

Error compiling C++/WinRT runtime class: function does not take N arguments

 devblogs.microsoft.com/oldnewthing/20200402-00

April 2, 2020



Raymond Chen

You're coding up a C++/WinRT runtime class, and the compiler spits out an error message:

```
error C2660: 'MyClass::Thing1': function does not take N arguments
```

If you're really unlucky, you'll get

```
error C2064: term does not evaluate to a function taking N arguments
```

where N is probably 0 or 1.

You may have failed to implement all the necessary overloads of a method.

C++/WinRT events are represented by a pair of overloads:

Operation	Signature (simplified)	Example
Register	<code>event_token Event(Delegate const& handler);</code>	<code>token = o.Event(handler);</code>
Unregister	<code>void Event(event_token const& token);</code>	<code>o.Event(token);</code>

C++/WinRT read/write properties are also represented by a pair of overloads:

Operation	Signature (simplified)	Example
Read	<code>T Property();</code>	<code>auto value = o.Property();</code>
Write	<code>void Property(T const& value);</code>	<code>o.Property(value);</code>

And naturally, if you have overloaded methods, then they are represented by, um, overloaded methods.

Signature (simplified)	Example
<code>void Method();</code>	<code>o.Method();</code>
<code>void Method(int32_t value);</code>	<code>o.Method(2);</code>
<code>void Method(int32_t value, hstring name);</code>	<code>o.Method(2, L"Bob");</code>

If you fail to implement all of the required methods, you will get an error when the C++/WinRT autogenerated code tries to call one of the missing overloads. When there is a mismatch between the call site and the function prototype, the compiler assumes that the prototype is correct and the call site is wrong. Therefore, the error won't be "You forgot to declare the 1-parameter overload of this method." It'll be "The method I found doesn't take 1 parameter."

Raymond Chen

Follow

