Subject: Re: Structures
Posted by Michael Wallace on Wed, 02 Mar 2005 00:40:31 GMT
View Forum Message <> Reply to Message

Marshall Perrin wrote:
> Michael Wallace <mwallace.no.spam@no.spam.swri.edu.invalid> wrote:
>
>> I learned all this the hard way by trying to figure out why I kept
>> getting errors when I'd attempt to use a different array size for some
>> of my structure variables. I finally ran across this in the IDL
>> documentation, but there wasn't any actual reason given for such
>> draconian policy, especially when compared to the rest of the language.
>
>
> I *think* it's something along these lines: We want to be able to
> create arrays of structs, which (from a language designer's
> viewpoint!) is much, much easier when all the structs are the same
> size. If you let array sizes inside of structs vary, then you might
> have foo1.somearray with 10 elements, and foo2.somearray with 100.
> Then when the user does bar = [foo1,foo2], suddenly you're dealing
> with an array of heterogenous data types and memory access becomes a
> whole lot uglier and more inefficient. At that point, it's probably
> easiest just to switch to Perl. ;-)
>

I can understand their decision if it had to do with memory access.
With static sizes you can easily index into an array of structures
rather that having to dynamically determine where the data is in memory.
Actually it makes sense considering that IDL is array based.

> You can also create named structures containing pointers to arrays, too.
> This very problem is what finally pushed me over the potential barrier
> into learning how to do pointers in IDL!

Seems like I'm going down the same road you went. That little thought
about needing to use pointers in my structures has been trundling
through my mind the last couple days. :-)

-Mike