

## Re: Why is "False" = "-1"?

**Source:** <http://www.tech-archive.net/Archive/VB/microsoft.public.vb.general.discussion/2004-07/3353.html>

---

**From:** Duane Bozarth ([dp\\_bozarth\\_at\\_swko.dot.net](mailto:dp_bozarth_at_swko.dot.net))

**Date:** 07/25/04

Date: Sun, 25 Jul 2004 09:46:43 -0500

MikeD wrote:

>

> *"Jonathan Wood" <jwood@softcircuits.com> wrote in message*

> *news:OTgT39ccEHA.3480@TK2MSFTNGP11.phx.gbl...*

>>

>>> *It's always been my understanding that 'If A Then' executes slightly*

>> *faster*

>>> *than 'If A = 1 Then' (or 'If A = -1 Then' or 'If A = True Then').*

>>

>>> *This is not true. In both cases, A must be compared to zero or some other*

>> *value. The resulting code is the same.*

>

> *I don't quite understand your comment, at least as it relates to the point I*

> *was attempting to make.*

No, he (we) understood exactly...the point is your contention is simply wrong...

...snip babbling...

> *The point I was making was that 'If A Then' executes slightly faster than*

> *'If CBool(A) = True Then'. I guess I should have made that clearer. I can*

> *understand how my original post was confusing in this regard. Sorry about*

<IF> the above contention <were> to be true (I've not taken the time to actually look at the generated code), it would only be because of the (unnecessary) cast that is written into the latter that VB perhaps would not optimize out...

You would perhaps find it instructive to look at some of the details Dan Barclay has in his article to which he posted a link...

In a nutshell, writing "If A Then" is <precisely> identical to writing "If A<>0 Then" and will generate the same code and therefore the same execution time. (Excepting perhaps for a <very> dumb compiler and I believe Dan's article shows that the VB compiler isn't that dumb—actually a couple of his examples show better code generation than I had suspected). This is conclusion can be generalized to cover any

microsoft.public.vb.general.discussion: Re: Why is "False" = "-1"?

and all of the cases/pseudo-examples you've previously provided—they all boil down to either an eq or ne and there has to be something for the machine instruction to compare to—whether it's zero or nonzero makes no difference.