How do I pass a raw pointer to a Windows Runtime function?

devblogs.microsoft.com/oldnewthing/20221125-00

November 25, 2022

Raymond Chen

A customer was trying to create a Windows Runtime function that accepts a raw pointer, but they got an error:

```
namespace MyNamespace
{
    runtimeclass MyClass
    {
        void Transform(Int32 count, Int16* input, Int16* output);
    }
}
```

The compilation files with the error

```
MIDL2025 [msg]syntax error [context]: expecting an identifier near "*"
```

What’s going on?

The Windows Runtime does not support raw pointers at the projection level. Many programming languages don’t have pointers at all (such as JavaScript and Python), and others restrict their use (C#, Rust). The Windows Runtime tries to be friendly to a large number of languages, and pointers are somewhat out of fashion in modern programming languages because they are so easy to use incorrectly.

In this particular customer’s case, they wanted to pass a buffer into the method without copying, and to receive the results directly into a caller-provided buffer. Fortunately, these are already expressible with the existing array-passing patterns.

```
void Transform(Int32[] input, ref Int32[] output);
```

The input buffer is a read-only array, and the output buffer is a write-only array.

For output pointers to single elements, you can use the `out` keyword.

```
bool TryGetCount(out Int32 count);
```

Bonus reading: How does JavaScript represent output parameters in the Windows Runtime?

Bonus chatter: You may have noticed that I didn't mention input pointers to single elements. That's because you can just pass those by value.

Raymond Chen

Follow