

Why does each drive have its own current directory?

 devblogs.microsoft.com/oldnewthing/20101011-00

October 11, 2010



Raymond Chen

Commenter Dean Earley asks, “Why is there a ‘current directory’ AND an current drive? Why not merge them?”

Pithy answer: Originally, each drive had its own current directory, but now they don’t, but it looks like they do.

Okay, let’s unwrap that sentence. You actually know enough to answer the question yourself; you just have to put the pieces together.

Set the wayback machine to DOS 1.0. Each volume was represented by a drive letter. There were no subdirectories. This behavior was carried forward from CP/M.

Programs from the DOS 1.0 era didn’t understand subdirectories; they referred to files by just drive letter and file name, for example, `B:PROGRAM.LST` . Let’s fire up the assembler (compilers were for rich people) and assemble a program whose source code is on the A drive, but sending the output to the B drive.

```
A>asm foo the “.asm” extension on “foo” is implied
```

```
Assembler version blah blah blah
```

```
Source File: F00.ASM
```

```
Listing file [F00.LST]: NUL throw away the listing file
```

```
Object file [F00.OBJ]: B: send the object file to drive B
```

Since we gave only a drive letter in response to the `Object file` prompt, the assembler defaults to a file name of `F00.OBJ` , resulting in the object file being generated as

```
B:F00.OBJ .
```

Okay, now let’s introduce subdirectories into DOS 2.0. Suppose you have want to assemble `A:\SRC\F00.ASM` and put the result into `B:\OBJ\F00.OBJ` . Here’s how you do it:

```
A> B:
B> CD \OBJ
B> A:
A> CD \SRC
A> asm foo
Assembler version blah blah blah
Source File: F00.ASM
Listing file [F00.LST]: NUL
Object file [F00.OBJ]: B:
```

The assembler reads from `A:F00.ASM` and writes to `B:F00.OBJ`, but since the current directory is tracked on a per-drive basis, the results are `A:\SRC\F00.ASM` and `B:\OBJ\F00.OBJ` as desired. If the current directory were not tracked on a per-drive basis, then there would be no way to tell the assembler to put its output into a subdirectory. As a result, DOS 1.0 programs were effectively limited to operating on files in the root directory, which means that nobody would put files in subdirectories (because their programs couldn't access them).

From a DOS 1.0 standpoint, changing the current directory on a drive performs the logical equivalent of changing media. "Oh look, a completely different set of files!"

Short attention span.

Remembering the current directory for each drive has been preserved ever since, at least for batch files, although there isn't actually such a concept as a *per-drive current directory* in Win32. In Win32, all you have is *a* current directory. The appearance that each drive has its own current directory is a fake-out by `cmd.exe`, which uses strange environment variables to create the illusion to batch files that each drive has its own current directory.

Dean continues, "Why not merge them? I have to set both the dir and drive if i want a specific working dir."

The answer to the second question is, "They already are merged. It's `cmd.exe` that tries to pretend that they aren't." And if you want to set the directory and the drive from the command prompt or a batch file, just use the `/D` option to the `CHDIR` command:

```
D:\> CD /D C:\Program Files\Windows NT
C:\Program Files\Windows NT> _
```

(Notice that the `CHDIR` command lets you omit quotation marks around paths which contain spaces: Since the command takes only one path argument, the lack of quotation marks does not introduce ambiguity.

Raymond Chen

Follow

