Subject: memory allocation problem
Posted by lyubo on Thu, 21 Nov 2002 03:50:39 GMT

I have more than 300Mb free memory, but IDL cannot allocate
memory for a double array with dimensions 180, 180, 110.

array=DBLARR(180,180,110)

The array should take about 29Mb and I have more than enough
free memory. What is going on exactly? Is it because of memory
fragmentation or something else?

Is there any way around it?

Lyubo

Subject: Re: memory allocation problem
Posted by Randall Skelton on Thu, 21 Nov 2002 13:44:09 GMT

On Wed, 20 Nov 2002, lyubo wrote:

> I have more than 300Mb free memory, but IDL cannot allocate
> memory for a double array with dimensions 180, 180, 110.
>
> array=DBLARR(180,180,110)

What message are you getting? What operating system and IDL version? What
does 'help, /memory' show you?

Cheers,
Randall

Subject: Re: Memory allocation problem
Posted by Foldy Lajos on Wed, 26 Nov 2008 16:20:38 GMT

On Wed, 26 Nov 2008, kodark@gmail.com wrote:

> Hi all,
>
> I am a beginner-level user of IDL. Recently I run into the memory
> allocation problem: 'unable to allocate memory'. I loaded a large

> image (11820*11820, float, about 530M) into an array. Then when I
> tried to create two new arrays of the same size, the program said
> 'unable to allocate memory'.
>
> I have read a lot of articles about this problem on the net, but I
> still cannot solve this problem. My computer runs 32-bit Windows XP
> with 4GB RAM and I have assigned 10GB from hard drive to Virtual
> Memory. The virtual memory looks contiguous in the Disk Defragmenter
> program. I am curious about the following output.
>
> IDL> help,/memory
> heap memory used: 681301, max: 681324, gets: 765,
> frees: 413
>
> Then I executed the MemTest program provided by ittvis
>
> % Compiled module: MEMTEST.
> Memory block # 1: 806 Mb (total: 806 Mb)
> Memory block # 2: 390 Mb (total: 1196 Mb)
> Memory block # 3: 214 Mb (total: 1410 Mb)
> Memory block # 4: 143 Mb (total: 1553 Mb)
> Memory block # 5:   66 Mb (total: 1619 Mb)
> Memory block # 6:   59 Mb (total: 1678 Mb)
> Memory block # 7:   56 Mb (total: 1734 Mb)
> Memory block # 8:   40 Mb (total: 1774 Mb)
> Memory block # 9:   27 Mb (total: 1801 Mb)
> Memory block #10:   23 Mb (total: 1824 Mb)
>
> It is obvious that the second largest block is less than 530M.
>
> Then
> IDL> help,/memory
> heap memory used: 685533, max: 1913289245, gets: 789,
> frees: 433
>
> The heap memory max increased from 681324 to 1913289245.
>
> I want to know why the virtual memory is not recognized and used? why
> the heap memory max changed?
>

A process can use at most 2 GByte memory on a 32-bit Windows system.
10 GByte virtual memory means that you can run five 2 GByte process in
parallel.

regards,
lajos

Subject: Re: Memory allocation problem
Posted by Jean H. on Wed, 26 Nov 2008 16:45:00 GMT
View Forum Message <> Reply to Message

kodark@gmail.com wrote:
> Hi all,
>
> I am a beginner-level user of IDL. Recently I run into the memory
> allocation problem: 'unable to allocate memory'. I loaded a large
> image (11820*11820, float, about 530M) into an array. Then when I
> tried to create two new arrays of the same size, the program said
> 'unable to allocate memory'.
>
> I have read a lot of articles about this problem on the net, but I
> still cannot solve this problem. My computer runs 32-bit Windows XP
> with 4GB RAM and I have assigned 10GB from hard drive to Virtual
> Memory. The virtual memory looks contiguous in the Disk Defragmenter
> program. I am curious about the following output.

>
> Kodark

Hi Kodark,

Ok, 3 things here:
1) switch to Linux... you will have access to most of your 4 Gb of
memory, and, best of all, it's almost all continuous! I did it a few
years ago and have not regretted it!
2) With IDL 6 (and before), you can, under windows, have up to 3Gb of
memory (the segmentation problem is still there though). You need to
alter to IDL exe and to tell windows to start in 3Gb mode. The later is
easy (on your boot menu, add /3gb to the end). The former is slightly
more complicated... You need Visual Studio, use "editbin" on idl.exe,
then set the "large address space aware" flag to 1. Google it for more
info! BUT, your system may become unstable...
3)divide your work!... don't use 1 big array, but 4,9 or whatever it
takes smaller arrays! You can easily modify the memtest program to have
it return the available memory. Then, based on your knowledge of the
memory you will need for your computations, chunk the array accordingly!

Jean

Subject: Re: Memory allocation problem
Posted by Jean H. on Wed, 26 Nov 2008 16:47:20 GMT
View Forum Message <> Reply to Message

> 2) With IDL 6 (and before), you can, under windows, have up to 3Gb of

> memory ....

by the way, has anyone tried it with IDL 7?... I am not too sure of what
should be edited to get the 3gb working! The workspace? IDL? a mix of
both? Moreover, Eclipse is in Java and, from what my colleagues told me,
it's not possible to set the large address aware flag on Java... any info?

Jean

Subject: Re: Memory allocation problem
Posted by pgrigis on Wed, 26 Nov 2008 16:59:17 GMT
View Forum Message <> Reply to Message

But surely, windows IDL 64 bits with windows 64 bits should solve
this problem? Not that I have ever used either...

Cheers,
Paolo

Jean H. wrote:
>> 2) With IDL 6 (and before), you can, under windows, have up to 3Gb of
>> memory ....
>
> by the way, has anyone tried it with IDL 7?... I am not too sure of what
> should be edited to get the 3gb working! The workspace? IDL? a mix of
> both? Moreover, Eclipse is in Java and, from what my colleagues told me,
> it's not possible to set the large address aware flag on Java... any info?
>
> Jean

Subject: Re: Memory allocation problem
Posted by David Fanning on Wed, 26 Nov 2008 17:01:40 GMT
View Forum Message <> Reply to Message

Jean H. writes:

> 3)divide your work!... don't use 1 big array, but 4,9 or whatever it
> takes smaller arrays! You can easily modify the memtest program to have
> it return the available memory.

Maybe this is a dumb question, but where did you find MEMTEST?
A search of the ITTVIS web page and user contrib sites turns
up nothing (which, I admit, doesn't prove it's not there, but...).

Cheers,

David

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

Subject: Re: Memory allocation problem
Posted by R.G. Stockwell on Wed, 26 Nov 2008 17:15:29 GMT

"David Fanning" <news@dfanning.com> wrote in message
news:MPG.23972ed74e6c1fd498a566@news.giganews.com...
> Jean H. writes:
>
>> 3)divide your work!... don't use 1 big array, but 4,9 or whatever it
>> takes smaller arrays! You can easily modify the memtest program to have
>> it return the available memory.
>
> Maybe this is a dumb question, but where did you find MEMTEST?
> A search of the ITTVIS web page and user contrib sites turns
> up nothing (which, I admit, doesn't prove it's not there, but...).
>
> Cheers,
>
> David
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming: http://www.dfanning.com/
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

It was posted in this group, so i guess I can repost it.


```
pro memtest
  compile_opt idl2 ; set default integers to 32-bit and enforce [] for
indexing

  MB = long64(2)^20
  currentBlockSize = MB * 2047   ; 2 GB

print,'current block size = ',currentblocksize
  maxIterations = 10            ; Max loop iterations
```

```
  memPtrs = ptrarr(maxIterations)
  memBlockSizes = ulonarr(maxIterations)

 for i=0, maxIterations-1 do begin
 ; Error handler
   catch, err

   ; Sepcifically designed for "Failure to allocate memory..." error
   if (err ne 0) then begin
     currentBlockSize = currentBlockSize - MB     ; ...try 1 MB smaller
allocation
     if (currentBlockSize lt MB) then break  ; Give up, if currentBlockSize
< 1 MB
   endif

 ; This 'wait' enables Ctrl-Break to interrupt if necessary (Windows).
   wait, 0.0001

 ; Allocate memory (if possible)
   memPtrs[i] = ptr_new(bytarr(currentBlockSize, /nozero), /no_copy)
   memBlockSizes[i] = currentBlockSize   ; Store the latest successful
allocation size

 ; Print the current allocated block size and the running total, in Mb
   print, format='(%"Memory block #%2d: %6d Mb (total: %4d Mb)")', $
     i + 1, ishft(currentBlockSize, -20),
ishft(ulong(total(memBlockSizes)), -20)
 endfor

 ptr_free,memPtrs
end
```

Subject: Re: Memory allocation problem
Posted by David Fanning on Wed, 26 Nov 2008 17:31:05 GMT
View Forum Message <> Reply to Message

R.G. Stockwell writes:

> It was posted in this group, so i guess I can repost it.

Thank you.

Cheers,

David
--
David Fanning, Ph.D.

Subject: Re: Memory allocation problem
Posted by Rick Towler on Wed, 26 Nov 2008 17:43:19 GMT

View Forum Message <> Reply to Message

Fï¿½LDY Lajos wrote:
>
> On Wed, 26 Nov 2008, kodark wrote:
>> I am a beginner-level user of IDL. Recently I run into the memory
>> allocation problem: 'unable to allocate memory'. I loaded a large
>> image (11820*11820, float, about 530M) into an array. Then when I
>> tried to create two new arrays of the same size, the program said
>> 'unable to allocate memory'.
>>
>> I have read a lot of articles about this problem on the net, but I
>> still cannot solve this problem. My computer runs 32-bit Windows XP
>> with 4GB RAM and I have assigned 10GB from hard drive to Virtual
>> Memory. The virtual memory looks contiguous in the Disk Defragmenter
>> program. I am curious about the following output.
>>
>> IDL> help,/memory
>> heap memory used: 681301, max: 681324, gets: 765,
>> frees: 413
>>
>> Then I executed the MemTest program provided by ittvis
>>
>> % Compiled module: MEMTEST.
>> Memory block # 1: 806 Mb (total: 806 Mb)
>> Memory block # 2: 390 Mb (total: 1196 Mb)
>> Memory block # 3: 214 Mb (total: 1410 Mb)
>> Memory block # 4: 143 Mb (total: 1553 Mb)
>> Memory block # 5:   66 Mb (total: 1619 Mb)
>> Memory block # 6:   59 Mb (total: 1678 Mb)
>> Memory block # 7:   56 Mb (total: 1734 Mb)
>> Memory block # 8:   40 Mb (total: 1774 Mb)
>> Memory block # 9:   27 Mb (total: 1801 Mb)
>> Memory block #10:   23 Mb (total: 1824 Mb)
>>
>> It is obvious that the second largest block is less than 530M.
>>
>> Then
>> IDL> help,/memory
>> heap memory used: 685533, max: 1913289245, gets: 789,
>> frees: 433

>>
>> The heap memory max increased from 681324 to 1913289245.
>>
>> I want to know why the virtual memory is not recognized and used? why
>> the heap memory max changed?
>>
>
> A process can use at most 2 GByte memory on a 32-bit Windows system.
> 10 GByte virtual memory means that you can run five 2 GByte process in
> parallel.
>

To answer your other Q, the heap memory max changed because you ran
memtest.pro which allocated large chunks of memory right up to your
systems maximum (around 1.8GB).

This issue has been discussed at length and I suggest visiting the
ITTVIS website and reading their tech-tips and searching this newsgroup.

You may find this thread helpful:

 http://groups.google.com/group/comp.lang.idl-pvwave/browse_t
hread/thread/e185e0e1f243423c/7993eae2818df28b?hl=en&lnk
=gst&q=memtest#7993eae2818df28b

As of IDL 7 you'll still have to modify the executable to allow it to
benefit from the /3GB option. This isn't outlined in the thread above
but you should be able to google this easily.

-Rick

Subject: Re: Memory allocation problem
Posted by Rick Towler on Wed, 26 Nov 2008 18:07:25 GMT
View Forum Message <> Reply to Message

Jean H. wrote:

 > by the way, has anyone tried it with IDL 7?...

You set it on idl_opserver.exe.

Just tried it (for the record using the "command line" idl, not the
workbench):

pre-modification:

IDL> memtest
% Compiled module: MEMTEST.
current block size =   2146435072
Memory block # 1:   1178 Mb (total: 1178 Mb)
Memory block # 2:   212 Mb (total: 1390 Mb)
Memory block # 3:   116 Mb (total: 1506 Mb)
Memory block # 4:   114 Mb (total: 1620 Mb)
Memory block # 5:   66 Mb (total: 1686 Mb)
Memory block # 6:   52 Mb (total: 1738 Mb)
Memory block # 7:   36 Mb (total: 1774 Mb)
Memory block # 8:   32 Mb (total: 1806 Mb)
Memory block # 9:   26 Mb (total: 1832 Mb)
Memory block #10:   21 Mb (total: 1853 Mb)

post-modification:

IDL> memtest
current block size =   2146435072
Memory block # 1:   1178 Mb (total: 1178 Mb)
Memory block # 2:   751 Mb (total: 1929 Mb)
Memory block # 3:   212 Mb (total: 2141 Mb)
Memory block # 4:   116 Mb (total: 2257 Mb)
Memory block # 5:   114 Mb (total: 2371 Mb)
Memory block # 6:   66 Mb (total: 2437 Mb)
Memory block # 7:   52 Mb (total: 2489 Mb)
Memory block # 8:   36 Mb (total: 2525 Mb)
Memory block # 9:   32 Mb (total: 2557 Mb)
Memory block #10:   26 Mb (total: 2583 Mb)

Windows XP-32 "4GB" RAM, /3GB /USERVA=2800

Using editbin is simple:

c:\> editbin /LARGEADDRESSAWARE idl_opserver.exe

You will need your VS environment set up correctly (run vcvars32.bat if
needed) and then navigate to your bin\bin.x86 directory.

Not to state the obvious but make a copy before you modify anything!

-Rick

Subject: Re: Memory allocation problem

Thanks Rick! I will soon try it, as well as in the Workbench!
Jean

Rick Towler wrote:
> Jean H. wrote:
>
>> by the way, has anyone tried it with IDL 7?...
>
>
> You set it on idl_opserver.exe.
>
> Just tried it (for the record using the "command line" idl, not the
> workbench):
>
> pre-modification:
>
> IDL> memtest
> % Compiled module: MEMTEST.
> current block size =   2146435072
> Memory block # 1:   1178 Mb (total: 1178 Mb)
> Memory block # 2:   212 Mb (total: 1390 Mb)
> Memory block # 3:   116 Mb (total: 1506 Mb)
> Memory block # 4:   114 Mb (total: 1620 Mb)
> Memory block # 5:   66 Mb (total: 1686 Mb)
> Memory block # 6:   52 Mb (total: 1738 Mb)
> Memory block # 7:   36 Mb (total: 1774 Mb)
> Memory block # 8:   32 Mb (total: 1806 Mb)
> Memory block # 9:   26 Mb (total: 1832 Mb)
> Memory block #10:   21 Mb (total: 1853 Mb)
>
> post-modification:
>
> IDL> memtest
> current block size =   2146435072
> Memory block # 1:   1178 Mb (total: 1178 Mb)
> Memory block # 2:   751 Mb (total: 1929 Mb)
> Memory block # 3:   212 Mb (total: 2141 Mb)
> Memory block # 4:   116 Mb (total: 2257 Mb)
> Memory block # 5:   114 Mb (total: 2371 Mb)
> Memory block # 6:   66 Mb (total: 2437 Mb)
> Memory block # 7:   52 Mb (total: 2489 Mb)
> Memory block # 8:   36 Mb (total: 2525 Mb)
> Memory block # 9:   32 Mb (total: 2557 Mb)
> Memory block #10:   26 Mb (total: 2583 Mb)
>
>

> Windows XP-32 "4GB" RAM, /3GB /USERVA=2800
>
>
> Using editbin is simple:
>
> c:\> editbin /LARGEADDRESSAWARE idl_opserver.exe
>
>
> You will need your VS environment set up correctly (run vcvars32.bat if
> needed) and then navigate to your bin\bin.x86 directory.
>
> Not to state the obvious but make a copy before you modify anything!
>
> -Rick
>

Subject: Re: Memory allocation problem
Posted by pgrigis on Wed, 26 Nov 2008 22:39:57 GMT
View Forum Message <> Reply to Message

b_...@hotmail.com wrote:
> Maybe not helpful for your specific problem, but if you really need
> big arrays then just take advantage of a 64-bit OS and cheap RAM:
>
> IDL> memtest
> Memory block # 1: 2047 Mb (total: 2047 Mb)
> Memory block # 2: 2047 Mb (total: 4094 Mb)

Seems like that there's a bug in memtest,
this should probably go up to 6000 instead?
> Memory block # 3: 2047 Mb (total: 2045 Mb)
> Memory block # 4: 2047 Mb (total: 4092 Mb)
> Memory block # 5: 2047 Mb (total: 2043 Mb)
> Memory block # 6: 570 Mb (total: 2613 Mb)
>
> WinXP x64, 8GB RAM, IDL 7.

Subject: Re: Memory allocation problem
Posted by Wout De Nolf on Thu, 27 Nov 2008 09:34:50 GMT
View Forum Message <> Reply to Message

On Wed, 26 Nov 2008 10:15:29 -0700, "R.G. Stockwell"
<nothanks@noemail.com> wrote:

> ; Print the current allocated block size and the running total, in Mb

---

```
>    print, format='(%"Memory block #%2d: %6d Mb (total: %4d Mb)")', $
>      i + 1, ishft(currentBlockSize, -20),
> ishft(ulong(total(memBlockSizes)), -20)
```

Change ulong to ulong64 when on a 64bit machine with more than ~4GB memory.

Subject: Re: Memory allocation problem
Posted by Wout De Nolf on Thu, 27 Nov 2008 09:37:44 GMT

On Wed, 26 Nov 2008 14:39:57 -0800 (PST), Paolo <pgrigis@gmail.com> wrote:

> Seems like that there's a bug in memtest,
> this should probably go up to 6000 instead?

Last line: ulong -> ulong64 otherwise it will never go above 4095MB.

Subject: Re: Memory allocation problem
Posted by b_gom on Thu, 27 Nov 2008 18:19:02 GMT

On Nov 27, 2:37 am, Wox <s...@nomail.com> wrote:
> Last line: ulong -> ulong64 otherwise it will never go above 4095MB.

Yes, with this modification:

```
  print, format='(%"Memory block #%2d: %4d Mb (total: %6d Mb)")', $
    i + 1, ishft(currentBlockSize, -20), ishft(ulong64(total
(memBlockSizes)), -20)
```

% Compiled module: MEMTEST.
Memory block # 1: 2047 Mb (total: 2047 Mb)
Memory block # 2: 2047 Mb (total: 4094 Mb)
Memory block # 3: 2047 Mb (total: 6141 Mb)
Memory block # 4: 2047 Mb (total: 8188 Mb)
Memory block # 5: 2047 Mb (total: 10235 Mb)
Memory block # 6: 572 Mb (total: 10807 Mb)

Subject: Re: Memory allocation problem
Posted by pgrigis on Fri, 28 Nov 2008 16:04:24 GMT

There seems to be a problem with that on my machine.
memtest reports that it can allocate *any* memory size
I want (at least up to 2 TB).
Since that is larger than my hard drive, not even
reckless virtual memory usage could go that far...

```
IDL> help,!version,/st
** Structure !VERSION, 8 tags, length=104, data length=100:
  ARCH          STRING    'x86_64'
  OS            STRING    'darwin'
  OS_FAMILY     STRING    'unix'
  OS_NAME       STRING    'Mac OS X'
  RELEASE       STRING    '7.0.4'
  BUILD_DATE    STRING    'Sep  3 2008'
  MEMORY_BITS   INT           64
  FILE_OFFSET_BITS
                INT           64
```

Ciao,
Paolo

b_...@hotmail.com wrote:

>> Last line: ulong -> ulong64 otherwise it will never go above 4095MB.
>
>
> Yes, with this modification:
>
>   print, format='(%"Memory block #%2d: %4d Mb (total: %6d Mb)")', $
>   i + 1, ishft(currentBlockSize, -20), ishft(ulong64(total
> (memBlockSizes)), -20)
>
> % Compiled module: MEMTEST.
> Memory block # 1: 2047 Mb (total: 2047 Mb)
> Memory block # 2: 2047 Mb (total: 4094 Mb)
> Memory block # 3: 2047 Mb (total: 6141 Mb)
> Memory block # 4: 2047 Mb (total: 8188 Mb)
> Memory block # 5: 2047 Mb (total: 10235 Mb)
> Memory block # 6: 572 Mb (total: 10807 Mb)

Subject: Re: Memory allocation problem
Posted by Foldy Lajos on Fri, 28 Nov 2008 16:12:47 GMT

On Fri, 28 Nov 2008, Paolo wrote:

> There seems to be a problem with that on my machine.
> memtest reports that it can allocate *any* memory size
> I want (at least up to 2 TB).
> Since that is larger than my hard drive, not even
> reckless virtual memory usage could go that far...
>
> IDL> help,!version,/st
> ** Structure !VERSION, 8 tags, length=104, data length=100:
>   ARCH   STRING   'x86_64'
>   OS   STRING   'darwin'
>   OS_FAMILY   STRING   'unix'
>   OS_NAME   STRING   'Mac OS X'
>   RELEASE   STRING   '7.0.4'
>   BUILD_DATE   STRING   'Sep 3 2008'
>   MEMORY_BITS   INT   64
>   FILE_OFFSET_BITS
>   INT   64

A shot in the dark: memtest does not use the memory, only allocates it.
Probably Mac OS X is 'lazy' and it allocates memory pages when they are
first used. Try to remove the /nozero keyword in

  ; Allocate memory (if possible)
    memPtrs[i] = ptr_new(bytarr(currentBlockSize, /nozero), /no_copy)

regards,
lajos

Subject: Re: Memory Allocation Problem
Posted by Craig Markwardt on Tue, 14 Aug 2012 23:51:51 GMT
View Forum Message <> Reply to Message

On Tuesday, August 14, 2012 9:34:11 AM UTC-4, adh...@gmail.com wrote:
> I will paste the other part of this section of the program below. I thought I might need to use the
TEMPORARY command to free up some room but am not sure how to appropriately determine
the point at which this should be placed. Any suggestions would be greatly appreciated as always!

When memory constrained, use TEMPORARY() after the last time you use a variable. Example:
After you use the LATLON variable once, you never use it again.

Actually, I use David Fanning's UNDEFINE procedure because the semantics are better.   The
procedural statement,
   UNDEFINE, LATLON
just makes sense.

Craig

Hi Craig,

Thank you for your response. I am pretty new to IDL so I must ask a couple of questions that are possibly a bit uneducated:

1.) Looking at David Fanning's site I see that there is a whole little .pro for this UNDEFINE procedure. Do I need to incorporate that in to my current set of programs, or is that for a different use.

2.) If I am not required to use the whole .pro file where do I execute this UNDEFINE statement? When I attempt to incorporate it in the code itself it is not recognized.

Thank you,
Allisyn

On Tuesday, August 14, 2012 7:51:51 PM UTC-4, Craig Markwardt wrote:
> On Tuesday, August 14, 2012 9:34:11 AM UTC-4, adh...@gmail.com wrote:
>
>> I will paste the other part of this section of the program below. I thought I might need to use the TEMPORARY command to free up some room but am not sure how to appropriately determine the point at which this should be placed. Any suggestions would be greatly appreciated as always!
>
>
>
> When memory constrained, use TEMPORARY() after the last time you use a variable. Example: After you use the LATLON variable once, you never use it again.
>
>
>
> Actually, I use David Fanning's UNDEFINE procedure because the semantics are better.   The procedural statement,
>
>   UNDEFINE, LATLON
>
> just makes sense.
>
>
>
> Craig

Subject: Re: Memory Allocation Problem
Posted by Craig Markwardt on Thu, 16 Aug 2012 00:27:43 GMT

On Wednesday, August 15, 2012 1:59:38 PM UTC-4, adh...@gmail.com wrote:
> Hi Craig,
>
>
>
> Thank you for your response. I am pretty new to IDL so I must ask a couple of questions that are possibly a bit uneducated:
>
>
>
> 1.) Looking at David Fanning's site I see that there is a whole little .pro for this UNDEFINE procedure. Do I need to incorporate that in to my current set of programs, or is that for a different use.
>
>
>
> 2.) If I am not required to use the whole .pro file where do I execute this UNDEFINE statement? When I attempt to incorporate it in the code itself it is not recognized.

The shortest answer if you are just starting out, don't worry about TEMPORARY(), just set the variable to 0.
  LATLON = 0
It works 99.99999% as well. Who cares if it uses two bytes of storage?

The short answer is, download the file from Fanning's site and put it in your current directory. That works as long as you don't change to another directory.

The slightly longer answer is, download the file and put it with your IDL path. Learn about your IDL path.

The even longer answer is, download everything from Fanning's site and put it in your path!

Strangely, the longest answer is the shortest to write.

Craig

Subject: Re: Memory Allocation Problem
Posted by David Fanning on Thu, 16 Aug 2012 01:50:05 GMT

Craig Markwardt writes:

> The slightly longer answer is, download the file and put it with your IDL path. Learn about your

IDL path.
>
> The even longer answer is, download everything from Fanning's site and put it in your path!

Here are some articles you can read to learn about your
IDL path and how to get the IDL programs on it to
work like magic:

  http://www.idlcoyote.com/code_tips/installcoyote.php
  http://www.idlcoyote.com/misc_tips/coyote_project.php
  http://www.idlcoyote.com/misc_tips/idl_startupfile.php
  http://www.idlcoyote.com/tips/namefiles.html

Cheers,

David

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