

# How can I find out which processor architectures are supported via emulation by the current system?

[devblogs.microsoft.com/oldnewthing/20220209-00](https://devblogs.microsoft.com/oldnewthing/20220209-00)

February 9, 2022



Raymond Chen

A customer was writing a debugging tool and wanted to know in their installer which processor architectures are supported by the current system, both the native architecture as well as anything supported by emulation. That way, the tool could install the versions that apply to those architectures.

Okay, getting the native architecture is easy. You can call `GetNativeSystemInfo` to find out what the native system architecture is.

Getting the emulated ones is a little trickier. Back in the old days, you just hard-coded some knowledge. “Well, Windows on x86-64 can emulate x86-32. And Windows on AArch64 can emulate x86-32 as well as T32.” But that broke down with the introduction of HoloLens 2, which is natively AArch64 and omits the x86-32 emulation, and then broke down further with the introduction of x86-64 emulation on native AArch64 systems.

To find out whether a particular architecture is supported in emulation, you can pass it to `IsWow64GuestMachineSupported` and see what the answer is. Repeat for each architecture you are curious about.

There is no way to get a list of all the architectures supported by emulation, primarily because there’s no real use for it. I mean, suppose you learn that Windows now supports architecture number 31415. You have no idea what that number means, because it’s not on your list of known architectures. Even if you somehow figured out that 31415 is `IMAGE_FILE_MACHINE_SUPERMEGAGREAT`, that still does you no good because you don’t have any binaries that support the SuperMegaGreat. Just enumerate through all the ones you support and intersect that with the ones that the operating system supports, and that gives you the ones for which you should install your architecture-specific packages.

**Bonus chatter:** Sometimes, the kernel folks get a little cute when picking the ID numbers for new architectures. Here are the ones I was able to guess at:

| Architecture | Value (hex) | Proposed explanation  |
|--------------|-------------|---|
| CEF          | 0x0CEF      | Hex digits spell out name                                     |
| CEE          | 0xC0EE      | Why not 0x0CEE ?  |
| EBC          | 0x0EBC      | Hex digits spell out name                                     |
| AMD64        | 0x8664      | Also known as x <u>86</u> - <u>64</u>                         |
| ARM64        | 0xAA64      | Also known as <u>A</u> Arch <u>64</u>                         |
| MIPS16       | 0x0266      | There's clearly something going on here but I don't know what |
| MIPSFPU      | 0x0366      |   |
| MIPSFPU16    | 0x0466      |   |

Raymond Chen

**Follow**

