Subject: Re: accesing variables defined in a higher program level
Posted by David Fanning on Thu, 06 Feb 2003 16:52:36 GMT
View Forum Message <> Reply to Message

Isa Usman (eepisu@bath.ac.uk) writes:

```
>  I am trying to get a function to use variable values that are defined in a
>  higher level procedure. Since COMMON blocks can't be used in functions is
>  there a way round this? here is an example of what i am trying to do;
>
>  Pro high_level
>  x=1
>  y=1
>  z=1
>
>  I=integrate('f_x',z)
>  end
>
>  function integrate,z
>  ........
>  ........
>  call function f_x
>  i=........
>  return,i
>  end
>
>  function f_x, z
>      w=x+y+z
>  return, w
>  end
```

What makes you think COMMON blocks can't be used in functions?
They can be used in functions, procedures, main-level programs,
and--in fact--anywhere in IDL.

Rearranging your code the way it will eventually be in a file
(if you want it to work, that is), why not something like this:

```
function f_x, z
COMMON data, x, y
   w=x+y+z
return, w
end

function integrate, f_x, z
value = call_function(f_x, z)
print, value
return,value
```

```
end


Pro high_level
COMMON data, x, y
x=1
y=1
z=1
I=integrate('f_x', z)
end


--
David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
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: accesing variables defined in a higher program level
Posted by R.Bauer on Thu, 06 Feb 2003 16:53:37 GMT
View Forum Message <> Reply to Message

```
Isa Usman wrote:
>  Hello,
>
>  I am trying to get a function to use variable values that are defined in a
>  higher level procedure. Since COMMON blocks can't be used in functions is
>  there a way round this? here is an example of what i am trying to do;
>
>  Pro high_level
>  x=1
>  y=1
>  z=1
>
>  I=integrate('f_x',z)
>  end
>
>  function integrate,z
>  ........
>  ........
>  call function f_x
>  i=........
>  return,i
>  end
>
>  function f_x, z
```

>   w=x+y+z
> return, w
> end
>
> Many thanks
>
> Isa
>
>
>

Dear Isa

I don't like common blocks but I am a bit surprised that they should not
work in functions. I can't believe this.

What is the reason for using all these call functions ?

Why did you not define

```
function f_x, x,y,z
      w=x+y+z
  return, w
```

and call this directly by high_level like

```
  Pro high_level
   x=1
   y=1
   z=1

   I=f_x(x,y,z)
  end
```

regards

Reimar

--
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: accesing variables defined in a higher program level
Posted by David Fanning on Thu, 06 Feb 2003 16:55:55 GMT
View Forum Message <> Reply to Message

David Fanning (david@dfanning.com) writes:

> What makes you think COMMON blocks can't be used in functions?
> They can be used in functions, procedures, main-level programs,
> and--in fact--anywhere in IDL.

Whoops, that one got away from me before I was ready!

But I guess the essential ideas were there. :-)

Cheers,

David

--
David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
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: accesing variables defined in a higher program level
Posted by Craig Markwardt on Thu, 06 Feb 2003 17:57:57 GMT
View Forum Message <> Reply to Message

"Isa Usman" <eepisu@bath.ac.uk> writes:

> Hello,
>
> I am trying to get a function to use variable values that are defined in a
> higher level procedure. Since COMMON blocks can't be used in functions is
> there a way round this? here is an example of what i am trying to do;

As other readers have pointed out, you can use common blocks in
functions. But, that's still not the greatest programming practice.
For example, what if you end up needing a double integral? Common
blocks will not survive the nested call.

I have a function integrator on my web page named QPINT1D that allows you to pass arbitrary data to the function via two methods. The first method is to pass data in a structure. Obviously you can arrange the structure any way you like. The second way is to pass the data using keywords. This also involves passing in a structure, and then QPINT1D uses the _EXTRA mechanism, which causes the named structure elements to be converted to named keywords in the function call.

Good luck,
Craig

--
---------- ----------
Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
---------- ----------