Subject: Re: Function referencing/automatic defintion question.
Posted by Paul Van Delst[1] on Thu, 29 May 2003 19:40:32 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
>
> Paul van Delst (paul.vandelst@noaa.gov) writes:
>
>> What this tells me is that the "default action" for IDL in this case is to assume that my
>> function call is really an array operation where I'm using () instead of [] to subscript
>> the array despite the fact that a function with the same name is compiled and resolved in
>> the current scope. Huh?
>>
>> This totally bamboozles me since I have a load of other source code files (including the
>> main file for this little project) that have more than one pro/function in them (e.g.
>> widget code with all the event handlers up front) with the "main" routine at the end. This
>> is the *only* time I've ever had problems. My assumption that the compilation of automatic
>> structure definition source files (the XXX__define type) is handled in the same way as
>> other multi-pro/function source files is apparently wrong. If so, I wonder what bright
>> spark decided that that would be a good idea?
>
> Well, I'm a little confused, too. But I've been burned too
> many times to jump on the bash the good folks at RSI
> bandwagon just yet. Most of the time I end up finding
> something stupid in my own code. :-(

Oh yeah - I've been there and done that plenty of times myself. It's the "good judgment
comes from experience; experience comes from bad judgment" type of thing. :o) But you're
right about the invective - I should replace the feet in my mouth (and count to 100 before
posting stuff).

> I'm not sure the notion of "compiled and resolved
> in the current scope" is terribly helpful. (For one
> thing, I don't even know what it means.) My understanding
> of the IDL compiler is that when it encounters an unresolved
> token it checks (1) to see if something by that name
> is already compiled and saved in the IDL code area,

This is the case. #1

<snip>

> Given the convoluted way this function was called, and

Convoluted? If this function calling method is convoluted, then doesn't that say the same
for object methods too? Don't get me wrong, I'm as adept at writing convoluted code as the
next person, but this seems quite bone simple to me.

> the sort of one-thing-after-another way computer software
> is written, I think a plausible explanation might be that
> even though IDL has compiled the function, it hasn't yet
> had a chance to write the function on its function list,
> so that at the time the list is *checked*, it is not there.

But it is. If I print out the list of compiled functions immediately prior to the function call itself, the function in question is on it. The next line -- the function call -- fails because IDL thinks the function call is an array reference.

If the result of

```
  PRINT, ROUTINE_INFO( /FUNCTIONS )
```

contains the name of the function in question, then it has been compiled, right? And the fact that using

```
  COMILE_OPT STRICTARR
```

makes eveything work means the same thing. (Right? I think so.)

> I don't feel confident enough about this to bet the ranch,
> but I would wager a beer or two that the answer turns out
> to be something like this. :-)

Well the whole thing has got me beat.

cheers,

paulv

--
Paul van Delst
CIMSS @ NOAA/NCEP/EMC
Ph: (301)763-8000 x7748
Fax:(301)763-8545