

SimJT User's Guide

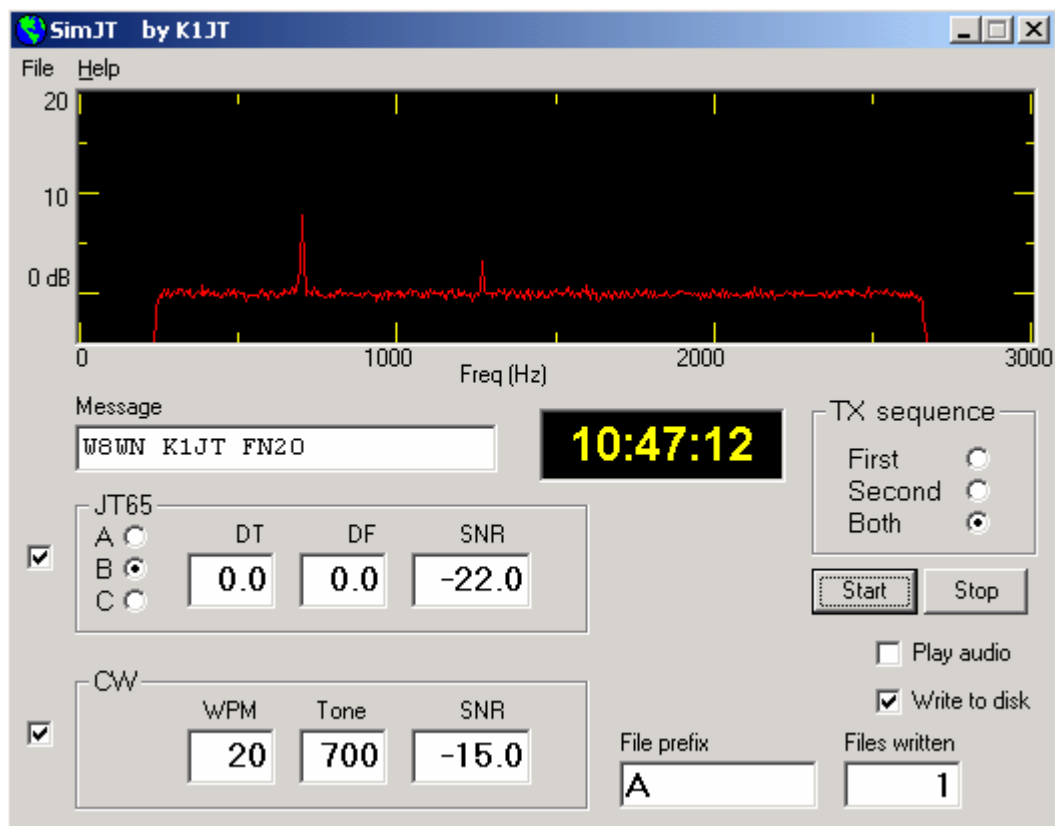
Copyright 2004 by Joe Taylor, K1JT

SimJT is a computer program designed as an aid for testing the JT65 modes of the weak-signal communication program WSJT. With SimJT you can generate JT65 test signals with specified signal-to-noise ratio, message content, and other parameters. For comparison purposes you can also generate Morse code messages with controllable parameters. The test signals can be played from one computer into another, or even back into the same computer, via the audio input/output ports. The signals can be saved on disk for later analysis by WSJT or other means.

Installation and Setup

1. Because SimJT is designed to work cooperatively with WSJT, it is assumed that you have already installed WSJT. You can download both programs and find instructions for WSJT at <http://pulsar.princeton.edu/~joe/K1JT> or the European mirror site, <http://www.vhfdx.de>.
2. Execute the downloaded file [SimJT050.EXE](#) (or a similar file with a later version number, if one exists) and install SimJT. Use the same directory in which you have installed WSJT.
3. To start the program, double-click on the desktop icon for SimJT.

On program startup you should see a screen display like the following (minus the red curve in the graphical display area):



Press the **Start** button, and after a few seconds press **Stop**. A red curve should appear as shown in the picture, and you will have generated a wave file named A00001 .WAV in your WSJT installation directory.

As indicated by information in the on-screen text boxes, the generated file contains the message “W8WN K1JT FN20” in JT65B format. It has time and frequency offsets **DT** and **DF** equal to zero, and signal-to-noise ratio **SNR** = -22 dB in a 2500 Hz bandwidth. With **DF** = 0 the frequency of the JT65 synchronizing tone is 1270.46 Hz. This signal is the weaker of the two red spikes in the picture.

The generated file also contains the same message sent repeatedly in CW at 20 words per minute, using an audio tone of 700 Hz and a key-down signal-to-noise ratio of -15 dB. This signal is the stronger of the two spikes on the red curve. Note that -15 dB in 2500 Hz bandwidth is equivalent to +2 dB in a 50 Hz bandwidth. By listening very carefully you may be able to hear and copy the CW at this level, but for most people the signal will be marginal. An external audio filter such as the ones used by many EME operators will make CW copy easier.

The red curve in the graphical display area shows the average spectrum of the generated audio file over the full transmission, plotted with 5.38 Hz resolution. The horizontal axis represents frequency from 0 to 3000 Hz, and the vertical scale indicates relative power on a logarithmic (dB) scale. Random background noise has been generated with Gaussian amplitude statistics and a flat spectrum between 200 and 2700 Hz. The spectrum has been shaped to roll off sharply below and above these limits, as can be seen in the picture.

The JT65 sync tone and the CW signal both have key-down “duty cycles” of about 50%, and most of the power of both signals is concentrated in a single frequency bin of the spectral display. The bandwidth ratio 2500/5.38 corresponds to +26.7 dB, and the 50% duty cycle adds -3 dB. Consequently, one should expect the JT65 and CW signals to appear at approximately $23.7 - 22 = +1.7$ dB and $23.7 - 15 = +8.7$ dB relative to the noise level in the spectral display—just about as observed.

The names of files recorded by SimJT begin with any text entered in the **File prefix** box. A five-digit serial number and the suffix “.WAV” will be appended to this prefix. In the illustrated example, the first files generated will be assigned names A00001 .WAV, A00002 .WAV, and so on.

To generate JT65 or CW signals alone, clear the on-screen checkbox to the left of the unwanted mode. To make real-time send-receive tests using SimJT as the transmitter and WSJT as the receiver, select the desired **TX sequence** in the box at upper right and check the **Play audio** box. For such tests you will need an audio cable to carry signals from “sound card out” to “sound card in,” the latter perhaps on a different computer.

That’s about all there is to it!