C++/WinRT gotcha: Setting properties incorrectly

devblogs.microsoft.com/oldnewthing/20200729-00

July 29, 2020

Raymond Chen

Getting and setting a Windows Runtime property looks like this:

Language	Getter	Setter
C#	`v = o.Property;`	`o.Property = 42;`
C++/CX	`v = o->Property;`	`o->Property = 42;`
C++/WinRT	`v = o.Property();`	`o.Property(42);`
JavaScript	`v = o.property;`	`o.property = 42;`
Python	`v = o.property`	`o.property = 42`

Somebody is the odd man out.

All the projections use a simple member access to read a property and a simple assignment statement to set a property, with the exception of C++/WinRT, which uses a function call in both places.

That's because the standard C++ language doesn't have "properties", and C++/WinRT is a projection of the Windows Runtime into standard C++. (C++/CX gets away with it because it's not standard C++.)

If you're translating existing code from one of the other languages to C++/WinRT, you may realize that properties need to change to function calls, but in your haste (or tiredness), you mistakenly convert `o.Property = 42` to

```
o.Property() = 42;
```

Fortunately, this gives you a compiler error because you cannot assign to an integer value.

Unfortunately, if the property has a non-primitive type, you don't get an error.

```
o.Name() = L"Fred";
// oops: Should be o.Name(L"Fred");

lv.Background() = greenBrush;
// oops: Should be lv.Background(greenBrush);
```

That's because you are assigning to the temporary object returned by the property getter method, and that temporary object has an assignment operator.

The above code breaks down like this:

```
auto name = o.Name();
name = L"Fred";
// destruct temporary "name"

auto background = lv.Background();
background = greenBrush;
// destruct temporary "background"
```

Congratulations, you updated a temporary that was immediately destructed. Total waste of time.

Raymond Chen

Follow