Why can't I do boolean logic on bytes?

Asked 15 years, 6 months ago Modified 15 years, 6 months ago Viewed 3k times

6

In C# (3.5) I try the following:

```
byte byte1 = 0x00;
byte byte2 = 0x00;
byte byte3 = byte1 & byte2;
```

and I get Error 132: "Cannot implicitly convert type 'int' to 'byte'. An explicit conversion exists (are you missing a cast?)". The same happens with | and ^.

What am I doing wrong? Why is it asking me about ints? Why can't I do boolean logic on bytes?

c# .net byte boolean-logic

Share Improve this question Follow

asked Jun 18, 2009 at 9:20

Simon

25.9k ● 45 ● 156 ● 270

2 All arithmetic in .NET is based on int. There is a related question with quite some discussion. It's not an answer to your question, but you might get some insight in how C# deals with arithmetic in general: stackoverflow.com/questions/941584/byte-byte-int-why – Dirk Vollmar Jun 18, 2009 at 9:26

"Because C# is not a perfect language", is how I would explain this. This is one of its warts (and compared to other languages, C# has by far the fewest and the least impactful warts...) – Roman Starkov Sep 25, 2012 at 11:19

2 Answers

Sorted by: Highest score (default)

12

Various operators aren't declared for `byte` - both operands get promoted to `int`, and the result is `int`. For example, addition:

```
byte byte1 = 0x00;
byte byte2 = 0x00;
byte byte3 = byte1 + byte2; // Compilation error
```

Note that compound assignments do work:

```
byte1 += byte2;
```

There was a recent SO question on this. I agree this is particularly irksome for bitwise operations though, where the result should always be the same size, and it's a logically entirely valid operation.

As a workaround, you can just cast the result back to byte:

```
byte byte3 = (byte) (byte1 & byte2);
```

Share

Improve this answer

Follow

edited May 23, 2017 at 9:57

Community Bot

1 • 1

answered Jun 18, 2009 at 9:23

Jon Skeet

1.5m • 889 • 9.3k • 9.3k

> It may be irksome that C# evaluates byte+byte as type "int", requiring a typecast if one wants to assign the result back to a byte, but vb.net is worse. Two variables or defined constants of type 'Byte' will be added as type 'Byte' (dying if the result is greater than 255 even if it's going to be assigned to an 'Integer') but numeric literals that are used for anything other than a direct assignment are regarded as integers. It's nice, though that vb.net handles Booleans ops mostly sensibly (the size of `x And y` should be, but isn't, that of the *smaller* unsigned operand, if any). – supercat Mar 1, 2012 at 18:50

Because byte (and short) types do not implement those operators

See Spec: 4.1.5

0

Share Improve this answer Follow

answered Jun 18, 2009 at 9:23

AnthonyWJones

189k • 35 • 235 • 307