Convert PDF to image with high resolution

I'm trying to use the command line program `convert` to take a PDF into an image (JPEG or PNG). Here is one of the PDFs that I'm trying to convert.

I want the program to trim off the excess white-space and return a high enough quality image that the superscripts can be read with ease.

This is my current best attempt. As you can see, the trimming works fine, I just need to sharpen up the resolution quite a bit. This is the command I'm using:

```
convert -trim 24.pdf -resize 500% -quality 100 -sharpen 0x1.0 24-11.jpg
```

I've tried to make the following conscious decisions:

- resize it larger (has no effect on the resolution)
- make the quality as high as possible
- use the `-sharpen` (I've tried a range of values)

Any suggestions please on getting the resolution of the image in the final PNG/JPEG higher would be greatly appreciated!

pdf imagemagick

edited Dec 16 '14 at 14:13
Kurt Pfeifle
54.5k ● 14 ● 148 ● 237

asked Jul 7 '11 at 1:52
JBWhitmore
4,404 ● 6 ● 24 ● 37

I don't know, you could also try link... – user826788 Jul 7 '11 at 1:56

See also: askubuntu.com/a/50180/64957 – Dave Jarvis Jun 22 '14 at 1:19

9 Answers

It appears that the following works:

```
convert           \
   -verbose       \
   -density 150   \
   -trim          \
    test.pdf      \
   -quality 100   \
   -flatten       \
   -sharpen 0x1.0 \
    24-18.jpg
```

It results in **the left image**. Compare this to the result of my original command (**the image on the right**):

(To ***really*** see and appreciate the differences between the two, right-click on each and select *"Open Image in New Tab..."*.)

Also keep the following facts in mind:

- The worse, blurry image on the right has a file size of 1.941.702 Bytes (1.85 MByte). Its resolution is 3060x3960 pixels, using 16-bit RGB color space.
- The better, sharp image on the left has a file size of 337.879 Bytes (330 kByte). Its resolution is 758x996 pixels, using 8-bit Gray color space.

So, no need to resize; add the `-density` flag. The density value 150 is weird -- trying a range of values results in a worse looking image in both directions!

edited May 24 at 18:21

answered Jul 7 '11 at 2:06
JBWhitmore
4,404 ● 6 ● 24 ● 37

40 The `density` parameter is a bit special in that in must come *before* the input file. As PDF is a vector based file format which does not have (much) notion of pixels, it says something like "page is 8in by 12in". If you want pixel, you use the `density` setting to tell it, how many pixels per inch you want to get in the output. E. g. with 150 you would get 8x150=1200 by 12x150=1800 pixels in the resulting image. That's also the amount of pixels the sharpen, contrast, compression etc. settings work on. – Daniel Schneller Aug 2 '13 at 7:30

5 It can result in black background on Mac OS (see stackoverflow.com/questions/10934456/…). To fix this, add `-flatten`. – Rolf Apr 16 '16 at 21:59

2 i got a black background on Mac OS when I tried to convert pdf to png, adding -flatten solved it. – olala Sep 1 '16 at 22:34

2 Wow! Just used the `-density` and `-flatten` option to reduce the size of a pdf (to another pdf). The `-flatten` option really helps a lot in reducing the total size. In my case, without visual disturbances. – parvus Oct 6 '16 at 8:45

2 The `-density` flag will likely give worse results on higher values if the quality of the starting image was lower than that. – parvus Oct 6 '16 at 8:48

Personally I like this.

```
convert -density 300 -trim test.pdf -quality 100 test.jpg
```

It's a little over twice the file size, but it looks better to me.

`-density 300` sets the dpi that the PDF is rendered at.

`-trim` removes any edge pixels that are the same color as the corner pixels.

`-quality 100` sets the JPEG compression quality to the highest quality.

Things like `-sharpen` don't work well with text because they undo things your font rendering system did to make it more legible.

If you actually want it blown up use resize here and possibly a larger dpi value of something like `targetDPI * scalingFactor` That will render the PDF at the resolution/size you intend.

Descriptions of the parameters on imagemagick.org are here

edited Mar 2 at 17:54

answered Dec 9 '12 at 4:52
majinnaibu
1,396 ● 11 ● 17

It's twice the size mostly because the output density has been doubled and the jpg compression quality is set on max (so not much compression). – rivimey Apr 15 '15 at 14:14

Little nitpick: JPG is still lossy at quality 100% – relgukxilef Feb 28 at 15:13

@relgukxilef Thanks, I've corrected the error. – majinnaibu Mar 2 at 17:55

I have found it both faster and more stable when batch-processing large PDFs into PNGs and JPGs to use the underlying `gs` (aka Ghostscript) command that `convert` uses.

You can see the command in the output of `convert -verbose` and there are a few more tweaks possible there (YMMV) that are difficult / impossible to access directly via `convert`.

However, it would be harder to do your trimming and sharpening using `gs`, so, as I said, YMMV!

answered Apr 19 '13 at 11:30

Coder
1,143 ● 1 ● 12 ● 19

I use `pdftoppm` on the command line to get the initial image, typically with a resolution of 300dpi, so `pdftoppm -r 300`, then use `convert` to do the trimming and PNG conversion.

answered May 31 '15 at 17:03

Norman Ramsey
151k ● 47 ● 304 ● 485

---

while not using Imagemagick, this solution seems most in the spirit of a transparent conversion. `pdftoppm` can also output JPEGs and PNGs. – Aaron Brick Sep 4 '16 at 5:16

---

normally I extract the embedded image with 'pdfimages' at the native resolution, then use ImageMagick's convert to the needed format:

```
$ pdfimages -list fileName.pdf
$ pdfimages fileName.pdf fileName   # save in .ppm format
$ convert fileName-000.ppm fileName-000.png
```

this generate the best and smallest result file.

Note: For lossy JPG embedded images, you had to use -j:

```
$ pdfimages -j fileName.pdf fileName   # save in .jpg format
```

On little provided Win platform you had to download a recent (0.37 2015) 'poppler-util' binary from: http://blog.alivate.com.au/poppler-windows/

edited Nov 11 '15 at 16:06

answered Oct 22 '15 at 21:12

Valerio
81 ● 1 ● 3

It also gives you good results:

```
exec("convert -geometry 1600x1600 -density 200x200 -quality 100 test.pdf
test_image.jpg");
```

edited May 15 '12 at 12:57

bluish
11.1k ● 13 ● 79 ● 139

answered May 15 '12 at 11:45

Preet Sandhu
176 ● 3 ● 7

One more suggestion is that you can use GIMP.

Just load the PDF file in GIMP->save as .xcf and then you can do whatever you want to the image.

answered Oct 24 '13 at 11:11

Armin Mustafa
464 ● 1 ● 6 ● 11

---

8 The reason for doing this via the command line is that I had thousands of pages that needed this process. – JBWhitmore Oct 24 '13 at 13:09

Also, GIMP renders the page on *loading*, so you'll want to set the resolution when you select the pages to load. It doesn't much matter what you set the output parameters to if you start with the 100 DPI default on loading. – Keith Davies Feb 9 at 1:29

PNG file you attached looks really blurred. In case if you need to use additional post-processing for each image you generated as PDF preview, you will decrease performance of your solution.

2JPEG can convert PDF file you attached to a nice sharpen JPG and crop empty margins in one call:

```
2jpeg.exe -src "C:\In\*.*" -dst "C:\Out" -oper Crop method:autocrop
```

edited Oct 13 '14 at 9:18
Mikhael Bolgov
21 • 4

answered Oct 13 '14 at 9:13
Mikhael
1 • 1

The blurriness in the original PNG is what inspired the question in the first place, and the PNG in the accepted answer is rather crisp. – JBWhitmore Oct 13 '14 at 17:35

Use this commandline:

```
convert -geometry 3600x3600 -density 300x300 -quality 100 TEAM\ 4.pdf team4.png
```

This should correctly convert the file as you've asked for.

edited Jul 21 '15 at 16:40
pid
8,592 • 5 • 17 • 42

answered Jul 21 '15 at 1:19

Aleksandrs Bogdanovs
1 • 1