# The MIDL compiler still has trouble with double greater-than signs, sadly

devblogs.microsoft.com/oldnewthing/20250623-00/?p=111295

June 23, 2025


The parser for the Microsoft MIDL compiler has long suffered from the problem of the double greater-than sign. This problem plagued C++ until C++11 added a special rule:[1]

> **[temp.names]** (4): When parsing a *template-argument-list*, the first non-nested > is taken as the ending delimiter rather than a greater-than operator. Similarly, the first non-nested >> is treated as two consecutive but distinct > tokens, the first of which is taken as the end of the *template-argument-list* and completes the *template-id*.

Microsoft's MIDL compiler predates C++11, and its parser treats two consecutive greater-than signs as a bitwise shift operator. This means that you cannot write

```
Windows.Foundation.IAsyncOperation<
    Windows.Foundation.Collections.IVector<Int32>>
```

when you want an asynchronous operation that produces a vector of 32-bit integers. The double-greater-than is interpreted as a shift operator, and you get a weird MIDL error message because a shift operator is not allowed there.

You must explicitly insert a space to force it to be parsed as two greater-than signs.

```
Windows.Foundation.IAsyncOperation<
    Windows.Foundation.Collections.IVector<Int32> >
```

Larry Osterman tells me that multiple people have attempted to fix this and allow the compiler to break up a shift operator into two greater-than signs "when appropriate".

All such attempts have failed.

I wonder if there is a comment in the source code similar to this legendary one:

```
//
// Dear maintainer:
//
// Once you are done trying to 'optimize' this routine,
// and have realized what a terrible mistake that was,
// please increment the following counter as a warning
// to the next guy:
//
// total_hours_wasted_here = 42
//
```

[1] The controversy is still not over, though, because the standard does not define what "non-nested" means. Issue 579 gives as an example the sequence

```
X<a ? b > c : d>
```

Is the first > a "nested" greater-than sign? This issue, opened in 2006, is still outstanding!