How can I work around the absence of default parameters in the Windows Runtime?

devblogs.microsoft.com/oldnewthing/20231213-00

December 13, 2023

Raymond Chen

The Windows Runtime does not support default parameters.

Wait, come back!

In some languages, you can specify that a call site can omit trailing parameters, in which case default values for those parameters will be auto-provided.

```
// C++
void DoSomething(int a = 1, int b = 2);

DoSomething(3, 4);
DoSomething(3); // rewritten as DoSomething(3, 2);
DoSomething();  // rewritten as DoSomething(1, 2);
```

The missing parameters are provided by the compiler at the call site, and at the end of the day, there is only one `DoSomething` function, which takes two parameters, but you can be lazy and specify fewer, and the compiler will fill in the missing ones. C# and Python also support default parameters, and JavaScript gained default parameters in ES6. Rust, on the other hand, doesn't support them.

The Windows Runtime doesn't have a facility for specifying default parameters, but you can fake it by using overloads:[1]

```
runtimeclass Widget
{
    void DoSomething(Int32 a, String b);
    void DoSomething(Int32 a);
    void DoSomething();
}
```

This is not quite the same as providing default parameters, because this declares three different methods with different numbers of parameters. The Windows Runtime API design conventions recommend that if you do this, then

- Each overload with more parameters matches the other overloads with fewer parameters for the parameters they have in common.
- Calling an overload with fewer parameters is equivalent to calling an overload with more parameters, just with some “obvious” defaults for the extra parameters.

In the above example, notice that the parameter lists for the overloads are compatible with each other where they overlap.

Method	Parameter 1	Parameter 2	Documentation
`DoSomething` (2 parameters)	`Int32 a`	`String b`	
`DoSomething` (1 parameter)			Equivalent to `DoSomething(a, "")`
`DoSomething` (0 parameters)			Equivalent to `DoSomething(42, "")`

By arranging the methods in this way, you can close one eye and pretend that you have

```
runtimeclass Widget
{
    void DoSomething(Int32 a = 42, String b = "");
}
```

Related reading: On the proper care and feeding of the `default_overload` Windows Runtime attribute.

[1] We saw this same trick earlier when we made a member function’s default parameter depend on `this`.