
Subject: Re: -32768

Posted by [Yngvar Larsen](#) on Thu, 20 Mar 2014 21:30:49 GMT

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

On Thursday, 20 March 2014 18:01:01 UTC+1, Karl wrote:

> On Thursday, March 20, 2014 3:24:59 AM UTC-6, Rob Klooster wrote:

>

>> The reason it does not work, is that you first try to define +32768 and then negate the result. Since +32768 can only be expressed as a long, you end up with a long variable. Using fix is a solution, but you could also use the bitwise not operator, if you feel adventurous:

>

>> IDL> a = not 32767

>> IDL> help, a

>> A INT = -32768

Just for completeness of the thread.

(1) A literal short fails here:

```
IDL> help, -32768S
```

```
help, -32768
```

```
^
```

% Integer constant must be less than 32768.

I consider this a bug.

(2) The following silly tricks work as expected:

```
IDL> help, -32767S-1
```

```
<Expression>  INT      = -32768
```

```
IDL> help, 32767S+1
```

```
<Expression>  INT      = -32768
```

(3) Generalizing to 32- and 64-bits integers:

```
IDL> help, -2147483648L
```

```
help, -2147483648
```

```
^
```

% Long integer constant must be less than 2147483648.

```
IDL> help, -9223372036854775808LL
```

```
<Expression>  LONG64   = -9223372036854775808
```

```
IDL> help, -9223372036854775809LL
```

```
<Expression>  LONG64   = 9223372036854775807
```

Thus, we have the same bug for 32-bit integers, but not for 64-bit!!

```
IDL> print, !version  
{ x86_64 linux unix linux 8.2.2 Jan 23 2013    64    64}
```

--

Yngvar
