
Subject: Re: Yet another bug??

Posted by [thompson](#) on Wed, 17 May 1995 07:00:00 GMT

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

phil@peace.med.ohio-state.edu (Phil) writes:

> Sorry to keep posting but I've just started developing some useful
> tools in IDL and it seems I run across a new 'bug' every day. Could
> someone tell me the sense behind the following:

> IDL> print, not 0

> -1

> IDL> print, not 1

> -2

> Shouldn't (not 0) = 1 and (not 1) = 0 or am i just missing something?

> Is there something in the manuals that describes this logic?

> Just wondering.

What the NOT function really does is to reverse all the bits in a number. Thus the bit pattern representing 0, which not surprising is all 0s, turns into all 1s which is the bit pattern for the integer -1. Such behavior is really useful when one wants to get down to the bit level of a number, and I think that other languages behave this way as well. Apparently, any even integer is treated as false and any odd integer is treated as true. For example,

```
IDL> for i = -5,5 do if i then print,i
```

```
-5
```

```
-3
```

```
-1
```

```
1
```

```
3
```

```
5
```

But the function `KEYWORD_SET`, which is also associated with boolean values, seems to behave differently. It only treats the value 0 as false, and anything else as true.

```
IDL> for i = -5,5 do if keyword_set(i) then print,i
```

```
-5
```

```
-4
```

```
-3
```

```
-2
```

```
-1
```

```
1
```

```
2
```

```
3
4
5
IDL> print, keyword_set(not 1)
1
```

I consider that a bug.

Also, floating point numbers work differently than integer numbers, and match the behavior of KEYWORD_SET.

```
IDL> print, not 0.
1.00000
IDL> print, not 1.
0.00000
```

and

```
IDL> for i = -5,5 do if float(i) then print,i
-5
-4
-3
-2
-1
1
2
3
4
5
```

Basically, it's a mess. I'm not too upset about the difference between floating point and integer behavior, but KEYWORD_SET really worries me now. If I use a statement like

```
MYPROG, TEST_KEY=(NOT A)
```

and I'm using KEYWORD_SET inside the routine to test whether or TEST_KEY was set, I'll get the wrong answer.

Bill Thompson

Subject: Re: Yet another bug??
Posted by [lmudge](#) on Fri, 19 May 1995 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

In article 95May17132737@peace.med.ohio-state.edu, phil@peace.med.ohio-state.edu (Phil) writes:

```

> Sorry to keep posting but I've just started developing some useful
> tools in IDL and it seems I run across a new 'bug' every day. Could
> someone tell me the sense behind the following:
>
> IDL> print, not 0
>   -1
> IDL> print, not 1
>   -2
>
> Shouldn't (not 0) = 1 and (not 1) = 0 or am i just missing something?
> Is there something in the manuals that describes this logic?
>
> Just wondering.
>
>
> --
> /*****
> Phil Williams
> Postdoctoral Researcher          "One man gathers what
> MRI Facility                     another man spills..."
> The Ohio State University        -The Grateful Dead
> email: phil@peace.med.ohio-state.edu
> URL: http://justice.med.ohio-state.edu:1525
> *****/

```

One must thing in binary terms to understand what is happening here. Integer variables in IDL uses twos complement binary arithmetic thus the binary representation of 0 is 0000000000000001. When we perform a not on this we get 1111111111111110 which is the twos complement representation for -1. Try doing this:

```
not float(1)
```

and

```
not float(2)
```

and see what you get.

Regards,

Leith Mudge

```

-----
Leith Mudge
lmudge@awadi.com.au
AWA Defence Industries
PO Box 161

```

Elizabeth SA 5112
Australia
Phone: +61 8 256 0952
Fax: +61 8 255 9117

Microsoft Network is prohibited from redistributing this work in any form,
in whole or in part. Copyright, Leith Mudge, 1995
License to distribute this post is available to Microsoft for \$1000.
Posting without permission constitutes an agreement to these terms.
