Subject: Re: Pointer syntax and IDL 4.0
Posted by John-David T. Smith on Wed, 19 Dec 2001 21:59:43 GMT
View Forum Message <> Reply to Message

```
tam wrote:
>
> Recently I upgraded the a couple of routines to use pointers
> in certain special cases.  However the pointer dereference operator
> is illegal prior to version 5.0, so the code fails to compile on
> older versions of IDL -- though the great majority of the code is still
> useful there. It would be nice to be able to use a single
> version of code to support all users, so I'm asking the question:
> Is there any way of addressing this, i.e., dereferencing a pointer
> in a way that will not cause a syntax error for earlier versions of IDL?
>
> An obvious solution would be if there were a dereferencing function
> as well as an operator...
>
>     x = ptr_val(some_pointer)
>
> would be the same as
>
>     x = *some_pointer
>
> but I don't think IDL supplies one.   If I write this one-liner
> myself, I may reduce the number of errors in the code to one
> but I'd prefer to make it completely transparent...
>
```

I recall Liam had some tools for this:

http://cimss.ssec.wisc.edu/~gumley/pointers.html

We recently had some trouble converting legacy handles to pointers. It
sounds like a trivial transformation, but here are the sticky spots:

1. Handles are long integers, pointers have their own special variable
type.

2. A good test for handle validity is "if handle ne 0L". This will
fail miserably for pointers, where "if ptr_valid(ptr)" is the
equivalent.

3. Handle storage slots in structures, object, etc. will consist of a
long integers. This will need to be changed to pointers to avoid
errors.

So, simple scripts which convert handle->pointer will typically not be effective, since they cannot account for the variety of ways handles are treated as simple long integers in practice.

Maybe we should just go back to using uvalues of unpopulated base widgets. (OK, I admit, that was actually before my time... we can solicit the full story from David).

Good luck,

JD