Subject: Pointer syntax and IDL 4.0
Posted by tam on Wed, 19 Dec 2001 20:43:24 GMT
View Forum Message <> Reply to Message

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?

Thanks,
Tom McGlynn
tam@lheapop.gsfc.nasa.gov

Subject: Re: Pointer syntax and IDL 4.0: summary
Posted by tam on Thu, 20 Dec 2001 17:39:19 GMT
View Forum Message <> Reply to Message

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?
>   Thanks,
>   Tom McGlynn

> tam@lheapop.gsfc.nasa.gov
>

Thanks to all who wrote responses. I'm not sure any do quite what I want but I now have a set of options...

1. I can just forget about backwards compatibilility with the old code -- figure v4 users can use earlier versions of my code.

2. Use library routines, like Liam Gumley's, which hide the derefenence in a single function call and hope that V4 users don't explicitly try to compile that function. In Liam's code the Pointer_setgetv5 routine has the dereference syntax. It means that the user has to download at least two separate files.

3. Use execute to do all dereferences in run-time compiled code. This works fine but may be inefficient in some cases since the routines may be called millions of times. It's not as bad as I originally thought though... Millions would be a rare case and I seem to get about 20K execute calls per second which would mean the overhead would be negligible most of the time.

4. Use run-time compilation (a la 3) but compile a function (as in 2) to do the dereference. This would be nice since it would combine efficiency and common v4/v5 code. Alas I can't get execute to compile a function, so this requires creating
a temporary file -- and that's a real pain. Does anyone know how/if you can compile a function in execute (or more generally without reference to a physical file)? I'm hoping there's some devious route around the limits that IDL seems to have here.

Thanks for the help,

Regards,
Tom McGlynn