Shared Folders in FreeDOS

If you are into retrocomputing, like me, you may sometimes want to run DOS-era software or games in their original environment. Instead of installing an old MS-DOS version, consider using something more modern, such as FreeDOS (FD). I have known FD for many years, but I remember that the last time I used it as a VM in VirtualBox (VB), it wasn't a fun experience because I couldn't establish a good mechanism to exchange files with the DOS environment.

So I decided to give it another try, installing FD in a VB VM. Here I'll share my most recent findings on how to have a *shared folder* between your host machine and a virtual one running FD.

VirtualBox has a great feature called *Guest Additions* that, among other things, allows you to share a folder in the *Host* machine with the *Guest*. It works like this:

```
+---------------+           +----------------+
| Host Machine  |           | Guest Machine  |
| Win 11        |           | Another OS     |
| +-------------------+     +----------+     |
| | Folder 'Share' | <---> | Unit 'Z' |     |
| +-------------------+     +----------+     |
+---------------+           +----------------+
```

The host machine has a folder, which I called *Share*, but one can specify any name, which can be mapped as a new storage unit inside the guest machine. This works perfectly when both Host and Guest OSs are supported by the *Guest Additions*. FD is not supported.

Searching on the internet, you'll find some approaches like setting up a floppy disk image or a CD iso. Other approaches needed more configuration, like setting up an FTP server or sharing a folder within the network. Then, I found the perfect solution: a community-written driver[1] for DOS-like systems to support VB Shared Folders!

The installation process is quite simple. I created a folder (`C:\VBSF`) in my VM to copy all the files from the floppy disk image. Next, you have to add `LASTDRIVE=Z` to the `CONFIG.SYS` file. This file is for MS-DOS, in FD we have `FDCONFIG.SYS` which is in the root of drive `C:\`. Using `edit FDCONFIG.SYS` I added the line to the very end of the file.

Every time you need to use the driver, you must run `VBSF.EXE install low` or just `VBSF.exe`. When you do this, the shared folder will be available inside FD under the drive `Z:`. You should test to make sure that everything is working as expected. Just type `Z:` to change the drive and then `dir` to confirm its content.

I don't have plans to change the shared folder or drive, so to avoid having to type the command all the time, I decided to add it to the `FDAUTO.BAT` (which is similar to the MS-DOS `AUTOEXEC.BAT`), and it is executed after the boot. Once again, I used `edit FDAUTO.BAT` and added the line `C:\VBSF\VBSF.EXE`. `VBSF\` is the folder that I created to copy all the files from the floppy disk. The command was added right before the `:END` label, as shown in Fig. 1.

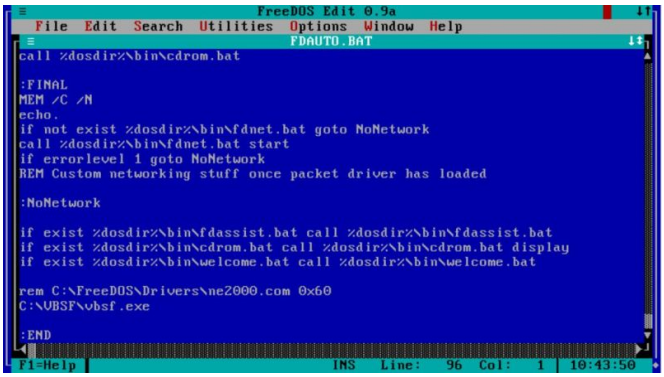

Figure 1: Editing `FDAUTO.BAT`.

Now, reboot the system to see if the auto-mount works. Type `reboot` for that.

After the boot, below the welcome message, you should see the message from the driver. In my case, it starts with `VBSharedFolders 0.67...` see the Fig. 2. The message acknowledges that the shared folder has been mounted as drive `Z:`.

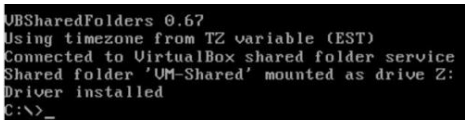

Figure 2: Shared folder auto-mounted.

You should check it once again using the `dir` command. If you want to have more fun, try to run a DOS-era software or game. This time, I checked by running the **IBM AntiVirus v.2.5.1** that was in the shared folder. Everything worked as expected, including the loading of the virus definitions file.

Figure 3: *IBM AntiVirus v.2.5.1* running from a shared folder

That's it, a simple setup! Have fun!

[1] Javier S. Pedro - https://git.javispedro.com/cgit/vbados.git/about/#vbsfexe-shared-folders