Subject: Fortran + IDL
Posted by fskhk on Tue, 14 Aug 2001 08:39:06 GMT
View Forum Message <> Reply to Message

I am working in Fortran (with a complicated program) and IDL. Up to now I put the output of the fortran data in a text file, go to IDL, read the text file in IDL and draw the graphs. For every small change in fortran I have to go in and out the fortran program and IDL program. Will it be easier and better to use ActiveX or Callable IDL under Windows? How and where can I learn to make the linkage?

Subject: Re: Fortran + IDL
Posted by Stein Vidar Hagfors H[1] on Mon, 20 Aug 2001 18:56:19 GMT
View Forum Message <> Reply to Message

fskhk@puknet.puk.ac.za (Helena Kruger) writes:

> I am working in Fortran (with a complicated program) and IDL. Up to
> now I put the output of the fortran data in a text file, go to IDL,
> read the text file in IDL and draw the graphs. For every small change
> in fortran I have to go in and out the fortran program and IDL
> program.

Why do you need to do that? You should be able to have one window with IDL running (constantly), and one window with compilation/running of the Fortran program... Recompile/run the Fortran program, then re-read the output file..

> Will it be easier and better to use ActiveX or Callable IDL
> under Windows? How and where can I learn to make the linkage?

I'd recommend either callable IDL or (even better) IDL Remote Procedure Calls (look in the External Development Guide - edg.pdf in the documentation library for IDL). It's possible then to make (C) wrappers that will send your Fortran variables directly to IDL, execute calls to plot data etc..

The reason why RPC is better than Callable for this instance is that you can have *two* (or more!) clients connect to the same idlrpc server, so you can do interactive work as well on the variables sent over by your Fortran program.

However, unless the above two paragraphs tickle your programmer's curiosity to learn something neat, I'd just stick with your current setup (but don't quit the IDL session every time!). Getting something up and going may take quite some time - so it all depends on how long you expect to be testing this out interactively...

--
---------- ----------
Stein Vidar Hagfors Haugan
ESA SOHO SOC/European Space Agency Science Operations Coordinator for SOHO

NASA Goddard Space Flight Center, Email: shaugan@esa.nascom.nasa.gov
Mail Code 682.3, Bld. 26, Room G-1, Tel.: 1-301-286-9028/240-354-6066
Greenbelt, Maryland 20771, USA. Fax: 1-301-286-0264
---------- ----------

Subject: Re: Fortran + IDL
Posted by Craig Markwardt on Mon, 20 Aug 2001 20:22:48 GMT
View Forum Message <> Reply to Message

fskhk@puknet.puk.ac.za (Helena Kruger) writes:
> I am working in Fortran (with a complicated program) and IDL. Up to
> now I put the output of the fortran data in a text file, go to IDL,
> read the text file in IDL and draw the graphs. For every small change
> in fortran I have to go in and out the fortran program and IDL
> program. Will it be easier and better to use ActiveX or Callable IDL
> under Windows? How and where can I learn to make the linkage?

Stein Vidar's response spawned a thought. Why aren't you using SPAWN?
One point of any programming language is to proceduralize mundane
tasks so you don't have to keep doing the tasks yourself.

You could start with something like this:

```
pro dofortran, array
  ;; Call fortran program
  spawn, 'runfortran', output

  ;; Determine number of elements in array - assume it's a 3xN array
  n = n_elements(output)
  array = fltarr(3,n)

  for i = 0L, n-1 do begin
     v0 = [0., 0., 0.]
     reads, output(i), v0
     array(*,i) = v0
  endfor

  return
end
```

Obviously you will have to customize for your program name and the

number of variables, etc.

Craig

Subject: Re: Fortran + IDL
Posted by A. D. & J.C. Cool on Tue, 21 Aug 2001 03:37:50 GMT
View Forum Message <> Reply to Message

Craig Markwardt wrote:
>
> fskhk@puknet.puk.ac.za (Helena Kruger) writes:
>> I am working in Fortran (with a complicated program) and IDL. Up to
>> now I put the output of the fortran data in a text file, go to IDL,
>> read the text file in IDL and draw the graphs. For every small change
>> in fortran I have to go in and out the fortran program and IDL
>> program. Will it be easier and better to use ActiveX or Callable IDL
>> under Windows? How and where can I learn to make the linkage?
>
> Stein Vidar's response spawned a thought. Why aren't you using SPAWN?
> One point of any programming language is to proceduralize mundane
> tasks so you don't have to keep doing the tasks yourself.
>
> You could start with something like this:
>
> pro dofortran, array
> ;; Call fortran program
> spawn, 'runfortran', output
>
> ;; Determine number of elements in array - assume it's a 3xN array
> n = n_elements(output)
> array = fltarr(3,n)
>
> for i = 0L, n-1 do begin
> v0 = [0., 0., 0.]
> reads, output(i), v0
> array(*,i) = v0
> endfor
>
> return
> end
>
> Obviously you will have to customize for your program name and the
> number of variables, etc.
>
> Craig

Helena hasn't said what platform she's actually running IDL on.

If she is running IDL on OpenVMS, then yes, she must exit her IDL
session
every time she re-compiles the Fortran. Alas IDL on VMS does not
release/delete/
overwrite the routine/memory (what's the jargon?) called by
CALL_EXTERNAL...

Andrew

andrew.cool@dsto.defence.gov.au

Subject: Re: Fortran + IDL
Posted by fskhk on Fri, 24 Aug 2001 10:53:18 GMT
View Forum Message <> Reply to Message

Craig Markwardt <craigmnet@cow.physics.wisc.edu> wrote in message
news:<onelq6a3c7.fsf@cow.physics.wisc.edu>...
> fskhk@puknet.puk.ac.za (Helena Kruger) writes:
>> I am working in Fortran (with a complicated program) and IDL. Up to
>> now I put the output of the fortran data in a text file, go to IDL,
>> read the text file in IDL and draw the graphs. For every small change
>> in fortran I have to go in and out the fortran program and IDL
>> program. Will it be easier and better to use ActiveX or Callable IDL
>> under Windows? How and where can I learn to make the linkage?
>
> Stein Vidar's response spawned a thought. Why aren't you using SPAWN?
> One point of any programming language is to proceduralize mundane
> tasks so you don't have to keep doing the tasks yourself.
>
> You could start with something like this:
>
> pro dofortran, array
> ;; Call fortran program
> spawn, 'runfortran', output
>
> ;; Determine number of elements in array - assume it's a 3xN array
> n = n_elements(output)
> array = fltarr(3,n)
>
> for i = 0L, n-1 do begin
> v0 = [0., 0., 0.]
> reads, output(i), v0
> array(*,i) = v0
> endfor
>
> return

> end
>
> Obviously you will have to customize for your program name and the
> number of variables, etc.
>
> Craig

I got an example of Callable IDL from internet, using the fortran program align2.f, a c-wrapper, written by Doug Loucks of RSI, in 1998 and the IDL program, called align.pro. I used these programs and tried to link them, but am still struggling. I use the following versions of software: Visual C++ 5.0, IDL 5.2.1 and Visual Fortran Professional Edition 5.0A, working under WINDOWS.

I can compile the fortran program and the c-wrapper (Callable.c) with no problems.

I get the following error messages:

```
--------------------Configuration: callable - Win32
Debug--------------------
Linking...
LINK : warning LNK4098: defaultlib "libc.lib" conflicts with use of
other libs; use /NODEFAULTLIB:library
callable.obj : error LNK2001: unresolved external symbol
_IDL_Win32Init
callable.obj : error LNK2001: unresolved external symbol
_IDL_ExecuteStr
callable.obj : error LNK2001: unresolved external symbol
_IDL_ImportNamedArray
callable.obj : error LNK2001: unresolved external symbol _IDL_Cleanup
align2.obj : error LNK2001: unresolved external symbol
_IDL_WIN32INIT@4
align2.obj : error LNK2001: unresolved external symbol
_IDL_IMPORTNAMEDARRAY@32
align2.obj : error LNK2001: unresolved external symbol
_IDL_EXECUTESTR@8
align2.obj : error LNK2001: unresolved external symbol _IDL_CLEANUP@4
Debug/callable.exe : fatal error LNK1120: 8 unresolved externals
Error executing link.exe.
```

Is the reason that it look for the IDL32.LIB file because of the different versions of software used or maybe the path? How can I bypass this linking problem?

Helena