

Detecting whether the `-opt` flag was passed to `cppwinrt.exe`: Using `__has_include`

devblogs.microsoft.com/oldnewthing/20190624-00

June 24, 2019



Raymond Chen

I was upgrading the [Window UWP Samples repo](#) to take advantage of the new `-opt` flag introduced in C++/WinRT 2.0. This provides performance improvements for accessing static class members, and avoids having to register the type in your manifest for strictly in-module consumption.

The new `-opt` flag enables these optimizations, but it also adds a new requirement: Your implementation file needs to `#include <ClassName.g.cpp>`. The problem is that I wanted to upgrade the samples one at a time, but that meant that the shared files needed to support both optimized and unoptimized builds, at least until I get them all converted.

I was at a bit of a loss, because there was no obvious `#define` in `winrt/base.h` that tells me whether the `-opt` flag was passed.

And then I realized: I could use `__has_include`.

C++17 introduced the `__has_include` preprocessor keyword which snoops around to determine whether a header file exists. The idea is that you could conditionalize based on whether an optional header file is present. For example, you might check for the presence of `xmmintrin.h` and conditionally enable SSE operations.

In my case, I wouldn't be probing for a system header file, but rather for a generated `.g.cpp` file produce by `cppwinrt.exe` in `-opt` mode.

```
#if __has_include(<MainPage.g.cpp>)\n#include <MainPage.g.cpp>\n#endif
```

If `cppwinrt.exe` were run with the `-opt` flag, then the `MainPage.g.cpp` file will exist in the `Generated Files` directory, and I can include it. If it were run without the `-opt` flag, then the `MainPage.g.cpp` file will not exist, and I skip over it.

[Raymond Chen](#)

Follow

