Subject: Re: CASE statement
Posted by Jeff Guerber on Sat, 08 Sep 2001 03:18:15 GMT
View Forum Message <> Reply to Message

On Fri, 7 Sep 2001, Brian Jackel wrote:

> CASE (1) OF
> (this OR that): x=1
> (something AND somethingelse): x=2
> (NOT theotherthing): x=3

Careful!!! Notice:

```
IDL> print, not 0, not 0b, not 1, not 1b
     -1 255      -2 254
IDL> print, (not (1 eq 2)) eq (1 ne 2)
  0
IDL>
```

Unfortunately, AND, OR, XOR, and NOT all operate _bitwise_, at least on integral types. The first two selectors in your CASE should be OK if the operands are all expressions that evaluate to 0 or 1 (which the relationals EQ, NE, GT, GE, LT, and LE all do; but watch out for the general case!). The third one very likely won't do what you intend.

This is why I've argued a couple times for a true logical type, like Fortran's, with "true" and "false" system constants and Boolean operators that always return them. More recently, though, I discovered some examples in the manual that use the 0 or 1 returned from the relationals in arithmetic expressions, so I guess we're stuck with them; I'd be mostly satisfied with operators that return only 0 or 1, and constants !true=1 and !false=0. (Hmmmm. Perhaps a type that can _only_ have the values 0 or 1, with operand conversion based on truth value??)

A while back, I tried to think up a function to evaluate the truth of its argument, for a general case, and return 0 or 1, for use in situations like this. The only thing I could come up with involved a loop over all its elements, enclosing an IF (or equivalent a?b:c) statement. OK for scalars, not so good for arrays.

> ELSE:MESSAGE,'Error- no match found for case statement'
> ENDCASE
>
> Kinda ugly, but gets the job done.

Agreed, and I use the idiom myself; but, you've got to be more careful with it than I for one think you should have to be!

Jeff Guerber
Raytheon ITSS
NASA Goddard Space Flight Ctr
Oceans & Ice Branch

P.S. I've long thought that a language where comparisons can be distributed, as in English, would be very handy:

if expression eq A or B or C or D then...
