Subject: Re: Problem running functions
Posted by David Fanning on Tue, 10 Sep 2013 02:48:37 GMT
View Forum Message <> Reply to Message

laura.hike@gmail.com writes:

> This is a problem I've had before several times that has caused me to change all my functions to procedures. This time I refuse! The basic problem is that I write a function, test it, have it work fine, then try to use it in a real program and it fails, generally with "Variable is undefined". Here are the details:
>
> I have a directory for IDL functions and procedures I have written. Every new code I write starts out with the line adding this directory to my path: !path = '/home/disk/margaret/idl:' + !path
>
> I wrote two new and very short functions in this directory. I tested them using a little program in a completely different directory: /home/dingo/2/snowmelt/MODIS.SW/data I ran this from the IDL command line, not IDLDE. They ran fine and gave the answers I expected.
>
> Then I switched to a different directory (/home/dingo/2/ceres.syn/ed3a) and started writing the full program I intended. I included use of one of the functions. When I tested the new program, I got the "Variable is undefined" error. WTF?? I checked the path, I checked the spelling and case of the function name, etc., but no luck. The typical error I read about on line is an order of compilation error, but I don't see where there should be any issue if the function worked fine in the test program. I mean, the new code just loads some variables from a file, then calls the function to linearly interpolate two of the values. How can the order be incorrect?
>
> Interestingly, I went back and tried to rerun the test program. I moved a copy to the second work directory (ceres etc.) and got an all-new error message: "Only 8 subscripts allowed." Huh?? I then tried running the test program in the directory where I had written it and it had previously worked fine and now also got the "Only 8 subscripts allowed." message. Note: I restart IDL _every_ time I run a program, so the memory is cleared.
>
> (Don't ask me why I wrote the test code in a different directory. I must have just confused my terminal windows.)
>
> Should I just give up again??

Yes, I think you should give up. Or, maybe just give up this unbelievably unorthodox and inefficient way of working with IDL. One or the other, I suppose. :-)

You restart IDL every time you run a program!? Wow.

How about you do this? Take that path statement out of all of your programs. God only knows what your IDL path looks like after you run a few programs.

Put your IDL programs, procedures and functions, in one directory. Add

that directory to your IDL path in some kind of normal fashion. Here are some articles you can read to give you some ideas for how to do this:

http://www.idlcoyote.com/code_tips/installcoyote.php
http://www.idlcoyote.com/misc_tips/idl_startupfile.php

In place of all that path mumbo-jumbo you were including in your files, add this line instead:

compile_opt strictarr

Now, read this article, particularly the part about naming files correctly:

http://www.idlcoyote.com/code_tips/mostcommon.html

OK, ready to go? Now, just call your function normally. It will work perfectly. No restart of IDL required, ever. :-)

result = MyFunction()

Hurrah!

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thue. ("Perhaps thou speakest truth.")