Subject: IDL calling C
Posted by nrk5 on Thu, 07 Nov 2002 19:58:15 GMT
View Forum Message <> Reply to Message

Hi,

I was wondering if anybody could point me to some resources for
calling C functions from IDL. I have never done this before myself and
am quite unfamiliar with C. I have somebody else's .h and .c files and
need to call their functions from IDL, but have little understanding
of the internals of their stuff. Thanks!

Nidhi

Subject: Re: IDL calling C
Posted by rmlongfield on Mon, 25 Nov 2002 09:33:42 GMT
View Forum Message <> Reply to Message

nrk5@cornell.edu (Nidhi Kalra) wrote in message
news:<6c4c9ef3.0211071158.2eea6820@posting.google.com>...
> Hi,
>
> I was wondering if anybody could point me to some resources for
> calling C functions from IDL. I have never done this before myself and
> am quite unfamiliar with C. I have somebody else's .h and .c files and
> need to call their functions from IDL, but have little understanding
> of the internals of their stuff. Thanks!
>
> Nidhi

Hi Nidhi (and everyone),
This is a bit late but I can help you with the C and IDL
interface using CALL_EXTERNAL. I have some sample files that I
need to make a little more user friendly and then I can send them
to you. They are written for SGI and Linux (with some necessary
keywords).
Below is a sample IDL code, I use it to call a C program
to call a Fortran program. Let me know if this is what you need
and I can send the rest.

Rose

```
PRO idl_rtau
;+
; NAME:
;      IDL_RTAU
;
```

```
; PURPOSE:  Demonstrate how one can run FORTRAN code from an IDL
session.
; AUTHOR:   Rose
; CATEGORY: CALL_EXTERNAL
; PROCEDURE:  IDL_RTAU does two things.
;  1) Runs a UNIX shell program through a SPAWN procedure.
;   This compiles the C and Fortran programs which will
;   be used later in the CALL_EXTERNAL.
;      If compilation has already been done, no need to recompile
;      This might save time for large compilation times.
;   Set compile_flag to zero.
;  2) Calls IDL Procedure, CALL_EXTERNAL, which accepts DOUBLE
;   input and returns DOUBLE output.  All variables must be
;   pre-defined.
; CALLING SEQUENCE:  idl_rtau
; MAJOR FUNCTIONS and PROCEDURES:
;  SPAWN
;  CALL_EXTERNAL
; NOTES:  If there are ANY modifications to the C or FORTRAN programs
;  one must exit IDL and then return  to run new executables.
;  Debugging should be done using accompanying wrapper routines.
; MODIFICATION HISTORY:  26 October 1999
;
; COMMON BLOCKS: none
;

print,'In idl_rtau: '
compile_flag = 1

IF(compile_flag  GT 0) THEN BEGIN
;
;-- Run make command which produces rtauc.o,rtauf.o,rtauc.so,
so_locations
;
sh_command = 'idl_rtau.sh'
SPAWN,sh_command
ENDIF ELSE BEGIN
;
print,'File is ok'
;
ENDELSE
; *** DEFINE variables for CALL_EXTERNAL ***
;     Must be Type double
;
surface_reflectivity = DOUBLE(.1)
nbcloud=14
tau = DBLARR(nbcloud)
reflectivity = DBLARR(nbcloud)
```

```
result_rtau =   CALL_EXTERNAL('rtauc.so','rtauc',surface_reflectivity,tau,re flectivity)

print,'Returned values from rtau: ',result_rtau
; Check results
IF(result_rtau EQ 0) THEN BEGIN
 FOR i = 0,N_ELEMENTS(tau)-1 DO BEGIN
 print,tau[i],reflectivity[i],FORMAT='(f6.2,1x,f6.2)'
 ENDFOR
ENDIF ELSE BEGIN
print, ' Well, something did not work'
ENDELSE
end
```

Subject: Re: IDL calling C
Posted by regnig on Sat, 07 Dec 2002 11:19:43 GMT
View Forum Message <> Reply to Message

There is also an excellent book called "Calling C from IDL; Using DLM's to extend your IDL code" by Ronn Kling. Look at http://kilvarock.com/books/callingCfromIDL.htm.

Mike

"Rose" <rmlongfield@yahoo.com> wrote in message news:5d5e16f6.0211250133.67ca32cf@posting.google.com...
> nrk5@cornell.edu (Nidhi Kalra) wrote in message news:<6c4c9ef3.0211071158.2eea6820@posting.google.com>...
>> Hi,
>>
>> I was wondering if anybody could point me to some resources for
>> calling C functions from IDL. I have never done this before myself and
>> am quite unfamiliar with C. I have somebody else's .h and .c files and
>> need to call their functions from IDL, but have little understanding
>> of the internals of their stuff. Thanks!
>>
>> Nidhi
>
> Hi Nidhi (and everyone),
> This is a bit late but I can help you with the C and IDL
> interface using CALL_EXTERNAL. I have some sample files that I
> need to make a little more user friendly and then I can send them
> to you. They are written for SGI and Linux (with some necessary
> keywords).
> Below is a sample IDL code, I use it to call a C program

```
> to call a Fortran program.  Let me know if this is what you need
> and I can send the rest.
>
> Rose
>
> PRO idl_rtau
> ;+
> ; NAME:
> ;       IDL_RTAU
> ;
> ; PURPOSE:  Demonstrate how one can run FORTRAN code from an IDL
> session.
> ; AUTHOR:   Rose
> ; CATEGORY: CALL_EXTERNAL
> ; PROCEDURE:  IDL_RTAU does two things.
> ; 1) Runs a UNIX shell program through a SPAWN procedure.
> ; This compiles the C and Fortran programs which will
> ; be used later in the CALL_EXTERNAL.
> ;   If compilation has already been done, no need to recompile
> ;   This might save time for large compilation times.
> ; Set compile_flag to zero.
> ; 2) Calls IDL Procedure, CALL_EXTERNAL, which accepts DOUBLE
> ; input and returns DOUBLE output.  All variables must be
> ; pre-defined.
> ; CALLING SEQUENCE:  idl_rtau
> ; MAJOR FUNCTIONS and PROCEDURES:
> ; SPAWN
> ; CALL_EXTERNAL
> ; NOTES:  If there are ANY modifications to the C or FORTRAN programs
> ; one must exit IDL and then return  to run new executables.
> ; Debugging should be done using accompanying wrapper routines.
> ; MODIFICATION HISTORY:  26 October 1999
> ;
> ; COMMON BLOCKS: none
> ;
>
> print,'In idl_rtau: '
> compile_flag = 1
>
> IF(compile_flag  GT 0) THEN BEGIN
> ;
> ;-- Run make command which produces rtauc.o,rtauf.o,rtauc.so,
> so_locations
> ;
> sh_command = 'idl_rtau.sh'
> SPAWN,sh_command
> ENDIF ELSE BEGIN
> ;
```

```
>  print,'File is ok'
>  ;
>  ENDELSE
>  ; *** DEFINE variables for CALL_EXTERNAL ***
>  ;     Must be Type double
>  ;
>  surface_reflectivity = DOUBLE(.1)
>  nbcloud=14
>  tau = DBLARR(nbcloud)
>  reflectivity = DBLARR(nbcloud)
>
>  result_rtau =
 CALL_EXTERNAL('rtauc.so','rtauc',surface_reflectivity,tau,re flectivity)
>
>  print,'Returned values from rtau: ',result_rtau
>  ; Check results
>  IF(result_rtau EQ 0) THEN BEGIN
>  FOR i = 0,N_ELEMENTS(tau)-1 DO BEGIN
>  print,tau[i],reflectivity[i],FORMAT='(f6.2,1x,f6.2)'
>  ENDFOR
>  ENDIF ELSE BEGIN
>  print, ' Well, something did not work'
>  ENDELSE
>  end
```