Subject: Re: AND statements
Posted by steinhh on Mon, 01 Mar 1999 08:00:00 GMT
View Forum Message <> Reply to Message

In article <36da962e.18718769@146.80.9.44> philaldis@geocities.com (Phil Aldis) writes:

> So, to avoid that you have to do some pretty messy code. Say for
> example I've got :
>
> IF Ptr_Valid(ThisPointer) THEN BEGIN
> IF Size(*ThisPointer, /type) EQ 10 THEN BEGIN

Yep, that's right. You do get used to it, though, even if you're grown up with C style logical operators, taking advantage of the non-evaluation of unnecessary parts for every little scrap of efficiency improvement.

> However, I want to execute the same bit of code if it fails the
> Ptr_Valid and the Size(*ThisPointer, type0 EQ 10, so as far as I can
> see, (and I realise that I may be missing something pretty blatent),
> you have to use flags

[..snip..]

> While obviously this is not the end of the world, there could be more
> complex examples, and the code does look messy.

Yes, though you learn to rewrite those statements somewhat, like this:

```
flag = NOT ptr_valid(thispointer)
IF NOT flag then flag = size(*thispointer,/type) EQ 10

IF NOT flag THEN BEGIN
  ;; Pointer is valid and points to type 10
END ELSE BEGIN
  ;; Pointer is not valid or doesn't point to type 10
END
```

There could be some improvement with the ?: construct:

```
flag = NOT (ptr_valid(thispointer) ? size(*thispointer,/type) eq 10 : 0b)
```

should be equivalent to the first two lines in my example.
In fact, you could rewrite your original code like this:

```
if (ptr_valid(ptr) ? size(*ptr,/type) eq 10 : 0b) then print,*ptr $
```

```
else                                    flag=1b
```

I'm not sure whether *I* would use the ?: construct in cases like this.... it looks more messy to me, in fact..

Regards,

Stein Vidar
