Subject: Re: Pointer syntax and IDL 4.0
Posted by Pavel A. Romashkin on Wed, 19 Dec 2001 21:01:54 GMT
View Forum Message <> Reply to Message

Obviously one can live just fine without spending a fortune on upgrades :)

tam wrote:
>
> 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?

It depends on how far back the code has to be backward compatible. Isn't
earlier IDL totally lacking pointers?
I'd look up !Version.Release in the code and use IF statements to switch
to Handles syntax.

Cheers,
Pavel

Subject: Re: Pointer syntax and IDL 4.0
Posted by Paul van Delst on Wed, 19 Dec 2001 21:07:33 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...
>

> Any ideas?

Have a lookie at:

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

--
Paul van Delst Religious and cultural
CIMSS @ NOAA/NCEP purity is a fundamentalist
Ph: (301)763-8000 x7274 fantasy
Fax:(301)763-8545 V.S.Naipaul

Subject: Re: Pointer syntax and IDL 4.0
Posted by thompson on Wed, 19 Dec 2001 21:11:10 GMT
View Forum Message <> Reply to Message

tam <tam@lheapop.gsfc.nasa.gov> writes:

> 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?

Try using an EXECUTE() statement. We've used that on occasion when
version-dependent code causes compilation problems.

Bill Thompson

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

Subject: Re: Pointer syntax and IDL 4.0
Posted by Craig Markwardt on Wed, 19 Dec 2001 23:36:16 GMT
View Forum Message <> Reply to Message

Paul van Delst <paul.vandelst@noaa.gov> writes:

> 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
...
>> Any ideas?
>
> Have a lookie at:
>
> http://cimss.ssec.wisc.edu/~gumley/pointers.html

Paul is right. Liam's code will work in IDL 4 and 5, and provides an interface that works under both versions of IDL too. Under IDL 4 there are no pointers, but "handles" are used instead, which have basically the same functionality. Basically you use POINTER_CREATE, POINTER_VALUE and POINTER_FREE in place of the normal pointer operations, and you are done.

Craig

--
---------- ----------
Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
---------- ----------