Subject: Re: how does /no_copy work???
Posted by Peter Mason on Thu, 03 Jun 1999 07:00:00 GMT
View Forum Message <> Reply to Message

davidf@dfanning.com (David Fanning) wrote:
<...>
> 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.

Further to what David has written, there is a way to capture the
"spirit" of NO_COPY, in general - wherever there's some kind of
assignment going on. Use the TEMPORARY() function. e.g., If you do
A=B then A is set up with a copy of B's stuff (B is left intact). If
you do A=TEMPORARY(B) then B's stuff is essentially "switched over" to A
(B is deleted).
This technique is only worthwhile in cases where the amount of data
concerned is *large* (e.g., large arrays), or in cases where the amount
of data is not insignificant and the operation is done very frequently.

Peter Mason

Sent via Deja.com http://www.deja.com/
Share what you know. Learn what you don't.