Subject: Re: How to link IDL files
Posted by Vapuser on Wed, 24 Mar 1999 08:00:00 GMT
View Forum Message <> Reply to Message

VU KHAC Tri <tvk@info.fundp.ac.be> writes:

> Hi,
> I'm am IDL newbie.
> I want to write some IDL procedures split in different files. How can I
> call a pro in a file from another pro in another file.
> Best regards,
>
>
> --
> #++ ++++++++#
> | VU KHAC TRI |

This is probably a FAQ, but I couldn't check the FAQ to be sure, so I'll answer it anyway.

The short, and slightly incorrect, answer is, put 1 procedure/function per file, give the file the same name as the procedure/function with a .pro extension, e.g. if the function is named 'my_funct' then name the file 'my_funct.pro' and make sure that the IDL variable !path contains the directory(ies) wherein these files reside. In UNIX, you do this last part by correctly defining the environmental variable IDL_PATH. If you do these things, IDL will find and compile the procedures automatically when it first sees them.

The long answer is:

Search in the IDL help file on the system variable '!path' and read it. Look at the part in you paper documentation about defining this variable and read it. In Unix land, this variable is controlled by the environment variable IDL_PATH which you should set in, if you're using csh or one of it's variants, either your .login or your .[t]cshrc, if not, your .profile or whichever shell initialization files you need. I don't know how it's done in Windows. (By the way, it would help if you gave us info about the system you're using. Do a

```
IDL> print,!version
```

and report the output.)

!path is a list of directories the IDL interpreter searchs when trying to resolve a procedure/function call that isn't internal to IDL. Upon encountering a unresolved IDL procedure/function call, the interpretor

looks for a file in the path with the same name and a '.pro'
extension. This places some restrictions on the names of
procedures/functions if they are to be automatically compiled in this
manner. For instance, you can't name a file 'a-b.pro', although this
is a legitimate file name in Unix, since the interpreter would
interpret the attempt at calling 'a-b,arg' as... well.. a syntax
error. Also be aware that in Unix, the case of the file name is
significant, but the case of the function/procedure call isn't!
(e.g. you can call the function a=ThisIsAFunction() but IDL will look
for the file 'thisisafunction.pro') So, best practice is to give unix
files all lowercase names. When the interpreter finds the file, it
compiles it, then executes the named procedure/function (provided the
procedure/function is defined by that file. So, best practice is to
name the procedure/function the same as the filename.) By the way,
you can define more than one procedure/function in a file, but you
should always have the procedure/function with the same name as the
file as the last one in that file.

A piece of syntactic sugar. In unix, when you set the path in the
shell initialization files, you can tell IDL to search all the
directories below a certain one by prepending a '+' to the path.

Thus, setenv

```
  IDL_PATH \+/some/directory/path:\+/some/other/path
```

means search *all* directories below the two directories
'/some/directory/path' and '/some/other/path'

This is probably true for windows/mac too, I don't know.

whd