Setting properties in C++/WinRT is done by a function call, but you need to call the function the right way

devblogs.microsoft.com/oldnewthing/20221028-00

October 28, 2022

Raymond Chen

For people transitioning to C++/WinRT from other languages, a common beginner mistake is setting properties incorrectly. Here are the ways of accessing a Windows Runtime property in various languages.

Language	Read a property	Write a property
C#	`oldValue = o.ActiveWidget;`	`o.ActiveWidget = newValue;`
VB	`oldValue = o.ActiveWidget`	`o.ActiveWidget = newValue`
JavaScript	`oldValue = o.activeWidget;`	`o.activeWidget = newValue;`
Python	`old_value = o.active_widget`	`o.active_widget = new_value`
C++/CX	`oldValue = o->ActiveWidget;`	`o->ActiveWidget = newValue;`
C++/WinRT	`oldValue = o.ActiveWidget();`	`o.ActiveWidget(newValue);`

Notice the odd one out in the above list.[1] Everybody uses the assignment operator to set a property, except C++/WinRT, which uses a function call.

If you're used to the other languages, you may incorrectly interpret the rule that "property access is a function call" People transitioning to C++/WinRT from other languages are tempted to write

```
o.ActiveWidget() = newValue;
```

in a misguided attempt to set the property to a new value. What this actually does is fetch the current value of the property, and then assign the new value to the temporary. It's a lot of work with nothing to show for it, and then you scratch your head wondering why the attempt to update the property appears to have been ignored.

I tried to update C++/WinRT to deny assignment to most types of rvalues, rendering the above a compile-time error, but the change had to be reverted for compatibility reasons.

There's no real value to creating a temporary only to assign a new value to it. You may as well just create the temporary with the desired value, and then use it.

```
// Pointless
DoSomething(o.ActiveWidget() = newValue);

// Just do this
DoSomething(Widget(newValue));

// Or possibly even just
DoSomething(newValue);
```

[1] Rust/WinRT also uses function calls to access properties:

Language	Read a property	Write a property
Rust/WinRT	`old_value = o.active_widget();`	`o.set_active_widget(new_value);`

However, it doesn't suffer from the same problem as C++/WinRT, because Rust disallows assignment to rvalues (known as *value expressions* in Rust-speak).

```
// error E0070: Invalid left-hand side of assignment
o.active_widget() = new_value;
```

The problem of assigning to a temporary is peculiar to C++.

Raymond Chen

Follow