

---

Subject: Re: Global variables and IDL

Posted by [rmlongfield](#) on Fri, 23 Apr 1999 07:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Hi all,

Whoah, this global variable stuff is getting heavy. I know that deep down in my soul, objects is the way to go. Objects seem to be similar to structures. This means that many different things associated with it can be lumped together in one group and then referenced only by passing around the name of the group and not what is in it.

Stein wrote:

> Again - singleton objects will do this for you. A singleton class  
> instance is, in fact, a kind of global variable. You can access  
> it from anywhere, simply by asking for an object of that  
> class. Since only one object of that kind exists, it must be the  
> same object that the other programs are talking to. If this  
> object is keeping track of your data/preferences/whatever, then  
> \*you\* can ask the object to return a data pointer, and your  
> widget program can ask for the same data pointer. Though you're  
> stuck with pointer notation....unless you want to use the common  
> block cache approach.

Are you telling us that we don't even have to know what the variable name is? We don't have to pass around ANY information? We just have to type in some routine that says, "See if anything is in this box " Aha, but we still have to know how box would be defined (class?), so this definition of box must be passed around. If we manage to do this correctly, and if it finds something, then can we assume that it is what we are looking for?

For example, many times I have used WIDGET\_CONTROL to get back a pointer and have found out that the wrong value is returned because I used the wrong widget ID. (Lost track of whether TOP or ID was holding the information I wanted). Can this still happen with OBJECTS or is it more "fool proof"? Another problem with WIDGET\_CONTROL (and I have seen others here with a similar problem when starting) is that the variable in the statement:

```
WIDGET_CONTROL,event.id,GET_UVALUE= variable
```

Can have ANY name. One can call it variablePTR or variableString or whatever. The name is completely irrelevant if what has been saved in this UVALUE is an integer.

Can objects solve this confusion? I do not know.  
But I am reading all this with great interest and appreciate the discussion.

Rose

