Subject: Re: how does /no_copy work???
Posted by davidf on Wed, 02 Jun 1999 07:00:00 GMT
View Forum Message <> Reply to Message

D. Mattes (dmattes@u.washington.edu) writes:

> i'm writing my own class methods, and i have a question regarding my
> SetPropery and GetProperty methods specifically. i want to implement the
> /no_copy keyword found in many of idl's functions. how is this
> implemented? it seems like the method should return a pointer to the
> variable requested, but idl seems able to get around that somehow, because
> no pointer-dereferencing is required to use that variable.

Uh, well, that's because what you are passing into and out of
procedures and functions by arguments and keywords *IS* a pointer.
That is to say, a variable in IDL is, among other things, a
C pointer. This is what is passed into a method like SetProperty:

myobject->SetProperty, Data=thisData

The variable thisData is, essentially, the pointer to the data.
We say the data is "passed by reference", meaning that what the
procedure received was the actual physical address of the data
in memory (I.e. the pointer to the data). If the data is copied
before it is passed into the procedure, we say it is "passed by
value". For example, to pass this data by value we could create \an
expression. Expressions are passed by value:

myobject->SetProperty, Data=thisData * 1

(An IDL variable is actually a structure that contains information
about the size and type of data, etc. as well as the actual C
pointer to the memory location of the data. So it is a little more
complicated than saying a variable is a pointer.)

Where No_Copy is useful is when you are transferring some
information from one memory location to another. For example,
from a local variable in an event handler to the user value
of the top-level base, or from a local variable to an IDL
pointer (heap variable). These operations actually copy the
data to another memory location, unless you tell IDL not
to with the NO_COPY keyword. Then all IDL transfers is the
C pointer to the data that already exists in memory.

I've never had occasion to need or use NO_COPY, except
where they are already supplied by IDL. Pass variables, or
pass pointers to variables, and you will be fine.

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155
