Subject: IDL Object for creating a Singleton
Posted by Phillip &amp; Suzanne on Mon, 22 Jun 1998 07:00:00 GMT
View Forum Message <> Reply to Message

I've been doing a fair amount of reading on OO design and methodology. One of the books I've had pointed out to me was 'Design Patterns', which deals with OO algorithm design quite nicely. One of the patterns the authors describe is a Singleton -- A class for which only one instance exists. If the sole instance exists, the constructor function (object's init function in IDL) will return a reference to the existing object. If none exists, a new object is created and returned.

This structure is particularly useful when there is no reason to have multiple objects around. I am trying to create a user preferences object that fits this criteria very nicely. I'd like to be able to store user preferences for screen size in an object. If that object exists, other screens can size themselves appropriately. The book gives sample C++ code that relies on class variables and class methods (i.e., not associated with a specific instance of the object, but with the object itself). IDL doesn't seem to have any way to do something similar. Does anyone know of a way to do this?

My first attempt uses a common block called preferencesclassvars to store an object reference. This common block is called from a function that, while not part of the object, is stored in the same file. If the reference contained in the common block is not an object reference, then I know the object hasn't been initialized, and I set it to a new instance of the object. Otherwise, I return the object reference from the common block.

Is there a better way to achieve the techniques of class variables and class methods?

Thanks for your thoughts.

Phillip David
IDL Tool Developer
XonTech, Inc.