# Why does my C++/WinRT project get errors of the form ‘winrt::impl::produce‘: cannot instantiate abstract class, missing method GetBindingConnector

devblogs.microsoft.com/oldnewthing/20190605-00

June 5, 2019

Raymond Chen

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

```
base.h(8208): error C2259: 'winrt::impl::produce<D, I>': cannot instantiate abstract
class
with
[
    D=winrt::YourNamespace::implementation::YourClass,
    I=winrt::Windows::UI::Xaml::Markup::IComponentConnector2
] (compiling source file YourClass.cpp)
base.h(8208): note: due to following members: (compiling source file YourClass.cpp)
base.h(8208): note: 'int32_t winrt::impl::abi<winrt::Windows::UI::Xaml::Markup::
IComponentConnector2, void>::type::GetBindingConnector(int32_t, void *, void **)
noexcept': is abstract (compiling source file YourClass.cpp)
```

Normally, the `GetBindingConnector` function is defined in `YourClass.xaml.g.hpp`, but that header file isn't being generated.

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

The problem is that you forgot to include the header file

```
#include "winrt/Windows.UI.Xaml.Markup.h"
```

Add that line to, say, your precompiled header file, and things should work again.

You are likely to run into this problem when upgrading a project from C++/WinRT 1.0 to C++/WinRT 2.0. The C++/WinRT 2.0 compiler is much better about reducing header file dependencies, which improves build times. If you forgot to include `winrt/Windows.UI.Xaml.Markup.h` in a C++/WinRT 1.0 project, you often got away with it, because some other C++/WinRT 1.0 header file you included happened to include `winrt/Windows.UI.Xaml.Markup.h` as a side effect. You were getting a free ride on the other header file.

Raymond Chen

**Follow**