Subject: Re: Global variables and IDL
Posted by Pavel Romashkin on Wed, 21 Apr 1999 07:00:00 GMT
View Forum Message <> Reply to Message

Hi Martin,

> In my oppinion it would be nice to be able to explicitely declare
> variables as global, such as
> GLOBAL glbarray=findgen(20)

I personally see no use for GLOBAL variables. I have been (and to some degree still am) a user of IGOR Pro for a long time and our applications relied heavily on the use of global variables. This led to lots of undesired results when the time had come to update applications: it was impossible to track down what GLOBALs were meant for, and required creating more GLOBALs to use old code. GLOBALs look to me like an inferior way of defining the required parameters, because then I lose the responsibility for "clean" programming.

> This would also help the problem discussed some months ago of how to
> retrieve information
> from widgets that are/were running in NO_BLOCK mode (e.g. data that has
> been manipulated).

I am sorry, I missed that, but why would you like to retrieve data from a widget into an interactive session before you are done with the widget?! II always save the data from the widget to a binary file and then restore it, if I want to manually play with it.

> and the program that wants to use that data can as always test for
> n_elements() gt 0.

Data can easily be requested by other programs from the widget's STATE that you keep around anyway in order to be able to act on the data from the event handling procedure.

> Also,
> if there are several instances of the widget application, the latest
> (current) application
> could overwrite the global variable by redefining it

That, in my opinion, is exactly where a lot of confusion will start from! How are you going to know which instance has just updated the GLOBAL? You will have to keep the track of it yourself.

> For me, a computer program is somewhat like a house: you have certain
> fixed structures
> like the doors and windows (which you can change but with some effort)

> and you have furniture
> that you can move around as you wish.

This sounds OK to me, but the way I found to be the best is to make "walls and doors" from the major DATA field in the widget STATE structure, and "furniture" was made of local variables inside daughter functions that branch from the main widget application. This way I don't need to worry that my GLOBAL "walls" can get altered by another process and my main program will not like that. Lack of the way to define session-global variables seems to be a good motivation to write self-consistent, independent and very flexible applications.