The misleading MIDL error message: MIDL5023: The arguments to the parameterized interface are not valid

devblogs.microsoft.com/oldnewthing/20210113-00

January 13, 2021

Raymond Chen

Suppose you're writing an IDL file for the Windows Runtime. You have a method that returns a vector view of strings, but for some reason the compiler tells you that "The arguments to the parameterized interface are not valid."

```
runtimeclass Widget
{
  Windows.Foudation.Collections.IVectorView<String> GetNames();
}
```

The argument to the parameterized interface is `String`, and that's certainly a valid argument for `IVectorView`, isn't it?

Yes, it's a perfectly fine argument for `IVectorView`.

But what you have there isn't the `IVectorView` you think you have.

```
  Windows.Foudation.Collections.IVectorView<String> GetNames();
```

You misspelled `Foundation`.

The compiler is technically correct: `String` is not a valid argument for the parameterized interface `Windows. Foudation. Collections. IVectorView`. But that's because *there are no possible valid arguments* for the parameterized interface `Windows. Foudation. Collections. IVectorView`, because `Windows. Foudation. Collections. IVectorView` is not a legal parameterized interface!

The order in which the compiler checks for validity happens to result in a misleading error message, making you believe that the error is in the argument, when in fact the error is in the interface.

Bonus chatter: You can avoid some of the risk of typos by taking advantage of the shorthand notation that lets you omit `Windows. Foundation. Collections`:

```
runtimeclass Widget
{
  IVectorView<String> GetNames();
}
```

Of course, you can still get the misleading error message if you typo `IVectorView` as, say, `IVetorView`. So the problem is still there. The shorthand just makes it a little bit less likely.

Raymond Chen

Follow