Subject: function in function
Posted by nobody@nowhere.com (S on Thu, 11 Oct 2001 20:31:53 GMT
View Forum Message <> Reply to Message

I wrote a function in my IDL program which opens a proprietary file format
and dumps the data into an array, as I had to do it many times, I wrote this
as a function. In this function, I find I had to convert fixed-point binary
numbers to floats, so I wrote a second function wich does this. it is called
within the first function. it all works fine, but when I first start IDL, and
compile, the program fails with an undefined reference to the second function.
after the second compile, it works fine. I inserted the FORWARD_FUNCTION
statement in the 1st function, and then the problem went away. I know that
if I compile a procedure, with references to other procedures, all will work
well. So I'm wondering why the function in a function gives this problem or
if others have encountered this? The IDL documentation says that the FORWARD
_FUNCTION statement merely helps IDL to decide if the call is a function or
an array, but it is only needed in the second function, so I don't think it
explains the observed behavior. At first I thought this was having something
to do with the order in which the procedures are compiled, e.g. you shouldn't
compile a procedure that makes reference to another procedure/function that
has yet to be defined. But I think I would be hard pressed to gaurantee that
all the miles of code I've written would actually observe this rule, yet they
all work just fine. So is there something different about functions vs.
procedures that gives this effect?
--
Steve S.

steve@NOSPAMmailaps.org
remove NOSPAM before replying

Subject: Re: function in function
Posted by Mark Hadfield on Thu, 11 Oct 2001 21:14:13 GMT
View Forum Message <> Reply to Message

From: "Steve Smith<steven_smith>" <nobody@nowhere.com>
>
> I wrote a function in my IDL program which opens a proprietary file format
> and dumps the data into an array, as I had to do it many times, I wrote
this
> as a function. In this function, I find I had to convert fixed-point
binary
> numbers to floats, so I wrote a second function wich does this. it is
called
> within the first function. it all works fine, but when I first start IDL,
and
> compile, the program fails with an undefined reference to the second
function.

> after the second compile, it works fine. I inserted the FORWARD_FUNCTION
> statement in the 1st function, and then the problem went away.
> ...

To avoid these problems, either

* Put each function in a separate .pro file (with name =
<function_name>.pro) and make sure both are on your path.

or if you don't want to call the second function from anywhere but inside
your first function

* Put the first function in a .pro file (name = first_function.pro) then add
the code for your second function BEFORE THIS in the file. In this case it's
best to give the second function a name based on the first function name so
it won't clash with anything else (eg first_function_binary_to_float)

The reason for putting auxiliary functions first is that when IDL is trying
to resolve first_function it searches for first_function.pro then compiles
it, stopping when it has found the code for first_function. Any later code
in the file is ignored.

Mark Hadfield
m.hadfield@niwa.cri.nz http://katipo.niwa.cri.nz/~hadfield
National Institute for Water and Atmospheric Research

--