
Subject: Re: Fortran External Calls

Posted by [rmlongfield](#) on Fri, 18 Dec 1998 08:00:00 GMT

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

In article <F43Aqt.I3H@midway.uchicago.edu>,

jpgrimes@midway.uchicago.edu (john peter grimes) wrote:

> Hi,
> I just need help with a simple (I htink) problem.
> Basically I am having a bit of trouble understanding the little
> bit of documentation in my idl manuals on call_extern. I just have a f90
> function that I want to call from idl.

Hi John Peter:

I found the IDL example files very confusing. Here is
a straightforward sample of a call_extern using a C program to call the
Fortran program.

- 1) IDL program: idl_ex.pro : Note ALL arrays and variables
defined FIRST in IDL
- 2) C program: exc.c
- 3) Fortran program: exf.f
- 4) make file to compile before running in IDL (Must exit IDL
if you recompile!!!): Make_ex

I have run it on an SGI and IDL 5.0 with Fortran 77. You should be
able to just run the make command (make -f Make_ex) and then run
idl_ex in IDL. I hope it works with Fortran 90. Please let me know. Good
Luck.

Rose

-- ---idl_ex.pro

```
PRO idl_ex
; Define arrays
a_surface = DOUBLE(.19)
sza = DOUBLE(45)
vza= DOUBLE(55)
ra = DOUBLE(65)
nbcloud=14
tau_c = DBLARR(nbcloud)
reflectivity = DBLARR(nbcloud)
print,'before call to external '
result_rtau = CALL_EXTERNAL('exc.so','exc',a_surface,sza,vza,ra, $
tau_c,reflectivity)
print,'Returned value from rtau: ',result_rtau
print,'-----Result of CALL_EXTERNAL-----'
print,'tau ', ' reflectivity'
```

```

FOR k=0,nbcloud-1 DO BEGIN
print,tau_c[k],reflectivity[k]
ENDFOR

```

```

end

```

```

---exc.c -----

```

```

/* This program serves as a wrapper for the fortran program exf.f */

```

```

#include <stdio.h>

```

```

#include <stdlib.h>

```

```

#include <math.h>

```

```

long exc(int argc,void *argv[])

```

```

{
extern void exf_();

```

```

static int i,j;

```

```

static double tau_c[14],reflectivity[14] ;

```

```

static double a_surface, sza_deg, vza_deg, ra_deg;

```

```

a_surface = *(double *)argv[0];

```

```

sza_deg= *(double *)argv[1];

```

```

vza_deg= *(double *)argv[2];

```

```

ra_deg= *(double *)argv[3];

```

```

/* Check to see if values correspond with IDL input */

```

```

fprintf(stderr,"%lf %lf %lf %lf \n",a_surface,sza_deg,vza_deg,ra_deg);

```

```

fprintf(stderr," -----Call fortran program----- \n");

```

```

    exf_(&a_surface,&sza_deg,&vza_deg,&ra_deg,tau_c,reflectivity);

```

```

fprintf(stderr,"----- Returned balues from Fortran: ----- \n");

```

```

for (i=0;i<14;++i)

```

```

{
    *((double *)argv[4] + i) = tau_c[i] ;
    *((double *)argv[5] + i) = reflectivity[i] ;
    fprintf(stderr," %lf %lf \n",tau_c[i] ,reflectivity[i]);
}

```

```

/* Send some strange number as a check to IDL */

```

```

return(33);

```

```

}

```

```

---exf.f -----

```

```

subroutine exf(albedo,solzen,satzen,phi,tau_c,
6    reflectivity)

```

```

integer i

```

```

double precision tau_c(14),reflectivity(14)

```

```

double precision albedo,solzen,satzen,phi

```

```

C Check to see if input values passed correctly

```

```

write(*,*)albedo,solzen,satzen,phi

```

C Calculate new values to be returned to C program and IDL

```
do 20 i=1,14
  tau_c(i) = i*2.0
  reflectivity(i) = i/10.
20 continue
return
end
```

---Make_ex -----

```
LDFLAGS= -IF77 -IU77 -II77 -lisam -lm OBJS = exc.o exf.o exc : $(OBJS) ld
-shared -o exc.so $(OBJS) $(LDFLAGS) exc.o : exc.c cc -c -KPIC exc.c -o
exc.o exf.o : exf.f f77 -c -KPIC exf.f -o exf.o
```

--

-----== Posted via Deja News, The Discussion Network ==-----

<http://www.dejanews.com/> Search, Read, Discuss, or Start Your Own
