
Subject: Re: IDL - freeing up used memory?
Posted by [Andy Sayer](#) on Fri, 31 May 2013 13:32:46 GMT
[View Forum Message](#) <> [Reply to Message](#)

As an update, in case anyone else has a similar issue in the future:

Using undefine helped somewhat (it got me further along before running into the issue).

However, and this may be something else I should have mentioned, I was using HDF5 files. Google searches suggest potential memory leaks with IDL and HDF5 (although I don't know how version/OS specific it is), e.g.: https://groups.google.com/forum/#!msg/comp.lang.idl-pvwave/9QJ_tcK7E2k/CLVPmPNVXJwJ

I had previously been using h5f_close to close each HDF5 file after I was done with it. Adding an additional h5_close statement (after h5f_close) resulted in my memory problem vanishing. From the documentation I don't know whether calling both h5f_close and h5_close are required, or what. I'm just happy that the code is working now.

Andy

On Wednesday, May 29, 2013 1:43:51 PM UTC-4, AMS wrote:

> Hi David,

>

>

>

> Thanks for the tip--I'll try using undefine and see whether that helps. Unfortunately, the data volume in the files is quite variable so the arrays will be quite different sizes.

>

>

>

> As an update, the same code runs without issue on IDL 7.1.1 on Mac OS 10.8. So perhaps it's in part related to CentOS?

>

>

>

> Thanks,

>

>

>

> Andy

>

>

>

> On Wednesday, May 29, 2013 1:20:46 PM UTC-4, David Fanning wrote:

>

>> AMS writes:

>

>>

```
>
>>
>
>>
>
>>
>
>>
>
>>> I'm running into a memory issue which I am not certain whether is related to my IDL code or
the machine the code is running on itself, and am wondering whether someone might be able to
help me out. To start with, I'm using IDL 7.1.1 on CentOS, invoking via command line.
>
>>
>
>>>
>
>>
>
>> I think your memory is getting fragmented. I would think about doing two
>
>>
>
>> things. First, I wouldn't set your unused variables to 0, as I think
>
>>
>
>> that promotes fragmentation. I would use the Coyote Library routine
>
>>
>
>> UNDEFINE to get rid of them:
>
>>
>
>>
>
>>
>
>> http://www.idlcoyote.com/programs/undefine.pro
>
>>
>
>>
>
>>
>
>> But, maybe better than this, I would try (assuming they are always the
```

```
>
>>
>
>> same size) to reuse the memory you have already set aside for reading in
>
>>
>
>> the variables:
>
>>
>
>>
>
>>
>
>> FOR j=0,n DO ....
>
>>
>
>> IF N_Elements(inputVar) EQ 0 THEN $
>
>>
>
>> inputVar = ... ELSE $
>
>>
>
>> inputVar = Temporary(inputVar) * 0
>
>>
>
>> ReadU, lun, inputvar
>
>>
>
>>
>
>> Cheers,
>
>>
>
>>
>
>>
>
>> David
```

>
>>
>
>> --
>
>>
>
>> David Fanning, Ph.D.
>
>>
>
>> Fanning Software Consulting, Inc.
>
>>
>
>> Coyote's Guide to IDL Programming: <http://www.idlcoyote.com/>
>
>>
>
>> Sepore ma de ni thue. ("Perhaps thou speakest truth.")
