Why does upgrading my project to Unicode cause Visual Studio to use a different version of the common controls?

devblogs.microsoft.com/oldnewthing/20171031-00

October 31, 2017

Raymond Chen

A customer was upgrading a program from Visual Studio 2008 to Visual Studio 2017. As part of the upgrade, they also decided to recompile it as a Unicode program rather than an ANSI program. And they bumped their `WINVER` from `0x400` to `0x600`. All together, these settings mean that they dropped support for Windows 95, Windows 98, Windows Me, as well as Windows NT 4.0, Windows 2000, Windows XP, and Windows Server 2003. Given that all of these operating systems left extended support quite some time ago (although Windows XP stubbornly held on way past its best-if-used-by date), these updates were long overdue.

The customer found that as a result of these upgrades, the project also switched from using version 5 of the common controls to version 6. This was not something they expected, since it's not anything they explicitly asked to be upgrade to. What happened?

It was the combination of the two other upgrades that triggered the switch from version 5 to version 6.

Version 6 of the common controls was introduced in Windows XP, so it has as a prerequisite that your `WINVER` be set to at least `0x0501`, since that is the value that corresponds to Windows XP. If you choose a lower value, then you are saying that you are targeting a of Windows that predates version 6 of the common controls, and therefore trying to use it will never work.

Version 6 of the common controls is also Unicode-only, with some concessions for older messages. If you build your project for ANSI, then the project can't use version 6 of the common controls because it's using the wrong character set.

But once you set your project parameters so both of the criteria are met, then version 6 of the common controls becomes available.

If that’s not what you want, then you’ll need to say, “Hey, like, I know I meet all the qualifications for using version 6 of the common controls, but I don’t want to.” It appears that if you create a new MFC project, then the IDE will enable version 6 of the common controls automatically once you enable Unicode, so that could be where the manifest magic is coming from.

Raymond Chen

Follow