

Turning Out the Lights in DOS

I thought I was the only contender still using DOS.

My professional life had not left a lot of time for operating or computers, and by the time I returned to contesting in late 2008 I was sure the world had passed me by. Software wasn't the problem. At the urging of Jim Neiger, N6TJ, I had joined the *TR LOG* camp a decade earlier and was comfortable with (addicted to?) its capabilities, especially in the fast moving world of SO2R SOAB CW contesting. Truth be told, the old dog was *not* excited about learning new tricks.

Goodbye, DOS and LPT. Hello, Windows and USB

The main factor driving the need for change was hardware. Some of you will recognize the signs. All of my operating was as a hired gun, and I took along my own laptops and trusty DX Doubler. From an earlier stint in the private sector, I had grown partial to the HP Omnibook 500 series (small and light; think netbook, but a decade ago) and had become best friends with PCMCIA serial port adapters, *Partition Magic*, docks with parallel (LPT) ports and the like. Now, however, these familiar companions were disappearing, one by one.

Feeding my long-suppressed inner geek, I went on an eBay buying spree and ended up with eight Omnibooks in various states of repair, plus an assortment of serial and LPT adapters. My wife accused me of having "computers on blocks," but at least they weren't sitting in the front yard. All that stemmed from the bleeding, but when Tree's (N6TR, *TR LOG*'s developer) stable of DOS experts threw up their hands over a PCMCIA issue, I knew the end was near.

Conversations with some top contesters at Dayton last spring at least provided some consolation that I wasn't alone, but these did not yield many easy solutions. For example, Mark, KD4D, who's a computer whiz, takes his own PCI card to the Caribbean and installs it on the host's desktop, along with his trademark LINUX-based partition program, on which he installs DOS. My skill set is such that this sort of approach would represent a "high-risk option."

So what to do? My goal was simple. I wanted to be able to do competitive SO2R CW using a modern laptop with only USB ports and a *Windows* contesting program.



Figure 1 — In a last-gasp effort to keep DOS going, a pile of Omnibook laptops were purchased on eBay.

Using the familiar DX Doubler would be a bonus.

I messed around with several *Windows* programs. *TR4W* was the obvious solution, but it required a *real* LPT port (while USB-to-serial adapters work well for contesting applications, USB-to-LPT adapters do *not*). I actually used *TR4W* in the 2010 IARU with a decent outcome (number 2 SOAB HP CW in US). As several people had warned me, however, I found the software clunky and a bit erratic in SO2R. Besides, much of the support details were in Russian.

I also played with *N1MM Logger* and *Win-Test*, but both seemed to be "CT-like," requiring me to relearn SO2R and give up the unique "Alt-D" *TR LOG* function, which I personally find the slickest and smoothest for fast-paced SO2R.

The catalyst for change came when Pete, N4ZR, decided to abandon SO2R [see "*Op Ed: Why I Went Back to SO1R*," by Pete Smith, N4ZR, elsewhere in this issue — Ed] and put his PIEXX SO2Rxlat up for sale on the PVRC reflector. This is

a USB device that's specifically designed to emulate the LPT functions required for CW keying, PTT, and — critically — for A/B radio switching. Pete had been the *N1MM Logger* beta tester for PIEXX when it was developing the SO2Rxlat, and he remains an *N1MM* guru, so I got a personal hardware consultant in the deal.

For software, my local host, Jim, W4RX, uses *Win-Test*. When the WRTC survey indicated that the majority of contestants were using *Win-Test*, I decided to start down that path and joined the *Win-Test* support group reflector. After a few challenging weeks, two breakthroughs occurred.

The initial challenges involved the hardware. I couldn't get the SO2Rxlat to key through the LPT port output, and for some reason there was a one second delay in transmitting on the second radio. Roger, VE3ZI, had joined the discussion and, while he wasn't having the first problem with his SO2Rxlat, he was seeing the delay. Enter Chris of PIEXX. Unbeknownst to Pete or me, Pete's SO2Rxlat was serial #1

or very close to it and needed a few modifications that had been incorporated in later versions. Chris took care of these in short order. That solved the keying issue. The delay issue proved to be more problematic. Again Chris jumped in and, working with *Win-Test* developer Larry, F6FVY, modified the firmware in short order to work with the open two-radio software protocol (OTRSP) developed by K1XM. *Hardware problems solved!* (Owners of earlier versions can download and flash existing units with the new firmware. Instructions are available on the PIEXX site, www.piexx.com.)

Meanwhile, on the software front, I had been corresponding with Bob, N6TV, both on and off the *Win-Test* reflector about some of the hardware issues, including the PIEXX modifications. I had met "TV Bob" at Visalia the year before and was greatly impressed with his rapid and effective response to questions on the reflector, which spanned the gamut from kindergarten to post-doctoral level. Bob makes the Energizer Bunny look like a piker. In Navy terms, N6TV is a true can-do guy who keeps a million balls in the air.

One day Bill, K5GA, a former *TR LOG* user who had heard of my quest, sent me an e-mail (after a frustrating NAQP experience with *Win-Test*) with the subject line "I have to have Alt-D!" I forwarded the e-mail to Bob who said, in effect, "We can do that." Bob promptly developed a first version of a *TR LOG*-like option for *Win-Test*, and the game was on. By request and on his own initiative, Bob developed a series of scripts to emulate key *TR LOG* functions within *Win-Test*. He even dug out an ancient post dealing with the best DX Doubler jumper settings to utilize the flexibility of *Win-Test*.

The acid test came during the 2011 ARRL International DX CW, with a number 1 US claimed score, operating SOAB HP as KØDQ from WW1WW and fewer than 100k points behind VY2ZM. The software played extremely well, and its operation was essentially the same as *TR LOG*'s.

Thanks to some great efforts and no small amount of teamwork, DOS has finally been relegated to the dustbin of contesting history for this operator. Most important, the core functionality not only has been retained, but the bells and whistles of *Windows* in general and of *Win-Test* in particular are sweet additions. Finally, parallel and serial ports are *out*, and USB ports rule. Anyone want to buy some old laptops...cheap?

Operating SO2R with Alt-D

So what's this "Alt-D" stuff, and why is it a big deal?

I'll concede that *TR LOG*, like single-malt scotch, is a preference, an acquired taste. Not all operators know or embrace it.



Figure 2 — Using the PIEXX SO2Rlat in conjunction with a serial-to-USB adapter, *Win-Test* was able to support the LPT-based DX Doubler. In addition, serial connections were available for CAT control of two radios.

Indeed, one of the world's premier SO2R operators, CT1BOH, was a major contributor to the development of *Win-Test*, and its stock SO2R functionality reflects his personal preference. But there's a good reason *TR LOG* remains popular. For those not of the *TR LOG* tribe, here, in a nutshell, is how it works.

The cardinal rule 1 of SO2R is *always* to be calling "CQ." *Always!* Cardinal rule 2 is always to be searching on the second radio for multipliers and QSOs. Here's how Alt-D works in a typical scenario.

You are CQing on your primary run radio, while searching continuously on the second radio. If the pileup on you is light, you can, with practice, run the pileup listening to one radio in each ear. If the pileup is heavy, you most likely will want both ears on the run radio, *except* when you are transmitting on it. Even then, you continue to search on the second radio during those brief periods, locating stations to work as the pileup allows.

You find a station you want to work on the second radio. Type "Alt-D," and a window opens. Enter the call sign of the station and hit Enter. The call sign is stored on deck, and you're primed to pull the trigger. You also can do this while the primary radio is calling "CQ" or sending a report.

Continue working the run on the first radio. When the timing is right, simply hit the space bar. This will send your call sign on the second radio. If the station responds, enter/verify the exchange and hit Enter.

This sends your exchange on the secondary radio, returns you to the run radio, and calls a quick "CQ" on the run radio.

If the station does *not* respond, you can (1) call again by pressing F4, or (2) hit Esc to return to your run on the first radio. In the latter case, the call will remain teed up for another call whenever you hit the space bar.

Alt-D makes it easy, even when you're dead tired and borderline sloppy. With practice, this becomes smooth and almost effortless. There are further refinements, but the above procedure will get you close enough to see if you like it.

Giving *Win-Test* a Try

Sound interesting, but complex? Don't want to spend a week figuring out the right boxes to check to set up *Win-Test*? Don't want to fork over \$\$\$ for something you may not like?

One of the biggest obstacles I faced in looking at the various *Windows* software was wading through all the setup options, so I could at least play with it and begin to understand the capabilities. That refrain is a perennial favorite on reflectors following many contests. Providentially, there's a way to test drive (or fly) *Win-Test*, and it requires about a half-hour of setup on your computer and no cash outlay.

A template available on the "Bonus Content" section of the *NCJ* Web site, www.ncjweb.com/bonus.php, will allow you download a trial copy of *Win-Test*, set it up

on your computer and play with the basic features — most notably data entry and the Alt-D function. *Win-Test* does not have a “simulator” mode (as *TR LOG* does), but the Secondary Radio window in *Win-Test* will allow you to watch the Receiver and Transmitter shift from Radio 1 to Radio 2, as you go through the logging process. Turn on your radio or find some contest audio on the Web and practice data entry in a simulated SO2R environment, and you should get enough of a sense of whether you like it or not.

Section 1 will lead you through computer setup to test drive *Win-Test*. Note that these are “KØDQ-preferred” options for the way I operate SO2R contests, but they’re close enough to the norm. Not surprisingly, a very rich menu allows you to tailor just about anything you want.

The bottom line is that *Win-Test* now has both *CT*-like and *TR LOG*-like features, depending on your personal preferences. To be clear, this is not an exact replica or complete menu of all the features of *TR LOG*. For example, one feature I miss is the “Shift key enable,” which allows tuning the RIT on certain radios up or down by using the left and right Shift keys. You can do this with *Win-Test*, but at this point this requires holding down two keys (eg, Shift + up arrow). I can live with that. Others may have other preferred features. Also, to be fair, it may well be that *N1MM* and other programs are similarly adaptable to have a *TR LOG*-like option.

If you like what you see, Section 2 leads you through hardware configuration for several scenarios involving (1) the DX Doubler (or other legacy SO2R box with a parallel port interface), (2) either a legacy LPT port or the PIEXX SO2Rxlat, and (3) configurations for two representative radio combinations ($2 \times K3$ or $2 \times FT-990$). Again, there are other hardware options, including SO2R boxes designed from the ground up for USB ports.

That’s it — a relatively painless effort to see if it appeals to you. If not, you’re wiser and out only an hour or so. If you like it, maybe you too will be able to chuck that old DOS box.

NCJ