# A brief tutorial on image format conversions

R (Chandra) Chandrasekhar

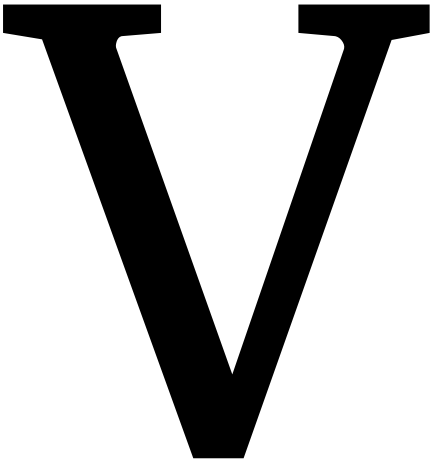
2021-03-07 | 2021-03-07

## Two varieties of images

Image formats come in two broad flavours:

- raster or bitmap graphics, and
- vector graphics.

The former leads to image blockiness or pixellation at high magnifications, while the latter scales without degradation when magnified.



Vector

Raster

### Raster Graphics

There are dozens of image formats, including:

1. The Tag(ged) Image File Format (TIFF) which is used in printing;
2. The Joint Photographic Experts Group(JPEG) format for scene and portrait image display and compression; and
3. The Portable Network Graphics (PNG) format, optimized principally for text-based image display and compression.

All three formats employ raster graphics.

### Vector Graphics

The two principal vector graphics formats are:

1. The Portable Document Format (PDF) format which is used primarily in archival quality electronic and printed documents; while
2. The Scalable Vector Graphics (SVG) format—interestingly text-based—is used for graphics display on web browsers.

Both these formats employ vector graphics.

## Format conversions

It is often necessary—for a variety of reasons—to convert from one image format to another. There are four possibilities:

a. raster to raster;
b. raster to vector;
c. vector to raster; and
d. vector to vector

We consider each of these in turn using platform-neutral open source tools. Since I run GNU/Linux on my desktop, my examples will feature commands from that setup.[1]

## Tools for image format conversion

There are four main tools for image format conversion:

1. ImageMagick
    - library plus executables
    - pixel based
    - converts raster to raster
2. cairo
    - library
    - few executablea
    - vector-based 2D drawing and rendering
    - used by other programs
3. poppler
    - library plus executables
    - vector-based
    - renders pdf
    - can use cairo as backend
4. Inkscape
    - GUI-based
    - vector-based SVG is default
    - two executables
    - export options
    - can use cairo for PDF export

Better top formalize this in a table.

[1] There are a good number of websites that promise conversion online, requiring you to upload the input file and download the output file. These *might be* fraught with security risks. Use them only if you trust that your documents will not be misused.

## ImageMagick: the Swiss Army knife

ImageMagick is the name given to a suite of image processing tools originally created in 1987 by John Cristy, then working for Du Pont. In 1990, it was freely released by Du Pont, who transferred copyright to ImageMagick Studio LLC who now maintain the project. It is distributed under a derived Apache 2.0 license. The authoritative source code repository shows active development even today, 34 years after the suite was first released [1].

ImageMagick is so versatile and useful that it may rightfully be called the Swiss Army knife of the image processing world.

ImageMagick comes with several command line utilities, each replete with options. Among these are:

- `convert` which converts from one format to another;
- `display` which displays one or more images;
- `identify` which identifies the type of image and displays its characteristics;
- `mogrify` which transforms an image, modifying its appearance; and
- `montage` which generates an image montage.

The above list is far from exhaustive. The interested reader is referred to the excellent online documentation for further details. The power of ImageMagick is enhanced with the magick-script Image Scripting Language. In the examples below, I will give both the command line invocations and scripts for performing image conversions.

## Cairo

`rsvg-convert` `pdftocairo`

https://en.wikipedia.org/wiki/Cairo_(graphics)

https://www.cairographics.org/

https://cgit.freedesktop.org/cairo

### Inkscape

https://wiki.inkscape.org/wiki/index.php/Tools

## pdf2svg

https://cityinthesky.co.uk/opensource/pdf2svg/

https://github.com/dawbarton/pdf2svg

https://inkscape.org/develop/about-svg/

## poppler

`pdftoppm`

pdftocairo

## Raster to raster

## Image to PDF

Still works. No strictures. But the PDF can get grungy. Use a pyramid of resolutions.

### Avoiding blurry PDFs

-units pixelsperinch -density 1200 etc., in conversion

Useful when a hgh resolution image is available. In any case: PDF and png/jpg sizes are similar.

## Choosing the optimal image resolution for a clear PDF

96dpi for screen 150 dpi default 300 dpi for print [give references]

```
convert -units pixelsperinch -density 300 file.png file.pdf
```

## PDF to image not supported

```
#! /bin/magick
convert file.pdf file.png
```

```
convert test.pdf test.png
convert: unable to open image 'test.pdf': No such file or directory @ error/blob.c/OpenBlob/3537.
convert: no images defined `test.png' @ error/convert.c/ConvertImageCommand/3304.
```

### Security considerations

ImageMagick is no more the famed Swiss army knife for conversions from PDFs to images.

Give references to security concerns.

## Enter poppler

### PDF to PNG

## PDF to SVG and vice versa

https://wiki.gnome.org/Projects/LibRsvg

```
pdftoppm -png ernst-heackel-medium.pdf ernst-heackel-medium.png
convert ernst-heackel-medium.jpg ernst-heackel-medium-direct.png
convert ernst-heackel-medium.jpg ernst-heackel-medium-direct.png
```

How to use resize etc.

## Appendix: ImageMagick's security vulnerabilities

Great power exacts a commensurate price. ImageMagick's great power and ease of use does come at a great price: vulnerability to exploits by malicious remote actors.

ImageMagick uses external libraries or *backend tools* which are called via `system()` commands in accordance with *delegated* command strings specified in a configuration file called `policy.xml`.

In April 2016, it was reported that because of insufficient validation of delegated command strings, it was possible for someone to execute malicious code remotely, to the detriment of the unwitting user of ImageMagick. This was revealed at a website, interestingly named ImageTragick to attract sufficient attention and remedial action to the discovered bug [2].

In November 2020, another security vulnerability was discovered [3]. It was reported and promptly patched by the ImageMagick maintainers [4].

Recent versions of the ImageMagick suite, bundled with major distributions, should have correctly configured `policy.xml` files that will block known exploits. Sandboxing is another technique to quarantine the system from possible vulnerabilities. Above all, it is vital to keep system and application software up to date to avail of evolutions in performance and security.

## Image used below

https://www.rawpixel.com/board/1236113/kunstformen-der-natur-ernst-haeckel-free-cc0-public-domain-animal-prints

**Feedback** Please email me your comments and corrections.

## References

[1] ImageMagick Studio LLC, 'ImageMagick 7.' [Online]. Available: https://github.com/ImageMagick/ImageMagick. [Accessed: 08-Mar-2021]

[2] —, 'ImageMagick is on fire—CVE-2016–3714,' 12-May-2016. [Online]. Available: https://imagetragick.com/. [Accessed: 08-Mar-2021]

[3] J. Leyden, 'ImageMagick PDF-parsing flaw allowed attacker to execute shell commands via maliciously crafted image,' 23-Nov-2020. [Online]. Available: https://portswigger.net/daily-swig/imagemagick-pdf-parsing-flaw-allowed-attacker-to-execute-shell-commands-via-maliciously-crafted-image. [Accessed: 08-Mar-2021]

[4] A. Inführ, 'ImageMagick - Shell injection via PDF password,' 21-Nov-2020. [Online]. Available: https://insert-script.blogspot.com/2020/11/imagemagick-shell-injection-via-pdf.html. [Accessed: 08-Mar-2021]