
Subject: Re: Pointer syntax and IDL 4.0: summary
Posted by [Craig Markwardt](#) on Thu, 20 Dec 2001 21:00:33 GMT
[View Forum Message](#) <> [Reply to Message](#)

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

> 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 compatibility 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 dereference
> 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.

You misinterpret Liam's library. The `POINTER_SETGETV5` procedure is *only* called when the user has IDL 5. Thus, in the normal course of events, and IDL 4 user can download the file but it will never be compiled or used. In the abnormal course of events, the user tries to compile the file and it fails, but that doesn't stop the program from working, since it will never be invoked.

I am renowned for keeping IDL 4 compatibility in my code. I still haven't changed to the `[]` syntax for array indexing (not broke, don't fix it). If you *do* use the `[]` notation, then forget IDL 4, stick with IDL 5 only.

Even so, the convenience of IDL 5 pointers is such that I am using them now in my new programs.

> 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.

There is no way to do this, other than dynamic compilation of a temporary file. Pavel has a secret about this :-)

Craig

--

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

Subject: Re: Pointer syntax and IDL 4.0: summary
Posted by [tam](#) on Fri, 21 Dec 2001 14:09:29 GMT
[View Forum Message](#) <> [Reply to Message](#)

>> 2. Use library routines, like Liam Gumley's, which hide the dereference
>> 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.

>>

>

> You misinterpret Liam's library. The POINTER_SETGETV5 procedure is
> *only* called when the user has IDL 5. Thus, in the normal course of
> events, and IDL 4 user can download the file but it will never be
> compiled or used. In the abnormal course of events, the user tries to
> compile the file and it fails, but that doesn't stop the program from
> working, since it will never be invoked.

>

I think that's what I said. Or at least what I meant!
Either the user has to store the function in a separate file or they'll get a compile time error. They also need to avoid explicitly compiling the code if it is in a separate file (e.g., if creating a save set).

I'll grant that the program will still run after such errors, but I wouldn't be comfortable distributing

code (or running code I'd received) which gave compile time errors.

One other aspect that I'm curious about is the extent to which this

incompatibility is 'planned'. This kind of incompatibility is in some ways very much in RSI/Kodak's interest. They could trivially have provided a functional equivalent of the dereference operator so that this kind of compile time incompatibility would be easy to avoid. But they don't want people to stay with IDL4.x.

Perhaps that's a little paranoid -- it's easy to overlook things like this. Who was it who said: "Never attribute to malice what can be accounted for by incompetence."?

Regards,
Tom McGlynn
