
Subject: Re: how can i call a compiled fortran code from IDL?
Posted by [David Fanning](#) on Thu, 06 Feb 2003 00:55:11 GMT
[View Forum Message](#) <> [Reply to Message](#)

mads (madhu@erc.msstate.edu) writes:

> I have a compiled FORTRAN code and I need to call it from a procedure in
> IDL. The FORTRAN code does not require any i/p. It reads everything from a
> data file and creates a data file as o/p.

SPAWN would be the easiest and most fool-proof way.

Cheers,

David

--

David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: how can i call a compiled fortran code from IDL?
Posted by [mads](#) on Thu, 06 Feb 2003 01:58:41 GMT
[View Forum Message](#) <> [Reply to Message](#)

This is what I need to do. I need to integrate the kuusk canopy reflectance model(5 .f files) into IDL code. I can use g77 and compile them into .exe and call from IDL - in this case how do i do it? or i can call the fortran code directly frm IDL. In this case how do I do it? If the explanation is too complicated where online can I find it. If its impossible I might as well recode all the fortran code in IDL . This will require that I learn FORTRAN first :(

Please help me in these 2 options. I have to further automate everything to train SNNS into recognising canopy types. So spawning and typing command will not serve my purpose. I tried all resources but couldnt find relevant info.

Please help

madhu

"David Fanning" <david@dfanning.com> wrote in message
news:MPG.18ab6059c38ab974989ac9@news.frii.com...

> mads (madhu@erc.msstate.edu) writes:

>

>> I have a compiled FORTRAN code and I need to call it from a
procedure in

>> IDL. The FORTRAN code does not require any i/p. It reads everything from
a

>> data file and creates a data file as o/p.
>
> SPAWN would be the easiest and most fool-proof way.
>
> Cheers,
>
> David
> --
> David W. Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Phone: 970-221-0438, E-mail: david@dfanning.com
> Coyote's Guide to IDL Programming: <http://www.dfanning.com/>
> Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: how can i call a compiled fortran code from IDL?
Posted by [David Fanning](#) on Thu, 06 Feb 2003 04:08:22 GMT
[View Forum Message](#) <> [Reply to Message](#)

mads (madhu@erc.msstate.edu) writes:

> This is what I need to do. I need to integrate the kuusk canopy reflectance
> model(5 .f files) into IDL code. I can use g77 and compile them into .exe
> and call from IDL - in this case how do i do it?

Uh, SPAWN.

> or i can call the fortran
> code directly frm IDL. In this case how do I do it?

You don't want to do this. :-)

> If the explanation is
> too complicated where online can I find it.

I'd try this:

IDL> ? SPAWN

This FORTRAN code takes no input and produces a file
as output, right? After it produces the file (and
SPAWN returns), you can just read the file into IDL.

> If its impossible I might as
> well recode all the fortran code in IDL . This will require that I learn
> FORTRAN first :(

Well, it is MUCH easier to learn FORTRAN than it is

to learn the IDL internals with the documentation available.

I really recommend SPAWN. Even I can understand how that works. :-)

- > Please help me in these 2 options. I have to further automate everything to
- > train SNNS into recognising canopy types. So spawning and typing command
- > will not serve my purpose.

Why would you be typing something? You will SPAWN the command that runs your FORTRAN program:

```
SPAWN, '/usr/local/me/myprogram.exe'  
; Now I am going to read the data file MYPROGRAM produced  
OpenR, lun, 'myprogram_output.dat', /Get_Lun  
etc.
```

- > I tried all resources but couldnt find relevant info.

It's a pain, sometimes.

Cheers,

David

--

David W. Fanning, Ph.D.

Fanning Software Consulting, Inc.

Phone: 970-221-0438, E-mail: david@dfanning.com

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

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

Subject: Re: how can i call a compiled fortran code from IDL?

Posted by [thompson](#) on Thu, 06 Feb 2003 14:27:45 GMT

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

"mads" <madhu@erc.msstate.edu> writes:

- > Hi,
- > I have a compiled FORTRAN code and I need to call it from a procedure in
- > IDL. The FORTRAN code does not require any i/p. It reads everything from a
- > data file and creates a data file as o/p.
- > Madhu

It's been a long, long time since I last did this, and there are probably more modern ways to do it, but I was able in the past to use LINKIMAGE to link Fortran procedures into IDL. The secret was to put a C wrapper around the Fortran. For example, here is a C wrapper for some software that I'm still using, where the IDL input and output arguments are converted via C into the kind of things that Fortran can understand. (For some strange reason, when the

Fortran routine is called from C, it ends up with an extra "_" appended to its name.) Probably something similar can be done with a DLM.

You can see all the C and Fortran code, plus make files for some different platforms, at

ftp://sohoftp.nascom.nasa.gov/solarsoft/gen/idl_external/

Bill Thompson

```
#include <stdio.h>
```

```
void f_median_c(argc, argv)
    int argc; /* The number of arguments */
    void *argv[]; /* The arguments */
{
    float *in, *out, *missing, *workspace;
    long *ndim1, *ndim2, *n_width1, *n_width2;

    /* Convert the IDL input parameters into FORTRAN parameters. */

    in = (float *) argv[0];
    out = (float *) argv[1];
    ndim1 = (long *) argv[2];
    ndim2 = (long *) argv[3];
    n_width1 = (long *) argv[4];
    n_width2 = (long *) argv[5];
    missing = (float *) argv[6];
    workspace = (float *) argv[7];

    /* Call the FORTRAN routine F_MEDIAN. */

    f_median_(in,out,ndim1,ndim2,n_width1,n_width2,missing,workspace);
    return;
}
```

Subject: Re: how can i call a compiled fortran code from IDL?

Posted by [thompson](#) on Thu, 06 Feb 2003 14:33:24 GMT

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

thompson@orpheus.nascom.nasa.gov (William Thompson) writes:

> "mads" <madhu@erc.msstate.edu> writes:

>> Hi,

>> I have a compiled FORTRAN code and I need to call it from a procedure in

>> IDL. The FORTRAN code does not require any i/p. It reads everything from a
>> data file and creates a data file as o/p.
>> Madhu

> It's been a long, long time since I last did this, ...

(rest deleted)

I haven't had my coffee yet, and I only just now realized that you state that everything is read from a data file, and written to a data file. In that case, I completely agree with David--all you want is SPAWN. Use IDL to write out the input file, spawn the command to run the compiled Fortran program, and then read the output file back into IDL. That's the simplest and most direct approach. Anything else would be a waste of time.

Bill Thompson

Subject: Re: how can i call a compiled fortran code from IDL?
Posted by [James Kuyper](#) on Thu, 06 Feb 2003 16:05:57 GMT
[View Forum Message](#) <> [Reply to Message](#)

William Thompson wrote:

...
> ... (For some strange reason, when the
> Fortran routine is called from C, it ends up with an extra "_" appended to its
> name.) ...

It's a naming convention issue, and the naming conventions are different on different platforms. On some platforms, an '_' is prepended, rather than appended, on some platforms, both will be needed.

If you have access to HDF, it provides a header "hdfi.h", containing a number of useful macros for encapsulating what the've learned about portable Fortran-C coding. FNAME(name) provides the C function name that will be callable as a Fortran function or subroutine called 'name'; it adds '_' wherever it's needed, if it's needed. In order to make a C function work as a Fortran function returning a specified type, you declared the return type as FRETVAL(type). It also defines a typedef named 'intf' for an integral C type that has the same representation as the Fortran INTEGER type.

Subject: Re: how can i call a compiled fortran code from IDL?
Posted by [David Fanning](#) on Thu, 06 Feb 2003 16:24:15 GMT
[View Forum Message](#) <> [Reply to Message](#)

James Kuyper (kuyper@saicmodis.com) writes:

- > It's a naming convention issue, and the naming conventions are different
- > on different platforms. On some platforms, an '_' is prepended, rather
- > then appended, on some platforms, both will be needed.
- >
- > If you have access to HDF, it provides a header "hdfi.h", containing a
- > number of useful macros for encapsulating what the've learned about
- > portable Fortran-C coding. FNAME(name) provides the C function name that
- > will be callable as a Fortran function or subroutine called 'name'; it
- > adds '_' wherever it's needed, if it's needed. In order to make a C
- > function work as a Fortran function returning a specified type, you
- > declared the return type as FRETVAL(type). It also defines a typedef
- > named 'intf' for an integral C type that has the same representation as
- > the Fortran INTEGER type.

Mads, do you see now why SPAWN is the solution of choice here? :-)

Cheers,

David

P.S. Let's just say learning FORTRAN is only the *start*
of your trouble. :-(

--

David W. Fanning, Ph.D.

Fanning Software Consulting, Inc.

Phone: 970-221-0438, E-mail: david@dfanning.com

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

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

Subject: Re: how can i call a compiled fortran code from IDL?

Posted by [mads](#) on Thu, 06 Feb 2003 17:40:26 GMT

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

hi all,

what if i have to use,

spawn, 'c:\hi\what to do now\something.exe'

now the space that is in "what now" causes a problem. Thats the reason i was
against spawning. i can solve that by just moving all the files into the
working directory and just use

spawn,'something.exe'

thats how i have solved the problem now

sincerely

madhu

"David Fanning" <david@dfanning.com> wrote in message

news:MPG.18ac3a1b6d55b87a989ace@news.frii.com...

> James Kuyper (kuyper@saicmodis.com) writes:

>

>> It's a naming convention issue, and the naming conventions are different

>> on different platforms. On some platforms, an '_' is prepended, rather

>> then appended, on some platforms, both will be needed.

>>

>> If you have access to HDF, it provides a header "hdfi.h", containing a

>> number of useful macros for encapsulating what the've learned about

>> portable Fortran-C coding. FNAME(name) provides the C function name that

>> will be callable as a Fortran function or subroutine called 'name'; it

>> adds '_' wherever it's needed, if it's needed. In order to make a C

>> function work as a Fortran function returning a specified type, you

>> declared the return type as FRETVAL(type). It also defines a typedef

>> named 'intf' for an integral C type that has the same representation as

>> the Fortran INTEGER type.

>

> Mads, do you see now why SPAWN is the solution of choice here? :-)

>

> Cheers,

>

> David

>

> P.S. Let's just say learning FORTRAN is only the *start*

> of your trouble. :-(

>

> --

> David W. Fanning, Ph.D.

> Fanning Software Consulting, Inc.

> Phone: 970-221-0438, E-mail: david@dfanning.com

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

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

Subject: Re: how can i call a compiled fortran code from IDL?

Posted by [Paul Van Delst\[1\]](#) on Thu, 06 Feb 2003 18:02:04 GMT

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

mads wrote:

>

> hi all,

> what if i have to use,

> spawn, 'c:\hi\something.exe'

what about something like:

```
execution_string = 'c:\hi\' + execution_directory + '\something.exe'
```

```
spawn, execution_string
```

You have to specify (somehow, e.g. via an argument with a default?) what is the directory that contains the executable, but you would have to identify it somehow in any case.

E.g.:

Given a utility routine:

```
FUNCTION Valid_String, Input_String
  IF ( N_ELEMENTS( Input_String ) EQ 0 ) THEN RETURN, 0
  IF ( STRLEN( Input_String ) EQ 0 ) THEN RETURN, 0
  RETURN, 1
END
```

then:

```
PRO myidlpro, Execution_Directory = Execution_Directory
```

```
  Default_ExeDir = 'what_now'
  IF ( Valid_String( Execution_Directory ) ) THEN $
    ExeDir = Execution_Directory $
  ELSE $
    ExeDir = Default_ExeDir
```

```
  ....do some other stuff...
```

```
  Execution_String = 'c:\hi\' + ExeDir + '\something.exe'
  SPAWN, Execution_String
```

```
  ...do some more stuff
```

```
END
```

So if you don't specify the "Execution_Directory" optional argument, it will default, but you can specify an alternative if required.

You should search the IDL online dos for "noninteractive use of spawn".

paulv

```
> now the space that is in "what now" causes a problem. Thats the reason i was
> against spawning. i can solve that by just moving all the files into the
> working directory and just use
> spawn,'something.exe'
> thats how i have solved the problem now
> sincerely
> madhu
```

--

Paul van Delst
CIMSS @ NOAA/NCEP/EMC
Ph: (301)763-8000 x7274
Fax:(301)763-8545

Subject: Re: how can i call a compiled fortran code from IDL?
Posted by [James Kuyper](#) on Thu, 06 Feb 2003 18:09:23 GMT
[View Forum Message](#) <> [Reply to Message](#)

mads wrote:

>
> hi all,
> what if i have to use,
> spawn, 'c:\hi\what to do now\something.exe'
> now the space that is in "what now" causes a problem. Thats the reason i was
> against spawning. i can solve that by just moving all the files into the
> working directory and just use
> spawn,'something.exe'
> thats how i have solved the problem now

Then your problem is with the command line you're using, not with spawn itself. Would the command

c:\hi\what to do now\something.exe

work at a command line in a DOS window? It's been a long time since I've used a DOS/Windows machine at that level, but naively I'd expect it to try to run a program named 'c:\hi\what', passing to it the argument list "to do now\something.exe". If that is what you want to do, then it should work equally well in the IDL spawn command. However, if there is no program named c:\hi\what, then you're going to have problems.

In Unix, if you have a directory name with spaces in it, you easiest way to refer to it is by using escape characters:

c:/hi/what\ to\ do\ now/something.exe

However, that obviously depends upon the fact that Unix uses '/' for it's directory delimiter, not '\'. I don't remember what the corresponding workaround is for DOS/Windows.

--

James Kuyper
MODIS Level 1 Lead
Science Data Support Team
(301) 352-2150

Subject: Re: how can i call a compiled fortran code from IDL?

Posted by [wmconnolley](#) on Thu, 06 Feb 2003 18:46:10 GMT

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

mads <madhu@erc.msstate.edu> wrote:

> hi all,

> what if i have to use,

> spawn, 'c:\hi\what to do now\something.exe'

Then you probably want

spawn, "c:\hi\what to do now\something.exe"

or

spawn, "c:\hi\what to do now\something.exe"

depending on what sort of quotes the Evil Empire likes.

-W.

--

William M Connolley | wmc@bas.ac.uk | <http://www.nerc-bas.ac.uk/icd/wmc/>

Climate Modeller, British Antarctic Survey | Disclaimer: I speak for myself

I'm a .signature virus! copy me into your .signature file & help me spread!
