## Subject: Re: array subscript conversion Posted by Foldy Lajos on Wed, 25 Jul 2007 18:04:10 GMT

View Forum Message <> Reply to Message

On Wed, 25 Jul 2007, mgalloy@gmail.com wrote:

```
> On Jul 25, 9:36 am, FÖLDY Lajos <fo...@rmki.kfki.hu> wrote:
>> Hi guys,
>>
>> according to the manual, array subscripts are converted to long (or long64
>> on 64 bit systems) before use if necessary, so an explicit conversion
>> should not affect the result.
>>
>> IDL> print, !version
>> { x86 linux unix linux 6.3 Mar 23 2006
                                             32
                                                   64}
>> IDL>
>> IDL> a=lindgen(10)
>> IDL> print, a[[long(-1ull)]]
>>
>> IDL> print, a[[-1ull]]
>>
>>
>> Is it a bug or I am missing something?
>>
>> regards,
>> lajos
>
  All I see in the manual about converting subscripts to longs is:
>
  "Subscripts can be any type of vector or scalar expression. If a
  subscript expression is not integer, a longword integer copy is made
  and used to evaluate the subscript."
>
  around the middle of this page:
>
>
   http://idlastro.gsfc.nasa.gov/idl_html_help/Understanding_Ar ray_Subscripts.html
>
> By "integer", I think they mean the more general any integer type:
> byte, integer, long, etc. And in your case, the type is "integer", so
> no conversion is made.
>
I am I missing another statement in the help that says something more
 explicit?
>
> Mike
> www.michaelgalloy.com
>
```

You are right. I remembered "not long" instead of "not integer". And also remembered IDL\_MEMINT from idl\_export.h, which represents memory offsets and sizes, and assumed that IDL always uses IDL\_MEMINT for subscripting internally.

## Second try:

```
IDL> a=lindgen(10)
IDL>
IDL> print, a[4294967296II]
0
IDL> print, a[[4294967296II]]
9
```

In the first case the scalar subscript is "integer", so no conversion is needed. But it is converted to LONG (=0). Why?

regards, lajos