CONVERTING POSTSCRIPT FILES TO IMAGES USING IMAGEMAGICK

By Jeff Guerber (jeff.guerber@nasa.gov), SSAI / GSFC code 614.1
This version, April 10, 2008.

PostScript files work very nicely for printing graphics, but to import them into a program such as PowerPoint or Word with good results, they should first be rendered into a raster image format such as JPEG or PNG. While at least some versions of PowerPoint (or Word, etc.) can now import PS directly, for me the results have been less than satisfactory.

(I believe this may not be necessary with Adobe's own programs such as Photoshop or Acrobat, but I have no experience with them.)

The ImageMagick package's "convert" program (sometimes installed as "image_convert"), can perform many types of image conversions, and can work with ghostscript to create good-quality raster image files from PostScript files. Although the default settings used by convert do not give good results, I have found the following to work well:

> CAUTION: DO NOT USE ONE OF THE HP-UX COMPUTERS FOR THIS! The versions of ImageMagick and Ghostscript on them are very old, and cannot easily be updated. Instead, use one of the Linux machines, or a PC with a *recent* version of this software.
>
> NOTE: If the PostScript file contains multiple pages, be sure to read the section "MULTIPLE POSTSCRIPT PAGES IN A FILE" below!

TO CREATE A JPEG FILE: In principle, JPEG is preferable for images. Note that JPEG uses a lossy compression scheme, and quality will decline the more times a JPEG image is opened and then saved.

```
convert -density 144x144 +antialias -quality 100 image.ps image.jpg
```

TO CREATE A PNG FILE: In principle, PNG (a successor and improvement to GIF) is preferable for line graphics. PNG compression is lossless.

```
convert -density 144x144 +antialias image.ps image.png
```

The resulting JPEG or PNG file can then be imported into PowerPoint or another program, positioned, resized, etc., like any image file.

EXPLANATION OF OPTIONS USED:

-density 144x144:

PostScript is a vector format oriented towards printers, and operates in _physical_ units on a page of paper: essentially it says, "position at (1.5",.5") from the corner of the page, then draw a line to (3",1.25") from the corner". (Normally, the units used are actually "points", 1 pt = 1/72"). When rendering to a raster image, these page locations must be converted into pixel positions. convert's default is

to use 72 output pixels per inch of the input (ie., one pixel per point), but this does not provide sufficient resolution to render fine detail like text well. "-density 144x144" doubles this resolution, which I have found to produce good results. I have not seen noticeable improvement by going higher. (PowerPoint's direct import of PostScript seems to always use 72 ppi.)

The size of the resulting image in pixels will be (PS plot size) * (density), so IDL's default PS plot size of 7"x5", converted with a density of 144x144, will create a raster image with 1008x720 pixels. At 72 ppi, the image will be only 504x360. At 216 ppi (= 3*72), the image will be 1512x1080, which is often larger than the screen size of a PowerPoint page (and unless you have a large monitor, very likely larger than the entire screen!).

+antialias:

(Note: PLUS sign.) Turns OFF antialiasing of the output raster image. Antialiasing is a procedure that slightly smooths a raster image, to reduce the jaggedness of curves (including letters) and skewed lines. Although it can improve the appearance of a plot, I found that some fine detail became harder to see (although I don't think any was lost), and lines tended to become washed out (black becoming noticeably gray, for instance). Turning off antialiasing resulted in much crisper images. Furthermore, PowerPoint appears to apply some antialiasing to imported images itself.

-quality 100:

(JPEG output only) JPEG uses a lossy compression algorithm. "-quality 100" says to produce the best quality output, at the expense of less compression (and thus a larger file).

Optional but possibly useful option:

-geometry <width>x<height>:

Before writing the output file, resamples the raster image to be <width> pixels wide by <height> pixels high. Use this if you want the final image to be a different size than was produced by the PostScript rendering step. Note that convert will normally maintain the aspect ratio (width:height) of the original image, fitting it within the <width>x<height>-pixel box. Append an "!" to force the image to exactly the number of pixels specified, even if that means changing its shape (eg., "-geometry 1600x900!").

All convert options are documented on the ImageMagick Web site:

http://www.imagemagick.org/script/command-line-options.php.

TO CONVERT MULTIPLE FILES IN BATCH: You can use a simple shell loop. Here's how to do it in tcsh (I think csh is the same):

```
$ foreach f ( psfile*.ps )
```

```
foreach? convert -density 144x144 +antialias -quality 100 $f ${f:r}.jpg
foreach? end
$
```

"${f:r}" removes the filename extension from the variable $f.

MULTIPLE POSTSCRIPT PAGES IN A FILE

When a PostScript file contains multiple pages, ImageMagick convert is supposed to put them into separate raster-image files. However, I have not been able to get this to work, only the first page renders correctly. There is another utility that can extract the individual pages into separate PostScript files:

```
psselect -p1 in.ps out.ps
```

will extract the first page of in.ps and write only that page to out.ps. To extract the second page, use -p2; the third page, -p3. Etc. Unfortunately it does not seem to be able to extract multiple pages to separate files at once; if you have many files, or one file containing many pages, you may need to use a shell loop or write a small script. psselect is from Angus Duggan's PSUtils package.

ghostview, and gv where available, can also extract individual pages. In ghostview, first display the page you want, mark it with menu item Page > Mark, then choose File > Save marked pages... and select the output file name. Finally, remove the page mark with menu item Page > Unmark. Unfortunately ghostview's file-selection dialog is clunky in the extreme, so if you're extracting more than one or two pages, you'll probably find psselect easier.

In gv, mark the page or pages you want by right clicking on the page numbers in the list in the lower left. A red rectangle will appear next to the number of each marked page. Then choose menu item File > Save marked pages... and select the output file name. Right click on the page numbers again to remove the marks.

GOING THE OTHER WAY: USING convert OR CUPS TO PRINT IMAGE FILES

ImageMagick convert can also be used to print .png, .jpg, or other image files on a PostScript printer:

```
convert in.jpg PS:- | lp -d ourprinter
```

will print file "in.jpg" on the printer called "ourprinter". This command encloses the image's raster data in a simple PostScript wrapper that tells the printer how to use it. (Note that it does *not* create an actual PostScript plot, it's still fundamentally a raster image.) You may need to experiment with convert options such as -geometry, -page, or -resize to get the sizing and positioning correct.

Although wildcards can be used in the file name to print multiple files, I don't recommend this because all the images will be combined into a

single PostScript stream going to the printer. It's better to do each file separately.

On Linux (but *NOT* on HP-UX!), many types of image file can just be printed directly, eg.:

```
lp -d ourprinter in.jpg                # Note: Linux only!
```

This is a function of the CUPS printer driver system used by Linux; see "man lp" on Linux for available options. You probably get more control using convert, however. I understand MacOSX also uses CUPS, so this feature is probably available there too. Be careful trying this for less common file types, though: printing an unsupported image type will result in the deaths of many trees (not to mention greatly annoying your colleagues)! If in doubt, immediately go to the printer so you can quickly hit the "Cancel Job" or "Stop" button if it's spewing garbage.

HP 4700 printers, such as icesatl4, are supposed to be able to print PDF files directly, in addition to PS, PCL, and text, but I haven't tested this.