Installing Winlink, VARA, and ARDOP on an M1-, M2-, or M3-based Mac

Apple Silicon Mac Winlink Installation Instructions

Updated for Winlink Hamweather Version 1.0.0.0 and Winlink Hamweather Version 1.0.0.0 and Winlink Hamweather Version 1.0.0.0

This guide will walk you through installing the Winlink, VARA, and ARDOP on Mac that you can run Winlink on the latest Mac hardware.

AUTHOR
Carl Hoover

PUBLISHED
December 27, 2022

I am a member of the Bear Mountain ARES group here in Sioux falls, and we have been encouraging our members to do emergency communications using digital modes. This seems to be important where all emergency, we want to make sure it's delivered intact and accurately. Winlink is a great emergency communications tool.

Winlink only supports Mac hardware that Apple Silicon, commonly has pretty much transitioned on a program and protocol called Winlink. There's a port for "Windows Lite," that's not my hard shack with a M1 Mac mini.

Now before everyone says "just go buy a 8 Mini Windows laptop." I'd note that 1) Hobard configure one of those to be a dedicated Winlink, 2) I'm not a fan of Windows 10 and 11 and 3) I'm not a fan of Microsoft and I'm not a fan of the rumors that come from running Windows operationally, especially during emergency ops. So I took it upon myself to come up with a Mac solution.

Now, the Winlink Mac version, unfortunately, is not a native application. It's a Windows executable for the Mac that runs in Wine. Wine is a compatibility layer that allows Windows software to run on Linux. Wine software I need to sit. Unfortunately, however, Parallels doesn't currently quite get you to a working Winlink installation. Parallels is a great piece of software, and I use it extensively on my Mac for a lot of other services, and the partial ones I phone home about regarding that. Winlink Express for Mac runs Common Winlink Clients can't properly do operational of any radio, And equally unfortunately, Winlink Express for Mac doesn't quite work with Wine. Fortunately, there's a Winlink Mac port that's based on another solution called Winlink Express. And, however, the chance of dragging along a copy of Windows just to speak a radio network protocol just feels wrong.

So, if just so to the closer, I have a working Winlink system that runs natively on my Mac. It includes a full Winlink client (including Telnet, Packet, and ARDOP support) for ARDOP and VARA HF modems, and VARA FM modems. It also includes the Winlink Express Mac port, and it's all free. It's a little clunky, I admit, and it's not as polished as the Mac, and it's missing a few things, but it's a working Winlink system, and it's free.

However, before I get to how you can install this for yourself, I'll provide a little background based on the research I did to get this to work.

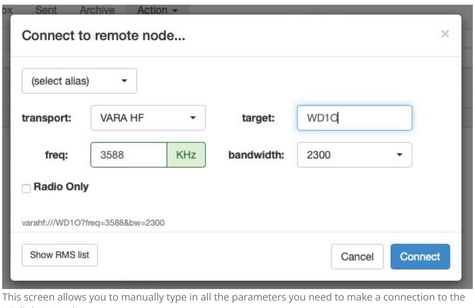
Wait: Why Not Just Use Winlink Express?

Winlink CMS Express runs in Wine's full official (sorry, typically just call it Winlink Express) is an Intel instruction set Windows only program. While Winlink Express runs just fine in a virtual machine under Parallels on an M1/M2 Mac running Intel, and Parallels VARA would speak just about fine on the Mac, Winlink Express for Mac doesn't quite work with Wine. Winlink Express for Mac is currently crashing if you attempt to use the market with it. Winlink is running Windows 11 VARA.

What about the ARM and Crossover systems for running Windows apps, Winlink Express runs just fine in Crossover, and Crossover is a great piece of software. Crossover is a commercial version of Wine, and it's a great piece of software. Crossover runs just fine on Mac Silicon and runs Winlink Express successfully in that environment. Winlink Express claims to support Apple Arm processors, but just as with Wine, Winlink Express for Mac doesn't quite work with Crossover. Winlink Express for Mac doesn't configure port right using Crossover, and it's a mess to troubleshoot. Finally, because Winlink Express was never meant to be run under Wine, it can't properly launch the modem programs.

So, what, then, is a poor ham to do? Well, I'm glad you asked. Winlink Express for Mac runs just fine in Wine, and Winlink Express for Mac can't properly launch the modem programs. Wine can properly launch Windows apps, and Wine can properly do serial port emulation, so I'm using Wine to run Winlink Express for Mac. Wine can't

Click on the *Action* menu, and there you'll be offered the opportunity to compose new emails for sending or to connect to servers. If you click on *Connect...*, you'll be shown the screen in Figure 16.



This screen allows you to manually type in all the parameters you need to make a connection to the Winlink network.

Making a Telnet connection.

The simplest possible connection is one that doesn't involve your radio at all, but that will test your `pat` configuration. Click on the dropdown menu with the *(select alias)* inside it, and you should see an alias for telnet. This alias was originally configured in your pat config.json file and if you select it, all the other parameters necessary to make a connection over the network to winlink.org should automatically fill in as shown in Figure 17 below.

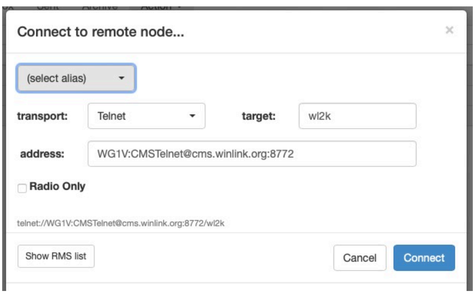


Figure 17: Clicking on the telnet alias should auto-populate all the fields you need to connect to winlink.org.

If those all look reasonable, click the *Connect* button at the bottom, and you should be connected to winlink.org and `pat` will send and receive any messages you may have queued.

Making a Radio connection.

Assuming you can make telnet connections, now it's time to test out all the other components of the system. Instead of typing in all the parameters for a connection, I recommend clicking on the box labelled *Show RMS list*. That will show a list of stations, their frequencies, their protocols, and their distances from you as shown in Figure 18 below:

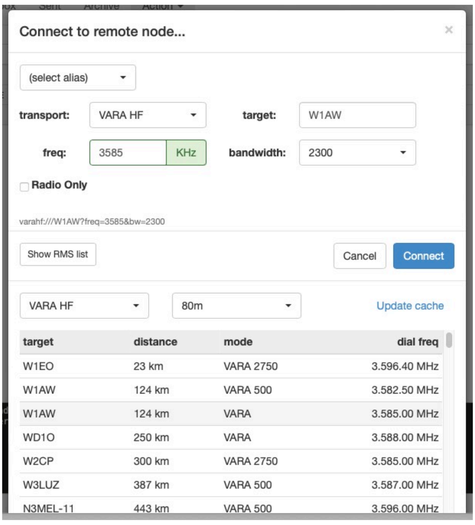


Figure 18: Clicking on the Show RMS list box will give you a list of stations to connect to.

If you single-click on any station in the list, `pat` should automatically tune your radio to the appropriate frequency, but won't make the connection until you click *Connect*. If your radio properly tunes to the frequency shown, you now know that `rigctld` properly configured for your rig.

At this point, you should check the VARA or ARDOP waterfalls to see if they are displaying whatever audio they are hearing. If they remain black, something isn't configured or connected correctly. However, if they are showing signals, now is the time to try hitting the *Connect* button.

Once you hit *Connect*, the log at the bottom of the `pat` browser window should show actions that `pat` is taking to make connections. Your radio should begin alternating between transmitting and listening for answers. If it does, then you have at least the beginnings of a working Winlink system on an M1/M2/M3 Mac!

What If It Doesn't Work?

Debugging this type of complex system is really beyond the scope of this blog post. However, let me give you some ideas for how you might continue testing your system and possibly get it to connect to other stations.

1. **Start with telnet**. The telnet test doesn't use your radio, but does test whether `pat` is properly configured and whether your user name and password for Winlink are valid. Write some test emails and try sending them to your Internet home address. Once you have that working, you can then move on to debugging the other bits of the system.
2. **Ensure your rig is in upper sideband digital mode.** Winlink always uses upper sideband modulation, regardless of band. You may also need to do something to select the USB audio input to get the signal from your computer to the radio. Some radios refer to this as USB-D or DIGU mode. Also, ensure that you have any audio equalization disabled.
3. **Look at the pat log and the `rigctld` output**. Both programs log their activities, so check their outputs to see if they are doing what you expect and if they are generating any errors.
4. **Try another station and/or another band**. Not all stations in the list are running Winlink all the time, nor is propagation always going to allow you to connect to every station. Start with stations that are close enough that you can hit them with ground wave and only target far away stations once you have everything working.
5. **Check to make sure your rig is keying up to transmit**. `Pat` may think it is transmitting, but if you don't have push-to-talk capability on your CAT interface, you may need to tell `rigctld` to use another device for PTT. Check the `rigctld` manual for how to do this. Again, `rigctld` logs every change of state for PTT, so you can check the `rigctld` console output to see if PTT requests are getting through.
6. **If VARA doesn't work, try ARDOP (and vice versa)**. The two modems have different signal characteristics, so one may work when the other doesn't.

Conclusion

Clearly, running `pat` on a M1/M2/M3 Macintosh takes a bit of work. Winlink Express and the VARA and ARDOP modems really only work in a Windows and Intel processor environment. What we've done here is to configure `pat`, a cross-platform Winlink client to use network connections to communicate with VARA and ARDOP modems and with `rigctld`, a daemon that controls your radio. We simultaneously run the VARA and ARDOP modems under Crossover, which simulates an Intel-based Windows environment for those PC-only programs. I plan to write a follow-up blog post on how to launch and use `pat` with these components. However, for the moment, I hope this how-to document provides enough information and detail for determined hams with Macs to get going with `pat` and try to get on the air with Winlink.

Carl WG1V

Footnotes

1. If you own an Intel-based Mac (I still have a 2012 Macbook Pro that still works great, so they still are out there), you have some advantages. You can run Winlink Express under the Windows in a Parallels virtual machine, and it will work properly. You also have a greater choice of which Crossover or WINE versions you use because you don't require instruction set translation. Regardless, I believe that the process described in this document will work just fine on an Intel-based Mac.↩