Subject: common blocks
Posted by craig on Tue, 06 Dec 1994 19:06:25 GMT
View Forum Message <> Reply to Message

Hello

A quick question with presumably a quick answer...

I am running PVwave 4.01 on a SPARC IPX.

Why can't I change a common block after first defining
it? The manual doesn't cover this as far as I can see.
Except to say that I can't change a common block.

Is there a reason why I can't or have I just not found
the correct command?

email replies if you could...
craig@arfa.civil.ubc.ca

thanks alot

Craig Stevens

Subject: Re: Common blocks
Posted by rivers on Tue, 02 May 1995 07:00:00 GMT
View Forum Message <> Reply to Message

In article <9505021847.AA14200@peace.med.ohio-state.edu>, phil@peace.med.ohio-state.edu (Phil) writes:
> I got a ? for y'all. I know how to use common blocks in procedures
> and functions but don't know hoe to do the following. Here's the
> scenario:
>
> 1) run program w/ a common block (it's a widget)
> 2) exit the widget
> 3) access the common block from the command line via
> IDL> common blockname
>
> Here's my ?. How do I get out of the common block and back to the
> 'main' variable space? I have tried retall, but that does not work.
> Any help would be appreciated.

You are not "in the common block". You ARE in the "main" variable space.
However, that variables in that common block are now "visible" in the main
program space, just like any other main program variable.

This can be useful in many circumstances. I don't think it is possible to make
that common block "invisible" again. It should not be a problem unless you try
to use a variable whose name is the same as one in the common block, which
might then have undesirable side-effects.

Mark Rivers (312) 702-2279 (office)
CARS (312) 702-9951 (secretary)
Univ. of Chicago (312) 702-5454 (FAX)
5640 S. Ellis Ave. (708) 922-0499 (home)
Chicago, IL 60637 rivers@cars3.uchicago.edu (Internet)

Subject: Re: Common blocks
Posted by phil on Wed, 03 May 1995 07:00:00 GMT
View Forum Message <> Reply to Message

In article <D7z1G2.5JI@midway.uchicago.edu> rivers@cars3.uchicago.edu (Mark Rivers) writes:

>>>>
>>>> Here's my ?. How do I get out of the common block and back to the
>>>> 'main' variable space? I have tried retall, but that does not work.
>>>> Any help would be appreciated.
>>>
>>> You are not "in the common block". You ARE in the "main" variable space.
>>> However, that variables in that common block are now "visible" in the main
>>> program space, just like any other main program variable.
>>>
>>> This can be useful in many circumstances. I don't think it is possible to make
>>> that common block "invisible" again. It should not be a problem unless you try
>>> to use a variable whose name is the same as one in the common block, which
>>> might then have undesirable side-effects.
>>>

Like trying to compile other functions/pros that use the said variables!

--

/** ****************/
Phil Williams
Postdoctoral Researcher "One man gathers what
MRI Facility another man spills..."
The Ohio State University -The Grateful Dead
email: phil@peace.med.ohio-state.edu
URL: http://justice.med.ohio-state.edu:1525
/** ****************/

Subject: Re: Common blocks
Posted by chase on Fri, 05 May 1995 07:00:00 GMT
View Forum Message <> Reply to Message

>>>> > "Phil" == Phil <phil@peace.med.ohio-state.edu> writes:
In article <PHIL.95May2201303@peace.med.ohio-state.edu> phil@peace.med.ohio-state.edu (Phil) writes:

Phil> In article <D7z1G2.5JI@midway.uchicago.edu> rivers@cars3.uchicago.edu (Mark Rivers) writes:

Phil> Like trying to compile other functions/pros that use the said variables!

A common block that is declared at the main level is not within the scope of compiled procedures or functions unless it is declared within those procedures or functions. If a common block is declared within a procedure/function then variable name conflict can be avoided simply by giving unique names for each variable in the common block definition for that routine. (A new IDL behavior for common blocks is that when no variable names are given the variables assume the names of those used in the first compiled procedure or function to define the common block).

In regards to one of the earlier posts, once a common block is declared at the main level I do not believe that it can be removed.

I find that common blocks are very handy for implementing a scope for global variables. In contrast, system variables are not limited in their global scope.

I have found many benefits by using a single variable in my common blocks. This variable is an anonymous structure. In the fields of this structure I put all the variables that would normally be placed individually in the common block. This has two advantages:

1) The structure can be redefined. For example, additional fields can be added to the structure. IDL does not let you redefine the number of elements in a common block within the life of the IDL session. To change the common block one has to restart IDL. Redefining the structure does not require restarting IDL. Additionally, I can add additional fields to the structure without affecting the older routines that access (but do not rewrite) previously defined fields in this structure. Sometimes this avoids having to edit alot of procedures.

2) This structure as a global variable helps to avoid namespace conflicts with other variables that might otherwise occur if the fields had instead been declared as individual variables in the

common block. This benefit is similar to C++ class static
variables and modules in other languages.

The disadvantage of this method is when storing very large variable arrays in the structure fields. IDL does not have the ability of taking the address of fields or subarrarys. This means that any field or subarray expression must copy the data. This may be undesirable in certain circumstances. Because of this, I was forced in one particular program to use an additional variable in the common block to hold an extremely large array.

In general, when global variables are needed I have found a single structure in a common block useful because of the above benefits.

Chris Chase

--
==========
Bldg 24-E188
The Applied Physics Laboratory
The Johns Hopkins University
Laurel, MD 20723-6099
(301)953-6000 x8529
chris.chase@jhuapl.edu

Subject: Re: Common blocks
Posted by Fergus Gallagher on Thu, 11 May 1995 07:00:00 GMT
View Forum Message <> Reply to Message

phil@peace.med.ohio-state.edu (Phil) wrote:
> I got a ? for y'all. I know how to use common blocks in procedures
> and functions but don't know hoe to do the following. Here's the
> scenario:
>
> 1) run program w/ a common block (it's a widget)
> 2) exit the widget
> 3) access the common block from the command line via
>   IDL> common blockname
>
> Here's my ?. How do I get out of the common block and back to the
> 'main' variable space? I have tried retall, but that does not work.
> Any help would be appreciated.
>

the command line above IS in the 'main' variable space - typing 'common ..' puts those variables in 'main'. You haven't lost any memory, since

those variables are there anyway, albeit invisible.

If you do need to get memory back, just set any (large) arrays in the common block to be smaller. This doesn't always work, but that's another story.....

Fergus

==========
| Fergus Gallagher |
| Remote Sensing Applications Development Unit |
| British National Space Centre |
| Monks Wood |
| Huntingdon PE17 2LS / UK |
| |
| F.Gallagher@nerc.ac.uk |
| http://uh.nmt.ac.uk/bnsc/fgg.html |
==========

Subject: Re: Common blocks
Posted by sjt on Fri, 12 May 1995 07:00:00 GMT
View Forum Message <> Reply to Message

Phil (phil@peace.med.ohio-state.edu) wrote:
: In article <D7z1G2.5JI@midway.uchicago.edu> rivers@cars3.uchicago.edu (Mark Rivers) writes:

<deletion>

: >>> This can be useful in many circumstances. I don't think it is possible to make
: >>> that common block "invisible" again. It should not be a problem unless you try
: >>> to use a variable whose name is the same as one in the common block, which
: >>> might then have undesirable side-effects.
: >>>

: Like trying to compile other functions/pros that use the said variables!

Not a problem - common blocks are only visible in the routines they're declared in (for this purpose MAIN is just anouther routine).

: --
: /********** **********/
: Phil Williams
: Postdoctoral Researcher
: MRI Facility
: The Ohio State University

"One man gathers what another man spills..."
-The Grateful Dead

: email: phil@peace.med.ohio-state.edu
: URL: http://justice.med.ohio-state.edu:1525
: /** ***************/

--
 +----------------------+--------------------------------- --+---------+
James Tappin,	School of Physics & Space Research	O__
sjt@star.sr.bham.ac.uk	University of Birmingham	-- \/`
Ph: 0121-414-6462. Fax: 0121-414-3722	
 +-- --+---------+

Subject: Re: common blocks
Posted by Alex Schuster on Wed, 14 Feb 2001 16:56:15 GMT
View Forum Message <> Reply to Message

Christophe Fraser wrote:

> Can I reset the common blocks in my program (so that I can change them)
> without closing and restarting IDL?

IDL >= 5.3 has the .reset_session command.

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

Subject: Re: common blocks
Posted by David Fanning on Wed, 09 Jun 2004 19:35:19 GMT
View Forum Message <> Reply to Message

lyubo writes:

> Is there a way to free the dynamic memory allocated to a Common Block at
> run-time?

It appears to me from a couple of simple tests that
variables in common blocks behave identically to
variables anywhere else in IDL. That is to say,
you aren't *ever* going to "free" dynamic memory.
But you can do intelligent things with variables
(e.g., UNDEFINE them) so that the memory you have
already allocated to the IDL process can be used
as efficiently as possible.

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
