Reimar Bauer wrote:

> Martin Downing wrote:
>>
>> First comment is to only use global variables sparingly, sometimes they are
>> appropriate but try not to rely on them to make programming simpler.
>> I know of three ways, the SYSTEM variables are fully global, and may be user
>> defined, COMMON BLOCK variables are shared between those routines which
>> include a
>> declaration for the common block, these are very suitable for those programs
>> where you cant find a way of writing your code without creating global
>> variables, but do not want the variables to be seen by any old routine.
>>
>> system variable eg:
>> ------------
>> IDL>filepaths = {tag:"FILEPATHS", bin:bin)
>> IDL>defsysv, '!fpaths',filepaths ; a variable for local paths
>>
>> now you can use this variable from anywhere in your code as:
>>
>> IDL> print, !fpaths.bin
>> d:\martin\bin\
>> ------------
>>
>> For common block variables you need a block identifier then a list of
>> variables in the block. Note that the order is all that is relevent and that
>> the first declaration defines the size of the block: eg
>>
>> pro foo
>>   COMMON FOO_COMMON_VARIABLES, a,b,c,d
>>   a = "first"
>>   b="second"
>>   c="third"
>>   d="fourth"
>>   print_foo
>> end
>>
>> pro print_foo
>>   ; note later calls do not have to decare all block variables
>>   COMMON FOO_COMMON_VARIABLES, d,c,var3
>>   print, d
>>   print, c,
>>   print, var3
>> end

```
>>
>> the result of calling this procedure is demonstrates that there is no
>> association between variable names inthe common block just the variable
>> order.
>> IDL> foo
>> first
>> second
>> third
>>
>> ----------------------------
>> Lastly there is a *really* nice way of passing around all the 'global'
>> variables you need in your widget programs using get/set_uvalue which. This
>> is my favourite tip learnt from David's books so I shall leave him to
>> describe. In essence you call:
>> WIDGET_CONTROL, base, SET_UVALUE=info_structure
>> WIDGET_CONTROL, event.top, GET_UVALUE=info_structure
>>
>> cheers
>>
>
> The fourth way is to use Pointers
>
> value=10
> ptr=ptr_new(value)
>
> help, *ptr
>
> Reimar
>
> --
> Reimar Bauer
>
> Institut fuer Stratosphaerische Chemie (ICG-1)
> Forschungszentrum Juelich
> email: R.Bauer@fz-juelich.de
> http://www.fz-juelich.de/icg/icg1/
> =========================================================== ======
> a IDL library at ForschungsZentrum Juelich
>  http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_lib_intro.h tml
>
> http://www.fz-juelich.de/zb/text/publikation/juel3786.html
> =========================================================== ======
>
> read something about linux / windows
> http://www.suse.de/de/news/hotnews/MS.html

Ok, thanx.
I prefer the de defsysv method.
```

```
--
+- - - - - - - - - - - - - - - - - - - - - -+
| Miguel Angel Cï¿½rdoba    cordoba@ehma.upc.es |
|                               |
| http://campus.uab.es/~2034008  |
|                               |
| Grup de Modelitzaciï¿½ Hidrometeorolï¿½gica (UPC) |
| (http://www.upc.es/ehma/gmh)         |
+- - - - - - - - - - - - - - - - - - - - - -+
```