
Subject: Re: assignment inside boolean expression
Posted by [Liam E. Gumley](#) on Mon, 10 Jul 2000 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Patrick Broos wrote:

> I was wondering if it's common knowledge 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.

Curious: I've never even considered using this syntax.

Enclosing a statement inside parentheses turns it into an expression,
which has a type and a value, e.g.

```
IDL> help, (v = 100)
<Expression>  INT      =    100
```

The variables in the right hand side of the statement must necessarily
be defined:

```
IDL> help, (zv = tv + vt)
% Variable is undefined: VT.
% Execution halted at: $MAIN$
```

If you take the following statements:

```
IDL> if (v = 0) then print, 'True'
IDL> help, v
V          INT      =    0
```

```
IDL> if (v = 1) then print, 'True'
True
IDL> help, v
V          INT      =    1
```

and remove the parentheses, the equivalent code is

```
IDL> v = 0
IDL> if (v) then print, 'True'
IDL> help, v
V          INT      =    0
```

```
IDL> v = 1
```

```
IDL> if (v) then print, 'True'  
True  
IDL> help, v  
V          INT      =      1
```

Recall that in IDL, integers with odd non-zero values are Boolean 'True'. Beware of floats and doubles though, where any non-zero value is Boolean 'True'.

Cheers,
Liam.
<http://cimss.ssec.wisc.edu/~gumley>
