Subject: Re: Static Variables in IDL
Posted by rivers on Wed, 13 Mar 1996 08:00:00 GMT
View Forum Message <> Reply to Message

In article <4i7ara$606@vixen.cso.uiuc.edu>, santanu@ux7.cso.uiuc.edu (Santanu Bhattacharyya) writes:
> Hi,
> Is there a way to statically allocate variables in IDL ? I would
> like to define a static variable in a .pro which is repeatedly called
> from an upper level. I am specifically looking for the IDL counterpart
> of
> static int testint;
>

Yes, it is easy. Use a common block. Make the name of the common block unique enough that you can be sure it won't conflict with common blocks used in other procedures.

```
common unique_name, testint
testint = 14
```

testint will have the value 14 the next time the .pro file is called.

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: Static Variables in IDL
Posted by steinhh on Thu, 14 Mar 1996 08:00:00 GMT
View Forum Message <> Reply to Message

In article <4i7ara$606@vixen.cso.uiuc.edu>, santanu@ux7.cso.uiuc.edu (Santanu Bhattacharyya) writes:
|>
|> Hi,
|> Is there a way to statically allocate variables in IDL ? I would
|> like to define a static variable in a .pro which is repeatedly called
|> from an upper level. I am specifically looking for the IDL counterpart
|> of
|> static int testint;
|>
|> I would appreciate any help in this regard,

The IDL counterpart is to put the variable in a common block:

```
pro test
common test_static,testint

  if N_elements(testint) eq 0 then testint = <init_value>

  ;; Use your testint variable for anything.
  ;; It keeps it's value between each call
end
```

Stein Vidar

Subject: Re: Static Variables in IDL
Posted by David Foster on Thu, 14 Mar 1996 08:00:00 GMT
View Forum Message <> Reply to Message

santanu@ux7.cso.uiuc.edu (Santanu Bhattacharyya) wrote:
>
> Hi,
> Is there a way to statically allocate variables in IDL ? I would
> like to define a static variable in a .pro which is repeatedly called
> from an upper level.

As much as I hate to use COMMON blocks unless I absolutely have to,
this would be one way of doing this. Another way would be to use
handles (HANDLE_VALUE(), HANDLE_CREATE(), etc.).

"Static" would be a great new reserved word for the next IDL version.

Dave Foster
UCSD Brain Image Analysis Lab
foster@bial1.ucsd.edu

Subject: Re: Static Variables in IDL
Posted by steinhh on Thu, 14 Mar 1996 08:00:00 GMT
View Forum Message <> Reply to Message

In article <4i8geg$5fv@vixen.cso.uiuc.edu>, santanu@eehpx22.cen.uiuc.edu (S Bhattacharyya)
writes:
|>
|>
|> Pardon me if I appear a bit obtuse, but I am still a little confused. I am
|> under the impression that the common block declaration is equivalent to
|> C's global declaration. What I would like to have is a bit of non re-entrant

```
|> code in a standalone function (.pro). I want an IDL .pro that does the
|> following::
|>
|> main()
|> {
|>  for(;;) non_rEntrant();
|> }
|> non_rEntrant()
|> {
|>  static int block=1;
|>
|>  if (block == 1){
|>    puts("This is executed only once");
|>    block=0;
|>        }
|>  puts("And this is done over and over again");
|> }

pro main
  while 1 do non_rEntrant
end

pro non_rEntrant
common non_rEntrant_private_others_keep_off,block

  if N_elements(block) eq 0 then block=1   ;; Instead of "static .... =1"

  if block eq 1 then begin
     print,"This is executed only once"
     block = 0
  end
  print,"And this is done over and over again"
end
```

This will do what you want. Common blocks are not quite like
global declarations. I can have e.g., a variable called block
in the main program without having a conflict. I can even
have a variable called block at the interactive prompt level,
and still have no conflict.

```
|>  call_procedure,'non_rEntrant'
```

You'll get speedier programs by using just

```
   non_rEntrant
```

as in the example above 8-)