

# What does the CompileAsWinRT Visual Studio project option mean?

 [devblogs.microsoft.com/oldnewthing/20211118-00](https://devblogs.microsoft.com/oldnewthing/20211118-00)

November 18, 2021



Raymond Chen

A customer was investigating some build problems in their C++/WinRT project related to mismatches in the state of `WINRT_NO_MAKE_DETECTION`. Along the way, they found that in their project file, they did not set `CompileAsWinRT` to `true`. Is that a problem?

The `CompileAsWinRT` property controls whether the Microsoft C++ compiler enables C++/CX language extensions. It corresponds to the command line switch `/ZW`. This is confirmed by the `cl.xml` file buried deep inside Visual Studio:

```
<BoolProperty
  Name="CompileAsWinRT"
  DisplayName="Consume Windows Runtime Extension"
  Description="Consume the Windows Run Time languages extensions. (/ZW)"
  Category="General"
  Switch="ZW"
  F1Keyword="VC.Project.VCCLCompilerTool.CompileAsWinRT">
</BoolProperty>
```

(I like how they spelled it as *Run Time* in the description, but used the compound word *Runtime* in the display name.)

C++/CX was the first projection of the Windows Runtime into C++, so it got first dibs and took the good name. It's now a decade later, and C++/CX is no longer the Windows Runtime projection of choice, but the name of the property can't be changed for compatibility reasons.

So don't worry, you don't need to (and indeed shouldn't) turn on `CompileAsWinRT` for your C++/WinRT projects. Turning it on enables the old-and-busted projection of the Windows Runtime.

**Bonus chatter:** So what's the compiler switch to enable C++/WinRT?

That's a trick question. There is no special compiler switch needed because C++/WinRT is written in standard C++. No nonstandard language extensions are required. Just include the C++/WinRT header files and go.

Raymond Chen

**Follow**

