
Subject: Re: multi conditional for loops

Posted by [Craig Markwardt](#) on Wed, 21 Jun 2000 07:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

K Mankoff <mankoff@colorado.edu> writes:

```
> Hi *,
> I am interested in coding with a c/c++ style multi-conditional for
> loop.
>
> c++ ex:
> for ( int i = 0; ( ( i < 10 ) && ( j <= 4 ) ); i++ )
> {
>     /* do something */
> }
>
> Here is sample IDL code. It runs but produces very interesting
> results. (this is IDL5.3 on Linux RH6.2)
>
> PRO test
> j = 0
> FOR i = 0, ((j LE 4) AND 10) DO BEGIN
>     j = j + 2
>     PRINT, i, j
> ENDFOR
> END
```

Your IDL snippet does not correspond to the C code at all. As already mentioned, in a FOR statement the two arguments are the first and last values of the iteration variable. The IDL FOR loop is not nearly as sophisticated as the C version. However, you can achieve the same thing with a WHILE loop:

```
j = 0
i = 0
while (j LE 4) AND (i LT 10) do begin
    j = j + 2
    print, i, j
endwhile
```

Also note that the expression you had, ((j LE 4) AND 10), is completely different than the expression (j LE 4) AND (i LT 10), even in C.

Finally, IDL does not distinguish between *logical* boolean operators and *bitwise* boolean operators, whereas C does. For example C has && for logical AND and & for bitwise AND. In IDL it is always bitwise. This means that an IDL expression like (i AND j) will fail for (i=1,j=2) while for C, (i && j) will succeed.

Craig

--

Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
