

What does it mean when the compiler tells me that `promise_type`: is not a member of `coroutine_traits`?

 devblogs.microsoft.com/oldnewthing/20221212-00

December 12, 2022



Raymond Chen

A customer using C++/WinRT found that their attempt to perform a `co_await` on an `IAsyncAction` failed to compile with the error

```
promise_type: is not a member of coroutine_traits<void>
```

What does this error mean? “We included `<winrt/Windows.Foundation.h>`, which is allegedly [how to fix this weird error message](#), but that didn’t fix it.”

This is a case of skipping the *reading the explanation* part of looking for a solution to your problem and skipping directly to the *how to fix it* part, without checking that the *how to fix it* applies to your specific situation. “Somebody called in to a car repair show because their car didn’t start, and the answer was that they had loose battery cables. My car won’t start, and I tightened the battery cables, but it didn’t fix it.” Well yeah, because you skipped over the part of the call where they did the troubleshooting.

As I noted in the earlier article, the `co_await` keyword triggers the coroutine transformation which rewrites the function performing the `co_await` into a state machine, and one of the pieces of the state machine is the `promise_type` associated with the coroutine’s return type and parameters.

The `coroutine_traits` specialization in the error message is `coroutine_traits<void>`, so the return type is `void` and there are no parameters. In other words, the function performing the `co_await` looks something like this:

```
void DoSomething()  
{  
    co_await MyFunctionAsync();  
}
```

The C++/WinRT library provides support for producing coroutines returning `winrt::Windows::Foundation::IAsyncAction` and related interfaces, as well as the special type `winrt::fire_and_forget`. It does not, however, add support for coroutines returning `void`.

The `winrt/Windows.Foundation.h` header file defines the Windows Runtime coroutines, but `void` is not a Windows Runtime coroutine type, so including Windows Runtime headers isn't going to help. The code probably meant for `DoSomething()` to return a type that can be used for coroutines, most likely `IAsyncAction` or `fire_and_forget`.

Now, you could always write your own support for producing coroutines returning `void`, say, by having it behave roughly the same as `fire_and_forget`. It's not hard.

But probably not advisable.

Raymond Chen

Follow

