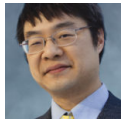


Why does my C++/WinRT project get errors of the form “Unresolved external symbol void* __cdecl winrt_make_YourNamespace_YourClass(void)“?

 devblogs.microsoft.com/oldnewthing/20190606-00

June 6, 2019



Raymond Chen

So your C++/WinRT project gets build failures of the form

```
unresolved external symbol "void * __cdecl winrt_make_YourNamespace_YourClass(void)"
(?winrt_make_YourNamespace_YourClass@YAPAXXZ) referenced in function
void * __stdcall winrt_get_activation_factory(class std::basic_string_view<wchar_t,
struct std::char_traits<wchar_t> > const &)" (?winrt_get_activation_
factory@@YGPAXABV?$basic_string_view@_WU?$char_traits@_W@std@@@std@@@Z)
```

What’s going on, and how do you fix it?

The problem is that you used the `-opt` flag with `cppwinrt.exe`, but didn’t do the work necessary to supoprt those optimizations.

To each of your implementation files (such as `YourClass.cpp`), add the line

```
#include "YourClass.g.cpp"
```

If your project defines classes in multiple Windows Runtime namespaces, then the inclusion should be

```
#include "Sub/Namespace/YourClass.g.cpp"
```

If you specified the `-prefix` option, then the inclusion should be

```
#include "Sub.Namespace.YourClass.g.cpp"
```

(Personally, I put the `#include` directive immediately after the inclusion of the corresponding `YourClass.h` header file.)

In a Visual Studio project, you can enable optimizations by setting

```
<CppWinRTOptimized>true</CppWinRTOptimized>
```

in your project file.

To turn on dotted prefixes, you can set

```
<CppWinRTUsePrefixes>true</CppWinRTUsePrefixes>
```

The main optimization enabled by the `-opt` flag in C++/WinRT 2.0 is bypassing the call to `RoGetActivatationFactory` if the class is implemented in the same module. Instead, the call goes directly to the implementation. This also removes the need to declare the runtime class in your manifest if it is used only within the module (say, by XAML binding).

Bonus reading: [Meet C++/WinRT 2.0: Optimizing Components.](#)

[Raymond Chen](#)

Follow

