

# Why am I being told that my message ID is too large?

 [devblogs.microsoft.com/oldnewthing/20160513-00](https://devblogs.microsoft.com/oldnewthing/20160513-00)

May 13, 2016



Raymond Chen

A customer asked for help with an error message in the message compiler.

Our message file goes like this:

```
MessageIdTypedef=DWORD
MessageId = 0x10001
SymbolicName = MSG_ERROR
Language = English
Error %1
.
```

When we compile it, we get this error:

```
MC : error : Message Id value (10001) too large
```

Our understanding is that we defined our message ID as a `DWORD` (which is an unsigned 32-bit integer), and the value `0x10001` easily fits inside an unsigned 32-bit integer. Is there a command line switch we need to pass to `mc.exe`? What are we missing?

What they're missing is that message identifiers are limited to the range 0 through 65535.

One place this limitation is visible is in the header file generated by the message compiler:

```

//
// Values are 32 bit values laid out as follows:
//
// 3 3 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1
// 1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3 2 1 0
// +-+-+-+-----+-----+
// |S|R|C|N|r|   Facility   |           Code           |
// +-+-+-+-----+-----+
//
// where
//
//   S - Severity - indicates success/fail
//
//       0 - Success
//       1 - Fail (COERROR)
//
//   R - reserved portion of the facility code, corresponds to NT's
//       second severity bit.
//
//   C - reserved portion of the facility code, corresponds to NT's
//       C field.
//
//   N - reserved portion of the facility code. Used to indicate a
//       mapped NT status value.
//
//   r - reserved portion of the facility code. Reserved for internal
//       use. Used to indicate HRESULT values that are not status
//       values, but are instead message ids for display strings.
//
//   Facility - is the facility code
//
//   Code - is the facility's status code
//

```

Observe that the `Code` field is a 16-bit value.

A copy of this header block is also included in the [documentation on event identifiers](#). Though you have to realize that the thing that the message compiler spits out are event identifiers.

The 16-bit limit is also called out in the C# version of the trace logging interface: `Event - Attribute.EventId` says that “This value should be between 0 and 65535.”

But again, this requires that you realize that message files are related to event logging.

One thing that may nudge you to that realization is that [the documentation for Message Files](#) is in the *Event Logging* section of MSDN, and it opens with the sentence, “Each [event source](#) should register message files that contain...”

But perhaps the smoking gun is that the documentation for the Message File syntax says Any value specified must fit in 16 bits.

Raymond Chen

**Follow**

