

Did the Windows 95 setup team forget that MS-DOS can do graphics?

 devblogs.microsoft.com/oldnewthing/20250211-00

February 11, 2025



One of the reactions to my discussion of [why Windows 95 setup used three operating systems](#) (and oh there were many) was my explanation that an MS-DOS based setup program would be text-mode. But c'mon, MS-DOS could do graphics! Are you just a bunch of morons?

Yes, MS-DOS could do graphics, in the sense that it didn't actively prevent you from doing graphics. You were still responsible for everything yourself, though. There were no graphics primitives aside from a BIOS call to plot a single pixel. Everything else was on you, and you didn't want to use the BIOS call to plot pixels anyway because it was slow. If you wanted any modicum of performance, you had to access the frame buffer directly.

Okay, so now you have to write a graphics library for drawing things fancier than a single pixel. Fortunately, Windows 95 required a VGA video card at a minimum, so didn't have to worry about CGA or EGA. Mind you, the VGA adapter required you to deal with planar modes, so that was annoying. Fortunately, you have a team of folks expert in VGA planar modes sitting down the hall working on Windows video drivers who can help you out.

But the setup program needs more than just pixels. It also wants dialog boxes, so you'll have to write a window manager to sit on top of your graphics library so you can show dialog boxes with a standard GUI dialog interface, which includes keyboard support for tabbing between elements and assigning hotkeys to fields.

You'll also have to add support for typing characters in non-alphabetic languages like Japanese. Fortunately, you have a team of folks expert in Japanese input sitting in the Tokyo office working on Windows input methods who can help you out, though the time zone difference between Tokyo and Redmond is going to slow you down.

You also want to take advantage of those fancy new controls that the UI team has been making, so maybe you can walk down the hall and ask them if they could port their controls library to your custom UI framework.

The setup program also wants to do simple animations, so you'll need a scheduler that can trigger events based on the system hardware timer.

So now you're going to write all this code for your setup program, none of which is actually involved in setting up Windows 95, but is just the infrastructure needed to run the setup program at all! There's a lot of stuff here, and you probably won't be able to cram all of it into 640KB of memory. So now you need to write a protected mode manager (also known as an MS-DOS extender) so you can take advantage of the larger address space afforded by protected mode.

Now take a step back and look at what you're doing. You're writing an operating system. (Or, if you are being uncharitable, you're writing an MS-DOS shell.)

An operating system with exactly one application: Windows 95 Setup.

What if I told you that Microsoft already had an operating system that did all the things you are trying to do, and it's fully debugged, with video drivers, a graphics library, a dialog manager, a scheduler, a protected mode manager, and input methods. And it has a fully staffed support team. And that operating system has withstood years of real-world usage? And Microsoft fully owns the rights to it, so you don't have to worry about royalties or licensing fees? And it's a well-known system that even has books written about how to program it, so it'll be easier to hire new people to join your team, since you don't have to spend a month teaching them how to code for your new custom Setup UI miniature operating system.

Go and grab a copy of the Windows 3.1 runtime.

Bonus chatter: If you committed to the custom operating system route, you'd have to make sure your miniature operating system could run in a Windows 3.1 MS-DOS session in case somebody wanted to install Windows 95 as an upgrade from Windows 3.1, and in a Windows 95 MS-DOS session in case somebody wants to do a repair install of Windows 95. And then you'd have this weird setup experience where Windows 95 setup is running inside an MS-DOS session.

Bonus bonus chatter: Windows setup still follows this pattern of installing a miniature operating system to bootstrap the setup program. But today, the miniature operating system is Windows PE, the Windows Preinstallation Environment.

Related reading: [Dear sir, you have built a compiler.](#)