
Subject: IDL memory limitation?

Posted by [IDLmastertobe](#) on Thu, 15 Sep 2005 08:08:37 GMT

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

Hi, I am trying to visualize some 3D data using IDL and I found when my data size grow big, the program runs out of memory and won't work. Indeed, the data size is not too big yet, it is about 50MB max while some of my other data could be on the level of few or more GigaBytes in the future. Does anyone have any idea if IDL has any certain limitations on memory allocation? Thank you.

Subject: Re: IDL memory limitation?

Posted by [raval.chintan](#) on Sat, 17 Sep 2005 08:04:39 GMT

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

If you can provide some code so that we can test it out over here...

Chintan

Subject: Re: IDL memory limitation?

Posted by [Mark Hadfield](#) on Mon, 19 Sep 2005 00:32:49 GMT

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

raval.chintan@gmail.com wrote:

> If you can provide some code so that we can test it out over here...

Good advice. Also, trim it down to a problem you *can* handle and see how much memory is used relative to the size of the data you are visualising. You might then try stepping through the objects in your heap (as listed by the OBJ_VALID and PTR_VALID function) and determine how much memory each of them uses. You could do this by deleting each object in turn and seeing how much memory is freed (but remembering that some objects delete others as part of their cleanup). This may all be a bit tedious, but once you've established where the memory is being consumed you may be able to do something about it. Remember Karl's advice, some graphics objects use much more memory than the data they're representing, some don't.

--

Mark Hadfield "Kei puwaha te tai nei, Hoesa tahi tatou"

m.hadfield@niwa.co.nz

National Institute for Water and Atmospheric Research (NIWA)

Subject: Re: IDL memory limitation?

Posted by [Karl Schultz](#) on Mon, 19 Sep 2005 17:46:50 GMT

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

On Fri, 16 Sep 2005 17:18:09 -0400, IDLmastertobe wrote:

> Thank you everyone, here is my test result from memtest:
>
> Memory block # 1: 569 Mb (total: 569 Mb)
> Memory block # 2: 382 Mb (total: 951 Mb)
> Memory block # 3: 196 Mb (total: 1147 Mb)
> Memory block # 4: 183 Mb (total: 1330 Mb)
> Memory block # 5: 128 Mb (total: 1458 Mb)
> Memory block # 6: 88 Mb (total: 1546 Mb)
> Memory block # 7: 63 Mb (total: 1609 Mb)
> Memory block # 8: 59 Mb (total: 1668 Mb)
> Memory block # 9: 40 Mb (total: 1708 Mb)
> Memory block #10: 37 Mb (total: 1745 Mb)
>
> before I was visualizing 3-D data on the size of say
> 50x50x10, now im visualizing data on the size of 250x250x1 and it is
> giving me error saying it can't allocate memory to create array. I check

Posting some more details or part of the actual code would really help here.

You say that you have 3D data that is 50x50x10 and then you say 250x250x1, which isn't clear to me. Did you mean 250x250x10? What is the type of the data in the array? Byte? Float?

> the memory by using "help, /memory" and found the heap memory is used
> up. I am taking in 3D data and visualizing it by using IDLgrContainer.
> I created IDLgrModel and IDLgrAxis etc and put them together to
> visualize them. I can rotate it or flip it any way I want. It is a
> real time visualization. Does anyone know how I can overcome this
> memory problem? I currently have 1GB of RAM installed and 2GB of Virtual
> Memory allocated.
> Thanks.

IDLgrContainer is a container object and isn't capable of visualizing anything on its own. There needs to be some sort of graphic primitive like IDLgrSurface, IDLgrContour, IDLgrVolume, etc. Which one are you using?

You mentioned the IDLgrAxis, but that's not the object that is handling your actual data.

The amount of RAM and Virtual Memory that you have doesn't matter here if you are asking for a block of contiguous memory that is larger than the

largest available free block. In your case, the largest free block WAS 569MB at the time you ran memtest. The size of the largest free block may have been different at the time your application needed it. We won't know until you tell us how you are visualizing your data.

Karl
