

---

Subject: Re: hexadecimal variables

Posted by [Vapuser](#) on Thu, 22 Apr 1999 07:00:00 GMT

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

---

Pavel Romashkin <promashkin@cmdl.noaa.gov> writes:

```
> Hi,  
> Is there a nice way to convert hexadecimal variables? Here's what I am  
> trying to do:  
>  
> This is received from the outside.  
>   h_string = '40000FC0DA'  
>  
>   h_long = long(h_string)  
> This, obviously, fails to recognize alphabetical hexadecimal part and  
> sets h_long to 40,000.  
>  
>   temp = execute("h_long = long(' "+h_string+" 'x)")  
> This, naturally, works, setting h_long to the desired 1032410,
```

This is not what h\_long equals; 1032410 = 'fc0da'x. You've only gotten the first 32 bits. See below.

```
> but looks  
> quite ugly. Besides, it is 3 times slower than without "execute" - not  
> really a concern on fast machines, but inefficient code just bugs me.  
>  
> I am sure I am missing something here. Can anyone suggest a neater way  
> to do this? Maybe, there is a way to use Z-formatted READ here?  
>  
> Thank you,  
> Pavel  
>
```

40000FC0DA is larger than 32 bits. Unless you're using idl 5.2, having access to 64 bit integers, you're out of luck, without kluging together something having the same flavor as a 'far' pointer in old MS-DOS.

The general method, however, is to use reads, which works on scalars and arrays. Here's an example, using IDL 5.2, and 64 bit integers.

```
IDL> h='40000FC0DA'  
IDL> x=0LL  
IDL> help,x  
X          LONG64  =          0  
IDL> reads,h,x,form='(Z)'
```

```
IDL> print,x,form='(z)'
      40000fc0da
IDL> print,x
      274878939354
IDL>
```

If you try this in idl<5.2,(with x=0L) you'll only get the right most 32 bits, i.e. X = 'fc0da'x. Witness...

```
IDL> x1=0L
IDL> reads,h,x1,form='(Z)'
IDL> print,x1
      1032410
IDL> print,x1,form='(z)'
      fc0da
```

If you really need the > 32 bits, you're going to have to use idl 5.2, or read it, split it, and recombine it as a double?

--

William Daffer  
vapuser@catspaw.jpl.nasa.gov

---