

## Ease of use Beta 6 DriveWire editions

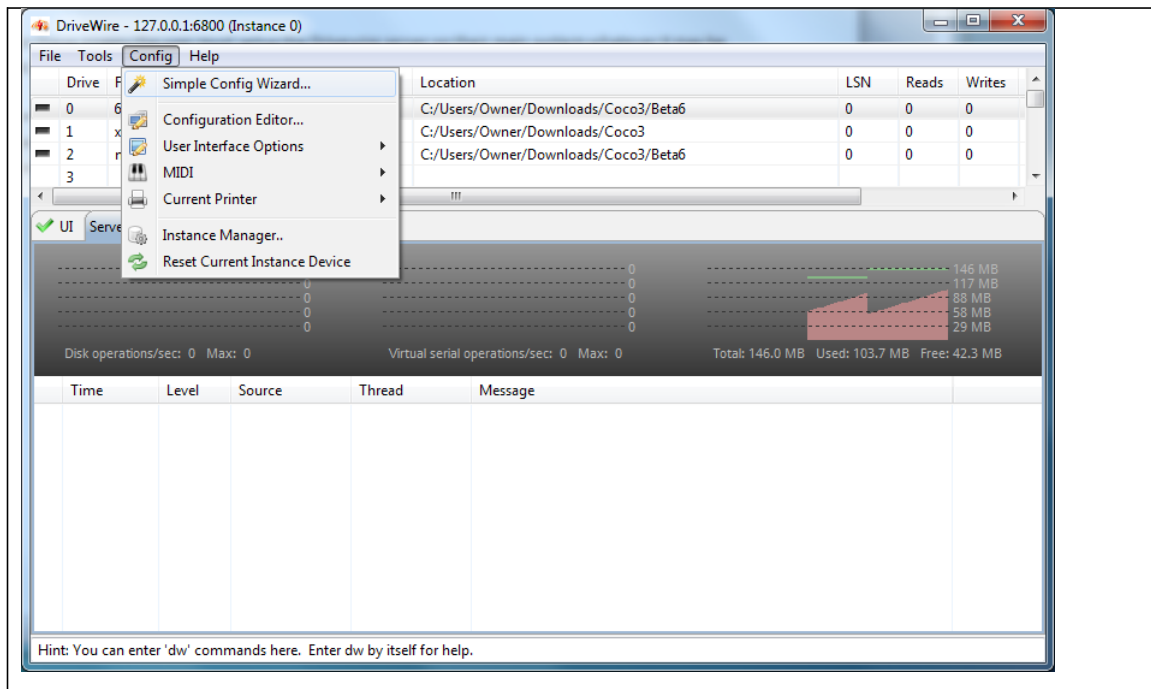
Ease of use has now added DriveWire clients to it's available images. To get one of the flavors of DriveWire going, you must first install a DriveWire server (below you will find instructions for both **DriveWire 4** (which requires Java) and **pyDriveWire** (which requires Python)). After getting the server installed, you must then set up the proper Boot file set to run it on the Coco (see the instructions for the **SWAPBOOT** command at the end of this document).

Here is a quick overview of how to install and set up **DriveWire 4**:

- ☐ Adds access to DriveWire servers drives 0-3 as /X0-X3
- ☐ Keeps access to SDC controllers SD card as /H0-/H1
- ☐ Adds full Midi support to DriveWire servers Midi access.
- ☐ Full DriveWire printer support.

For these images the user must setup the DriveWire server on their main system whatever it may be (Mac, Windows, Linux etc...). To do this follow these simple steps for use on a real Coco 3:

- ☐ Download the DriveWire server at: <https://sourceforge.net/projects/drivewireserver/files/>
- ☐ Install on what the machine you will use as the DriveWire server.
- ☐ Configure using the Simple Config server option:

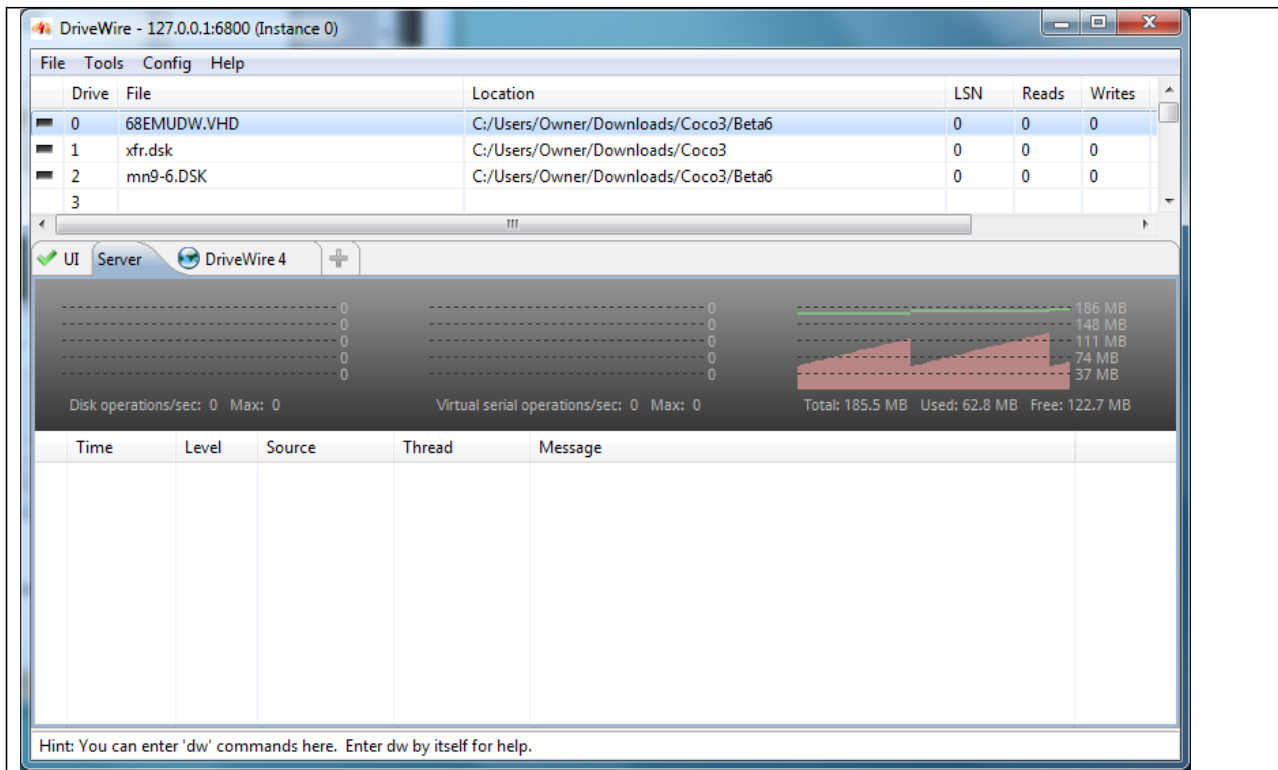


- ☐ Set the server as a Coco 3:

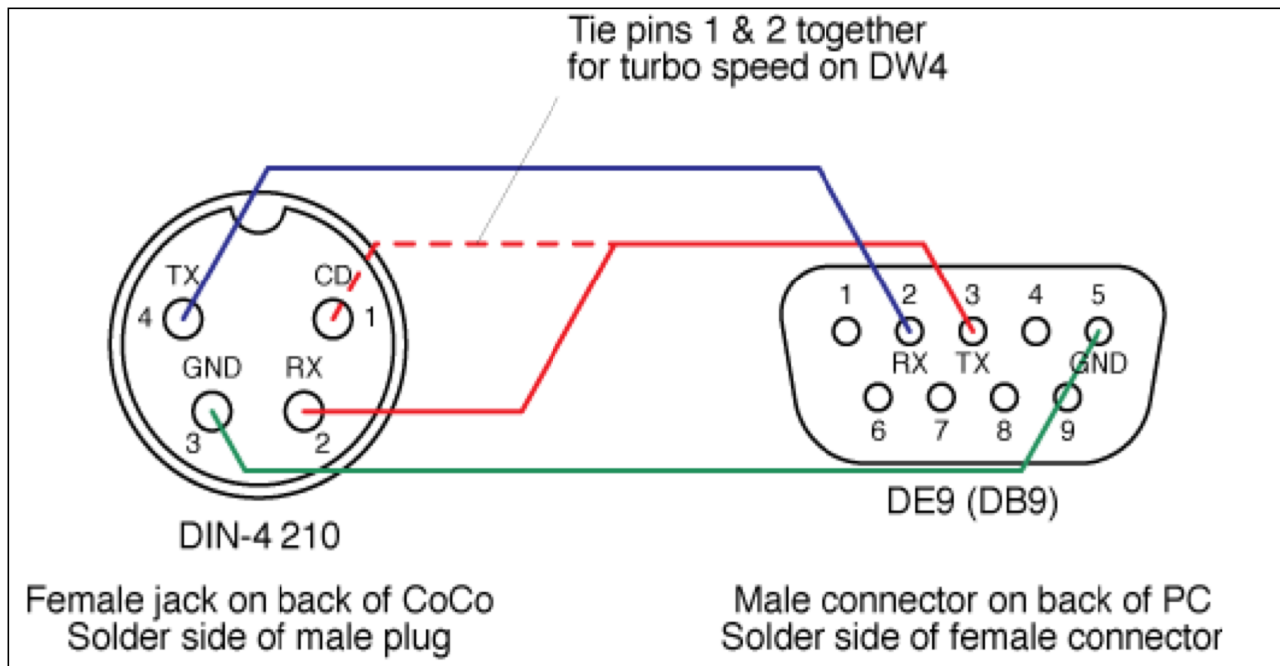


- ☐ Select the proper Com/TTY port for you client machine to use.
- ☐ That are the only things needed to be done on the Server side. The rest of the Simple Config Wizard you would just respond yes to.

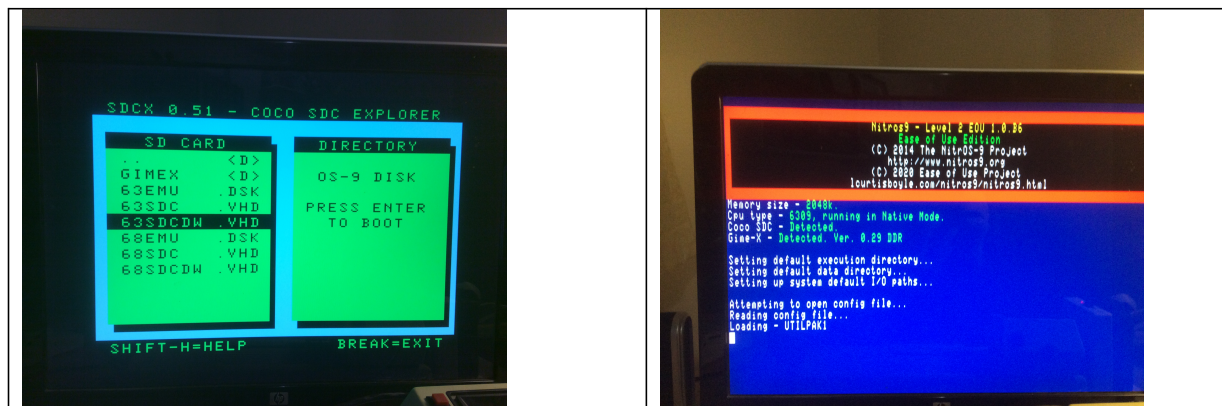
- ❑ Once complete you can now add drives from 0-3 in the Server that Ease of use Beta 6 will recognize (once you run **SWAPBOOT** - see **EOU Update notes for Beta 6, item #53**).



- ❑ Next on your Coco 3 you will need the standard DriveWire cable from the 4 pin DIN to a serial FTDI adapter to the server ready and working.



- ☐ DriveWire 4 cable support will be coming soon. This hasn't been fully tested yet.
- ☐ Once that is ready and connected you can now boot into Ease of use on your Coco 3 using one of these images:
  - o 63SDCDW.VHD
  - o 68SDCDW.VHD



- ☐ Once selected hit enter, or mount it and type DOS.



Here is a quick overview of how to install and set up **pyDriveWire**:

**INSTALL FIRST:** pyDriveWire Installer for Windows, MacOS, Linux, and Raspberry Pi OS:

<https://github.com/n6il/pyDriveWire/releases/> Once pyDriveWire is installed, and you've run a drivewire cable to a USB to serial adapter, follow these beginner instructions:

1. \*Windows: Open a Command Prompt (with admin rights) from the folder where pyDriveWire.exe lives and paste in:  
**pydrivewire --ui-port 6800 --port COM4 --speed 115200**
2. \*\*Open browser to load your DSKs and VHD (Bookmark it):  
<http://localhost:6800/>
3. Click on "x" Drive Letter from Gshell, dir /x0, x1, etc. Enjoy!

More advanced users can set up a config file, and/or auto-mount disks at launch – see PyDriveWire instructions for more details.

### Notes:

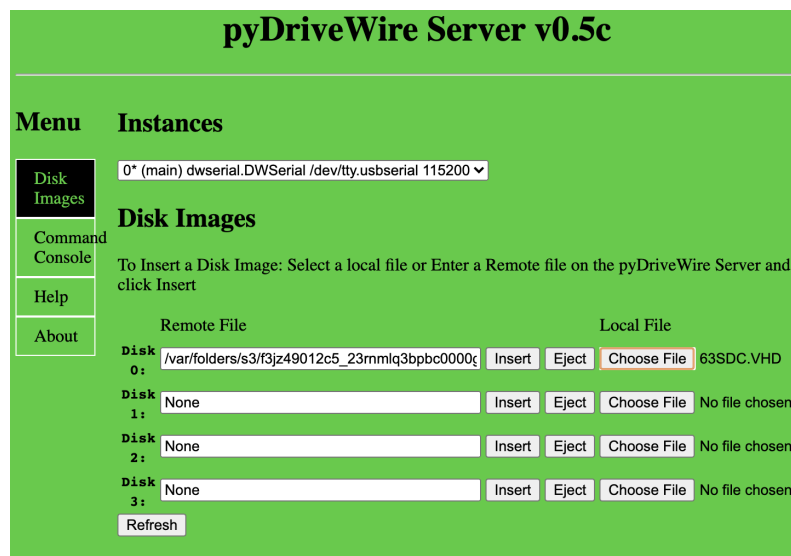
If necessary, replace the number 4 with your COM[#] port used – shown next to your USB Serial Adapter in Windows Device Manager.

\*MacOS/Linux/Pi users: replace **COM4** above with: **/dev/[devicename]**. Therefore, paste and edit the following command line in Terminal:

**Pydrivewire --ui-port 6800 --speed 115200 --port /dev/tty.usbserial**

Finding your device name: To do this, open a Terminal window, type: **ls /dev/tty.usb\*** and press enter. The device name in the MacOS example above is **tty.usbserial**.

\* \*



### Emulators:

*Becker Port (MAME/VCC/OVCC) Example: Run pyDriveWire with the HTTP UI on port 6800 and the Becker port connection on port 65504:*

***pyDriveWire --ui-port 6800 --accept --port 65504***

### DriveWire server Special Notes:

There are several versions of DriveWire servers available, each with their own unique characteristics. The ones named "DriveWire" require Java, but if Java is not something you have (or want) on your system, pyDriveWire is an alternative that is written in Python. Capabilities wise, it is in between DriveWire 3 and DriveWire 4. The table below shows the capabilities of each server:

Server Software	Windows	OSX	Linux	Virtual Drives	Remote files	TCP/IP	Printer	Real time clock	MIDI	Virtual terminals
DriveWire 3	X	X	X	X			X***	X		
DriveWire 4	X	X	X	X	X	X	X	X	X	X
pyDrivewire	X*	X*	X*	X	X	X	X****	X		

#### NOTES:

\* Requires Python installed on the host OS.

\*\* Printer to text file only

\*\*\* Printer to text file or emulated Epson FX-80 Dot Matrix Printer

\*\*\*\* Printer to text file or PDF only

### SwapBoot utility

**SwapBoot** is a utility that lets you swap between not only **OS9Boot** (which is the main operating system, including all hardware drivers) file itself, but also your boot environment (which includes the startup file script, and the your env.file (environment file), including the new enhancements to the environment added in **EOU Beta 6**. There are no command line parameters in **SwapBoot**; it is presented by menus, making it easier to use. It also checks to make sure that all 3 parts of a “set” are present before it lets you select them, thus cutting down on issues of incomplete sets.

**SwapBoot** does this by checking for complete sets of 3 files (each with the same matching extension & not case sensitive):

OS9Boot.<boot set name>  
startup.<boot set name>  
SYS/env.file.<boot set name>

If any of these are missing, it will not let you select them. If it finds a complete set, it will add it to the interactive menu of boot sets that you can choose from (maximum of 20 at any given time). You can then pick one to be your next active boot, and it will set up the hard drive image for booting to the selected set. It will then reboot the Coco to DECB, where you can type DOS to boot NitroS-9 with your new settings. By default, **EOU Beta 6** contains four sets:

**Standard SDC (.sdc)**

**SDC with DriveWire (.dw)**

**Emulator with software clock (.emusoftclock)**

**Emulator with hardware clock (.emucloud9clock)**

You can add your own, which can be special hardware drivers, or slimmed down boot files with less windows for a memory hungry program. It could be the same OS9Boot (just copied over) but with a custom **startup** file with less fonts, different windows launched, and/or custom **env.file** with different boot screen settings, colors, mice and keyboard settings, etc. Please note that any sets that contain ‘emu’ in it’s extension will be interpreted as being for the emulators (**VCC** or **MAME**), and the OS9Boot files in those cases apply to the **d0** floppy drive (68EMU.DSK or 63EMU.DSK). All others are assumed to be for the SDC, and will apply to the **dd** hard drive (68SDC.VHD or 63SDC.VHD). Startup files and env.file’s are always done on the hard drive image (**dd**).

This is still considered somewhat experimental, but seems to be working across multiple beta testers. Please report any bugs you find to: [curtisboyle@sasktel.net](mailto:curtisboyle@sasktel.net).

It should be mentioned here that you must have a **DriveWire** server running before you boot with the DriveWire set; the DriveWire set will not only let you access DriveWire virtual drives, but also get the date and time from the DriveWire server (so you don’t have to type it in by hand anymore).