

---

Subject: Re: GUI size on different screens  
Posted by [Harvey Rarback](#) on Fri, 01 Dec 2006 00:07:38 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

"David Fanning" <[news@dfanning.com](mailto:news@dfanning.com)> wrote in message  
news:[MPG.1fd7eedbd3e10bde989e01@news.frii.com](mailto:MPG.1fd7eedbd3e10bde989e01@news.frii.com)...  
> Harvey Rarback writes:  
>  
>> And now for the totally anal:  
>>  
>> I like to create resizable windows with resizable fonts so that the  
>> initial  
>> TLBs fit on the screen, but can be resized to suit the user's preference.  
>> This scheme works as follows.  
>  
> Harvey, where in the world have you been!? I can't believe  
> it takes resizeable fonts to draw you out of the shadows. :-)  
>  
> By all means, let's see that GET\_FONTS function. That's  
> the one we are ALL missing!  
>  
> I'd be delighted to host this if it is something that  
> can be made available on my web page. I think we are  
> now only one or two tools away from cracking this nut.  
> (Although Dick and Andrew are still discussing a two  
> pixel discrepancy in screen size that none of us can  
> account for. Andrew calls them pixels, and I think he  
> may be right.)

This is so trivial that I am almost embarrassed to display the code here,  
but you asked for it. Its origin is lost in the mists of time, but I  
believe Mark Rivers was responsible for this. We didn't document back then  
;-)

```
function get_fonts, font_size=font_size
compile_opt idl2

if n_elements( font_size ) eq 0 then font_size = 2
fonts = {application_fonts,
          $                               default:   get_font_name( /helvetica,    size=font_size ),  $
          $                               menu:      get_font_name( /bold,        size=font_size ),  $
          $                               big_menu:   get_font_name( /bold,        size=font_size+1 ), $
          $                               label:     get_font_name( /bold, /italic, size=font_size ),  $
          $                               big_label:  get_font_name( /bold, /italic, size=font_size+1 ), $
          $                               button:    get_font_name(           size=font_size ),  $
          $                               big_button: get_font_name(           size=font_size+1 ), $
          $                               text:      get_font_name(           size=font_size ),  $
          $                               big_text:  get_font_name(           size=font_size+1 ), $
```

```

message:    get_font_name(           size=font_size-1 ), $  

symbol:     get_font_name( /symbol,   size=font_size ), $  

big_symbol: get_font_name( /symbol,   size=font_size+1 ) $  

}  
  

return, fonts  

end  
  

function get_font_name,                      $  

    helvetica=helvetica, times=times, courier=courier, symbol=symbol, $  

    tiny=tiny, small=small, medium=medium, large=large, huge=huge,   $  

    giant=giant,                      $  

    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 New' else $
    if keyword_set(symbol)  then font = font + 'Symbol' 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']
    font_size_strings = ['8', '12', '16', '20', '24']
    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 ; Assume Motif
    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   $
    if keyword_set(symbol)  then font = font + 'symbol-' 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 keyword_set(giant) then size=5
if (n_elements(size) eq 0) then size=2
font_size_strings = ['80-', '100-', '120-', '140-', '180-', '240-']
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 + '*-*-*'
if keyword_set(symbol) then font = font + '*-*-*-*' else $
    font = font + '*-*-*iso8859-1'

return, font
endelse

end

```

---