

# Why do operations on "byte" result in "int"?

 [devblogs.microsoft.com/oldnewthing/20040310-00](http://devblogs.microsoft.com/oldnewthing/20040310-00)

March 10, 2004



Raymond Chen

(The following discussion applies equally to C/C++/C#, so I'll use C#, since I talk about it so rarely.)

People complain that the following code elicits a warning:

```
byte b = 32;
byte c = ~b;
// error CS0029: Cannot implicitly convert type 'int' to 'byte'
```

“The result of an operation on ‘byte’ should be another ‘byte’, not an ‘int’,” they claim.

Be careful what you ask for. You might not like it.

Suppose we lived in a fantasy world where operations on ‘byte’ resulted in ‘byte’.

```
byte b = 32;
byte c = 240;
int i = b + c; // what is i?
```

In this fantasy world, the value of *i* would be 16! Why? Because the two operands to the + operator are both bytes, so the sum “*b+c*” is computed as a byte, which results in 16 due to integer overflow. (And, as I noted earlier, integer overflow is the new security attack vector.)

Similarly,

```
int j = -b;
```

would result in *j* having the value 224 and not -32, for the same reason.

Is that really what you want?

Consider the following more subtle scenario:

```
struct Results {
    byte Wins;
    byte Games;
};
bool WinningAverage(Results captain, Results cocaptain)
{
    return (captain.Wins + cocaptain.Wins) >=
           (captain.Games + cocaptain.Games) / 2;
}
```

In our imaginary world, this code would return incorrect results once the total number of games played exceeded 255. To fix it, you would have to insert annoying int casts.

```
return ((int)captain.Wins + cocaptain.Wins) >=
       ((int)captain.Games + cocaptain.Games) / 2;
```

So no matter how you slice it, you're going to have to insert annoying casts. May as well have the language err on the side of safety (forcing you to insert the casts where you know that overflow is not an issue) than to err on the side of silence (where you may not notice the missing casts until your Payroll department asks you why their books don't add up at the end of the month).

[Raymond Chen](#)

**Follow**

