-click moves both Rx and Tx frequencies and decodes there. Again, decoded text appears in the left window.
   7. Clicking **Erase** clears the left window. Double-clicking **Erase** clears both text windows.
3. Don’t forget to re-enter your own callsign as “My Call”. Click the **Monitor** button to return to normal receive operation. Using the receiver gain control(s) and/or the Windows mixer controls, set the background noise level to around 30 dB on the thermometer scale at lower left of the *WSJT-X* main screen. If necessary you can also use the slider next to the thermometer scale. The overall dynamic range will be best with the slider close to mid-scale. When this is true, the dB scale is calibrated relative to the least significant bit of 16-bit samples from the soundcard.
4. You should now be ready to make QSOs with the JT9 modes in *WSJT-X*.

# Operating Hints

By longstanding tradition, a minimal valid QSO requires the exchange of callsigns, a signal report or some other information, and acknowledgments. *WSJT-X* is designed to facilitate making such minimal QSOs with very weak signals, and the process works best if you follow standard operating practices.

The recommended basic QSO goes something like this:

1. CQ K1ABC FN42

2. K1ABC G0XYZ IO91

3. G0XYZ K1ABC -19

4. K1ABC G0XYZ R-22

5. G0XYZ K1ABC RRR

6. K1ABC G0XYZ 73

These standard, structured messages consist of two callsigns (or CQ, QRZ, or DE and one callsign) followed by a grid locator, a signal report, or the acknowledgement “RRR” or sign-off “73”. Such messages are compressed and encoded in a highly efficient and reliable way. Free-format messages such as “TNX JOE 73 GL” can also be transmitted, up to a maximum of 13 characters. Users often add some friendly chit-chat as a final transmission, perhaps instead of sending standard message #6. It should be obvious that JT9 is not a mode suitable for extensive conversations or rag-chewing.

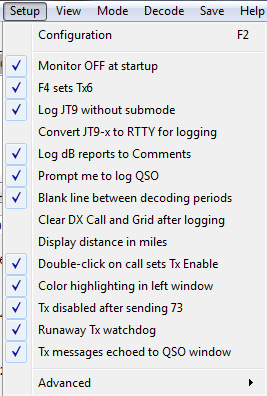
# Menus

Program menus offer many options for configuration and operation. You should explore them and test the resulting program actions. The actions caused by most menu items should be self-explanatory.

**File menu**



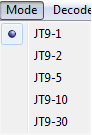
**Setup menu**



**View menu**



**Mode menu**



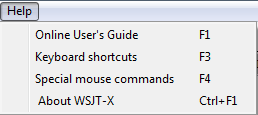
**Decode menu**



**Save menu**



**Help menu**



# Keyboard Shortcuts

**F1** Online User's Guide

**F2** Open configuration window

**F3** Display keyboard shortcuts

**F4** Display special mouse commands

**Alt+F4** Exit program

**F6** Open next in directory

**F11** Move Rx frequency down 1 Hz

**Ctrl+F11** Move Rx and Tx frequencies down 1 Hz

**F12** Move Rx frequency up 1 Hz

**Ctrl+F12** Move Rx and Tx frequencies up 1 Hz

**Shift+F6** Decode remaining files in directory

**Ctrl+F1** About WSJT-X

**Alt+D** Decode again at QSO frequency

**Shift+D** Full decode (both windows)

**Alt+E** Erase()

**Ctrl+F** Edit the free text message box

**Alt+G** Generate standard messages

**Alt+H** Halt Tx()

**Ctrl+L** Lookup callsign in database, generate standard messages

**Alt M** Monitor()

**Alt+N** Enable Tx()

**Alt+Q** Log QSO()

**Alt+S** Stop()

**Alt+T** Tune

**Alt+V** Save the most recently completed \*.wav file

# Special Mouse Commands

|  |  |
| --- | --- |
| **Mouse-click on** | **Action** |
| **Waterfall** | Set Rx frequency  Double-click to set Rx frequency and decode there  Ctrl-click to set Rx and Tx frequencies and decode |
| **Decoded text** | Double-click to copy second callsign to Dx Call and locator to Dx Grid; change Rx and Tx frequencies to decoded signal's frequency; generate standard messages. If first callsign is your own, Tx frequency is changed only if Ctrl is held down when double-clicking. |
| **Erase button** | Click to erase QSO window  Double-click to erase QSO and Band Activity windows |
| **Tx5 Entry** | Right-click to select a macro message |

# Acknowledgments

Many users of *WSJT*, too numerous to mention here individually, have contributed suggestions and advice that have greatly aided the development of *WSJT-X* and its sister programs. Since 2005 the overall project (including *WSJT*, *MAP65*, *WSPR*, and *WSPR-X*) has been “open source”, all code being licensed under the [GNU Public License (GPL)](http://www.gnu.org/licenses/gpl.html). For *WSJT-X* in particular, I wish to acknowledge the many recent contributions from PY2SDR, VK4BDJ, AC6SL, and AF5X. They have helped to bring the program’s design, code, and documentation to its present state.

# Appendix A: Installed and Generated Files

After installing *WSJT-X* as described in steps 1 and 2 on page 4, the following files will be present in the installation directory:

afmhot.dat Data for AFMHot palette

blue.dat Data for Blue palette

CALL3.TXT Callsign database

hamlib-alinco.dll Hamlib libraries

hamlib-amsat.dll ...

hamlib-flexradio.dll

hamlib-icom.dll

hamlib-jrc.dll

hamlib-kachina.dll

hamlib-kenwood.dll

hamlib-kit.dll

hamlib-tapr.dll

hamlib-tentec.dll

hamlib-winradio.dll

hamlib-yaesu.dll

jt9.exe Executable for JT9 decoder

libfftw3f-3.dll Optimized FFT library

libgcc\_s\_dw2-1.dll gcc runtime

libhamlib-2.dll

libstdc++-6.dll standard C function library

libusb0.dll USB interface functions

mingwm10.dll MinGW library

mouse\_commands.txt Special mouse commands

palir-02.dll Linrad functions

PSKReporter.dll Library for PSK reporter

QtCore4.dll Qt core library

QtGui4.dll Qt GUI library

QtNetwork4.dll ...

QtSvg4.dll

qwt.dll Qwt library

save Directory for saved \*.wav files

shortcuts.txt Keyboard shortcuts

unins000.dat

unins000.exe Executable for uninstalling WSJT-X

wsjt.ico WSJT icon

wsjtx.exe Executable for WSJT-X

You might be curious about additional files that appear in the *MWSJT-X* installation directory after using the program for a while. These include:

kvasd.dat Data for the Koetter-Vardy decoder

ALL.TXT Log of all received and transmitted messages

decoded.txt Decoded text from the most recent Rx interval

timer.out Diagnostic information for decoder optimization

wsjtx.ini Saved configuration parameters

wsjtx\_status.txt Information sent to companion program JT-Alert-X

map65\_tx.log Log of all transmitted messages

prefixes.txt List of available add-on DXCC prefixes

timer.out Profile showing times in decoder routines

tmp26.txt Intermediate file used by decoder

# Appendix B: The JT9 Protocol

JT9 is a mode designed for making QSOs at MF and LF. The mode uses essentially the same 72-bit structured messages as JT65. Error control coding (ECC) uses a strong convolutional code with constraint length K=32, rate r=1/2, and a zero tail, leading to an encoded message length of (72+31) × 2 = 206 information-carrying bits. Modulation is 9-FSK: 8 tones are used for data, one for synchronization. Sixteen symbol intervals are used for synchronization, so a transmission requires a total of 206 / 3 + 16 = 85 (rounded up) channel symbols. Symbol durations are approximately (TRperiod - 8) / 85, where TRperiod is the T/R sequence length in seconds. Exact symbol lengths are chosen so that nsps, the number of samples per symbol (at 12000 samples per second) is a number with no prime factor greater than 7. This choice makes for efficient FFTs. Tone spacing of the 9-FSK modulation is df = 1 / tsym = 12000 / nsps, equal to the keying rate. The total occupied bandwidth is 9 × df. The generated signal has continuous phase and constant amplitude, and there are no key clicks.

Parameters of five JT9 sub-modes are summarized in the following table, along with approximate S/N thresholds measured by simulation on an AWGN channel. Numbers following “JT9-” in the sub-mode names specify TRperiod in minutes.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Submode | nsps | Symbol Duration  (s) | Tone Spacing (Hz) | Signal Bandwidth (Hz) | S/N Threshold\* (dB) | QSO Time (minutes) |
| JT9-1 | 6912 | 0.58 | 1.736 | 15.6 | -27 | 6 |
| JT9-2 | 15360 | 1.28 | 0.781 | 7.0 | -30 | 12 |
| JT9-5 | 40960 | 3.41 | 0.293 | 2.6 | -34 | 30 |
| JT9-10 | 82944 | 6.91 | 0.145 | 1.3 | -37 | 60 |
| JT9-30 | 252000 | 21.00 | 0.048 | 0.4 | -42 | 180 |

\* Noise power measured in 2500 Hz bandwidth.

# Appendix C: Source Code

*WSJT-X* is an open-source program released under the [GNU General Public License](http://www.gnu.org/copyleft/gpl.html). Source code is available from the public repository at <http://developer.berlios.de/projects/wsjt/>. To compile the program you will need to install open source packages for Subversion, QtSDK, qwt, g++, g95 or gfortran, portaudio, fftw3, and hamlib. For compiling in Windows I recommend installing the MinGW package.

The full source code for *WSJT-X* can be downloaded by using the command

svn co svn://svn.berlios.de/wsjt/branches/wsjtx