Subject: Re: Logical vs. Bitwise boolean operators
Posted by Jeff Guerber on Fri, 01 Feb 2002 20:47:33 GMT
View Forum Message <> Reply to Message

On Fri, 1 Feb 2002, James Kuyper wrote:

> Fortran has the logical boolean operators .AND., .OR., and .NOT., and I
> naively expected the correspondingly named IDL operators to be the same.
> However, in IDL, those are bitwise operators, rather than logical ones
> (my copy of the manual just calls them 'boolean', without specifying
> whether they are logical or bitwise). As a result, the following code
> did not do what I expected:
>
> IF (!D.FLAGS AND 8) AND (ratio LE minrat) THEN BEGIN

James, as an aside, this statement tries use AND both bitwise _and_ logically. But I understand your point...

> I can always code around this limitation. The solution I came up with
> was to change it to:
>
> IF ((!D.FLAGS AND 8) EQ 8) AND (ratio LE minrat) THEN BEGIN

When confronted with a similar situation I like to use "IF ((!D.FLAGS AND 8) NE 0) ...", which is a little more general.

> However, it seems to me that there should be a better way to do this.

I agree! I've argued for logical operators before, and also for a true logical variable type like Fortran's. (I find the practice of using integers for this to be the source of considerable confusion, especially when coupled with the arcane rules for what's true and what's false, and the bitwise operators.) But, so far Boulder has not heard.

Jeff Guerber
Raytheon ITSS
NASA/Goddard, code 971

Subject: Re: Logical vs. Bitwise boolean operators
Posted by James Kuyper on Fri, 01 Feb 2002 21:17:39 GMT
View Forum Message <> Reply to Message

Jeff Guerber wrote:

> On Fri, 1 Feb 2002, James Kuyper wrote:

...
 >> IF (!D.FLAGS AND 8) AND (ratio LE minrat) THEN BEGIN
 >
 >
 > James, as an aside, this statement tries use AND both bitwise _and_ logically.
 > But I understand your point...

I realized after I sent the message that I'd used a messy example. It doesn't reflect any single, consistent mis-interpretation. I now fully understand what's going on, but that code represents an intermediate stage where I was just plain confused. :-)

...
 > When confronted with a similar situation I like to use "IF ((!D.FLAGS
 > AND 8) NE 0) ...", which is a little more general.

I used that construct two more times today, and by the time I'd written it the third time, I'd figured out the advantage of "NE 0".
