Subject: Widgets break CALL_EXT?
Posted by mallozzi on Thu, 01 Feb 1996 08:00:00 GMT
View Forum Message <> Reply to Message

Hi all,

I believe that there is a problem with CALL_EXTERNAL under some operating systems. For a FORTRAN subroutine that requires termial I/O, an error is generated if a widget is created before the call to CALL_EXTERNAL. The READ statement in the FORTRAN subroutine generates an error. However, if no widgets are used, the call succeeds. I did not test with a C routine, but would be interested to see if it produces the same error.

I tested this on four different configurations:

OpenVMS AXP, v6.2
IDL v3.6
WORKS CORRECTLY

SunOS v4.1A.3
IDL v4.0.1, IDL v3.6
WORKS CORRECTLY

SGI IRIX v5.3
IDL v4.0.1
** DOES NOT WORK CORRECTLY

SunOS v5.4 (Solaris)
IDL v4.0.1, IDLv3.6
** DOES NOT WORK CORRECTLY

Included below are the .pro and .for files I used for the test.
If anyone is interested, I can also supply the Makefile I used.

Robert Mallozzi
mallozzi@ssl.msfc.nasa.gov

- - - - - - - - - - - - - BEGIN TEST_CALL_EXTERNAL.PRO - - - - - - - - - - - - -

```
;       This is a simple subroutine that demonstrates an error with
;       IDL's CALL_EXTERNAL on some operating systems.  Normally, the
;       call will work, but if a widget is created first (and then perhaps
;       destroyed), and then the CALL_EXTERNAL is executed, the READ
```

```
;       statement in the FORTRAN subroutine generates
;       an error, and the subroutine does not wait for user input.

ANS = 'NO'
READ, 'Create a widget (Y/N)? ', ANS

IF (STRUPCASE(STRMID(ANS, 0, 1)) EQ 'Y') THEN BEGIN
   BASE = WIDGET_BASE()
   BUTTON = WIDGET_BUTTON(BASE)
   WIDGET_CONTROL, /DESTROY, BASE
;  DIALOG = WIDGET_MESSAGE('Test widget.', /INFO, TITLE = ' ')
ENDIF

PRINT
PRINT, 'Executing CALL_EXTERNAL.'
PRINT
STATUS = CALL_EXTERNAL('TEST_SHARE', 'test_call_external_')
PRINT
PRINT, 'CALL_EXTERNAL finished.'



END
- - - - - - - - - - - - - END TEST_CALL_EXTERNAL.PRO - - - - - - - - - - - - -


- - - - - - - - - - - - - BEGIN TEST_CALL_EXTERNAL.FOR - - - - - - - - - - - - -


 SUBROUTINE  TEST_CALL_EXTERNAL
C
C       This is a simple subroutine that requires terminal I/O,
C       used to demonstrate an error with IDL's CALL_EXTERNAL.
C       Link this subroutine into a shared object and call through
C       IDL.  Normally, the call will work, but if a widget is
C       created first in IDL (and then perhaps destroyed), and then
C       this subroutine is called, the READ statement generates
C       an error, and the subroutine does not wait for user input.
C
C
 CHARACTER*10 INPUT
C
       WRITE (6, *) '+++++ START FORTRAN TEST SUBROUTINE +++++'
C
       WRITE (6, *) 'Enter INPUT'
 READ (5, 8020, ERR=8030, END=9000) INPUT
 8020   FORMAT (A)
       GOTO 9999
```

```
C
 8030   WRITE (6, *) 'TOOK ERR= BRANCH'
        GOTO 9999
C
 9000   WRITE (6, *) 'TOOK END= BRANCH'
C
 9999   WRITE (6, *) 'INPUT = ', INPUT
        WRITE (6, *) '----- END OF FORTRAN TEST SUBROUTINE -----'
        END
- - - - - - - - - - - - - - - END TEST_CALL_EXTERNAL.FOR - - - - - - - - - - - - -
```