Subject: Getting filename from PS device
Posted by breid on Mon, 03 Aug 1998 07:00:00 GMT
View Forum Message <> Reply to Message

Hi all.

How can I determine the filename that the PS device is using? I'd like to write a timestamp contining the output filename when I close the PS device, but for convenience, I would like the timestamp routine to be able to figure out the output filename without having to create a new variable.

Now I know that IDL knows what filename it's using, since the help, /device command will show it, but how can I access that filename? Is there a system variable that contains the filename?

--Bob

Robert J. Reid, Research Specialist | Imager for Mars Pathfinder /
breid@lpl.arizona.edu | Surface Stereo Imager & Robotic Arm
http://www.lpl.arizona.edu/~breid/ | Camera for Mars Polar Lander
---
Lunar and Planetary Lab, The University of Arizona, Tucson, AZ 85721 USA
telephone: 520-621-2499 | fax: 520-621-2994

Subject: Re: Getting filename from PS device
Posted by Martin Schultz on Wed, 12 Aug 1998 07:00:00 GMT
View Forum Message <> Reply to Message

Bob Reid wrote:

> How can I determine the filename that the PS device is using? I'd like
> to write a timestamp contining the output filename when I close the
> PS device, but for convenience, I would like the timestamp routine to be
> able to figure out the output filename without having to create a new
> variable.

... since Brian and David already answered the technical question, I simply attach a pair of routines that are designed to do exactly what you point out. You can pass an optional label together with the timestamp, and you can use these routines also to handle a window device. Also attached is strdate.pro which returns a formatted date string and is used by close_device. What I normally do is:

```
pro myprogram,ps=ps

  open_device,ps=ps,filename='thisfile.ps' [,...]

  ; call program to make plot or insert plotting commands

  close_device,/timestamp

return
end
```

Then 'myprogram' will produce screen output, whereas 'myprogram,/ps'
creates a postscript file. If you like to give your postscript file the
same name as your routine (in this case 'myprogram.ps'), check out
routine_name.pro from my webpage
  http://www-as.harvard.edu/people/staff/mgs/idl/

Hope this helps,
Martin.

--
 -- -------
Dr. Martin Schultz
Department for Earth&Planetary Sciences, Harvard University
109 Pierce Hall, 29 Oxford St., Cambridge, MA-02138, USA

phone: (617)-496-8318
fax : (617)-495-4551

e-mail: mgs@io.harvard.edu
Internet-homepage: http://www-as.harvard.edu/people/staff/mgs/
 -- -------

```
 ;------------------------------------------------------------ --
;+
; NAME:
;       CLOSE_DEVICE
;
; PURPOSE:
;       CLOSE_DEVICE closes the PostScript device and spawns
;       a print job to the printer specified by the user or
;       it can be used to close a graphics window.
;
; CATEGORY:
;       Input/Output
;
```

```
; CALLING SEQUENCE:
;       CLOSE_DEVICE [,OLD_DEVICE] [,keywords]
;
; INPUTS:
;       OLD_DEVICE (str) -> Name of device that shall become the active
;           plotting device. If omitted, "X", "WIN" or "MAC" will be
;           set depending on !VERSION.OS_FAMILY. Hence, specification of
;           OLD_DEVICE is only rarely needed.
;           This parameter works together with the OLD_DEVICE parameter
;           of OPEN_DEVICE which returns the device name before the
;           postscript device (or a graphics device) is opened.
;           The OLD_DEVICE parameter can be misused to set the output
;           device to anything! Therefore, it's probably safest to not
;           use it and stick with the operating system dependent default.
;
; KEYWORD PARAMETERS:
;       PRINTER    (str) -> name of the printer queue to send output to.
;           Default is 'none', i.e. the postscript file will only be
;           closed and can then be manually printed e.g. using the Unix
;           lpr command. Direct printing only works in Unix environments.
;
;       WINDOW -> window number to be closed (or -1 if current)
;
;       LABEL -> a string that contains an annotation for a postscript
;           plot (usually a user name and/or filename). The current
;           data and time will be appended if TIMESTAMP is set.
;           NOTE: TIMESTAMP will produce an annotation even if LABEL
;           is not provided. The annotation is only added to postscript
;           plots and only if the ps file is actually open.
;
;       /TIMESTAMP  -> add date and time to the LABEL on the plot.
;           If no LABEL is provided, the value of FILENAME will be used
;           or the filename of the current postscript file will be
;           added. The timestamp is only added to postscript plots.
;
;       _EXTRA -> any additional keywords to CLOSE_DEVICE will be
;           passed on to STRDATE which is used to format the date
;           and time string. Possible values are: /SHORT, /SORTABLE,
;           /GERMAN.
;
;       LCOLOR -> the color value for the LABEL (default 1).
;
;       LPOSITION -> the position of the LABEL in normalized coordinates
;           (a two-element vector with default [0.98,0.007]).
;
;       LSIZE -> the charcter size of the LABEL (default 0.7).
;
;       LALIGN -> the alignment of the LABEL (default 1).
```

```
;
;       FILENAME   (str) -> name of the PostScript file.
;            This keyword is now obsolete since the name of the postscript
;            file is determined automatically. It is kept as a surrogate
;            for LABEL to ensure compatibility with older implementations.
;
; OUTPUTS:
;       If postscript device is active, a *.ps file will be closed and
;       optionally sent to the printer.
;
; SUBROUTINES:
;
; REQUIREMENTS:
;       Needs STRDATE for the TIMESTAMP option.
;
; NOTES:
;       The WINDOW keyword is only evaluated if the current device supports
;       windows [!D.FLAGS AND 256) GT 0]. If you only want to close a
;       postscript file and don't fuss around with your screen windows
;       then simply don't set this keyword.
;
; EXAMPLE:
;       CLOSE_DEVICE
;          closes a postscript file (if opened) and resets the current
;          plotting device to 'X', 'WIN', or 'MAC' depending on the
;          OS_FAMILY.
;
;       CLOSE_DEVICE, PRINTER='hplj4', LABEL='mgs', /TIMESTAMP
;          If current device name is PS then the postscript file will
;          be closed, a user, date and time label will be added and
;          the file will be spooled to the  printer queue 'hplj4'.
;          NOTE: Printing of the file fails if you specified FILENAME as
;          a relative path in the OPEN_DEVICE command and you changed your
;          working directory while the device was opened.
;
;       CLOSE_DEVICE, WIN=-1
;          If current device name is PS then the postscript file will
;          be closed. If the current device is a screen device (that
;          supports windows), then the active window will be deleted.
;
; MODIFICATION HISTORY:
;       bmy, 18 Aug 1997: VERSION 1.00
;       bmy, 19 Aug 1997: VERSION 1.01
;       mgs, 20 Aug 1997: VERSION 1.02
;       mgs, 08 Apr 1998: VERSION 2.00
;          - automatic filename detection
;          - default device depending on OS_FAMILY
;       mgs, 21 May 1998:
```

```
;          - added L.. keywords to control label and timestamp output
;
;-
; Copyright (C) 1997, 1998, Bob Yantosca and Martin Schultz,
; Harvard University.
; This software is provided as is without any warranty
; whatsoever. It may be freely used, copied or distributed
; for non-commercial purposes. This copyright notice must be
; kept with any copy of this software. If this software shall
; be used commercially or sold as part of a larger package,
; please contact the author to arrange payment.
; Bugs and comments should be directed to mgs@io.harvard.edu
; with subject "IDL routine close_device"
 ;------------------------------------------------------------ --


pro close_device, OLD_DEVICE, PRINTER=PRINTER,  $
              LABEL=LABEL, TIMESTAMP=TIMESTAMP, LCOLOR=LCOLOR, $
              LPOSITION=LPOSITION,LSIZE=LSIZE, $
              LALIGN=LALIGN, $
              _EXTRA=e, WINDOW=WINDOW,  $
              FILENAME=FILENAME

     on_error, 2


     ; determine default device
     case strupcase(!VERSION.OS_FAMILY) of
       'UNIX'    : defdev = 'X'
       'WINDOWS' : defdev = 'WIN'
       'MACOS'   : defdev = 'MAC'
       else      : begin
                   print,'*** CLOSE_DEVICE: unknown operating system ! ***'
                   defdev = 'NULL'
                   end
     endcase

     ; set default value of OLD_DEVICE
     if (n_params() le 0) then OLD_DEVICE = defdev

     ; set default for printer queue
     if (n_elements(PRINTER) eq 0) then PRINTER  = ''

     ; set default position, charsize and color for label
     if (n_elements(LPOSITION) ne 2) then LPOSITION = [0.98,0.007]
     if (n_elements(LSIZE) eq 0) then LSIZE = 0.7
     if (n_elements(LCOLOR) eq 0) then LCOLOR = 1
     if (n_elements(LALIGN) eq 0) then LALIGN = 1
```

```
;  ============================================================
if (!d.name eq 'PS') then begin   ; if postscript device active

   ; determine if ps file was opened
   if (!D.UNIT gt 0) then begin
      ; extract current filename
      r = fstat(!D.UNIT)
      CFILENAME = r.name

      ; add label and timestamp if desired
      addlabel = 0   ; default: no label
      if (n_elements(LABEL) eq 0) then begin
         ; for compatibility set FILENAME as default label
         if (n_elements(FILENAME) gt 0) then $
            LABEL = FILENAME $
         else $    ; use actual filename as label
            LABEL = CFILENAME
      endif else $
         addlabel = 1   ; add label in any case

      if(keyword_set(TIMESTAMP)) then begin
         LABEL = LABEL + ', ' + strdate(_EXTRA=e)
         addlabel = 1   ; add label
      endif

      if (addlabel) then $
        xyouts,lposition(0),lposition(1),LABEL,color=lcolor,   $
            align=lalign,/norm,charsize=lsize

      ; close postscript file
  device, /close

      ; spawn postscript file to printer
      if (PRINTER ne '') then begin
       TRIM_PRINTER = strtrim(PRINTER,2)
         print, 'Sending output to printer ' + TRIM_PRINTER
         spawn, 'lpr -P ' + TRIM_PRINTER + ' ' + CFILENAME
      endif
   endif else $   ; ps file was not open
      device,/close   ; only close it


;  ============================================================
endif else begin     ; no postscript device active

   ; check if device supports windows and if a window
   ; shall be closed
```

```
        if(n_elements(window) gt 0) then begin
           if(window lt 0) then window = !D.WINDOW
           if( (!D.FLAGS AND 256) GT 0 AND window ge 0) then $
              wdelete,window
        endif

     endelse
     ;  ============================================================


     ; make OLD_DEVICE (usually default screen) active
     set_plot, OLD_DEVICE

return
end



 ;------------------------------------------------------------ --
;+
; NAME:
;       OPEN_DEVICE
;
; PURPOSE:
;       If hard copy is to be generated, OPEN_DEVICE opens the
;       PostScript device.  Otherwise OPEN_DEVICE opens an Xwindow.
;
; CATEGORY:
;       Input/Output
;
; CALLING SEQUENCE:
;       OPEN_DEVICE [,OLD_DEVICE] [,keywords]
;
; INPUTS:
;
; KEYWORD PARAMETERS:
;       PS        (int)  -> will send PostScript file to printer
;
;       COLOR     (int)  -> will enable PostScript color mode
;
;       LANDSCAPE (int)  -> will enable PostScript landscape mode
;
;       PORTRAIT  (int)  -> will enable PostScript portrait mode
;
;       FILENAME  (str)  -> The name to be given the PostScript file.
;                 Default: idl.ps
;
;       WINPARAM  (int)  -> An integer vector with up to 5 elements:
```

```
;               WINPARAM(0) = window number  (if negative, a window
;                   will be opened with the /FREE option.
;               WINPARAM(1) = X dimension of window in pixels (width)
;               WINPARAM(2) = Y dimension of window in pixels (height)
;               WINPARAM(3) = X offset of window (XPOS)
;               WINPARAM(4) = Y offset of window (YPOS)
;
;       TITLE -> window title
;
;       _EXTRA -> additional keywords that are passed to the call to
;               the DEVICE routine (e.g. /ENCAPSULATED)
;
; OUTPUTS:
;       OLD_DEVICE (str) -> stores the previous value of !D.NAME for
;          use in CLOSE_DEVICE. Note that CLOSE_DEVICE will automatically
;          set the default screen device if OLD_DEVICE is not provided,
;          hence it will only rarely be used.
;
; SUBROUTINES:
;
; REQUIREMENTS:
;
; NOTES:
;       If PS=0 then
;          Open Xwindow WINPARAM(0), which is WINPARAM(1) pixels wide
;          in the X-direction, and WINPARAM(2) pixels wide in the
;          Y-direction.
;
;       If PS=1 then
;         depending on /PORTRAIT or /LANDSCAPE and /COLOR
;         postscript is enabled in either portrait or landscape
;         mode as color or b/w plot
;
;       The key parameter which determines whether to open a postscript
;       file or a screen window is PS. Therefore, e.g. in a widget
;       application, you can pass a standard set of parameters for both,
;       postscript and screen, to this routine, and determine the device
;       to be chosen by a button state or checkbox which is passed into PS.
;
;
; EXAMPLE:
;       OPEN_DEVICE, WINPARAM=[0,800,800]
;          opens a screen window of size 800x800 pixels at the default
;          position
;
;       OPEN_DEVICE, OLD_DEVICE, /LANDSCAPE, FILENAME='myplot.ps'
;          opens a postscript file named myplot.ps in b/w and landscape
;          orientation
```

```
;
;       OPEN_DEVICE, OLDDEV, PS=PS, /PORTRAIT, /COLOR, WIN=2
;          depending on the value of PS either a color postscript file
;          named idl.ps is opened or screen window number 2 in default
;          size.
;
;
; MODIFICATION HISTORY:
;       bmy  15 Aug 1997: VERSION 1.00
;       bmy, 19 Aug 1997: VERSION 1.01
;       mgs, 20 Aug 1997: VERSION 1.02
;       mgs, 09 Apr 1998: VERSION 1.10
;          - added 2 more parameters for WINPARAM and TITLE keyword
;
;
;-
; Copyright (C) 1997, 1998, Bob Yantosca and Martin Schultz,
; Harvard University
; This software is provided as is without any warranty
; whatsoever. It may be freely used, copied or distributed
; for non-commercial purposes. This copyright notice must be
; kept with any copy of this software. If this software shall
; be used commercially or sold as part of a larger package,
; please contact the author to arrange payment.
; Bugs and comments should be directed to mgs@io.harvard.edu
; with subject "IDL routine open_device"
 ;-------------------------------------------------------------- --


pro open_device, OLD_DEVICE,                          $
              PS=PS, COLOR=COLOR, FILENAME=FILENAME,   $
              LANDSCAPE=LANDSCAPE, PORTRAIT=PORTRAIT,  $
              WINPARAM=WINPARAM, TITLE=TITLE, _EXTRA=E

     on_error, 2   ; return to caller

     OLD_DEVICE = !D.NAME    ; retrieve current device


     if (not keyword_set(FILENAME)) then FILENAME = 'idl.ps'
     if (not keyword_set(COLOR))    then COLOR    = 0
     if (not keyword_set(PORTRAIT)) then LANDSCAPE = 1 ; default

     if (keyword_set(PS)) then begin
        set_plot, 'PS'

        if (keyword_set(LANDSCAPE)) then begin       ;Landscape mode
           device, /landscape, color=COLOR, $
```

```
            bits=8, filename=FILENAME, _EXTRA=e

        endif else begin    ; Portrait mode
           device, color=COLOR, bits=8, /portrait,   $
              /inches, xoffset=0.25, yoffset=0.25,   $
              xsize=8.0, ysize=10, filename=FILENAME, _EXTRA=e
        endelse


    endif else begin      ; no postscript desired
                     ; only action if winparam given
                     ; and device supports windows

        if ((!D.FLAGS AND 256) gt 0 AND $
           n_elements(WINPARAM) gt 0) then begin
        ; WINPARAM must have 1, 3, or 5 elements, otherwise, default
        ; values for YSIZE and YOFFSET are used
           if (n_elements(winparam) gt 5) then winparam = winparam(0:4)

           if (n_elements(winparam) eq 4) then $
              winparam = [ winparam, 500 ]

           if (n_elements(winparam) eq 2) then $
              winparam = [ winparam, fix(winparam(1)/1.41) ]

           case n_elements(winparam) of
             5 :  window,winparam(0),FREE=(winparam(0) lt 0), $
                     xsize=winparam(1)>60, ysize=winparam(2)>10, $
                     xpos=winparam(3), ypos=winparam(4), $
                     TITLE=TITLE
             3 :  window,winparam(0),FREE=(winparam(0) lt 0), $
                     xsize=winparam(1)>60, ysize=winparam(2)>10, $
                     TITLE=TITLE
             1 :  window,winparam(0),FREE=(winparam(0) lt 0), $
                     TITLE=TITLE
              else : print,'*** OPEN_DEVICE: UNEXPECTED ERROR ! ***'
           endcase

        endif

    endelse


return
end


 ;------------------------------------------------------------ --
```

```
;+
; NAME:
;       STRDATE
;
; PURPOSE:
;       format a "standard form" date string
;
; CATEGORY:
;       date and time functions
;
; CALLING SEQUENCE:
;       result=STRDATE([DATE][,keywords])
;
; INPUTS:
;       DATE --> (optional) Either a up to 6 element array
;          containing year, month, day, hour, minute, and secs
;          (i.e. the format returned from BIN_DATE) or
;          a structure containing year, month, day, hour, minute, seconds
;          (as returned from tau2yymmdd) or a date string in "standard"
;          format as returned by SYSTIME(0).
;          If DATE is omitted, STRDATE will automatically
;          return the current system time.
;
; KEYWORD PARAMETERS:
;       SHORT --> omit the time value, return only date
;
;       SORTABLE --> will return 'YYYY/MM/DD HH:MM'
;
;       GERMAN --> will return 'DD.MM.YYYY HH:MM'
;
;       IS_STRING --> indicates that DATE is a date string rather
;          than an integer array. This keyword is now obsolete but kept
;          for compatibility.
;
; OUTPUTS:
;       A date string formatted as 'MM/DD/YYYY HH:MM'.
;       If SHORT flag is set, the format will be 'MM/DD/YYYY'
;
; SUBROUTINES:
;
; REQUIREMENTS:
;
; NOTES:
;       /GERMAN and /SORTABLE will have effect of SORTABLE but
;       with dots as date seperators.
;
; EXAMPLE:
;
```

```
; MODIFICATION HISTORY:
;       mgs, 11 Nov 1997: VERSION 1.00
;       mgs, 26 Mar 1998: VERSION 1.10
;           - examines type of DATE parameter and accepts structure input.
;
;-
; Copyright (C) 1997, Martin Schultz, Harvard University
; This software is provided as is without any warranty
; whatsoever. It may be freely used, copied or distributed
; for non-commercial purposes. This copyright notice must be
; kept with any copy of this software. If this software shall
; be used commercially or sold as part of a larger package,
; please contact the author to arrange payment.
; Bugs and comments should be directed to mgs@io.harvard.edu
; with subject "IDL routine strdate"
 ;--------------------------------------------------------- --


function strdate,date,is_string=is_string, $
          short=short,sortable=sortable,german=german


; on_error,2


if(n_elements(date) gt 0) then begin

    ; analyze format of DATE
    s = size(date)
    dtype = s(s(0)+1)
    if (dtype eq 7) then is_string = 1 else is_string=0
    if (dtype eq 8) then is_stru = 1 else is_stru=0

    if(is_string) then bdate=bin_date(date) $
    else if(is_stru) then begin
            bdate = intarr(6)
            for i=0,5 do bdate(i) = fix(date.(i))
        endif else bdate = date

endif else bdate = bin_date()    ; insert system time


; in case of not enough elements pad with zero's
tmp = intarr(6)
bdate = [bdate,tmp]

; convert to formatted string items
bdate = strtrim(string(bdate,format='(i4.2)'),2)
```

```
; and compose result string
; determine date seperator
if(keyword_set(german)) then sep='.' else sep='/'
; default : US format
   result = bdate(1)+sep+bdate(2)+sep+bdate(0)
; german format, day first
if (keyword_set(german)) then $
   result = bdate(2)+sep+bdate(1)+sep+bdate(0)
; sortable: year month day
if(keyword_set(sortable)) then $
   result = bdate(0)+sep+bdate(1)+sep+bdate(2)

if (not keyword_set(SHORT)) then  $
    result = result+' '+bdate(3)+':'+bdate(4)

return,result

end
```

File Attachments

1) close_device.pro, downloaded 118 times
2) open_device.pro, downloaded 108 times
3) strdate.pro, downloaded 126 times