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

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

> 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,
>

The first thing is to make sure that all your files are named properly. If you write a procedure like this:

```
PRO test
print, 'Hello World'
END
```

....................then save it in a file called test.pro. The reason for this is as follows. When you type the name of a procedure or a function that is not yet compiled IDL goes out and searches for a file named   (What you typed).pro . This means that you could start up a new session of IDL and before you had even compiled our test program you could type
IDL>test
 .... and it would find test.pro, compile it and then call a now compiled procedure called test.

A word of warning. Wherever you store all your programs, you must make sure it is in the list of paths that IDL searches in. Go to the file menu and select preferences (or is it options - one of the two). In the path tab, you will find a box where you can add paths using a file selector and it will look there for your programs.

Okay, so that's all very well you're thinking, but how the hell does all of this help me (and you may have known the filenaming stuff already anyway). Well, once you have renamed all your files to the right names, it is extremely easy to call other procedures from within procedures. If we create a new procedure:

```
PRO HelloWorld
test
END
```

 ....and save it as HelloWorld.pro (somewhere in your search path), then when at the prompt you type

```
IDL> HelloWorld
%Compiled function helloworld.pro
%Compiled function test.pro
Hello World
```

..........you will get a response like this. (or there abouts,
I'm not at my own PC so I can't remember exactly all the prompts and
stuff look like!!) Calling functions is pretty similar but the syntax
is slightly different. If test had been like this:

```
FUNCTION test
RETURN, "Hello World"
END
```

.... then hello world would have
been like this:

```
PRO HelloWorld
temp = test()
print, temp
END
```

The temp variable receives the variable passed back by the function.

Finally, maybe you have some files in which you have several
different procedures/functions. One is the main procedure/function
which you call and then it calls others which are never called by the
user only by this particular procedure/function. You might not be
quite sure what exactly you should call the file when you save it, to
make it work like before, (or what the hell I'm talking about).

Well, if we take the current situation. If we change the situation
slightly. When we call HelloWorld, if we set a keyword then it calls
test2, if not it calls test1. These two are both functions which are
never used by any other programs, only by HelloWorld. There's no point
in creating separate files for each of test1 and test2. Instead you
can write the file like this:

```
FUNCTION test1
    RETURN, "This is test1 - Hello World"
END

FUNCTION test2
   RETURN, "This is test2 - Hello World"
END

PRO HelloWorld, WHICHTEST=WhichTest
   IF Keyword_Set(WhichTest) THEN temp = test1() ELSE temp = test2()
   print, temp
END
```

........................you then save it in a file called
"HelloWorld.pro". Open a new IDL session and type

```
IDL>  HelloWorld, WHICHTEST=1
%Compiled function test1
%Compiled function test2
%Compiled function HelloWorld
This is test1 - Hello World
```

You may have worked out from the behaviour what it's doing. It compiles any functions or procedures it finds until it reaches one which is called the same name as the filename, i.e. HelloWorld.

So make sure that the procedure/function you are calling is placed right at the bottom of the file.

I hope this explains it all. Perhaps you know a lot of it already a simply wanted to know the syntax of temp=test1() and test1, but it's better to have too much help than not enough.

Anyway,
Cheers,
Phil Aldis