Subject: Re: global variables in IDL
Posted by David Fanning on Tue, 25 Dec 2001 17:25:07 GMT
View Forum Message <> Reply to Message

Gert (gert.van.de.wouwer@NO_SPAMpandora.be) writes:

> I want to keep some global variables in an IDL program containing different
> widgets without having to pass the variables each time a new base widget is
> created. It can be done with common blocks, but then you can have only 1
> instance of the program running.
> I was thinking of creating a widget (called Globals) and copy the variables
> in a struct to its uvalue. If another widget then wants these variables, I
> need to get a widget identifier to Globals and retrieve the struct. But how
> can I get this widget identifier?

It's odd, isn't it, how good ideas keep being rediscovered
over and over again. :-)

It sounds like what you need is, uh, a pointer!

```
IDL> info = Ptr_New({all_your_stuff_in_here}, /No_Copy)
```

Then, when you create your widgets, you simply
pass the pointer to each program that needs it:

```
IDL> widget_program_1, info
IDL> widget_program_2, info
IDL> widget_program_3, info
```

The syntax of using pointers to structures is a little
weird until you get used to it:

```
TV, (*info).image, Order=(*info).order
```

Alternatively, you can create your pointer in the first
widget program, and pass it to the other programs that
are called from the first, etc. There are lots of ways
to do it, but all rely on a pointer of some sort.

Cheers,

David

--
David W. Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: global variables in IDL
Posted by Gert on Thu, 27 Dec 2001 01:02:48 GMT
View Forum Message <> Reply to Message

David,

thanks for the tip - I could off course do it this way, but then i need to
pass the pointer around to anyone who might need it. What i wanted to is

pro Widgetpro,

widgetID = getIDofGlobalswidget
WIDGET_CONTROL, widgetID , GET_UVALUE=sGlobals, /NO_COPY

print, sGlobal.threshold

...
end

is this possible? How can I do the 'getIDofGlobalswidget'?

Gert

"David Fanning" <david@dfanning.com> wrote in message
news:MPG.16926460c37f95ec9897bb@news.frii.com...
> Gert (gert.van.de.wouwer@NO_SPAMpandora.be) writes:
>
>> I want to keep some global variables in an IDL program containing
different
>> widgets without having to pass the variables each time a new base widget
is
>> created. It can be done with common blocks, but then you can have only 1
>> instance of the program running.
>> I was thinking of creating a widget (called Globals) and copy the
variables
>> in a struct to its uvalue. If another widget then wants these variables,
I
>> need to get a widget identifier to Globals and retrieve the struct. But
how
>> can I get this widget identifier?
>

> It's odd, isn't it, how good ideas keep being rediscovered
> over and over again. :-)
>
> It sounds like what you need is, uh, a pointer!
>
> IDL> info = Ptr_New({all_your_stuff_in_here}, /No_Copy)
>
> Then, when you create your widgets, you simply
> pass the pointer to each program that needs it:
>
> IDL> widget_program_1, info
> IDL> widget_program_2, info
> IDL> widget_program_3, info
>
> The syntax of using pointers to structures is a little
> weird until you get used to it:
>
> TV, (*info).image, Order=(*info).order
>
> Alternatively, you can create your pointer in the first
> widget program, and pass it to the other programs that
> are called from the first, etc. There are lots of ways
> to do it, but all rely on a pointer of some sort.
>
> Cheers,
>
> David
>
> --
> David W. Fanning, Ph.D.
> Fanning Software Consulting
> Phone: 970-221-0438, E-mail: david@dfanning.com
> Coyote's Guide to IDL Programming: http://www.dfanning.com/
> Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: global variables in IDL
Posted by Kristine Hensel on Thu, 27 Dec 2001 01:39:56 GMT
View Forum Message <> Reply to Message

Gert wrote:
>
> David,
>
> thanks for the tip - I could off course do it this way, but then i need to
> pass the pointer around to anyone who might need it. What i wanted to is
>
> pro Widgetpro,

>
> widgetID = getIDofGlobalswidget
> WIDGET_CONTROL, widgetID , GET_UVALUE=sGlobals, /NO_COPY
>
> print, sGlobal.threshold
>
> ...
> end
>
> is this possible? How can I do the 'getIDofGlobalswidget'?

If your global widget is registered using XMANAGER, i.e.
 xmanager, 'global widget', base_id_of_global_widget
then the name and ID are stored in the MANAGED common block, which is used by XREGISTERED.

To get the ID of the global widget, a calling procedure needs to use this common block:
 COMMON MANAGED, ids, names, outermodal

You can then look through the list of registered names to find the widget(s) that you want:

```
    validids = WHERE(ids ne 0, answer)
    if (answer ne 0) then begin
      registered = WHERE(names(validids) EQ 'global widget', answer)
      if (answer ne 0) then begin
        registered = validids(registered)
                           ; get the IDs registered under this
                           ; name:
        global_ids = ids[registered]
      endif                  ; name is registered?
    endif                    ; some valid IDs?
```

I use something like this for an error message widget. All (ahem) the error messages get stuck in the same widget, rather than popping up individual messages, and I use this code to figure out whether an error message widget has already been opened or whether a new one should be created.

Kristine

--
Kristine Hensel e-mail: kristine@esands.com
Environmental Systems & Services phone: +61-(0)3-9835-7901
20 Council St., Level 3 fax: +61-(0)3-9835-7900
Hawthorn East, VIC, Australia 3124

Subject: Re: global variables in IDL
Posted by Pavel A. Romashkin on Thu, 27 Dec 2001 05:02:30 GMT
View Forum Message <> Reply to Message

I agree with Kristine. Working around Commons writing a global variable
code, if you use Xmanager, is kind of silly becasue Xmanager uses Common
blocks in the first place. If you are a purist and don't use Xmanager to run
your widget code, then check out

http://spot.colorado.edu/~romashki/idl/single_set.pro
http://spot.colorado.edu/~romashki/idl/sobj_new.pro

Even if you dislike the idea itself, at least the syntax with these will be
pretty much exactly what you desire :)

Cheers,
Pavel

"Gert" <gert.van.de.wouwer@NO_SPAMpandora.be> wrote in message
news:YquW7.16901$Fe3.1736@afrodite.telenet-ops.be...
> David,
>
> thanks for the tip - I could off course do it this way, but then i need to
> pass the pointer around to anyone who might need it. What i wanted to is
>
> pro Widgetpro,
>
> widgetID = getIDofGlobalswidget
>

Subject: Re: global variables in IDL
Posted by John-David T. Smith on Fri, 28 Dec 2001 16:22:23 GMT
View Forum Message <> Reply to Message

Pavel Romashkin wrote:
>
> I agree with Kristine. Working around Commons writing a global variable
> code, if you use Xmanager, is kind of silly becasue Xmanager uses Common
> blocks in the first place.

This objection, though oft-repeated, is somewhat unfounded, I think.
The chief problem with common blocks for global state data is that only
a single such block exists. For example, if you store the widget id of
a button in:

```
common mywidget, mybutton
mybutton=widget_button(base,value='Foo')
```

Then `mybutton' will be the *only* button that can exist for your program in a single IDL session. You won't be able to invoke your program more than once at a time. Actually, you will, and that single button variable slot will get overwritten, and very bad things will happen.

XManager does use common blocks, but, by design, it uses them intelligently. Instead of overwriting common block variables, it maintains global lists of all the widgets it needs to manage, extending and pruning the lists as necessary. It's an example of common block usage which doesn't prevent multiple copies of a single widget program from running. (Here I gloss over the subtlety that widget programs run with /NO_BLOCK are actually handled by the command-line loop code, and not XManager at all.)

The fact that common blocks, in their basic usage, can cause this subtle problem doesn't imply that this problem is endemic to their use: XManager is an example of a more intelligent setup which capitalizes on, rather than suffers from, the single global instance status of common block variables.

JD

Subject: Re: global variables in IDL
Posted by Pavel A. Romashkin on Fri, 28 Dec 2001 16:39:45 GMT
View Forum Message <> Reply to Message

JD Smith wrote:

> This objection, though oft-repeated, is somewhat unfounded, I think.
> The chief problem with common blocks for global state data is that only
> a single such block exists. For example, if you store the widget id of
> a button in:
>
> common mywidget, mybutton
> mybutton=widget_button(base,value='Foo')
>
> Then `mybutton' will be the *only* button that can exist for your
> program in a single IDL session.

Oh geez. Did I ever advocate a primitive way of using Common blocks? All I was saying is, if you want to not have *any* common blocks, then you will have them if you use Xmanager.

> XManager does use common blocks, but, by design, it uses them
> intelligently. Instead of overwriting common block variables, it

> maintains global lists of all the widgets it needs to manage

Absolutely. This is why I am saying that if one needs any sort of global access in IDL, Common blocks are the way to go. I myself chose this way when I needed random global access (http://spot.colorado.edu/~romashki/idl/display.pro). But if you absolutely hate them, there are other ways.

As a matter of fact, I now think I found a way to achieve all the intelligencee of Xmanager style of Common usage but having nothing more than a Long integer in one Common variable. I will post the code when it is past the idea stage just to get input from the NG.

Cheers,
Pavel
