Subject: Re: variables in common blocks
Posted by David Fanning on Fri, 04 May 2007 13:45:43 GMT
View Forum Message <> Reply to Message

Lasse Clausen writes:

> al long time ago, somebody asked the following question in this
> newsgroup but never got an answer. I was wondering if any of you knew:
>
>> Date: 1995/05/16
>> Subject: Common blocks revisited...
>>
>> 1) Is there a way to list only variables in a common block?
>>
>> 2) Is there a way to list the currently defined common blocks?

Having spent a couple of weeks trying to deconvolve a program
with Common blocks, I can understand why someone might
want something like this, but I'm afraid adding the capability
would only encourage this kind of nonsense. :-)

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: variables in common blocks
Posted by lasse on Fri, 04 May 2007 14:06:56 GMT
View Forum Message <> Reply to Message

On 4 May, 14:45, David Fanning <n...@dfanning.com> wrote:
> Lasse Clausen writes:
>> al long time ago, somebody asked the following question in this
>> newsgroup but never got an answer. I was wondering if any of you knew:
>
>>> Date: 1995/05/16
>>> Subject: Common blocks revisited...
>
>>> 1) Is there a way to list only variables in a common block?
>
>>> 2) Is there a way to list the currently defined common blocks?
>
> Having spent a couple of weeks trying to deconvolve a program

> with Common blocks, I can understand why someone might
> want something like this, but I'm afraid adding the capability
> would only encourage this kind of nonsense. :-)
>
> 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.")

I won't be drawn into a discussion whether COMMON blocks are a good
things or not.

And anyway, I found out that you can list all variables using

help, names='*', output=out

For common block variables the output string contains the name of the
common block in brackets, hence the functionality I would like to have
is easily achieved by parsing through the output.

Ha!

Cheers
Lasse

Subject: Re: variables in common blocks
Posted by David Fanning on Fri, 04 May 2007 14:11:53 GMT
View Forum Message <> Reply to Message

Lasse Clausen writes:

> I won't be drawn into a discussion whether COMMON blocks are a good
> things or not.

Shoot. I thought my Friday troll was less transparent than that. :-)

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: variables in common blocks
Posted by lasse on Fri, 04 May 2007 14:42:57 GMT
View Forum Message <> Reply to Message

On 4 May, 15:11, David Fanning <n...@dfanning.com> wrote:
> Lasse Clausen writes:
>> I won't be drawn into a discussion whether COMMON blocks are a good
>> things or not.
>
> Shoot. I thought my Friday troll was less transparent than that. :-)
>
> 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.

Ok. I'll describe what I am using common blocks for: Whatever day
during the week, I am doing one of the following:
1) read data
2) process data
3) plot data
Since there are many different ways to process or plot data, it is
nice to read the data into common blocks, so that when I decide to
process or plot the data differently, I do not have to read it in
again.

If you have an easy solution on how to avoid common blocks, shoot. And
why are common blocks bad? Is it the same reason "goto" statements are
bad?

Cheers
Lasse

Subject: Re: variables in common blocks
Posted by Maarten[1] on Fri, 04 May 2007 15:07:12 GMT
View Forum Message <> Reply to Message

On May 4, 2:42 pm, Lasse Clausen <l...@lbnc.de> wrote:

> If you have an easy solution on how to avoid common blocks, shoot. And
> why are common blocks bad? Is it the same reason "goto" statements are
> bad?

Common variables are really global variables. They allow a function to have side effects, and make it hard to debug a program. For sanity's sake try to avoid common blocks, unless you really have a good reason to use them, and clearly label the variables in them as such, so that you have a clear warning when dealing with them.

In general is it easier to pass variables as formal parameters.

The book I learnt C from started out with local variables, and only mentioned that global variables even existed in chapter 6. By that time I was acustomed to pass variables as function arguments, and hardly ever used globals, except perhaps as a global program kill switch in GUI applications. The Mac way of things infected me with aonther (pseudo)Hungarian notation: all my globals have a name starting with 'g', all constant start with 'k'. If you use a common block, make sure you do something similar.

Maarten

Subject: Re: variables in common blocks
Posted by David Fanning on Fri, 04 May 2007 15:16:10 GMT
View Forum Message <> Reply to Message

Lasse Clausen writes:

> Ok. I'll describe what I am using common blocks for: Whatever day
> during the week, I am doing one of the following:
> 1) read data
> 2) process data
> 3) plot data
> Since there are many different ways to process or plot data, it is
> nice to read the data into common blocks, so that when I decide to
> process or plot the data differently, I do not have to read it in
> again.
>
> If you have an easy solution on how to avoid common blocks, shoot. And
> why are common blocks bad? Is it the same reason "goto" statements are
> bad?

Oh, oh. *I* didn't want to get into it. I just wanted
to light the fire. :-(

OK, there is *nothing* inherently evil about either Common

blocks or GOTO statements. Both have a purpose, and I have used both (I think, although I can't remember a specific instance of either).

But what generally makes both Common blocks and GOTO statements pernicious is how easy it is for inexperienced programmers to substitute them for clear thinking and well-written programs. The program I have been working on for several weeks was brought to me because it is nearly impossible for two "expert" users to get the program to run the same way twice in a row, or--sometimes--to even run at all. It depends entirely on what order things are done and the current phase of the moon. Mostly, this is because of the indiscriminate use of Common blocks, and the fact that the left hand doesn't always know what the right hand is doing.

If I saw a COMMON block in a program written by JD Smith or Craig Markwardt, I probably wouldn't even think twice. But when I open an e-mail and there is someone pleading for "help" and I see Common blocks, I head straight for the liquor cabinet.

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: variables in common blocks
Posted by Foldy Lajos on Fri, 04 May 2007 15:36:44 GMT
View Forum Message <> Reply to Message

On Fri, 4 May 2007, Lasse Clausen wrote:

> why are common blocks bad?

They is an area where they are really useful: preserving the state of a user routine (like static local variables in C/C++):

pro mypro
common mypro_state, a, b, c

```
if size(a, /type) eq 0 then begin
   ; first call, initialize
endif

...

end
```

regards,
lajos

Subject: Re: variables in common blocks
Posted by lasse on Fri, 04 May 2007 15:56:50 GMT
View Forum Message <> Reply to Message

On 4 May, 16:16, David Fanning <n...@dfanning.com> wrote:
> Lasse Clausen writes:
>> Ok. I'll describe what I am using common blocks for: Whatever day
>> during the week, I am doing one of the following:
>> 1) read data
>> 2) process data
>> 3) plot data
>> Since there are many different ways to process or plot data, it is
>> nice to read the data into common blocks, so that when I decide to
>> process or plot the data differently, I do not have to read it in
>> again.
>
>> If you have an easy solution on how to avoid common blocks, shoot. And
>> why are common blocks bad? Is it the same reason "goto" statements are
>> bad?
>
> Oh, oh. *I* didn't want to get into it. I just wanted
> to light the fire. :-(
>
> OK, there is *nothing* inherently evil about either Common
> blocks or GOTO statements. Both have a purpose, and I have
> used both (I think, although I can't remember a specific
> instance of either).
>
> But what generally makes both Common blocks and GOTO
> statements pernicious is how easy it is for inexperienced
> programmers to substitute them for clear thinking and
> well-written programs. The program I have been working
> on for several weeks was brought to me because it is
> nearly impossible for two "expert" users to get the program

> to run the same way twice in a row, or--sometimes--to
> even run at all. It depends entirely on what order things
> are done and the current phase of the moon. Mostly, this
> is because of the indiscriminate use of Common blocks,
> and the fact that the left hand doesn't always know what the
> right hand is doing.
>
> If I saw a COMMON block in a program written by JD Smith
> or Craig Markwardt, I probably wouldn't even think twice.
> But when I open an e-mail and there is someone pleading for
> "help" and I see Common blocks, I head straight for the
> liquor cabinet.
>
> 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.")

Ok, agreed. It is not that I am not on your side, it is just that I
use common blocks for the above stated reasons and I cannot think of
any other way of doing it without reading in the data every time I
want to plot it in a different way.

Anyway, have a good weekend.

Cheers
Lasse

Subject: Re: variables in common blocks
Posted by David Fanning on Fri, 04 May 2007 16:03:35 GMT
View Forum Message <> Reply to Message

Lasse Clausen writes:

> Ok, agreed. It is not that I am not on your side, it is just that I
> use common blocks for the above stated reasons and I cannot think of
> any other way of doing it without reading in the data every time I
> want to plot it in a different way.

Uh, well, a data object comes to mind. :-)

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: variables in common blocks
Posted by Christopher Thom on Fri, 04 May 2007 16:40:45 GMT
View Forum Message <> Reply to Message

Quoth Lasse Clausen:

> Ok. I'll describe what I am using common blocks for: Whatever day
> during the week, I am doing one of the following:
> 1) read data
> 2) process data
> 3) plot data
> Since there are many different ways to process or plot data, it is
> nice to read the data into common blocks, so that when I decide to
> process or plot the data differently, I do not have to read it in
> again.
>
> If you have an easy solution on how to avoid common blocks, shoot. And
> why are common blocks bad? Is it the same reason "goto" statements are
> bad?

Well...I tend to store all my reduced data in IDL .sav files, then read them whenever I want to run a routine. So which is more evil...sav files or common blocks? *ducks* :-)

On the GOTO side of things, I often use them as a get-out-of-jail-free statement inside several layers of nesting, when I take user input. I put in a "quit now and I really mean it" option for myself, then jump to a cleanup section at the end, so I don't leave a mangled plotting window and file handles lying around.

cheers
chris

Subject: Re: variables in common blocks
Posted by David Fanning on Fri, 04 May 2007 16:48:31 GMT
View Forum Message <> Reply to Message

Christopher Thom writes:

> Well...I tend to store all my reduced data in IDL .sav files, then read
> them whenever I want to run a routine. So which is more evil...sav files
> or common blocks? *ducks* :-)

I'm dealing with save files in *combination* with common
blocks. This is truly a lethal combination unless you
can train yourself to NEVER CHANGE THE FRIGGIN DEFINITON
of the common block.

Oh, sorry...

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.")
