Subject: Re: yet another idl memory question
Posted by ben.bighair on Mon, 15 Jun 2009 19:07:46 GMT
View Forum Message <> Reply to Message

On Jun 8, 3:57 pm, Paolo <pgri...@gmail.com> wrote:
> On Jun 8, 3:05 pm, David Fanning <n...@dfanning.com> wrote:
>
>> Paolo writes:
>>>> Of course, data=3D0B doesn't free *all* the memory,
>>>> and doing this many times leads, I suspect, to the memory
>>>> fragmentation that is the heart of the problem. I suggest
>>>> you use UNDEFINE. That really does release *all* the memory
>>>> associated with a variable.
>
>>> Really? Certainly it's not a substitute for ptr_free, is it?
>>> At least not in my system:
>
>> Well, I think you are confusing "variable", which is what
>> I claim, with "pointer to a variable", which I admit UNDEFINE
>> doesn't free. (I think it was written *before* pointers, to
>> tell you the truth!)
>
> Well, for me pointers are just another kind of variables :)
>
> In fact, I don't think IDL has anything like classic
> "pointers to variables": pointers are just references to
> data in memory (similar to regular variabes), because a
> command such as ptr_new(A) just duplicate the contents of A
> to a new memory location, so there is no such a thing as a
> true pointer to variable A, right?
>

Hi,

I don't know about the technical deep down stuff, but don't forget
that PTR_NEW accepts the /NO_COPY keyword. From the online help... "If
the NO_COPY keyword is set, the value data is taken away from the
InitExpr variable and attached directly to the heap variable. This
feature can be used to move data very efficiently. However, it has the
side effect of causing the InitExpr variable to become undefined."

Cheers,
Ben