Subject: Re: Function referencing/automatic defintion question.
Posted by Paul Van Delst[1] on Fri, 30 May 2003 13:49:04 GMT
View Forum Message <> Reply to Message

JD Smith wrote:
>
> On Thu, 29 May 2003 12:40:32 -0700, Paul van Delst wrote:
>
>> 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.)
>
> There are routines which IDL knows about but hasn't compiled. They're
> called "unresolved".

That's what I meant in my previosu posts when I said they were compiled and resolved.

> Try this:
>
> print,routine_info(/FUNCTIONS,/SOURCE)

Here is the snippet of code where the weird stuff occurs (I commented out the COMPILE STRICTARR statement):

```
  ; ---------------------------------------------
  ; Create and allocate the coefficient structure
  ; ---------------------------------------------
  ; -- Create the structure
  EmisCoeff = { EmisCoeff }

  print,routine_info(/FUNCTIONS,/SOURCE)
  stop ;***NOTE the STOP ***
  ; -- Allocate it
  Result = Allocate_EmisCoeff( n_Wind_Speeds, n_Coefficients, n_Channels, $
                               EmisCoeff )
  IF ( Result NE SUCCESS ) THEN $
    MESSAGE, 'Error allocating EmisCoeff structure', $
             /NONAME, /NOPRINT
```

The output upon execution is (I put each listing of the ROUTINE_INFO output on a separate line so it's readable):

```
IDL> .reset_session
IDL>  print, compute_emissivity_coefficients( 'airsM9_aqua.SensorEmissivity.nc',
EmisCoeff, /pause)
% Compiled module: COMPUTE_EMISSIVITY_COEFFICIENTS.
```

% Compiled module: VALID_STRING.
% Compiled module: READ_NCDF.
% Compiled module: IS_NCDF.
% Compiled module: EMISCOEFF__DEFINE.
{ ALLOCATE_EMISCOEFF
 /usr2/wd20pd/idl/Emissivity/Sensor_Emissivity_Model/emiscoef f__define.pro}
{ASSIGN_EMISCOEFF
 /usr2/wd20pd/idl/Emissivity/Sensor_Emissivity_Model/emiscoef f__define.pro}
{ASSOCIATED_EMISCOEFF
 /usr2/wd20pd/idl/Emissivity/Sensor_Emissivity_Model/emiscoef f__define.pro}
{ COMPUTE_EMISSIVITY_COEFFICIENTS
 /usr2/wd20pd/f90/Emissivity/Sensor_Emissivity_Model/Regress_
Sensor_Emissivity/compute_emissivity_coefficients.pro}
{ COMPUTE_EMISSIVITY_FIT
 /usr2/wd20pd/f90/Emissivity/Sensor_Emissivity_Model/Regress_
Sensor_Emissivity/compute_emissivity_coefficients.pro}
{ COMPUTE_THETA_COEFFICIENTS
 /usr2/wd20pd/f90/Emissivity/Sensor_Emissivity_Model/Regress_
Sensor_Emissivity/compute_emissivity_coefficients.pro}
{ DESTROY_EMISCOEFF
 /usr2/wd20pd/idl/Emissivity/Sensor_Emissivity_Model/emiscoef f__define.pro}
{ MPCURVEFIT }
{SPLINE }
{ UNIQ }
% Stop encountered: COMPUTE_THETA_COEFFICIENTS 204
/usr2/wd20pd/f90/Emissivity/Sensor_Emissivity_M
   odel/Regress_Sensor_Emissivity/compute_emissivity_coefficien ts.pro

So, immediately before the call to Allocate_EmisCoeff(), it is in the compiled function
list, along with it's source file. I then type .cont:

IDL> .cont
% COMPUTE_THETA_COEFFICIENTS: Variable is undefined: ALLOCATE_EMISCOEFF.
% COMPUTE_EMISSIVITY_COEFFICIENTS: Error computing emissivity vs. theta fit coefficients.
         -1

> I don't think there's anything wrong with your setup. I can put:
>
> function stfunction,a
>   return,a^2
> end
>
> pro stprocedure,b
>   return
> end
>
> pro st__define,a
>   a={ST,b:0}

> end
>
> in st__define.pro, and then:
>
> IDL> a={st}
>
> compiles the listed procedure and function by side-effect, and they work
> fine. The place this technique can go quite wrong, as it can for objects,
> is if the structure in question is already defined by some other means.

It isn't. This is the *only* place in the code where I do
EmisCoeff = { EmisCoeff }

I never create named structures for in-line structure definitions since I am always adding/changing stuff to/in them. But using the __define method creates a named structure so I know where it's happening. If I understand it correctly, a named structure is a different beastie from an unnamed one so it should be quite easy to tell the difference.

And when it reaches this part of the code I always get a

% Compiled module: EMISCOEFF__DEFINE.

message since all my tests have been preceded by a ".reset_session" to ensure I'm not shooting myself in the foot.

> Then IDL does not feel compelled to compile your __define fine, and your
> utility routines remain hidden. Any chance you use a full structure
> definition in creating a struct of this type anywhere else?

No, unfortunately (I say unfortunately because if my brainfade was the cause, that would make me happy).

> You obviously can't mix the two methods for this to work.

O.k. I can see that. But, again, the fact that everything works when I stick in a COMPILE_OPT STRICTARR statement suggests that a hidden, in-line definition that I forgot about somewhere else is not the problem.

Thanks for the info re: the ROUTINE_INFO. Although I'm even more bamboozled now - everything tells me the function is all ready to be used but IDL keeps thinking the call is an array reference rather than a function call.

cheers,

paulv

--
Paul van Delst

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