
Subject: Re: 2 questions

Posted by [wgallery](#) on Tue, 11 Dec 2007 21:00:31 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Dec 11, 3:00 pm, David Fanning <n...@dfanning.com> wrote:

> R.G. Stockwell writes:

>> I have never been clear on what makes a ps file be an 'eps' file,

.....

> Most EPS files, though, also contain, in addition to
> the PostScript part of the file, another part that
> allows the graphic to be "previewed" in applications.
> So, if you had created your PostScript file in IDL
> with the ENCAPSULATED and PREVIEW keywords set
> appropriately, and you include THAT file in your
> Word document, you might see more than a big
> rectangle with an X in it. If you had been living
> a pious life, you might even see something that looked
> like the graphic you intend to print.

>

> However, when you send that Word file to a PS printer,
> the graphic will use the PostScript part to render it,
> not the low-level preview part.

>

> Of course, IDL preview images suck, but I--like you--
> have never been too bothered by that. They print correctly,
> that's the main thing. :-)

>

As David F. stated, an idl-created preview to an eps file sucks.

However, you can include your own high quality preview to an eps file which will look good both on screen and printed to a postscript printer. Details below. Note that as of Microsoft Office 2002, Word discards any preview supplied with an .eps file and substitutes its own even suckier preview (see: <http://support.microsoft.com/kb/290362/en-us>).

My solution to this problem is to create an eps file without a preview and convert it to a high resolution (300 dpi) .png file using epstool. This file looks good both on screen (even in Word) and printed. Details below.

To add a high quality (i.e., high resolution) preview to the .ps file idl.ps:

```
epstool -w --dpi 300 idl.ps idl.eps
```

You can use epstool either on a Linux/Unix box or in Windows under cygwin.

For epstool, see: <http://pages.cs.wisc.edu/~ghost/gsvieview/epstool.htm>

To create a high-quality .png file from an .eps file:

```
gs -sDEVICE=png256 -r300 -dBATCH -dNOPAUSE -sEPSCrop -q -
sOutputFile=idl.png idl.eps
```

For ghostscript (gs) see: <http://pages.cs.wisc.edu/~ghost/doc/AFPL/index.htm>

Finally, I have an idl procedure `eps_to_png.pro` which does the creation of a high-quality .png file. It works on both Windows and Linux/Unix (but requires ghostscript to be installed.) Here it is:

```
-----
;+
;$Name: $
;$Id: eps_to_png.pro,v 1.7 2007/07/30 15:17:31 wgallery Exp $
;
; NAME:
;   eps_to_png
;
; PURPOSE:
;   To convert an postscript plot file (.ps or .eps) to a Portable
Network Graphics
;   file (.png).
;
; CATEGORY:
;   Graphics
;
; CALLING SEQUENCE:
;   eps_to_png, filename, error, delete = delete, resolution =
resolution, true = true
;
;
; INPUTS:
;   filename = string: fully qualified name of the postscript file.
The .png file will have
;   the same base name but with .ps or .eps replaced with .png.
;
; KEYWORD PARAMETERS:
;   delete: if set and spawn executes sucessfully, then delete the .eps
file
;   resolution: int: resolution of the .png file, in pixels-per-inch,
default = 300
;   true: if set, then the .png file is in 24 bit truecolor. Default: 8
bit color.
;
; OUTPUTS:
;   error: 0: no error occurred, 1: an error occurred in processing
```

```

;
;
; PROCEDURE:
; This procedure should work on either Windows or Unix (Linux)
platforms that have the
; program ghostscript installed (program name: Windows=gswin32c.exe,
Unix=gs). It runs
; ghostscript with the proper parameters to convert the postscript
file (.ps or .eps) to
; a .png file. The resolution of the .png file is by default 300
pixels-per-inch which
; is sufficient for inclusion in Word or Powerpoint documents. For
multipage .ps files,
; each page will be sent to separate file with a sequence number
before the .png. E.g.,
; a 3 page .ps file named foo.ps will produce foo_01.png, foo_02.png,
and foo_03.png.
;
;
; MODIFICATION HISTORY:
; Created:
; Oct. 31, 2006 William Gallery, AER, Inc wgallery@aer.com
; Jan. 31, 2006 William Gallery,
; Added capability of converting multipage .ps file to
sequential .png files
;-
pro eps_to_png, filename, error, delete = delete, resolution =
resolution, true = true

error = 0

;;Check that the file exists
r = file_test(filename, /read)
if r ne 1 then begin
    print, 'Error: file does not exist or is not readable, file: ',
filename
    error = 1
    return
endif

;;Separate the file root from the extension
separate_filename_parts, filename, name = name, ext = ext, path =
path, drive = drive

if ext ne 'eps' and ext ne 'ps' then begin
    print, 'Error: file does not have .eps or .ps extension, file: ',
filename
    error = 1
    return
endif

```

```
;;Get the absolute path: the relative path will not work on Windows
path = file_expand_path(path)
```

```
;;In Windows, file_expand_path prepends the drive so don't include it
again
case ext of
  'eps': png_filename = path+path_sep()+name+'.png'
  'ps': png_filename = path+path_sep()+name+'_%02d.png' ;add sequence
number
endcase
```

```
case strupcase(!version.os_family) of
  'UNIX': gs_name = 'gs'
  'WINDOWS': gs_name = 'gswin32c.exe'
  else: begin
    print, 'OS not recognized, OS: ', !version.os_family
    error = 1
    return
  end
endcase
```

```
;;Options for the gs command. Note: capitalization is important
if n_elements(resolution) gt 0 then res = strtrim(fix(resolution), 2)
else res = '300'
if keyword_set(true) gt 0 then out_dev = 'png16m' else out_dev =
'png256'
```

```
gs_options = ['-sDEVICE='+out_dev, '-r'+res, '-dBATCH', '-dNOPAUSE ',
'-sEPSCrop ']
```

```
;;Run spawn with the /noshell option to make it run faster.
;;(Note: in this form, it hangs idl!!!????)
; cmd_ns = [gs_name, gs_options, '-sOutputFile='+png_filename,
filename]
; spawn, cmd_ns, /noshell, $
;   sp_out, sp_err_out, $
;   count = sp_count, exit_status = exit_status
cmd = gs_name+' '+strjoin(gs_options, ' ')+ $
  '-sOutputFile='+png_filename+' '+ $
  file_expand_path(filename)
```

```
case strupcase(!version.os_family) of
  'UNIX': begin
    spawn, cmd, sp_out, sp_err_out, $
    count = sp_count, exit_status = exit_status
  end
  'WINDOWS': begin
```

```
        spawn, cmd, sp_out, sp_err_out, $
            count = sp_count, exit_status = exit_status, $
            /hide
    end
endcase

if exit_status ne 0 then begin
    print, 'Error: from eps_to_png: exit status = ', exit_status
    print, sp_out
    print, sp_err_out
    error = 1
    return
endif

if keyword_set(delete) then file_delete, filename

return

end
```
