Subject: Re: Different Platforms
Posted by rivers on Sat, 22 Nov 1997 08:00:00 GMT
View Forum Message <> Reply to Message

In article <3473138C.B39FFA08@ssec.wisc.edu>, Liam Gumley <Liam.Gumley@ssec.wisc.edu> writes:
> 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

For widgets I use the following methods:

- Use a function which returns font names in a system-dependent manner.
```
   default_font = get_font_name(/SMALL, /HELVETICA)
   button_font = get_font_name(/LARGE, /HELVETICA, /BOLD)
   label_font = get_font_name(/MEDIUM, /HELVETICA)

  widget_control, default_font=default_font
  Use button_font and label_font when creating buttons and labels, etc.
```
  My get_font_name.pro is attached. It works on Unix, VMS and PC. Have not
  had a Mac to try it on.

- When laying out widgets of different types (e.g. buttons, labels and
  droplists) which need to line up in columns or rows do the following:

   1) Create dummy widgets with the same commands you will be using for your
      real widgets before you call WIDGET_CONTROL, /REALIZE (that way you won't
      see them).
   2) Use geometry = widget_control(id, /geometry) to determine the size of
      each widget in pixels (i.e. geometry.scr_xsize and geometry.scr_ysize)
   3) Step 2 will tell you which widget is the biggest.
   4) Now create your real widgets, using the SCR_XSIZE and SCR_YSIZE keywords
      to explicitly set the size of all of the widgets to the size of the
      biggest one.

This has worked fine for me on PCs, VMS and Unix.

Here is an example from one program:
```
;************************************************************ *
   row = widget_base(base, /row, /frame)
   ; Determine height for all of the widgets in this row
```

```
    dummy = widget_text(base, xsize=6)
    geometry = widget_info(dummy, /geometry)
    widget_control, dummy, /destroy
    scr_ysize = geometry.scr_ysize
    scr_xsize = geometry.scr_xsize

    col = widget_base(row, /column)
    t = widget_label(col, value='ROI', font=self.fonts.label)
    for i=0, nrois-1 do begin
       t = widget_label(col, value=strtrim(string(i),2), scr_ysize=scr_ysize)
    endfor

    col = widget_base(row, /column)
    t = widget_label(col, value='Use?', font=self.fonts.label)
    for i=0, nrois-1 do begin
       cal.widgets.use_flag[i] = $
          widget_droplist(col, value=['No','Yes'], scr_ysize=scr_ysize)
       widget_control, cal.widgets.use_flag[i], $
                set_droplist_select=cal.roi[i].use
    endfor
 ;************************************************************ *
```

Here is get_font_name.pro

```
function get_font_name, $
       helvetica=helvetica, times=times, courier=courier, $
       tiny=tiny, small=small, medium=medium, large=large, huge=huge, $
       size=size, $
       bold=bold, italic=italic, $
       dpi75=dpi75, dpi100=dpi100

; Returns the name of the font with the specified characteristics

if (!version.os_family eq 'Windows') then begin
font = ''
if keyword_set(helvetica) then font = font + 'Helvetica' else $
if keyword_set(times)     then font = font + 'Times' else $
if keyword_set(courier)   then font = font + 'Courier' else $
                               font = font + 'MS San Serif'

if keyword_set(bold) then font = font + '*Bold'
if keyword_set(italic) then font = font + '*Italic'
if keyword_set(tiny)   then size=0
if keyword_set(small)  then size=1
if keyword_set(medium) then size=2
if keyword_set(large)  then size=3
if keyword_set(huge)   then size=4
if (n_elements(size) eq 0) then size=2
```

```
font_size_strings = ['12', '14', '16', '18', '20']
size = (size > 0) < (n_elements(font_size_strings)-1)
font = font + '*' + font_size_strings(size)
return, font

endif else if (!version.os_family eq 'Mac') then begin
   font='Helvetica'
   return, font
endif else begin

; VMS and Unix
font = '-adobe-'
if keyword_set(helvetica) then font = font + 'helvetica-' else $
if keyword_set(times)     then font = font + 'times-' else $
if keyword_set(courier)   then font = font + 'courier-' else $
                               font = font + 'helvetica-'

if keyword_set(bold) then font = font + 'bold-' else font = font + 'medium-'
if keyword_set(italic) then font = font + 'o-' else font = font + 'r-'
font = font + 'normal--*-'

if keyword_set(tiny)   then size=0
if keyword_set(small)  then size=1
if keyword_set(medium) then size=2
if keyword_set(large)  then size=3
if keyword_set(huge)   then size=4
if (n_elements(size) eq 0) then size=2
font_size_strings = ['80-', '100-', '120-', '140-', '180-']
size = (size > 0) < (n_elements(font_size_strings)-1)
font = font + font_size_strings(size)

if keyword_set(dpi100) then font = font + '100-100-' else $
if keyword_set(dpi75) then font = font + '75-75-' else $
                           font = font + '*-*-'
font = font + '*-*-iso8859-1'
return, font
endelse

end
```