Subject: IDL5 and large structures: Program code area full
Posted by Alex Schuster on Tue, 09 Dec 1997 08:00:00 GMT
View Forum Message <> Reply to Message

Hi there!

There is this program, I didn't use it for a long time, and today I started it for the first time under IDL 5.02. I have heard about some strange bugs with structures containing many tags, but as I didn't have problems with IDL 4, I am surprised I got a 'Program code full error'.

Any ideas? Hey, didn't RSI tell us all limitations are gone now? And why do I have more limitations than before?

This is the code:

```
-------------------------snip------------------------------
some_variable = {_refima,   $
 disp  : intarr (256, 256, 3), $
 disp_3d  : intarr (256, 256, 3), $
 ident   :   0,  $
 image_type  :   0,   $
 filename  : '',  $
 pix  : fltarr(3),  $
 dim  : intarr(3), $
 dim_old  : intarr(3), $
 num_planes  :   0,   $
 start_plane  :   0,   $
 frame   :   0,  $
 frame_old :   0,  $
 num_frames  :   0,  $
 gate   :   0,  $
 gate_old :   0,  $
 num_gates  :   0,  $
 bed_pos  :   0,  $
 bed_pos_old :   0,  $
 num_bed_pos  :   0,  $
 header_size :   0L,  $
 offset   : fltarr(3),  $
 angles   : fltarr(3), $
 center  : fltarr(3), $
 interpolation :   0,  $
 inv_flag :   0,  $
 scale_factor : 0.0,  $
 min  :   0,  $
 max  :   0,  $
 scal  : intarr(2), $
 diff  : intarr(2), $
 surf  : intarr(2), $
 grad  : intarr(2),  $
 cont  : intarr(2),  $
 quant  : fltarr(10), $
 num_contours  :   0,  $
 contour_mode :   0,  $
 sigma  : 0.0,  $
 average  : intarr(3), $
 filter_type :   0,  $
 kernel_size :   0,   $
 color_table :   0,  $
 gamma  : 0.0,  $
 vox_flag :   0,  $
 render_control :   0,  $
 viewport : intarr (3),  $
 rotation : fltarr (3,3), $
 surf_levels  : intarr(2),  $
 mask   : intarr(2),  $
 int_off  :   0,  $
 int_len  :   0,   $
 annotation : ''  }
------------------------snap----------------------------
```

I just pasted into the shell, and IDL says:

```
[...]
IDL>  mask   : intarr(2),  $
IDL>  int_off  :   0,  $

 int_off  :   0,  $
            ^
% Program code area full.
IDL>  int_len  :   0,   $

 int_len  :   0,   $
            ^
% Syntax error.
IDL>  annotation : !STR80}

IDL> help, !version, /structure
** Structure !VERSION, 5 tags, length=40:
   ARCH            STRING    'sparc'
   OS              STRING    'sunos'
   OS_FAMILY       STRING    'unix'
   RELEASE         STRING    '5.0'
   BUILD_DATE      STRING    'Apr 28 1997'
```

Can anyone using IDL 5.02 be so kind check if this works? And, no, I
don't even think about modifying the program in any way.

Many thanks in advance,

Alex

--
Alex Schuster   Wonko@weird.cologne.de   PGP Key available
alex@pet.mpin-koeln.mpg.de

Subject: Re: IDL5 and large structures: Program code area full
Posted by R. Bauer on Wed, 17 Dec 1997 08:00:00 GMT

View Forum Message <> Reply to Message

Alex Schuster wrote:

> Wow, I get fast responses today!
>
> David Fanning helped me a lot when he wrote:
>
>> [reasonably long structure snipped]
>>
>>> Can anyone using IDL 5.02 be so kind check if this works?
>>
>> Well, it certainly works as you describe when it is "pasted"
>> onto the IDL command line, but why in the world would you
>> be doing this?
>
> This code is part of a script, which I invoke via @.
> The code I posted isn't the original code, I tried to make it more
> readable. In the original code, the variables (some_variable, str80
> etc.) are defined as system variables via DEFSYSV. I didn't like common
> blocks when I wrote the program, so I chose to use global system
> variables.
>
>>> And, no, I
>>> don't even think about modifying the program in any way.
>>
>> Well, I think I would modify it by adding an END statement
>> and at least running it as a main-level program. At least
>> then it will work. :-)
>>
>>   IDL> .Run structure_def
>
> Good idea! I really should have tried this myself.
> After some changes (I define the structure in the program, and then,
> outside, the system variable) the code now compiles completely, and

> works.
>

An other way is to make a procedure from it e.g.

```
pro some,some_variable
some_variable = {_refima,              $
       disp            : intarr (256, 256, 3), $
       disp_3d          : intarr (256, 256, 3), $
       ident            :   0,          $
       image_type       :   0 }
end
```

The main program can compile it by resolve_routine and use execute in this way.

```
pro main
resolve_routine,'some'
a=execute('some,some_variable')
help,some_variable,/str

end
```

I would prefer this solution because I don't like storing unused variables or common blocks

--
R.Bauer

Institut fuer Stratosphaerische Chemie (ICG-1)
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de

Subject: Re: IDL5 and large structures: Program code area full
Posted by R. Bauer on Wed, 17 Dec 1997 08:00:00 GMT

View Forum Message <> Reply to Message

David Fanning wrote:

> Alex Schuster (alex@rosa.mpin-koeln.mpg.de) writes:
>
>> There is this program, I didn't use it for a long time, and today I
>> started it for the first time under IDL 5.02. I have heard about some
>> strange bugs with structures containing many tags, but as I didn't have

>> problems with IDL 4, I am surprised I got a 'Program code full error'.
>>
>> Any ideas? Hey, didn't RSI tell us all limitations are gone now? And why
>> do I have more limitations than before?
>>
>> This is the code:
>
> [reasonably long structure snipped]
>
>> Can anyone using IDL 5.02 be so kind check if this works?
>
> Well, it certainly works as you describe when it is "pasted"
> onto the IDL command line, but why in the world would you
> be doing this?
>

I did the same whith idl5.03

```
IDL>  help,!version,/str
** Structure !VERSION, 5 tags, length=40:
   ARCH            STRING    'ibmr2'
   OS              STRING    'AIX'
   OS_FAMILY       STRING    'unix'
   RELEASE         STRING    '5.0.3'
   BUILD_DATE      STRING    'Nov  3 1997'
```

and there NO Problems.

R.Bauer

--
R.Bauer

Institut fuer Stratosphaerische Chemie (ICG-1)
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de