Subject: CALL_EXTERNAL - Fortran Troubles
Posted by Jim Brown on Tue, 07 Nov 1995 08:00:00 GMT
View Forum Message <> Reply to Message

Hello all:

Please forgive me if somebody else has posed a similar question, but I am stumped! I have obtained some code from a colleague that makes use of the CALL_EXTERNAL function to execute Fortran code. I have been able to understand hs example and make certain routines work, but I have hit a dead end.

Here is a segment of some Fortran code that I can get to work with CALL_EXTERNAL:

```
C********************************************************** **************
      subroutine psi( argc, argv )
C
C this section is a driver for IDL - callable fortran code.
C       this code is based on code obtained from andy loughe.
C
      Real argc, argv(*)              ! Argc and argv are reals

      Integer num_expected
      parameter (num_expected  =  3)    ! Number of parameters
                                ! expected by programmer

C  Obtain # of arguements passed-in and check that this is correct.
      j = LOC(argc)
      if (j .ne. num_expected) then
        return
      endif


C  Call subroutine with two parameters passed in.
      call calc_psi( %val(argv(1)), %val(argv(2)), %val(argv(3)) )

      return
      end
C********************************************************** **************
C********************************************************** **************
      subroutine calc_psi( u1, v1, psi )
C
C      program psi
C
      real u1(145,73),v1(145,73),psi(145,73)

.... code goes here ...
```

```
C************************************************************ **************
```

Now I can get the above code (and other code for that matter) to work if I am only passing back arrays containing REALS. What I would like to do is to have a routine pass back both reals and a string such as the following shows:

```
 ************************************************************ *************
      subroutine psi( argc, argv )
C
C this section is a driver for IDL - callable fortran code.
C       this code is based on code obtained from andy loughe.
C
      Real argc, argv(*)              ! Argc and argv are reals

      Integer num_expected
      parameter (num_expected  =  4)    ! Number of parameters
                                ! expected by programmer

C  Obtain # of arguements passed-in and check that this is correct.
      j = LOC(argc)
      if (j .ne. num_expected) then
         return
      endif


C  Call subroutine with two parameters passed in.
      call calc_psi( %val(argv(1)), %val(argv(2)), %val(argv(3)),
     +               %val(argv(4)) )

      return
      end
 C************************************************************ **************
 C************************************************************ **************
      subroutine calc_psi( u1, v1, units, psi )
C
C      program psi
C
      real u1(145,73),v1(145,73),psi(145,73)
      character*6 units

.... code goes here ...

 C************************************************************ **************
```

Note, the only change was the addition of the string variable, units. Everything compiles great, but I cannot get the value of the variable

named units passed back into my IDL routine.

If anyone has successfully passed both real arrays and character strings back into a IDL routine from a FORTRAN subroutine, I would appreciate some tips on how this can be accomplished.

Many thanks in advance for any hints on this topic.

Jim

==========
James T. Brown
CIRES - Cooperative Institute for Research in Environmental Sciences
University of Colorado

email: jtb@cdc.noaa.gov