Why are there two kinds of Universal Windows apps, one for Windows 8 and another for Windows 10?

devblogs.microsoft.com/oldnewthing/20180215-00

February 15, 2018

Raymond Chen

If you create a new Windows app (as opposed to a Win32 program), you have a few choices. One set of choices is under the *Windows Universal* node, which lets you create a variety of programs described as *Universal Windows*. Another set of choices is under the *Store Apps*, *Windows Apps* node, which lets you create a variety of programs described as *Universal Apps*.

Why are there two kinds of Universal apps, and what's the difference, and which one should I be creating?

Confusingly, there has been more than one type of project with the name "universal".

Windows 8 universal projects consist of three sub-projects: One for desktop-specific code, one for phone-specific code, and one for common code. The idea was that you put most of your app in the "common" project, with a small amount of code in the desktop-specific and phone-specific projects. When you built the solution, you got two packages, one for Windows 8 desktop and one for Windows 8 phone. It was universal in the sense that you wrote the code only once, but it still produced two packages that had to be submitted separately. The project is universal but the packages aren't.

Windows 10 universal projects are a single project that produce a single package that runs on both Windows 10 desktop and Windows 10 phone (and any other Windows 10 platforms). Both the project and package are universal.

If you ask for the Windows universal project template, you get a Windows 10 style universal project, which is almost certainly the one you want. If you ask for the Windows 8 universal project template, you get a Windows 8 style universal project, which is not as interesting nowadays.

The Windows 8 folks may have been a bit too ambitious calling their projects "universal", since they weren't actually creating projects that ran on all Windows systems. I mean, it was certainly better than what we had before, so it was "more universal", you might say. But with

that name they painted themselves into a corner, because when the Windows 10 Universal app model came on the scene, you had a program that ran on all Windows 10 systems, and you would want to call these things "Universal Apps", except the name was already taken by that other thing that wasn't as universal as the new thing.

Raymond Chen

Follow