Subject: Discussion on global variables in IDL
Posted by alt on Wed, 01 Aug 2001 05:34:55 GMT
View Forum Message <> Reply to Message

Hello,

I'd like to discuss global variable management in IDL. I've read a pair of NG threads on that theme, and would like to add some consideration and share experience. Some time ago I've sent a letter to RSI with some proposals, but since then two version have been released and I don't see any movement in the direction of improving global vars management. It seems that everyone is satisfied. Is it really so?
First of all, I like IDL. And the most remarkable IDL feature, IMHO, is an opportunity to develop program at the same time you think. You have an idea, you type pair of strings, run it, look how it works, debug it, correct it and so on. If you need variable, you define it at once. I don't need declare it somewhere, define its type, etc. All that you have to point is variable name. Its type, structure can be changed during run flow. It can become even undefined. It's okay. I like it. Sometimes it leads to errors, that couldn't happen in C for example, but everyone choose his own way.
And it's okay while you are dealing with a pair of procedures without widgets. But when you create big widget project you start to have problems with global data. The only way that IDL offers is COMMON block. One can argue that's good enough. May be. Tastes differ. But one thing may be said exactly. COMMON block doesn't allow having many instances of same widget. First time I ran into it I was surprised that I have to put my global data in structure, then put it into some button user value, and get it from there on event. I disliked this method strongly, so I've written several very simple procedures that help me to manage global data. I have been using last version of them for about a year and a half and I'm satisfied. Not always. Sometimes I use COMMONs. It depends on situations, but at least I have a choice. May be this ideas will be useful for someone else.

So, everything is based on pointers. We have main global data pointer p that points on all global data heap.
p = ptr_new({ parr:ptrarr(Q), namearr:strarr(Q) })
Parr points on global variables, namearr contains global var names. This p I pass as parameter to every procedure where I need global data.

So, how it looks for the user. Very simple in my opinion.

ini, p ; global data initialization
sav, p, var1, 'var1_global_name' ; save var1 to global heap as var1_global_name

var, p, var1, 'var1_global_name' ; restore var1_global_name from
global heap as var1
del, p ; clean up heap

You can save and restore several variables at once.
sav, p, var1, 'var1', var2, 'var2', var3, 'var3'
var, p, var1, 'var1', var2, 'var2'

You can restore var as pointer to avoid copy huge arrays copying.
var, p, var1, '*var1'
(*var1)[100,200] = 20
The vars to be saved can be undefined.

So the widget application looks like that.

```
pro main
 ini, p
 sav, p, var1, 'var1'
 ....
 baseID = widget_base( uvalue = p )
 ....
 sav, p, var2, 'var2'
 ....
 xmanager, 'main', baseID, cleanup = 'main_clean'
end
pro main_event, ev
 widget_control, ev.top, get_uvalue = p
 ....
 proc1, p, par1, par2
 proc2, p, par3, par4
end
pro proc1, p, par1, par2
 ...
 var, p, var1, 'var1'
 ...
 sav, p, var1, 'var1'
end
pro proc2, p0, par1, par2
 ini, p ; initialization of new global data heap for another widget,
for example
 ....
end
pro main_cleanup, id
 ....
 widget_control, id, get_uvalue = p
 del, p
 ....
end
```

This technique main advantages are:
1) We create, save and restore global variables just where we need them
2) We can have multiple instances of the same widget
Disadvantages:
1) We can't restore a group of variables at once as in case COMMON block.
2) It is more time consuming. So it is better not to use this method in time critical parts.

Implementing this technique in IDL kernel can solve these disadvantages. So we could get FLOATING COMMON block with pointer on it.

If someone is interested in those programs you can e-mail me or I can put them as is on our server.

Thank you for your attention.

Altyntsev Dmitriy
Remote Sensing Center of ISTP, Irkutsk
alt@iszf.irk.ru
