Subject: Screen Printing
Posted by your name on Wed, 23 Jun 1999 07:00:00 GMT
View Forum Message <> Reply to Message

It's me again (I'm not really "enter your name" - see below).

I previously posted a request on obtaining a screen dump of several draw widgets. This was answered by David Foster, complete with an IDL routine to print a window.

Unfortunately, David, this routine only works for a single draw widget window and I have several such widgets all required on one sheet of A4!!!

Perhaps the following code will give you an insight into the method I currently use albeit not very elegant.

```
PRO Top_screen
!P.BACKGROUND = 255                     ; Black on white
!P.COLOR = 0
DEVICE, GET_SCREEN_SIZE = Disp_scr_size         ; Get the current
screen size
Disp_widget_size = Disp_scr_size * 0.98         ; Allow room for
scroll bars
Base = LONARR(4)
Picture = LONARR(4)
Window = LONARR(4)
Offsets = [[1, 1], $                    ; Top LH
corner
        [Disp_scr_size[0]/2, 1], $              ; Top RH
corner
        [1, Disp_scr_size[1]/2], $              ; Bottom LH
corner
        [Disp_scr_size[0]/2, Disp_scr_size[1]/2]]       ; Bottom RH
corner

; Create 4 base widgets each containing a scrollable draw widget.
; These are top-level bases to allow users to move, size, minimize
; and maximize each one separately.

FOR Screen = 0, 3 DO BEGIN
  Base[Screen] = WIDGET_BASE(XSIZE = Disp_widget_size[0]/2, $
    YSIZE = Disp_widget_size[1]/2, $
    XOFFSET = Offsets[0,Screen], $
    YOFFSET = Offsets[1,Screen], $
    TITLE = 'Window ' + STRING(Screen))
  Picture[Screen] = WIDGET_DRAW(Base[Screen], $
    X_SCROLL_SIZE = Disp_widget_size[0]/2, $
```

```
    Y_SCROLL_SIZE = Disp_widget_size[1]/2, $
    XSIZE = Disp_widget_size[0]/2, $
    YSIZE = Disp_widget_size[1]/2, /SCROLL)
  WIDGET_CONTROL, Base[Screen], /REALIZE     ; Let's see it
  WIDGET_CONTROL, Picture[Screen], $
    GET_VALUE = Window_num                   ; We need window number
  Window[Screen] = Window_num                ; Save it for later
ENDFOR

; Let's plot something really unuseful in each of the widgets

WSET, Window[0]
PLOT, [1, 2], [1, 2], TITLE = 'Plot 0'
WSET, Window[1]
PLOT, [10, 20], [10, 20], TITLE = 'Plot 1'
WSET, Window[2]
PLOT, [100, 200], [100, 200], TITLE = 'Plot 2'
WSET, Window[3]
PLOT, [1000, 2000], [1000, 2000], TITLE = 'Plot 3'

; Now user requests a screen dump (this would normally be done in the
; event handler attached to a "Screen Dump" button, but is placed here
; for convenience).

Old_display = !D.NAME                        ; Remember what we were
Scale = 20                                   ; Output scaling
SET_PLOT, 'PS'                               ; Could also be other
formats
DEVICE, FILENAME = 'User_output.lis', SCALE_FACTOR = Scale, /LANDSCAPE

HELP, /DEVICE, OUTPUT = Device_data          ; Need this for plot size
info
READS, STRMID(Device_data[5],17,11), Xmax, Ymax      ;Extract the sizes
Ymax = Ymax * Scale                          ; Allow for scaling

FOR Screen = 0, 3 DO BEGIN                   ; Loop round each plot
  SET_PLOT, Old_display                      ; Normal display device
  WSET, Window[Screen]                       ; Change to window of
interest
  Image = TVRD()                             ; Get its image...
  Details = WIDGET_INFO(Base[Screen], /GEOMETRY)  ; ... and its info

; We need information about the base, because the user may have
; moved and/or resized it since it was drawn.

  SET_PLOT, 'PS'                             ; Change to output file
  TV, Image, $                               ; and draw the image...
    Details.XOFFSET, $                       ; ...with these offsets...
```

```
    Ymax - Details.YOFFSET, $
    XSIZE = Details.XSIZE, $              ; ...and these sizes
    YSIZE = Details.YSIZE
ENDFOR
DEVICE, /CLOSE_FILE                       ; Close file to enable
printing
SET_PLOT, Old_display                     ; Go back to the normal
display
Command = 'xxxxxxxxxxxxx'                 ; system dependant print
command
SPAWN, Command                            ; Do the print
WIDGET_CONTROL, /RESET                          ; Clean up
END
```

Surely it should be possible for IDL to "dump" a whole screen to a file/printer.
After all, with RETAIN=2, IDL keeps a screen map in its memory why not use this?

If anybody has a better workaround for this, I'd be very grateful.

Regards,
Ian

!=====!=====!=====!=====!=====!=====!=====!=====!=====!
!   Ian Dean   Marconi Electronic Systems   !
! Ian.Dean@GECM.COM   Elettra Avenue, Waterlooville   !
!   +44(0)1705 260186   Hampshire, England PO7 7XS   !
!=====!=====!=====!=====!=====!=====!=====!=====!=====!=====!