

User's Guide for PSKimmer

I wrote PSKimmer as a supplement to Digipan. I found Ham Radio Deluxe too “busy”, and I really prefer Digipan's interface. I wound up getting several new countries on PSK thanks to PSKimmer. PSKimmer will run independently of Digipan. I'm using XP Pro. (If it works, don't fix it.)

The two shortcomings of the otherwise excellent Digipan 2.0 are that the reports aren't piped out to the internet, and the log files have to be doctored up before they can be read by TQSL. The two functions aren't closely related, but I bundled them in the same executable.

To make it work, you can open the installation file, setup.msi, or, using a more direct method, put the executable (PSKimmer.exe), PSKCore.dll and PSKReporter.dll all in the same directory somewhere on your computer. Put msvcrt100.dll in c:\WINDOWS\system32. (You may also need winmm.dll, but if you're fooling with sound cards, you probably already have it.) Then, execute PSKimmer.exe, and you're off. It'll make its own registry entries. A few other things are going on in the PSKimmer directory, which I'll explain later.

The main dialog for PSKimmer has 5 buttons, 4 text areas, and a combo box.

The buttons are:

- USB/LSB toggle. This should match the setting on your rig.
- Toggle for the mode, PSK31, PSK63, PSK125, and Auto. In the first 3 modes, all 50 channels in PSKCore are allocated to the same mode. In Auto, 28 channels are allocated to PSK31, 14 channels to PSK64, and 8 channels to PSK125.
- Start/Running/Halted/No Audio toggle. This starts the receive process, and also allows you to halt the receiver while you change other settings. If the text on the button ever goes to “No Audio”, you've lost the sound card. You may have to restart the PC.
- The “Fix ADI File” button creates a file open / file save dialog pair so you can read and write a “fixed” adi file. This feature is useful because Digipan was written before the adi file specification was finalized. The files PSKimmer writes have the updated mode designators.
- The “Exit” button does the obvious.

The text areas are:

- Entry for your callsign and grid square. This data is ported to the pskreporter website.
- Entry for the dial frequency, in MHz. This should match your rig's dial frequency.
- The output window shows the callsign, frequency, baud rate, time, and signal level (00 to 99) of each incoming signal. You may select, copy, and paste the

contents.

The combo box is a drop-down for selecting the sound card. Use the same one as Digipan. They should be able to share it without fighting.

The program decodes the incoming audio and looks for “ DE “ (not case sensitive) in the incoming text stream. It then does some simple parsing to determine if what follows is a callsign. When it gets what it believes to be a valid callsign, it does a few things. First, it refreshes the output text box. It also sends a report via the internet to the pskreporter server. The last thing it does is write the raw text to “uploads.txt”. This file can be ftp-ed to a web server along with the accompanying index.html to allow you to see your own reports directly – without processing by the pskreporter server.

The reason I wanted a customized web presence was that the pskreporter server delays the reports, and also suppresses some of them. When I'm operating remotely, I would rather look at all the reports in raw format than depend on the “filtered” version returned from pskreporter. If you want to do this for your station, edit index.html so your callsign and grid are in it. Then FTP it to your web site. Also, get an FTP app that allows you to “keep a directory up to date”. I use WinSCP, which is free. Just light it off to keep uploads.txt up to date on the remote directory. The file index.html is set up to reload the page every 60 seconds. I am not a seasoned web designer, and would welcome a more polished approach!

PSKimmer has a couple features found nowhere else (to the best of my knowledge). First, it can receive in 3 different PSK baud rates simultaneously. This has been really handy working stations running PSK63, which I seldom monitor. Also, it reports the actual frequencies of the reports – not just the dial frequencies. This allows you to locate the station in the waterfall.

Dynamic linked libraries (DLL's) were already available to decode the PSK audio (PSKCore by Moe Wheatley, AE4JY mwheatley@moetronix.com), and then port signal reports to the pskreporter web site (PSKReporter by Philip Gladstone, N1DQ). Both are extremely well written and documented. All that remained was a little windows programming and some text-mashing.

I'm including PSKimmerVersion1r0.exe (version 1.0) for completeness. It uses the default sound card.

Please let me know if you have any trouble – and please share success stories, too!