

Hi mark

please tell a bit more

help, !version,/str

and please explain a bit more why you are using perl for doing this.

Reimar

mshep wrote:

```
> Hi,  
>  
> I did a quick search on the news group but did not find anything  
> recent.  
>  
> I thought the "Program code area full" when explicitly defining arrays  
> was resolved back in IDL 5.3, but I ran into it again.  
>  
> Here is a simple example to demonstrate my issue:  
>  
> IDL> .r full_test.pro  
> % Compiled module: FULL_TEST.  
> IDL>  
> IDL> full_test,  
> [0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9  
> ,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9],  
> $  
> IDL>  
> [0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9  
> ,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9],  
> $  
> IDL>  
> [0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9  
> ,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9],  
> $  
> IDL>  
> [0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9  
> ,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9],  
> $  
> IDL>  
> [0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9  
> ,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9],
```

```

> $
>
>
> [0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9
,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9],
> $
>
>
>          ^
> % Program code area full.
> IDL>
>
> -----
>
> PRO full_test, a_arr, b_arr, c_arr, d_arr, e_arr, f_arr , g_arr
>
> END
>
> -----
>
> I am generating arrays in a perl script and passing them into IDL so
> simply setting:
>
> a_arr =
> [0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9
,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9]
>
> and
>
> full_test, a_arr, ....
>
> is not an option for me.
>
> Any insights would be greatly appreciated.
>
> Thanks,
> mark
>

```

--

Reimar Bauer

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

a IDL library at ForschungsZentrum Juelich
http://www.fz-juelich.de/icg/icg-i/idl_icglib/idl_lib_intro.html
=====

Subject: Re: IDL 6.2 Array Definition Question (Program code area full)
Posted by [Ken Mankoff](#) on Wed, 19 Apr 2006 16:16:30 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Wed, 19 Apr 2006, mshep wrote:

> I am generating arrays in a perl script and passing them into IDL
> so simply setting:
>
> a_arr = [0,1,2,...,9,0,1,2,3,4,5,6,7,8,9]
> and
> full_test, a_arr,
>
> is not an option for me.

I'm not clear why this is not an option. Why exactly does perl or
external array generation place this limitation on you?

-k.

Subject: Re: IDL 6.2 Array Definition Question (Program code area full)
Posted by [mshep](#) on Wed, 19 Apr 2006 17:49:04 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Reimar,

The main question is why does IDL place a limit on the amount of array
elements being passed explicitly?

help, !version, /str

IDL> help, !version, /str

** Structure !VERSION, 8 tags, length=104, data length=100:

```
ARCH      STRING  'x86_64'
OS        STRING  'linux'
OS_FAMILY  STRING  'unix'
OS_NAME    STRING  'linux'
RELEASE    STRING  '6.2'
BUILD_DATE STRING  'Jun 20 2005'
MEMORY_BITS INT     64
FILE_OFFSET_BITS
```

IDL>

As for the reason why I am passing this many elements explicitly is that the arrays are being generated in a perl script and then passed to an IDL program.

Below is a simple example where I am explicitly passing arrays from perl to IDL, but I run into a limitation on the amount of elements that I can pass explicitly:

```
#!/usr/bin/perl -w
```

```
@a_arr =
(0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9);
@b_arr =
(0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9);
@c_arr =
(0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9);
@d_arr =
(0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9);
@e_arr =
(0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9);
@f_arr =
(0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9);
@g_arr =
(0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9);

$str_a = "@a_arr";      # put array into a string with blanks
between elements
$str_a =~ s/ /,/'g;      # substitute
$str_a = "[".$str_a."]"; # Add IDL brackets and end quotes
$str_b = "@b_arr";      # put array into a string with blanks
between elements
$str_b =~ s/ /,/'g;      # substitute
$str_b = "[".$str_b."]"; # Add IDL brackets and end quotes
$str_c = "@c_arr";      # put array into a string with blanks
between elements
$str_c =~ s/ /,/'g;      # substitute
$str_c = "[".$str_c."]"; # Add IDL brackets and end quotes
$str_d = "@d_arr";      # put array into a string with blanks
between elements
$str_d =~ s/ /,/'g;      # substitute
$str_d = "[".$str_d."]"; # Add IDL brackets and end quotes
$str_e = "@e_arr";      # put array into a string with blanks
between elements
$str_e =~ s/ /,/'g;      # substitute
$str_e = "[".$str_e."]"; # Add IDL brackets and end quotes
```

```
$str_f = "@f_arr";      # put array into a string with blanks
between elements
$str_f =~ s/ /','/g;    # substitute
$str_f = "[".$str_f."]"; # Add IDL brackets and end quotes
$str_g = "@g_arr";      # put array into a string with blanks
between elements
$str_g =~ s/ /','/g;    # substitute
$str_g = "[".$str_g."]"; # Add IDL brackets and end quotes

# Use a here document to run IDL from linux/unix command line

$idl_prog_full_test = "full_test";
$idl_run = ".r $idl_prog_full_test";
$exit = "exit";
$idl_command_try_2 = "$idl_prog_full_test, $str_a, $str_b, $str_c,
$str_d, $str_e, $str_f, $str_g";
system("idl
<<IDL_INPUT\n$idl_run\n$idl_command_try_2\n$exit\nIDL_INPUT\n ");

exit;
```

Subject: Re: IDL 6.2 Array Definition Question (Program code area full)
Posted by [mshep](#) on Wed, 19 Apr 2006 17:58:08 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Ken,

Thanks for the reply.

As I mentioned in a response to Reimar, the main question is why does IDL place a limit on the amount of array elements being passed explicitly?

As for the reason why I would like to do it, please refer to the perl example I provided in my response to Reimar.

I could just write out the arrays in perl to a file and then read the file in using IDL, but I would like to know why I cannot just pass them using explicit arrays.

Thanks,
Mark

Subject: Re: IDL 6.2 Array Definition Question (Program code area full)
Posted by [news.qwest.net](#) on Wed, 19 Apr 2006 18:28:23 GMT

"mshep" <mshep@aer.com> wrote in message
news:1145469488.757805.179980@i40g2000cwc.googlegroups.com.. .

>
> Hi Ken,
>
> Thanks for the reply.
>
> As I mentioned in a response to Reimar, the main question is why does
> IDL place a limit on the amount of array elements being passed
> explicitly?

Everything limits their input buffer, or else a user could crash the
computer
if they "accidentally" gave too large of an input. It is a famous bug from
the old
days. There are just better ways to handle the situation. It is like you
are
trying to put your luggage into your glove box instead of the trunk. :)

> I could just write out the arrays in perl to a file and then read the
> file in using IDL

sounds good.

Or use your existing perl script to create a main level IDL code where
you define your arrays:

Pro call_full_test

```
a_arr =  
[0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9 ,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9]  
b_arr =  
[0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9 ,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9]  
c_arr =  
[0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9 ,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9]  
d_arr =  
[0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9 ,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9]  
e_arr =  
[0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9 ,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9]  
f_arr =  
[0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9 ,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9]  
g_arr =  
[0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9 ,0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5,6,7,8,9]
```

full_test, a_arr, b_arr, c_arr, d_arr, e_arr, f_arr , g_arr

end

```
Then have your script call "call_full_test"
$idl_prog_full_test = "call_full_test";
$idl_run = ".r $idl_prog_call_full_test";
$exit = "exit";
$idl_command_try_2 = "$idl_prog_call_full_test";
```

Cheers,
bob

Subject: Re: IDL 6.2 Array Definition Question (Program code area full)
Posted by [mshep](#) on Wed, 19 Apr 2006 19:10:16 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Bob,

Thanks for the quick response and suggestions.

For my current application it is probably the best to just write out the arrays into a file and then read them in. However, I just wanted to let you know that your suggestion of using the existing perl script to write a main level IDL code with the defined arrays is also a very good approach that I had not thought of previously.

Thanks again,
Mark

Subject: Re: IDL 6.2 Array Definition Question (Program code area full)
Posted by [JD Smith](#) on Thu, 20 Apr 2006 21:12:34 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Wed, 19 Apr 2006 12:10:16 -0700, mshep wrote:

>
> Hi Bob,
>
> Thanks for the quick response and suggestions.
>
> For my current application it is probably the best to just write out
> the arrays into a file and then read them in. However, I just wanted to
> let you know that your suggestion of using the existing perl script to
> write a main level IDL code with the defined arrays is also a very good

> approach that I had not thought of previously.

A little known fact: if you set !EDIT_INPUT=0, the input buffer becomes unlimited (well, memory limited). This also turns off background widget event processing, so I wouldn't recommend it for general use.

JD
