Subject: Re: Structure field concatenation
Posted by John-David T. Smith on Wed, 06 Sep 2000 20:34:41 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
>
> Martin Schultz (martin.schultz@dkrz.de) writes:
>
>> The only reason to do this that I could accept without further
>> quirking is if you tell me there is a lot of penalty if you manually
>> deallocate and reallocate the memory instead of letting IDL do it.
>> Haven't tested, but I would doubt that it makes a big difference.
>
> I really don't think there is any difference at all.
> If it makes you feel safer, by all means free pointers
> yourself. I just wanted to make the point once again that
> IDL really does have some nice features. This aspect
> of pointer memory management is one of them. :-)

I just wanted to point out that this technically isn't pointer memory management at all. Rather, it's simply the same old variable memory management we know and love:

```
IDL> a=[1,2,3]
IDL> a=1
```

...no memory loss there! The only difference is that *heap* variables are being handled in the pointer case. So Martin, if you're happy with this, you should be happy with David's method. Of course you might always use "delvar,a", but somehow I doubt it.

I'm being pedantic only to prevent readers (especially the Java-enabled among them) getting confused about what kind of memory management IDL really provides. The best way to stay clear on the issue is to think about pointers as what they are: references to IDL variables which are exactly the same as any other variable except for being hidden ("on the heap") and persistent ("not cleaned up by function/procedure exits"). They do *not* point directly to memory (just as variables like "a" above don't directly map to memory -- thankfully for us).

JD

--
J.D. Smith /*\ WORK: (607) 255-6263
Cornell University Dept. of Astronomy */ (607) 255-5842
304 Space Sciences Bldg. /*\ FAX: (607) 255-5875
Ithaca, NY 14853 */