

---

Subject: How to link IDL files

Posted by [VU KHAC Tri](#) on Wed, 24 Mar 1999 08:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

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 |

---

---

Subject: Re: How to link IDL files

Posted by [Phillip & Suzanne](#) on Wed, 24 Mar 1999 08:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Vapuser wrote:

- >
- > 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.

There is a way of allowing similar behavior on windows and mac, but it's set in the preferences file instead of the environment.

Phillip

---

---

Subject: Re: How to link IDL files

Posted by [Phillip & Suzanne](#) on Wed, 24 Mar 1999 08:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

David Fanning (davidf@dfanning.com) picked a few nits, but left himself open for some nit picking himself... He wrote:

>  
> And to just make this absolutely crystal clear, the program  
> module that has the same name as the file should be the  
> LAST program module in the file. If it is not, you will  
> NEVER get automatic compilation (which has got to be one  
> of the best features of IDL) to work correctly. There  
> should really never be any reason at all to hand compile  
> programs before they are used if your files are structured  
> and named correctly.

However, it may be appropriate to write test drivers for your programs, and in fact, I recommend this procedure highly. In this case, the test driver gets put AFTER the program matching the name of the file. When you want to test changes, you compile the file, and the test driver gets compiled as well. When you are simply trying to use the program, though, the test driver doesn't get compiled. This keeps the namespace from getting cluttered, and even occasionally prevents an error by calling a routine named the same as your test driver. I know I use the name "test" frequently, and wouldn't want my test driver to get replaced by another test driver for another file.

Phillip

---

---

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 your 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 its 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 searches when trying to resolve a procedure/function call that isn't internal to IDL. Upon encountering an unresolved IDL procedure/function call, the interpreter 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

---

---

Subject: Re: How to link IDL files  
Posted by [davidf](#) on Wed, 24 Mar 1999 08:00:00 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Phil Aldis ([philaldis@geocities.com](mailto:philaldis@geocities.com)) covers just about all the ground here, but I want to make one or two niggling points. He writes:

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

In fact, you should probably save this file as "helloworld.pro" (all lowercase letters) and \*ESPECIALLY\* if you are on a UNIX system, in which filenames are case sensitive. When IDL goes looking for a file:

```
IDL> JunK_It
```

It doesn't care how you spelled the command on the command line, but it does always look for \*lowercase\* file names.

I would also encourage you to NOT put spaces in file names, although it is allowed on some operating systems. It will make your code less portable and I find it causes problems even on those operating systems that supposedly support it.

And to just make this absolutely crystal clear, the program module that has the same name as the file should be the LAST program module in the file. If it is not, you will NEVER get automatic compilation (which has got to be one of the best features of IDL) to work correctly. There should really never be any reason at all to hand compile programs before they are used if your files are structured and named correctly.

Cheers,

David

P.S. If Vu Khac Tri is a student, I still have several cheap copies of my book hanging about. :-)

--

David Fanning, Ph.D.

Fanning Software Consulting

Phone: 970-221-0438 E-Mail: [davidf@dfanning.com](mailto:davidf@dfanning.com)

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Toll-Free IDL Book Orders: 1-888-461-0155

---

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](mailto: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

---

Subject: Re: How to link IDL files  
Posted by [Ivan Zimine](#) on Wed, 24 Mar 1999 08:00:00 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

VU KHAC Tri 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.

You make a call to your procedures and functions the same way as you call any IDL routine. The routines must be compiled before the first call. If they are not, then IDL will try to find a routine\_name.pro file in the directories specified in !path variable. To make sure all your routines are compiled you can add .compile command to your IDL\_STARTUP file or write a batch file which will compile specific files with routines needed to your particular application.

I advise you to visit  
IDL FAQ at [http://www.ivsoftware.com/pub/idl\\_faq.html](http://www.ivsoftware.com/pub/idl_faq.html)  
and  
Coyote's Guide at <http://www.dfanning.com/>

best regards  
Ivan

--

Dpt. of Radiology (MRI), Geneva University Hospitals  
email: [ivan.zimine@physics.unige.ch](mailto:ivan.zimine@physics.unige.ch)  
tel. : (+41 22) 372 70 70

---

Subject: Re: How to link IDL files  
Posted by [davidf](#) on Thu, 25 Mar 1999 08:00:00 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

David Foster ([foster@bial1.ucsd.edu](mailto:foster@bial1.ucsd.edu)) writes:

> You shouldn't ever need to hand compile files, but you might want  
> to "include" them in your current program using something like:  
>  
> @setarray\_utils.pro ; Compiles file  
>  
> In this case "setarray\_utils.pro" is a collection of functions that  
> I want to group together because they have a common purpose, and I  
> don't necessarily want all it's routines cluttering up my directory  
> as separate files.

Sigh...

With modern computer technology, you could probably group  
all your program files in a separate sub-directory. :-)

Cheers,



David

--

David Fanning, Ph.D.

Fanning Software Consulting

Phone: 970-221-0438 E-Mail: davidf@dfanning.com

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Toll-Free IDL Book Orders: 1-888-461-0155

---

---

Subject: Re: How to link IDL files

Posted by [David Foster](#) on Thu, 25 Mar 1999 08:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

David Fanning wrote:

>

> <snip>

>

> There

> should really never be any reason at all to hand compile

> programs before they are used if your files are structured

> and named correctly.

Vu -

You shouldn't ever need to hand compile files, but you might want to "include" them in your current program using something like:

```
@setarray_utils.pro      ; Compiles file
```

In this case "setarray\_utils.pro" is a collection of functions that I want to group together because they have a common purpose, and I don't necessarily want all it's routines cluttering up my directory as separate files.

If you use this, you can easily compile all routines that are in another file and then call them. Used this way, the routines you call don't necessarily have to be in a file having the same name.

Dave

--

~~~~~  
David S. Foster Univ. of California, San Diego  
Programmer/Analyst Brain Image Analysis Laboratory  
foster@bial1.ucsd.edu Department of Psychiatry  
(619) 622-5892 8950 Via La Jolla Drive, Suite 2240  
La Jolla, CA 92037

~~~~~

---

Subject: Re: How to link IDL files  
Posted by [davidf](#) on Thu, 25 Mar 1999 08:00:00 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Phillip David (pdavid@earthling.net) writes:

> David Fanning (davidf@dfanning.com) picked a few nits, but left himself open  
> for some nit picking himself... He wrote:  
>>  
>> And to just make this absolutely crystal clear, the program  
>> module that has the same name as the file should be the  
>> LAST program module in the file. If it is not, you will  
>> NEVER get automatic compilation (which has got to be one  
>> of the best features of IDL) to work correctly. There  
>> should really never be any reason at all to hand compile  
>> programs before they are used if your files are structured  
>> and named correctly.  
>  
> However, it may be appropriate to write test drivers for your programs, and in  
> fact, I recommend this procedure highly. In this case, the test driver gets  
> put AFTER the program matching the name of the file. When you want to test  
> changes, you compile the file, and the test driver gets compiled as well.  
> When you are simply trying to use the program, though, the test driver doesn't  
> get compiled. This keeps the namespace from getting cluttered, and even  
> occasionally prevents an error by calling a routine named the same as your  
> test driver. I know I use the name "test" frequently, and wouldn't want my  
> test driver to get replaced by another test driver for another file.

Now, honestly, Phillip. This is just too sophisticated for  
the usual riff-raff and falls under the header of giving away  
all our secrets. I am strongly in favor of banning this kind  
of posting.

Cheers,

David

P.S. You know, if more people wrote drivers for their  
programs, I'll bet more people would use them. :-)

--

David Fanning, Ph.D.  
Fanning Software Consulting  
Phone: 970-221-0438 E-Mail: [davidf@dfanning.com](mailto:davidf@dfanning.com)  
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Toll-Free IDL Book Orders: 1-888-461-0155

---