

Why are there separate Program Files and Program Files (x86) directories?

devblogs.microsoft.com/oldnewthing/20220329-00

March 29, 2022



Raymond Chen

On Windows editions that support x86 emulation, there are two directories for program files. The `C:\Program Files` directory is for programs in the native system bitness, and the `C:\Program Files (x86)` directory is for programs that run under the x86-32 emulator. But why are they separate directories? Why can't we combine them? If you have 32-bit Contoso and 64-bit LitWare, you can still put them in `C:\Program Files\Contoso` and `C:\Program Files\LitWare`, and they won't bother each other.

That's true, but it does create problems if a program is available in both 32-bit and 64-bit versions, such as Microsoft Office or Visual Studio. You couldn't install both versions side-by-side; you'd have to pick one.

And it so happens that Windows itself comes with a lot of programs that are available in both 32-bit and 64-bit versions, like Internet Explorer and WordPad. Those programs could have installed themselves into separate directories like `C:\Program Files\Internet Explorer` and `C:\Program Files\Internet Explorer (x86)` to avoid the conflict, but then that runs into compatibility issues with apps that do things like launch `%ProgramFiles%\Internet Explorer\iexplore.exe` and then start manipulating the Internet Explorer process expecting it to be the same bitness as the program that did the launching.

This sounds sort of like a lazy answer, seeing as this problem could be solved with sufficient application of application compatibility shims. But application compatibility shims are really a solution of last resort, because by their nature, they can be applied only to programs where the problem has already been identified. If there's a way you can structure your system so that these problems never arise in the first place, then you don't have to worry about identifying the programs that suffer from the problem: You fixed all of them at once!

Yet another problem is with the pesky `Common Files` subdirectory under `Program Files`. This is a directory that is intended for the case of separately-installed programs which nevertheless share some things in common. For example, Contoso might have two

products, Contoso Standard and Contoso Deluxe, and both of the programs use the Contoso Framework. The Contoso Framework files can go into the `C:\Program Files\Common Files\Contoso` subdirectory.

If the 32-bit and 64-bit directories were combined into a single `C:\Program Files` directory, then you would have a mix of 32-bit and 64-bit components in `C:\Program Files\Common Files\Contoso`, and that makes nobody happy. Whoever installs first wins, and whoever installs second gets errors.

Keeping the two Program Files directories separate avoids this problem. `C:\Program Files\Common Files\Contoso` contains the 64-bit Contoso Framework files, and `C:\Program Files (x86)\Common Files\Contoso` contains the 32-bit Contoso Framework files.

After I wrote up this answer, I realized that I already answered it some time ago, so here's the video:



Watch Video At: <https://youtu.be/qRb6otsHG5c>

[Raymond Chen](#)

Follow

