Subject: assignment inside boolean expression
Posted by Patrick Broos on Mon, 10 Jul 2000 07:00:00 GMT
View Forum Message <> Reply to Message

I was wondering if it's common knowlege that one can put an IDL
assignment inside
a boolean expression (like in the C language). For example

if (v = 0) then ... assigns v and does not execute the "then"
statement, while
if (v = 1) then ... assigns v and does execute the then.

Just as in C I find this leads to really nasty bugs.

--
==========
Patrick S. Broos, Systems Analyst/Programmer patb@astro.psu.edu

Department of Astronomy and Astrophysics My office 814-863-7947
Penn State University
525 Davey Lab FAX 814-863-8686
University Park, PA 16802-6305
http://www.astro.psu.edu Group office 863-9550
==========

Subject: Re: assignment inside boolean expression
Posted by Martin Schultz on Tue, 18 Jul 2000 07:00:00 GMT
View Forum Message <> Reply to Message

Patrick Broos wrote:
>
> I was wondering if it's common knowlege that one can put an IDL
> assignment inside
> a boolean expression (like in the C language). For example
>
> if (v = 0) then ... assigns v and does not execute the "then"
> statement, while
> if (v = 1) then ... assigns v and does execute the then.
>
> Just as in C I find this leads to really nasty bugs.
>
> --

One application where I am actually using this is token search in
strings. It
can save you a few keystrokes and (in my opinion) make code more

readable, because
the temporary variable p is only used within the IF construct:

```
   if ((p=StrPos(test,'Martin')) ge 0) then $
     tail = StrMid(test,p,999)
```

as opposed to

```
   p = StrPos(test,'Martin')
   if p ge 0 then tail = StrMid(test,p,999)
```

--
[[[
[[Dr. Martin Schultz   Max-Planck-Institut fuer Meteorologie   [[
[[Bundesstr. 55, 20146 Hamburg   [[
[[phone: +49 40 41173-308   [[
[[fax: +49 40 41173-298   [[
[[martin.schultz@dkrz.de   [[
[[[

Subject: Re: assignment inside boolean expression
Posted by Martin Schultz on Tue, 18 Jul 2000 07:00:00 GMT
View Forum Message <> Reply to Message

Ben Tupper wrote:
>
> Craig Markwardt wrote:
>
>> Not to undercut you, but will (X AND 1) do the trick?
>>
>
> Thanks to Ken and Craig. I think for my purposes the following should suffice (I
> guess as long as I make sure that I'm working with an integer/long/byte type.)
>
> X = Indgen(6) - 2
>
> For i = 0, N_elements(X)-1 Do $
>   If X then Print, X[i], ': Odd' Else print, X[i], ': Even'
>
> -2: Even
> -1: Odd
> 0: Even
> 1: Odd
> 2: Even
> 3: Odd
>

> Thanks again,
>
> Ben

but if you start increasing the number of elements of X to say 1000000,
you are
certainly better off with:
answer=['even','odd']
print,answer[(x and 1)]

no loop ;-)

Example:
IDL> x=lindgen(20)-5
IDL> answer=['even','odd']
IDL> print,answer[x and 1]
odd even odd even odd even odd even odd even odd even odd even odd even
odd even odd even

BTW: X MOD 2 does not work for negative numbers !!!
IDL> print,answer[x mod 2]
even even even even even even odd even odd even odd even odd even odd
even odd even odd even

Cheers,
Martin
--
[[[[[[[[[
[[Dr. Martin Schultz Max-Planck-Institut fuer Meteorologie [[
[[Bundesstr. 55, 20146 Hamburg [[
[[phone: +49 40 41173-308 [[
[[fax: +49 40 41173-298 [[
[[martin.schultz@dkrz.de [[
[[[[[[[[[