

In my Visual Studio project, I set my Conformance mode to permissive, but it's still not permissive

 devblogs.microsoft.com/oldnewthing/20240725-00

July 25, 2024



Raymond Chen

Visual Studio lets you configure the conformance mode of your C++ project under Configuration Properties, C/C++, Language, Conformance mode. But if you set it to “No (/permissive)”, it might still not be permissive. What’s going on?

The documentation for the /permissive flag says

```
The /permissive- option is implicitly set by the /std:c++latest option starting in Visual Studio 2019 version 16.8, and in version 16.11 by the /std:c++20 option. ... The /permissive option must come after any option that sets /permissive- implicitly.
```

I created a project, set the C++ Language Standard to “ISO C++20 Standard (/std:c++20)”, and set the Conformance mode to “No (/permissive)”. I confirmed that the compiler was still not running in permissive mode by using the first example on the page and seeing whether the compiler accepted it.

After confirming that permissive mode was disabled, I looked at the Command Line page to see what flags were ultimately being passed to the compiler. My guess was that the /permissive flag was being passed too soon, and was being subsequently being turned off by the /std:c++20 flag.

And my theory was correct:

```
/JMC /permissive /ifcOutput "x64\Debug\" /GS /Wall /Zc:wchar_t /ZI /Gm- /Od /sdl  
/Fd"x64\Debug\vc143.pdb" /Zc:inline /fp:precise /D "_DEBUG" /D "_CONSOLE" /D  
"_UNICODE" /D "UNICODE" /errorReport:prompt /WX- /Zc:forScope /RTC1 /Gd /MDd  
/std:c++20 /FC /Fa"x64\Debug\" /EHsc /nologo /Fo"x64\Debug\" /Fp"x64\Debug\Test.pch"  
/diagnostics:column
```

The solution was to add `/permissive` as an explicit flag in the “Additional Options” box at the bottom. These flags appear to go at the end, so we can turn permissive mode back on after the `/std:c++20` flag turned it off.

This is arguably a bug in the build system (putting the `ConformanceMode` property too early on the command line), but at least now you have a workaround.