
Subject: Re: resizing system variable

Posted by [Craig Markwardt](#) on Thu, 20 Dec 2012 20:28:45 GMT

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

On Thursday, December 20, 2012 2:38:00 PM UTC-5, Bob Plano wrote:

> My apologies if this has been answered elsewhere, or if the answer is
>
> obvious, but...
>
>
>
> I use defsysv to define an array. Later on, I want to set that system
>
> variable equal to an array of another size, but I get an error message
>
> that the system variable and second array are conflicting data
>
> structures:
>
>
>
> pro one
>
> defsysv,!var',fltarr(10)
>
> !var= findgen(20) ;this gives an error.
> end
>
>
>
> I've tried setting !var=!null, but I get an error message to the
> effect the !null is undefined. Which is true, in a way, of course.
>
> Does anyone know of a fix for this? Could I define !var as a pointer,
> which could point at one or another array?

You can only set the dimensions of a system variable once, the first time you initialize it. You can get around it using a pointer.

It sounds to me you are trying to use a system variable like a common block (i.e. global variable). See if you can rethink how you do it without a system variable, because it will probably come back to bite you.

Craig

Subject: Re: resizing system variable

Posted by [astroboy.20000](#) on Thu, 20 Dec 2012 21:44:37 GMT

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

On Thursday, December 20, 2012 3:28:45 PM UTC-5, Craig Markwardt wrote:

> On Thursday, December 20, 2012 2:38:00 PM UTC-5, Bob Plano wrote:

>

>

> You can only set the dimensions of a system variable once, the first time you initialize it. You can get around it using a pointer.

>

>

>

> It sounds to me you are trying to use a system variable like a common block (i.e. global variable). See if you can rethink how you do it without a system variable, because it will probably come back to bite you.

>

>

>

> Craig

Thanks Craig,

I have a program that I call in nearly every program I write, that defines about one hundred global variables, such as the date, time, working directory, default color table, etc., using the `defsyst` command. In any given program I'd use only a handful of these, but I wouldn't necessarily know which ones until the program was finished.

Based on David's books, it seems to me that using a common block in such a case would be a real pain.

Am I missing something as regards global variables?

Thanks,

Mark

Subject: Re: resizing system variable

Posted by [David Fanning](#) on Thu, 20 Dec 2012 23:14:47 GMT

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

Craig Markwardt writes:

> You can only set the dimensions of a system variable once, the first time you initialize it. You can get around it using a pointer.

I notice the documentation doesn't say you can create a system variable as a pointer. And, I can think of hundreds of ways you could get into trouble by **making** one a pointer! But, it does seem possible. :-)

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.")

Subject: Re: resizing system variable

Posted by [Craig Markwardt](#) on Fri, 21 Dec 2012 04:33:00 GMT

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

On Thursday, December 20, 2012 4:44:37 PM UTC-5, astrobo...@gmail.com wrote:

> On Thursday, December 20, 2012 3:28:45 PM UTC-5, Craig Markwardt wrote:

>> It sounds to me you are trying to use a system variable like a common block (i.e. global variable). See if you can rethink how you do it without a system variable, because it will probably come back to bite you.

...

> I have a program that I call in nearly every program I write, that defines about one hundred global variables, such as the date, time, working directory, default color table, etc., using the defsystv command. In any given program I'd use only a handful of these, but I wouldn't necessarily know which ones until the program was finished.

>

> Based on David's books, it seems to me that using a common block in such a case would be a real pain.

Actually, I think that David would argue that system variables are inappropriate for use as global variables for exactly the same reason as common blocks. Global variables can pin you down in painful ways when you try to scale/enhance your program.

[Don't tell David I mentioned this, but common blocks are actually more flexible than system variables because common block variables can be redefined - even redimensioned whenever you wish.]

You've got a system that works, I would stick with it now.

When designing a new system, I usually define a structure with all of those quantities and then pass that structure around.

[Actually, if I were designing my IDL library from scratch, I would start with Python. :-]

Craig

Subject: Re: resizing system variable

Posted by [David Fanning](#) on Fri, 21 Dec 2012 05:06:23 GMT

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

Craig Markwardt writes:

> Actually, I think that David would argue that system variables
> are inappropriate for use as global variables for exactly
> the same reason as common blocks. Global variables can
> pin you down in painful ways when you try to scale/enhance
> your program.
>
> [Don't tell David I mentioned this, but common blocks
> are actually more flexible than system variables because
> common block variables can be redefined - even redimensioned
> whenever you wish.]

You know I'm standing here, right?

> You've got a system that works, I would stick with it now.
>
> When designing a new system, I usually define a structure
> with all of those quantities and then pass that structure around.

I wish I knew my own mind as well as everyone else seems to know it. I'm pretty sure you could count the times I have used a common block on one hand. But, truthfully, I could say the same about defining my own system variable.

I suppose I would use them more if I could think of some good reason to do so. But, nothing seems to come to mind. ;-)

> [Actually, if I were designing my IDL library from scratch, I would start with Python. :-]

You guys really do make me smile, you know. I wish all of you a Happy Christmas and a Merry New Year!

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.")

Subject: Re: resizing system variable
Posted by [Craig Markwardt](#) on Fri, 21 Dec 2012 06:50:10 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Friday, December 21, 2012 12:06:23 AM UTC-5, David Fanning wrote:
> Craig Markwardt writes:

> I suppose I would use [common blocks] more if I could think of some
> good reason to do so. But, nothing seems to come to mind. ;-)

I use it for internal persistent state. A procedure needs to precompute some expensive function and then have it available for the next time it is called. Common block.

[And I'm a bad boy and use it within MPFITFUN, because there's nothing faster or more reliable.]

> You guys really do make me smile, you know. I wish all of you
> a Happy Christmas and a Merry New Year!

Same to you David :-)

May you find peace on earth, in whatever map projection it might be.

Craig

Subject: Re: resizing system variable
Posted by [Paul Van Delst\[1\]](#) on Fri, 21 Dec 2012 16:29:25 GMT
[View Forum Message](#) <> [Reply to Message](#)

On 12/21/12 01:50, Craig Markwardt wrote:
> On Friday, December 21, 2012 12:06:23 AM UTC-5, David Fanning wrote:
>> Craig Markwardt writes:
>
>> You guys really do make me smile, you know. I wish all of you
>> a Happy Christmas and a Merry New Year!
>
> Same to you David :-)
>

> May you find peace on earth, in whatever map projection it might be.

HA!

Craig! I'm wearing a new shirt and now it's got coffee all over it! :oD

Oh, and +1 on the Happy Merry Peace stuff too. To everyone.

cheers,

paulv
