Subject: Re: Different Platforms
Posted by Liam Gumley on Wed, 19 Nov 1997 08:00:00 GMT
View Forum Message <> Reply to Message

Neil Winrow wrote:

> I have written a number of widget programs, which are visually correct
> on my PC, however when they are run on the silicon graphics machines the
> layout starts to go terribly wrong. The character sizes are wrong, and
> the labelling carried out using the 'XYOUTS' call is all wrong. The
> whole window sizing falls down. The programs are going to be used on
> PC's, silicon graphics, and MAC's. Could anyone offer me a few pointers
> on how to correct these problems to run the programs on the different

(1) If your development platform is a PC, then make sure that once a day you test the program on a Unix box, and a Mac. Unless you do this, it is easy in your program design to go down a path from which there is no return.

(2) As David Fanning suggested, having the commands

```
Device, Set_Character_Size = [ 6, 9 ]
Widget_Control, Default_Font = '7x13'
```

in your IDL startup file will guarantee consistent graphics font and widget font sizes on all *Unix* platforms. However, these commands are ignored on PCs. You should try David's STR_SIZE program.

(3) You may wish to look into using *hardware* fonts for graphics. I've been experimenting with them in Unix, and getting pretty good results. You can see the list of available hardware fonts in IDL by using the command

```
Device, Get_Fontnames = Fontnames
```

which returns a string array of all the font names defined on your system (Unix or PC or Mac). You can then select a font that looks consistent on all platforms (say a 14 point Helvetica font), and make it the default graphics and widget font by the commands

```
device, font = name    ; set graphics font
!p.font = 0    ; use hardware fonts for graphics instead of vector fonts
widget_control, default_font = name    ; set the default widget font
```

The major drawback to using hardware fonts is that they do not rotate automatically, e.g.

```
plot, indgen(10), xtitle = 'X AXIS', ytitle = 'Y AXIS'
```

when using hardware fonts will give you the Y title plotted vertically, but not rotated. You can get around this by creating the Y axis title in a pixmap, reading an array of byte data from the pixmap, rotating the array using ROTATE, and TVing the rotated array next to your Y axis. This takes a bit of messing around, but the resulting graphics plots look *much* more professional than the default vector fonts.

(4) When creating widget programs, be very careful about using XSIZE and YSIZE keywords. I try to use them only for WIDGET_DRAW, WIDGET_LABEL, and WIDGET_BUTTON sizing.

(5) Rely on the ROW, COLUMN, and alignment keywords when creating widget bases to position your widgets.

Cheers,
Liam.
PS I've attached a Unix hardware font selection routine below - I'll be modifying it soon to work on PC and Mac.

```
PRO SELECT_FONT, HELVETICA = HELVETICA, TIMES = TIMES, $
  PALATINO = PALATINO, COURIER = COURIER, BOLD = BOLD, ITALIC = ITALIC, $
  SIZE = SIZE, NAME = NAME

;+
; Purpose:
;    Select a Unix hardware font for IDL graphics.
;
; Usage:
;    SELECT_FONT
;
; Input:
;    None required.
;
; Optional Keywords:
;    /HELVETICA    Select a Helvetica font (default)
;    /TIMES        Select a Times font
;    /PALATINO     Select a Palatino font
;    /COURIER      Select a Courier font
;    /BOLD         Select a bold font (default is no bold)
;    /ITALIC       Select and italic font (default is no italics)
;    SIZE          If set to a named variable, sets the font size
(default=12)
;    NAME          If set to a named variable, returns the font name
selected
;
; Revised:
```

```
;    17-OCT-1997 Liam Gumley, CIMSS/SSEC
;    Created
;
; Notes:
;    (1) This procedure currently only works on Unix IDL platforms.
;    (2) The NAME value returned by SELECT_FONT can be used to set the
;        default widget font by the command
WIDGET_CONTROL,DEFAULT_FONT=NAME
;
; Example:
; !P.MULTI=[0,1,2,0,0]
; PLOT,INDGEN(10)
; SELECT_FONT,/BOLD
; PLOT,INDGEN(10)
;-

;- this version is only for Unix at the moment

if !version.os_family ne 'unix' then begin
  message, /continue, 'Only works on Unix at the moment'
  return
endif

;- check keyword flags

if not keyword_set( helvetica ) then helvetica = 0
if not keyword_set( times ) then times = 0
if not keyword_set( palatino ) then palatino = 0
if not keyword_set( courier ) then courier = 0
if not keyword_set( bold ) then bold = 0
if not keyword_set( italic ) then italic = 0

;- check keyword values

if n_elements( size ) eq 0 then size = 12

;- set keyword return values

name = ''

;- create font search string

case 1 of
  helvetica : search = '*helvetica*'
  times     : search = '*times*'
  palatino  : search = '*palatino*'
  courier   : search = '*courier*'
  else      : search = '*helvetica*'
```

```
endcase

if bold then begin
  search = search + 'bold-'
endif else begin
  search = search + 'medium-'
endelse

if italic then begin
  search = search + 'o-normal*'
endif else begin
  search = search + 'r-normal*'
endelse

;- open a graphics window

window, /free, /pixmap

;- get list of font names matching search string

device, font = search, get_fontnames = fontnames

;- find a font size that matches

fontstring = '--' + strcompress( long( size > 8 ), /remove_all ) + '-'
index = strpos( fontnames, fontstring )
loc = where( index ne -1, count )

;- use font if it was found, or else set graphics font size

if count ge 1 then begin
  name = fontnames( loc(0) )
  device, font = name
  !p.font = 0
endif else begin
  message, /continue, 'Requested font was not found - using graphics font
instead'
endelse

;- close graphics window

wdelete, !d.window

end
```