Subject: Re: Function referencing/automatic defintion question.
Posted by Paul Van Delst[1] on Thu, 29 May 2003 18:26:50 GMT
View Forum Message <> Reply to Message

Robert Moss wrote:
>
> Paul van Delst wrote:
>>
>> Thanks very much for the FORWARD_FUNCTION tip. That worked....but I
>> don't understand in the least why it should be necessary.
>>
>> cheers,
>>
>> paulv
>>
>>
>
> I have run across the same problem, Paul, and I agree that it should not
> *be* a problem. I have gotten in the habit of using
>
> COMPILE_OPT IDL2
>
> in virtually all of my programs. Since this forces one to properly use
> brackets for array indices and parentheses for function calls, it
> obviates the need for the use of FORWARD_FUNCTION. I'd be curious to
> know if this solution would also solve your problem... I suspect it would.

Yep, you're right. I removed the FORWARD_FUNCTION statement and replaced it with a COMPILE_OPT STRICTARR and everything worked fine. Thanks very much.

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?

cheers,

paulv

--

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