

---

Subject: Re: large array

Posted by [Dick Jackson](#) on Tue, 12 Oct 2004 17:43:48 GMT

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

---

Hi Wolf,

IDL's WHERE function recently (6.0?) received a new keyword, /L64, with this listing in the Online Help:

L64

By default, the result of WHERE is 32-bit integer when possible, and 64-bit integer if the number of elements being processed requires it. Set L64 to force 64-bit integers to be returned in all cases.

Note

Only 64-bit versions of IDL are capable of creating variables requiring a 64-bit result. Check the value of !VERSION.MEMORY\_BITS to see if your IDL is 64-bit or not.

Perhaps something in this will help you out.

Cheers,

--

-Dick

Dick Jackson / [dick@d-jackson.com](mailto:dick@d-jackson.com)  
D-Jackson Software Consulting / <http://www.d-jackson.com>  
Calgary, Alberta, Canada / +1-403-242-7398 / Fax: 241-7392

"Wolf Schweitzer" <[shwi@irm.unizh.ch](mailto:shwi@irm.unizh.ch)> wrote in message  
news:416c0503\$1@idnews.unizh.ch...

> Update.

>

> I believe that IDL can STILL only work with arrays of 2 GB size due to  
> internal representation even on systems where more space is available.  
> Positioning along any huge file using point\_lun and the Long64-function  
> and then reading a <2GB-part is no problem.

>

> <http://www.rsinc.com/services/techtip.asp?ttid=2597>

>

> But try:

>

> a = intarr (2000,2000,400)

> a [1999,1999,299] = 12

> b = where (a eq 12)

> help, b ; it is a LONG ... and that is a problem in my eyes;

> ; it shows they represent arrays with scalar counters and

```

> ; as long as these are LONG there is no way an array can be
> ; larger
>
>
> a = intarr (2000,2000,800)
> a [1999,1999,799] = 12
> b = where (a eq 12) ; (this really should crash your IDL)
>
> help, b ; i would be shocked if you get that far, but, please
> ; tell me what data type you obtain here
> print, b ; let me see this if you can, too
>
> Does anyone know when RSI will fix this serious limit? Until then, it'll
> probably be the chunk trick they wrote about.
>
>
>
> Wolf Schweitzer wrote:
>
>> I need to read a file that is ca. 7 GB large (the file size is defined as
>> 2048 x 2048 x 900 are the dimensions, x 2 (integers) + 512 bytes header).
>>
>> The format is known (.ISQ) and I have a routine that deals with the
>> header information efficiently; I have some similarly formatted files of
>> the same scan process around 800-900 MB that I can read without problems.
>>
>> Generally, I had no problems reading files up to 1.4 GB in size directly
>> into one variable under IDL.
>>
>> The data is read into an intarr (2048,2048,900) like this:
>>
>> imagearray = assoc (.., intarr(...), headersize)
>> imagearray = temporary (imagearray [0])
>>
>> Now, the reading takes a little while but it seems to work alright. The
>> machine - most of the time - does not crash (12 GB RAM, 64-bit AIX
>> version of IDL 6.1, IBM-Workstation).
>>
>> However, the visualisation of slice subscripts to this array later does
>> not display any interesting information; instead, the images of these
>> large data files look different each time and they do not reflect the
>> content of the data.
>>
>>
>> Question:
>>
>> What do I need to know in order to set up an array for very large data?

```

>> Is there a basic difference between arrays < 2 GB and arrays > GB of  
>> size? Are the subscript variables all multiplied before the array is  
>> really looked at, so do all of the subscript variables need to be Long64?  
>> Such as intarr (long64(x),long64(y),long64(z))?  
>>  
>> Thanks,  
>> Wolf.

---