

---

Subject: Re: passing parameters from base to base  
Posted by [Pavel A. Romashkin](#) on Tue, 03 Dec 2002 21:19:07 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

JD Smith wrote:

< Snip-snip...

> Suppose I  
> have 5 instances of a viewer running. Each little plug-in tool of the  
> viewer is a separate object which needs to contact a 'tvDraw' object  
> to get draw events. In your system, how does it go about doing that?  
> Well, if each tvDraw object had a unique ID, that would work. But how  
> do you distribute all those unique ID's?... and you're back in the same  
> boat again.

I agree with Stein Vidar, and I felt that this thread will inevitably  
end up as a singleton discussion. This is known to have happened before :-)

In fact, most of my own chots at it were exactly using the singleton  
systems of various sorts. The cornerstone of singletons is in my opinion  
that there is only ONE official way in IDL to have anything at all  
visible at the global level - Common blocks. Other than that? Well,  
Functions and Procedures are also visible globally, and I used them too  
to avoid Common blocks (one example is at  
[http://www.ainaco.com/idl/idl\\_library/sobj\\_new.pro](http://www.ainaco.com/idl/idl_library/sobj_new.pro)). However, I'd say  
that a Common is just as ugly as that :-)

Anyway, having zeroed in on the central message manager (a Singleton  
object, or, for that matter, even a pointer - one example at  
[http://www.ainaco.com/idl/idl\\_library/single\\_set.pro](http://www.ainaco.com/idl/idl_library/single_set.pro)) I think the rest,  
particularly what JD is asking, is not all that difficult. For what I  
tried, I didn't care really about how the messages are passed: they just  
had a destination, and the message manager simply passed them along. If  
the destination didn't exist, well - too bad. Message just went  
unnoticed, to the dismay of the user :-)

More seriously, for my purposes it was sufficient to indicate the  
recepient of the message by the mouse click - the receving graphics  
window, per say. From then on, the output from the several data  
processing widget programs was directed to that target. Then, I wanted  
to compare that picture with another - just clicked on another target,  
and exchange data with it from then on. And if there was no valid  
viewer, a new one was created.

Not extremely flexible but it works.

And, again, one example of self-managing, self aware command line system  
using Common blocks is my little Display program, which is the key  
building block of all of my recent 2D data plotters. It has no messaging  
as such (its much too simple for needing it, as it has no modules like  
David's recent system) in it but handles the directing of data input by itself.

Cheers,

