

Error C3867: non-standard syntax; use ‘&’ to create a pointer to member: What it means and how to fix it

devblogs.microsoft.com/oldnewthing/20201007-00

October 7, 2020

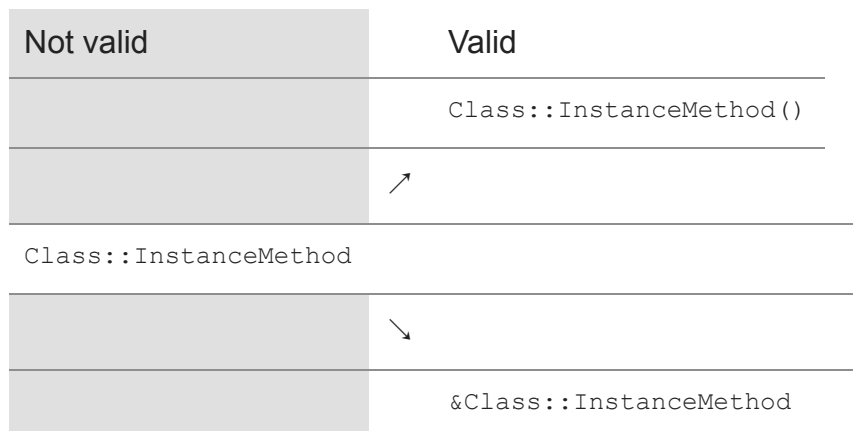


Raymond Chen

When you get the Microsoft compiler error message “non-standard syntax; use ‘&’ to create a pointer to member”, what is it trying to say?

It’s telling you that you are accessing an instance member function, but you didn’t finish the job and either take its address or invoke it.

```
class Class
{
    static void StaticMethod();
    void InstanceMethod();
};
```



C++ contains a curious discrepancy in how you create pointers to methods, depending on whether the method is instance or static.

For static methods, you can just say its name without any special decoration: `Class::StaticMethod`. You can put an ampersand in front if you like, but it’s not required.

For instance methods, you cannot say its name without decoration. You *must* take its address explicitly with the ampersand. The upper right corner of the diagram is illegal.

Syntax	Static	Instance
Without ampersand	Class::StaticMethod	Class::InstanceMethod
With ampersand	&Class::StaticMethod	&Class::InstanceMethod

This error message is telling you that it looks like you are trying to create a pointer to an instance member, but you forgot the ampersand. My guess is that the text “non-standard syntax” is an acknowledgement of the fact that very early versions of the Microsoft compiler allowed you to take the address of an instance member without the ampersand, but that is not standard syntax. (I don’t know if the early versions of the compiler actually did permit it, but the wording of the error message suggests it.)

In practice, when you get this error message, you usually aren’t actually trying to create a pointer to a member function. You just forgot to call the method.

```
Class c;
c.InstanceMethod; // error C3867
```

Notice that the Microsoft compiler gives you error C3867 in this case, even though there’s clearly no attempt to create a member function pointer. An ampersand here isn’t even legal!

I suspect that this is a case of a more general error case, like “method group not valid here”, and the error text was chosen to be more descriptive of one of the scenarios that would lead to that error, even though it ends up being more confusing for the other scenarios.

But at least now you know what the error is actually trying to tell you, and more importantly, what you probably need to do to fix it. You’re probably trying to call the method. Add the parentheses.

Raymond Chen

Follow

