Subject: Re: File sizes and the SAVE command Posted by Paolo Grigis on Wed, 22 Mar 2006 11:08:10 GMT

View Forum Message <> Reply to Message

```
Carsten Pathe wrote:
> Hi there.
>
> I am wondering about the IDL save command and the disk space of the
> created save files.
> Just an simple example:
>
> a = intarr(100000)
> tmp = size(a)
> print, string(format='(f10.3)',(tmp(1)*tmp(2))/(2.^10.))+' kbyte'
> ;195.313 kbyte
> save, a, filename='d:\temp\test\b.dat'
>
> b=fltarr(100000)
> tmp = size(b)
> print, string(format='(f10.3)',(tmp(1)*tmp(2))/(2.^10.))+' kbyte'
> ;390.625 kbyte
> save, a, filename='d:\temp\test\a.dat'
>
> c=dblarr(100000)
> tmp = size(c)
```

- > save, c, filename='d:\temp\test\c.dat'
 >
- > When you look at the created files and their sizes, you will see the

> print, string(format='(f10.3)',(tmp(1)*tmp(2))/(2.^10.))+' kbyte'

- > following:
- > a.dat 393 kb

> ;488.281 kbyte

- > b.dat 393 kb
- > c.dat 784 kb
- >
- > If you compare the file sizes to the sizes, the arrays were allocating
- > in the memory before they were save to disk, you see differences which
- > will cost you a lot of disk space when saving arrays of several hundred
- > megabytes.
- > Does anybody know, why the save command is producing files larger than
- > they should be?

Because (size(a))[2] is the type code, which has nothing to do with the byte size of each type, which is:

TYPE #BYTES

Byte 1

Integer 2
Unsigned Integer 2
Long 4
Unsigned Long 4
64-bit Long 8
64-bit Unsigned Long 8
Floating-point 4
Double-precision 8

Ciao, Paolo

>

- > PS: I know, that I can also use:
- > openw, 10, 'd:\temp\test\a.dat'
- > writeu, 10, a
- > close, 10
- > But when I want to restore the data, I have to know the structure of the
- > data to restore which is not always the case.

>

> Thanks a lot help