Subject: Re: IDL: from Sun to MsWindows
Posted by thompson on Thu, 03 Jun 1993 18:14:25 GMT
View Forum Message <> Reply to Message

knipp@ipi.uni-hannover.de (K. Knipp) writes:

> Hi,
> I've got two questions concerning the adaption of IDL-routines written on SunOs
> for MsWindows:

> - does anyone have a list of general hints to this problem, like
> ; avoid system-calls
> ; short (8 characters) routine-names
> ; ... ?

You can have routine names that are longer than eight characters. You just make the DOS name of the file the first eight characters of the routine. It is better not to have two or more routines which start with the first eight characters--however, you can deal with this too by simply concatenating the procedures with names that share the first eight characters into a single file. (If one procedure calls another, and their names start with the same eight characters, then the procedure being called should be first in the concatenated file). I will post a procedure file (CONCAT4DOS.PRO) that will automatically concatenate procedure files into a form that can then be simply copied to MsWindows/DOS machine.

I haven't had any luck so far adapting software that works through spawning OS commands, but maybe that will be easier in a future release of IDL for Windows.

Another thing to avoid is software that using all three mouse buttons (for example the routine ZOOM in the standard IDL user's library). Some PCs seem to be capable of supporting this, and some don't. It does appear, however, that one always has the left (!ERR=1) and right (!ERR=4) buttons available.

> - how can use non-printable characters under MsWindows, p.e.:
> ; print,'\007' ; (bell)
> ; print,'\x08' ; (move cursor left) ?
> MsWindows just prints ths string on the screen.

I haven't worried about ringing the bell, but I have come up with a solution for software that manipulates screen output. For example, consider a routine that continuously prints out the position of the cursor to the screen. (RDPIX in the standard IDL user's library is an example of such a routine. On Unix and VMS workstations this is accomplished with formatted output statements. However, this doesn't seem to work with IDL for Windows. The workaround I came up, and which I'm happy with, is to use a text widget instead of writing directly to the output log. I will also post an example (CRS.PRO) of how this

is done. (Note that in this example widgets are used without ever calling XMANAGER.)

Bill Thompson

Subject: Re: IDL: from Sun to MsWindows
Posted by thompson on Thu, 03 Jun 1993 18:31:36 GMT
View Forum Message <> Reply to Message

```
PRO CONCAT4DOS
;+
; NAME:
; CONCAT4DOS
; PURPOSE:
; Concatenates IDL procedure files together into a form suitable for
; copying to a DOS machine.
; CALLING SEQUENCE:
; CD, directory ;(go to desired directory)
; CONCAT4DOS
; PARAMETERS:
; None.
; OPTIONAL KEYWORD PARAMETERS:
; None.
; COMMON BLOCKS:
; None.
; SIDE EFFECTS:
; A "dos" subdirectory is created.  On VMS machines, a temporary command
; file called "CONCAT4DOS.COM" is created and then destroyed.
; RESTRICTIONS:
; None.
; PROCEDURE:
; All the .PRO files in the current directory are copied into a special
; "dos" subdirectory, with the following changes made:
;
;  1.  All filenames are truncated to eight characters.
;  2.  All procedure files with names beginning with the same
;      first eight characters are concatenated together into a
;      single file.
;
; MODIFICATION HISTORY:
; William Thompson, August 1992.
;-
;
 ON_ERROR,2
;
;  First make sure there are procedure files in the current directory.
;
```

```
 FILES = FINDFILE('*.pro',COUNT=N_FILES)
 IF N_FILES EQ 0 THEN MESSAGE,'No procedure files found'
;
;  Next, look for an existing "dos" directory.  On VMS machines, open up a
;  command file to store all subsequent commands.  All the commands will then
;  be executed at the end with a single spawn.
;
 IF !VERSION.OS EQ 'vms' THEN BEGIN
  OPENW,UNIT,'CONCAT4DOS.COM',/GET_LUN
  PRINTF,UNIT,'$ SET VERIFY'
  DOSDIR = 'DOS.DIR'
 END ELSE DOSDIR = 'dos'
 DOSDIRFILE = FINDFILE(DOSDIR,COUNT=N_FOUND)
;
;  If an existing directory was found, then warn the user that all the ".pro"
;  files in that subdirectory will be deleted, and ask if the user wants to
;  continue.  If yes, then delete the files.
;
 IF N_FOUND NE 0 THEN BEGIN
  PRINT,'DOS directory already found'
  PRINT,'All .PRO files in the DOS directory will be deleted.'
  ASK,'Continue? ',ANSWER,'YN'
  IF ANSWER EQ 'Y' THEN BEGIN
   IF !VERSION.OS EQ 'vms' THEN BEGIN
    PRINTF,UNIT,'$ DELETE/NOLOG/NOCONFIRM ' + $
     '[.DOS]*.PRO;*'
   END ELSE BEGIN
    COMMAND = 'rm dos/*.pro'
    PRINT,'$ ' + COMMAND
    SPAWN,COMMAND
   ENDELSE
  END ELSE RETURN
;
;  Otherwise, create the subdirectory.
;
 END ELSE BEGIN
  IF !VERSION.OS EQ 'vms' THEN BEGIN
   PRINTF,UNIT,'$ CREATE/DIRECTORY [.DOS]'
  END ELSE BEGIN
   COMMAND = 'mkdir dos'
   PRINT,'$ '+COMMAND
   SPAWN,COMMAND
  ENDELSE
 ENDELSE
;
;  For each file, determine the eight character equivalent, and copy all files
;  beginning with those eight characters into a single file.
;
```

```
 LAST = ''
 FOR I=0,N_FILES-1 DO BEGIN
  FDECOMP,FILES(I),DISK,DIR,NAME,EXT,VER
  NAME8 = STRMID(NAME,0,8)
  IF NAME8 NE LAST THEN BEGIN
   IF STRLEN(NAME8) EQ 8 THEN NAME9 = NAME8 + '*' ELSE $
    NAME9 = NAME8
   IF !VERSION.OS EQ 'vms' THEN BEGIN
    PRINTF,UNIT,'$ COPY ' + NAME9 + $
     '.PRO [.DOS]' + NAME8 + '.PRO'
   END ELSE BEGIN
    COMMAND = 'cat ' + NAME9 + '.pro > dos/' + $
     NAME8 + '.pro'
    PRINT,'$ ' + COMMAND
    SPAWN,COMMAND
   ENDELSE
  ENDIF
  LAST = NAME8
 ENDFOR
;
;  On VMS machines, tell the command file to delete itself after processing,
;  and execute it.
;
 IF !VERSION.OS EQ 'vms' THEN BEGIN
  PRINTF,UNIT,'$ DELETE/NOLOG/NOCONFIRM CONCAT4DOS.COM;*'
  FREE_LUN,UNIT
  SPAWN,'@CONCAT4DOS.COM'
 ENDIF
;
 RETURN
 END
```

Subject: Re: IDL: from Sun to MsWindows
Posted by thompson on Thu, 03 Jun 1993 18:32:24 GMT
View Forum Message <> Reply to Message

```
PRO  CRS,X_VALUE,Y_VALUE,PRINT_SWITCH,CONTINUOUS=CONTINUOUS,FONT= FONT
;+
; NAME:
; CRS
; PURPOSE:
; This procedure uses the routine CURSOR to find the coordinates,
; expressed in data units, of a point selected with the cursor.
; CATEGORY:
; CALLING SEQUENCE:
; CRS  [, X_VALUE  [, Y_VALUE  [, PRINT_SWITCH ]]]
; OPTIONAL INPUT PARAMETERS:
```

```
; PRINT_SWITCH - Switch used to control printing the values of
;     X_VALUE, Y_VALUE to the screen.  If not passed,
;     then assumed 0 (no printing) unless no parameters
;     are passed, in which case 1 (printing) is assumed.
; OPTIONAL OUTPUT PARAMETERS:
; X_VALUE  - X position in data coordinates of cursor.
; Y_VALUE  - Y position in data coordinates of cursor.
; OPTIONAL KEYWORD PARAMETERS:
; CONTINUOUS - If set, then a continuously updated display of the
;     cursor X and Y positions are written to the screen.
;     On systems which support widgets, the text is
;     displayed in a special widget.
;
;     In continuous operation pressing either the left or
;     middle mouse button will print out the current
;     position on a fresh line on the terminal screen.
;     Pressing the right mouse button quits the program.
;     The PRINT_SWITCH parameter controls whether or not
;     the last cursor position is printed or not.
;
;     When CONTINUOUS is set, the PRINT_SWITCH variable is
;     ignored--the position is always printed to the
;     screen.
;
; FONT  = Font to use when displaying the CRS widget.  Only
;     meaningful when the graphics device supports widgets,
;     and CONTINUOUS is set.  If not passed, then the first
;     available 20 point font is used.
; COMMON BLOCKS:
; None.
; SIDE EFFECTS:
; Using the CONTINUOUS keyword on a device without a mouse or trackball
; may not allow the user to exit the program.
; RESTRICTIONS:
; Use of the CONTINUOUS keyword may not be supported on some more
; primitive graphics terminals.
; PROCEDURE:
; CURSOR is used to get cursor position.
; MODIFICATION HISTORY:
; William Thompson Applied Research Corporation
; September, 1987  8201 Corporate Drive
;    Landover, MD  20785
;
; William Thompson, 13 May 1993, added CONTINUOUS and FONT keywords.
; William Thompson, 1 June 1993, changed to ignore PRINT_SWITCH when
;  using in CONTINUOUS mode.
;-
;
```

```
 ON_ERROR, 2
;
;  Check that a plot has been made.
;
 IF (!X.S(1)*!Y.S(1) EQ 0) THEN MESSAGE,  $
  'Data coordinates not initialized'
;
;  Assign the default value of PRINT_SWITCH.
;
 IF N_PARAMS(0) LT 3 THEN PRINT_SWITCH = 0
 IF N_PARAMS(0) EQ 0 THEN PRINT_SWITCH = 1
;
;  If the CONTINUOUS keyword was set, then show a continuous display of cursor
;  positions.
;
 IF KEYWORD_SET(CONTINUOUS) THEN BEGIN
;
;  If the current graphics device supports widgets, then display the text in
;  a special text widget.
;
  IF HAVE_WIDGETS() THEN BEGIN
   TEST = EXECUTE("BASE = WIDGET_BASE(" + $
    "TITLE='Cursor Position',/ROW)")
   TEXT = ' '
   IF N_ELEMENTS(FONT) NE 1 THEN FONT = '*20'
   TEST = EXECUTE("LABEL = WIDGET_TEXT(" +  $
    "BASE,VALUE=TEXT,FONT=FONT, XSIZE=30)")
   WIDGET_CONTROL,BASE,/REALIZE
  ENDIF
;
;  Keep reading the cursor until the right button is pressed.
;
  CR = STRING("15B)
  !ERR = 0
  PRINT,'Press left or center mouse button for new output line.'
  PRINT,'... right mouse button to exit.'
;
  WHILE !ERR NE 4 DO BEGIN
   CURSOR,X_VALUE,Y_VALUE,2
   TEXT = STRTRIM(X_VALUE,2) + ', ' + STRTRIM(Y_VALUE,2)
;
;  If either the left or middle mouse button was pressed, then display a fresh
;  line on the terminal screen.
;
   IF (!ERR AND 3) NE 0 THEN BEGIN  ;New line?
    IF HAVE_WIDGETS() THEN BEGIN
     PRINT,' Position:  ' + TEXT
    END ELSE BEGIN
```

```
     PRINT,FORMAT="($,A)",STRING("12B)
    ENDELSE
    WHILE (!ERR NE 0) DO BEGIN
     WAIT,0.1
     CURSOR,X_VALUE,Y_VALUE,0
    ENDWHILE
   ENDIF
;
;  Display the current cursor position.
;
   IF HAVE_WIDGETS() THEN BEGIN
    WIDGET_CONTROL, LABEL, SET_VALUE=TEXT
   END ELSE BEGIN
    PRINT,FORMAT="($,' Position:  ',A,'     ',A)",$
     TEXT,CR
   ENDELSE
  ENDWHILE
;
;  Close the continuous display.
;
  IF HAVE_WIDGETS() THEN BEGIN
   WIDGET_CONTROL, /DESTROY, BASE
   PRINT,' Position:  ' + TEXT
  END ELSE BEGIN
   PRINT,FORMAT="(/)"
  ENDELSE
;
;  If CONTINUOUS is not set, then simply get a single cursor position from the
;  screen.
;
 END ELSE BEGIN
  CURSOR,X_VALUE,Y_VALUE
;  IF (!D.NAME EQ 'SUN') OR (!D.NAME EQ 'X') THEN TVCRS,/HIDE
;
;  If requested, print the cursor position.
;
  IF PRINT_SWITCH NE 0 THEN BEGIN
   IF !D.NAME EQ 'REGIS' THEN PRINT,STRING(27B) + '[H'
   PRINT,' Position:  ' + STRTRIM(X_VALUE,2) + ', ' + $
    STRTRIM(Y_VALUE,2) + '     '
  ENDIF
 ENDELSE
;
 RETURN
 END
```

Subject: Re: IDL: from Sun to MsWindows
Posted by thompson on Thu, 03 Jun 1993 20:08:52 GMT
View Forum Message <> Reply to Message

The following routine may also be of use to those worrying about moving for
Unix/VMS to Microsoft Windows.

Bill Thompson

```
 PRO FLAG_LONG_NAMES
;+
; NAME:
; FLAG_LONG_NAMES
; PURPOSE:
; Flags pairs of IDL procedure names which have the same first eight
; characters.  These would appear to be the same file on DOS machines.
; CALLING SEQUENCE:
; CD, directory ;(go to desired directory)
; FLAG_LONG_NAMES
; PARAMETERS:
; None.
; OPTIONAL KEYWORD PARAMETERS:
; None.
; COMMON BLOCKS:
; None.
; SIDE EFFECTS:
; None.
; RESTRICTIONS:
; None.
; PROCEDURE:
; The names of each set of .PRO files with the same first eight
; characters are printed to the screen.
; MODIFICATION HISTORY:
; William Thompson, January 1993.
;-
;
 ON_ERROR,2
;
;  First make sure there are procedure files in the current directory.
;
 FILES = FINDFILE('*.pro',COUNT=N_FILES)
 IF N_FILES EQ 0 THEN MESSAGE,'No procedure files found'
;
;  For each file, determine the eight character equivalent, and look for
;  duplicates.
;
 LAST = ''
 DUPS = ''
 FOR I=0,N_FILES-1 DO BEGIN
```

```
  FDECOMP,FILES(I),DISK,DIR,NAME,EXT,VER
  NAME8 = STRMID(NAME,0,8)
  IF NAME8 NE LAST THEN BEGIN
   IF N_ELEMENTS(DUPS) GT 1 THEN $
    FOR J=0,N_ELEMENTS(DUPS)-1 DO PRINT,DUPS(J)
   DUPS = NAME
  END ELSE BEGIN
   DUPS = [DUPS,NAME]
  ENDELSE
  LAST = NAME8
 ENDFOR
;
 RETURN
 END
```

Subject: Re: IDL: from Sun to MsWindows
Posted by hadfield_m on Fri, 04 Jun 1993 01:12:20 GMT
View Forum Message <> Reply to Message

Bill Thompson writes:

>
> Another thing to avoid is software that using all three mouse buttons (for
> example the routine ZOOM in the standard IDL user's library). Some PCs seem to
> be capable of supporting this, and some don't. It does appear, however, that
> one always has the left (!ERR=1) and right (!ERR=4) buttons available.
>

I raised this with RSI (I have a Microsoft-pattern mouse) and got the following
response:

> From: RSI email support <support@rsinc.com>

> Subject: Re: Mouse in IDL for MS Windows

> DEAR MARK,

> I believe that you can overcome a missing center mouse button by pushing
> both buttons (right and left) at the same time. This should emulate the
> center button function.

> Anne Fegley

I must confess that I haven't actually tried it yet!

 ---------- ------
| Mark Hadfield hadfield@storm.greta.cri.nz |

| NIWA Oceanographic (Taihoro Nukurangi) |
| 310 Evans Bay Rd, Greta Point Telephone: (+64-4) 386-1189 |
| PO Box 14-901, Kilbirnie Fax: (+64-4) 386-2153 |
| Wellington, New Zealand |
---------- ------

Subject: Re: IDL: from Sun to MsWindows
Posted by thompson on Fri, 04 Jun 1993 13:54:25 GMT
View Forum Message <> Reply to Message

hadfield_m@kosmos.wcc.govt.nz writes:

> Bill Thompson writes:

>>
>> Another thing to avoid is software that using all three mouse buttons (for
>> example the routine ZOOM in the standard IDL user's library). Some PCs seem to
>> be capable of supporting this, and some don't. It does appear, however, that
>> one always has the left (!ERR=1) and right (!ERR=4) buttons available.
>>

> I raised this with RSI (I have a Microsoft-pattern mouse) and got the following
> response:

>> From: RSI email support <support@rsinc.com>

>> Subject: Re: Mouse in IDL for MS Windows

>> DEAR MARK,

>> I believe that you can overcome a missing center mouse button by pushing
>> both buttons (right and left) at the same time. This should emulate the
>> center button function.

>> Anne Fegley

> I must confess that I haven't actually tried it yet!

> ---------- ------
> | Mark Hadfield hadfield@storm.greta.cri.nz |
> | NIWA Oceanographic (Taihoro Nukurangi) |
> | 310 Evans Bay Rd, Greta Point Telephone: (+64-4) 386-1189 |
> | PO Box 14-901, Kilbirnie Fax: (+64-4) 386-2153 |
> | Wellington, New Zealand |
> ---------- ------

I did, but that doesn't seem to always work either. (What happens when IDL

starts running on a MacIntosh which only has a single mouse button--maybe IDL will start to understand double-clicks?)

Bill Thompson
