Subject: Re: keyword inheritance and object inheritance
Posted by Ted Cary on Sat, 23 Mar 2002 17:44:06 GMT
View Forum Message <> Reply to Message

Hi Ben,

You should use _REF_EXTRA when you have output keywords. It should therefore not be necessary in an object's SETPROPERTY method, but is almost indispensable in any subclass object's GETPROPERTY method.

Output keywords are possible exactly because IDL functions are normally passed keyword arguments by reference, so that changing the value of a keyword argument inside a function will change the value of the argument outside the function as well. When keywords are stored and passed along in the _EXTRA structures created in function declarations, however, they are stored in fields by value and the reference information is lost. This is fine if you only want to USE the value, but not if you want to CHANGE the value stored in the argument. In order to change the value in some other function, _REF_EXTRA allows you to pass along the output keywords by reference.

Use _REF_EXTRA instead of _EXTRA in GETPROPERTY function declarations of subclass 'child' objects (Mother/Father, MyObj/YourObj in your example) if you want to override the GETPROPERTY methods of their superclass 'parent' objects. In the call to a superclass (parent's) GETPROPERTY, pass the structure stored by _REF_EXTRA via the _EXTRA keyword. Your MOTHER::GETPROPERTY is written correctly.

To correct your code rewrite all subclass GETPROPERTY methods so that they are like your MOTHER::GETPROPERTY. You can remove the _REF_EXTRAs from all the SETPROPERTYs.

Another option which avoids function overriding and _REF_EXTRAs is to use the EXTRACT function on David Fanning's website. It takes advantage of the fact that self fields of subclass objects inherit the fields of their superclasses. This will work if your self fields ARE the properties you want to extract, as in your example. For more complicated objects, however, GETPROPERTY is better. HTH