

Why am I being told my `fire_and_forget` coroutine is not returning a value?

 devblogs.microsoft.com/oldnewthing/20210201-00

February 1, 2021



Raymond Chen

You're writing a C++/WinRT coroutine that nominally returns a `fire_and_forget`, meaning that the coroutine runs without notifying anybody when it is finished.

```
fire_and_forget DoSomething()  
{  
    DoFirstThing();  
    DoSecondThing();  
}
```

And when you compile this, you get

```
error C4716: 'DoSomething': must return a value
```

A `fire_and_forget` coroutine doesn't have a return value, since there is nobody awaiting it. So what's going on here?

What's going on is that the body of your `fire_and_forget` coroutine doesn't contain any `co_await` or `co_return` statements. A function must contain at least one `co_await` or `co_return` statement in order to be considered a coroutine. Since the function did neither of those things, the C++ language treats it not as a coroutine but a regular function.

A regular function that needs to return a `fire_and_forget` object, and you didn't do that.

You have a few options for fixing this.

One option is to add a `co_return;` statement at the end. Normally, falling off the end of a coroutine is equivalent to performing a `co_return;`, but in this case, you need to say `co_return` explicitly in order to make sure you have a coroutine at all!

Another option is to change your function to return `void`. After all, it doesn't contain any asynchronous operations, so the whole thing ran synchronously anyway. It never needed to be a coroutine in the first place.

Raymond Chen

Follow

