



PDFTron PDF2Image™ User Manual

Version 4.x

LEGAL STATEMENT AND COPYRIGHT NOTICE

PDFTron PDF2Image™ Command-Line Application User Manual
 Part number: PDFTRON-4-PDF2ImageCMD
 Last Updated: November 23, 2009

© 2004-2009 PDFTron Systems, Inc. All Rights Reserved.

All information contained herein is the property of PDFTron Systems, Inc. ("PDFTron"). No part of this publication (whether in hardcopy or electronic form) may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of PDFTron Systems, Inc..

The information in this publication is provided for informational use only, is subject to change without notice, and should not be construed as a commitment by PDFTron. PDFTron assumes no responsibility or liability for any loss or damage that may arise from the use of any information in this publication. The software described in this user manual is furnished under License (enclosed in the software package) and may only be used or copied in accordance with the terms of that License.

PDFTron and the names of PDFTron products referenced herein are either trademarks and/or service marks and/or registered trademarks of PDFTron Systems, Inc. PDFTron, PDFTron PDFSecure, PDF2SVG, PDF2Image, PDF2Text, XPSConvert, CosEdit, PDF PageMaster, PDFNet SDK, PDF2SVG SDK, PDF2Image SDK, PDF PageMaster SDK, PDF2Text SDK, PDFSecure SDK and associated Logos are either trademarks and/or service marks and/or registered trademarks of PDFTron Systems, Inc.

Any other brand or product names mentioned in this publication are the registered trademarks or trademarks of their respective holders. Mention of a product in this document does not necessarily imply endorsement of the product.

LEGAL STATEMENT AND COPYRIGHT NOTICE		2
1.	Introduction	5
1.1	An Introduction to PDFTron PDF2Image	5
1.1.1	Key Functions	5
1.1.2	What's New?	5
1.1.3	Common Use Case Scenarios	5
1.1.4	Operating Systems Supported	5
1.1.5	System Requirements	5
1.2	PDF2Image SDK	6
1.3	About This Manual	6
2.	Installing and Uninstalling PDF2Image	7
2.1	PDF2Image Installation	7
2.2	Demo Version Installation	7
2.3	Uninstalling PDF2Image	7
3.	Overview	8
3.1	Basic Syntax	8
3.2	Command-Line Summary	9
3.3	Basic Usage	12
3.3.1	How to save rasterized files in a given folder?	12
3.3.2	How can I control the output name for rasterized/converted images?	12
3.3.3	How do I convert PDF to JPEG, PNG, TIF or some other image format?	12
3.3.4	How do I convert PDF to multi-page TIF?	13
3.3.5	How do I open a password protected PDF?	13
3.3.6	How do I create grayscale images?	13
3.3.7	How do I specify which pages to convert?	13
3.3.8	How do I specify the resolution of the output image?	14
3.3.9	How do I specify dimensions of the output image in pixels?	14
3.3.10	How do I render only a subset of a given page?	15
3.3.11	How do I render very large images?	15
3.3.12	How do I batch convert files?	15
3.3.13	How do I specify compression ratio for JPEG format?	16
3.3.14	How can I rotate pages?	16
3.3.15	How can I show/hide crop marks or the trim region?	16
3.3.16	How do I render PDF as CCITT Group 4 FAX TIFF or monochrome PNG?	17
3.3.17	Does PDF2Image have any dependencies on third party components/software?	17
3.4	General Usage Examples	18
Example 1.	The simplest command line: Convert PDF to PNG.	18
Example 2.	Convert PDF to JPEG at 300 DPI and higher compression.	18
Example 3.	Convert a password protected file to a TIFF file of given pixel dimensions.	18

4

1.2 PDF2Image SDK

For developers who are looking for a software development component to integrate into their application, PDFTron also offers **PDF2Image SDK**, an easy-to-use, yet powerful software component for embedding into client and server based applications. PDF2Image SDK is available as a plain 'C DLL' and can be easily accessed from any programming language (including C#, VB.NET, C/C++, Java, VB6, Perl, Python, Ruby, Delphi, etc).

PDF2Image is based on **PDFNet SDK**, PDFTron's own core technology, which offers the same rasterization capability available in PDF2Image. PDFNet SDK is a comprehensive developer library for PDF creation, manipulation and rendering, offered on a wide range of platforms and programming environments. If you require rasterization or other functionality than what is provided as part of PDF2Image for embedding in your own applications, please contact a PDFTron representative or visit <http://www.pdftron.com/net> for more information.

1.3 About This Manual

This manual is intended as a guide to the installation and use of PDF2Image. It is intended for programmers and other users who are familiar with PDF documents, graphic image file creation, graphic file manipulation and general computer processes.

- Section 1 introduces PDF2Image and describes the manual.
- Section 2 explains how to install and uninstall PDF2Image.
- Section 3 covers basic use of PDF2Image.
- Section 4 is where you will find all the support information you may require, such as how to report a problem with the software.

3. Overview

PDFTron's PDF2Image is a command-line application designed to convert a selected PDF document file to one or more BMP, PNG8, JPEG, PNG, TIFF or RAW image files, while presenting several options to control Resolution, Color Bit Depth and other settings, depending on the output format selected. This section covers the basic use of PDF2Image explaining all the available options.

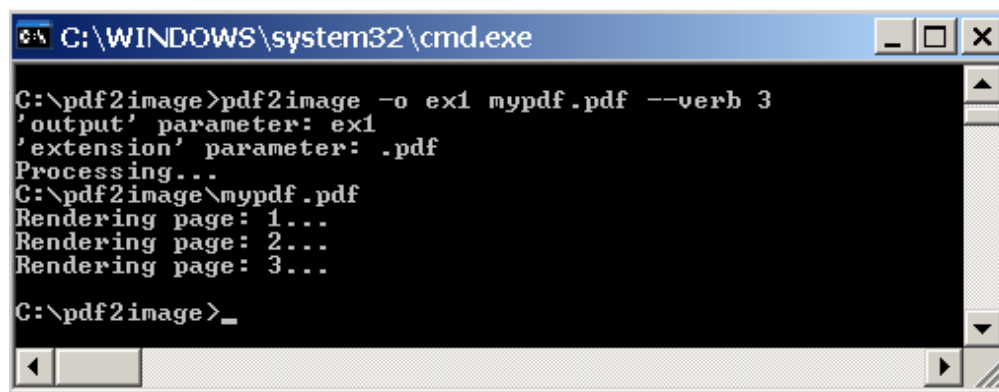


Figure 3.0 PDF2Image Command-line Application.

3.1 Basic Syntax

The basic command-line syntax is:

```
pdf2image [options] file1 file2 folder1 file3 ...
```


3.2 Command-Line Summary

The following command-line arguments are available for PDF2Image.

Option	Parameter	Description
-o or --output	e.g. -o myfolder	The output folder used to store rasterized files. By default, the currently selected working folder will be used to store converted image files.
--prefix	--prefix myprefix	The prefix for the output image file. The output filename will be constructed by appending the prefix string, the page number, and the appropriate image extension (e.g. myprefix1.png, myprefix2.png, etc). The prefix option should be used only for conversion of individual documents. By default, the each input filename will be used as a prefix.
--digits	--digits 4	The number of digits used in the page counter portion of the output filename. By default, new digits are added as needed; however this parameter could be used to format the page counter field to a uniform width (e.g. myfile0001.jpg, myfile0002.jpg, etc).
--subfolders		Process all sub-directory for every directory specified in the argument list. By default, sub-directories are not processed.
-f or --format	-f jpg -f bmp -f tif	Output image format. The following is a list of supported export formats: <ul style="list-style-type: none"> ■ png (Portable Network Graphics) ■ png8 (Palletized PNG) ■ jpg or jpeg (Joint Photographic Expert Group) ■ tif or tiff (Tagged Image File Format) ■ tif8 (Palletized Tagged Image File Format) ■ bmp (Windows Bitmap Format) ■ raw (raw RGB data) <p>The default output format is png.</p>
-d or --dpi	-d 300	The output resolution, from 1 to 1000, in Dots Per Inch (DPI). The higher the DPI, the larger the image. Resolutions larger than 1000 DPI can be achieved by rendering image in tiles or stripes. The default resolution is 92 DPI.
--hres	--hres 100	The width of the output image, in pixels.
--vres	--vres 100	The height of the output image, in pixels.
-a or --pages	Render page 1,3, and 10: -a 1,3,10	Specifies the list of pages to convert. By default, all pages are converted.

	<p>Render all even pages: -a even</p> <p>Render pages in the range from 3-11 and page 50: --pages 3-11,50</p> <p>Render all odd pages and all pages in the range from 100 to the last page: -a odd,100-</p>	
-b or --box	-b media	<p>Specifies the page box/region to rasterize. Possible values are:</p> <ul style="list-style-type: none"> ■ media ■ crop ■ trim ■ bleed ■ art <p>By default, page crop region will be rasterized.</p>
-c or --clip	-c 216,522,330,600	User definable clip box. By default, the clip region is identical to the current page 'box'.
-r or --rotate	-r 90	<p>Rotates all pages by a given number of degrees, counterclockwise. The allowed values are:</p> <ul style="list-style-type: none"> ■ 0 ■ 90 ■ 180 ■ 270 <p>The default value is 0.</p>
-g or --gray	--gray	Render and export the image in grayscale mode. Sets pixel format to 8 bits per pixel grayscale. By default, the image is rendered and exported in RGB color space.
-k or --cmyk	--cmyk -f tif	Render and export the image in CMYK mode. To export CMYK, the output image format must support CMYK pixel format. An example of image format that supports CMYK is TIFF (e.g. -f tif -k). By default, the image is rendered and exported in RGB color space.
--mono	--mono	Export the rendered image as 1 bit per pixel (monochrome) image. If the output format is TIFF, the image will be compressed using G4 CCITT compression algorithm. By default, the image is not dithered. To enable dithering use '--dither' option.
--dither	--dither	Enables dithering when the image is exported in palletized or monochrome mode (e.g. when export format is tif8, png8 or --mono).
--gamma	--gamma 0.3	Sets the gamma factor used for anti-aliased rendering.

		Typical values are in the range from 0.1 to 3. Gamma correction can be used to improve the quality of anti-aliased image output and can (to some extent) decrease the appearance common anti-aliasing artifacts (such as pixel width lines between polygons).
-q or --quality	-q 100	Compression quality is a number in the range from 1 to 100. Lower numbers usually result in better compression at the expense of image quality. The default setting is 80.
-m or --multipage	--multipage	If the output image format supports multi-page or multi-frame capability, store all output images in one file instead of separate files. Currently, this option is only relevant to TIFF output. By default, images will be saved in separate files.
--printmode		Renders annotations in the print mode. This option can be used to render 'Print Only' annotations and to hide 'Screen Only' annotations.
--noannots		Disables drawing of annotations and forms.
--nosmooth		Disables image smoothing.
--noprompt		Disables any user input. By default, the application will ask for a valid password if the password is incorrect.
-p or --pass	e.g. secret or "my pass"	The password for the input file. Not required if the input document is not secured.
--extension	--extension ".pdf"	The default file extension used to process PDF documents. The default is ".pdf".
-h or --help		Print a listing of available options.
-v or --version		Print the version information.
--verb	--verb 2	Set the verbosity level. Valid parameter values are 0, 1, and 2. The higher number results in more feedback. The default is 1.

3.3 Basic Usage

3.3.1 How to save rasterized files in a given folder?

By default, PDF2Image saves rasterized files in the current working folder. To specify another output location, use the '-o' (or --output) parameter. For example:

```
pdf2image -o "c:\My Output" 1.pdf 2.pdf 3.pdf
```

Note: If the specified path does not exist, PDF2Image will attempt to create the necessary folders.

3.3.2 How can I control the output name for rasterized/converted images?

By default, PDF2Image creates a separate image file for every page in the document. The output filename is constructed using the name of the input PDF file, page counter, and appropriate image extension. For example, the following command-line generates a sequence of image files starting with mydoc_1.jpg, mydoc_2.jpg, etc.:

```
pdf2image -f jpg mydoc.pdf
```

PDF2Image allows output filename customizations using the '--prefix' and '--digits' options. For example, the following command-line generates a sequence of image files starting with newname_0001.jpg, newname_0002.jpg, etc.:

```
pdf2image -f jpg --prefix newname --digits 4 mydoc.pdf
```

The '--digits' parameter specifies the number of digits used in the page counter portion of the output filename. By default, new digits are added as needed, however this parameter could be used to format the page counter field to a uniform width (e.g. myfile0001.jpg, myfile0010.jpg, instead of myfile_1.jpg, myfile_10.jpg, etc).

To avoid any ambiguities in file naming, the prefix option should be used only for conversion of individual documents.

If your output image format is TIFF, you can convert PDF to a single, multi-page TIFF document using the '--multipage' option (See 'How do I convert PDF to multi-page TIF?' for an example).

3.3.3 How do I convert PDF to JPEG, PNG, TIF or some other image format?

By default, PDF2Image automatically converts PDF to PNG. The output image format can be modified using the '-f' (or --format) option. For example,

```
pdf2image -f jpg in.pdf
```

will convert PDF to JPEG.

The '--format' parameter accepts any of the following output formats:

- **png** - (Portable Network Graphics)
- **png8** - (Palletized PNG)
- **jpg or jpeg** (Joint Photographic Expert Group)
- **tif or tiff** (Tagged Image File Format)
- **bmp** (Windows Bitmap Format)

- ### 3.3.4 How do I convert PDF to multi-page TIF?

For example:

3.3.5 How do I open a password protected PDF?

For unattended conversion, the password can also be specified directly on the command-line using the '-p' (or --password) option. For example:

The above command line will convert PDF to PNG and will use the provided password ('secret') to open the secured document (i.e. 'secured.pdf').

Note: PDF2Image supports all standard security options available in PDF, including 40 and 128 bit RC4 encryption, Crypt filters, and 128 AES (Advanced Encryption Standard) encryption.

3.3.6 How do I create grayscale images?

By default, PDF2Image uses the RGB color model for rasterization and image export. You can instruct PDF2Image to use single channel Device Gray color model for rasterization and image export using the '--gray' option. For example:

3.3.7 How do I specify which pages to convert?

By default, PDF2Image will rasterize and convert all PDF pages to output image format. You can specify a subset of pages to convert using the '-a' or '--pages' options. For example:

will convert only pages 1, 3, and 10. Please note that PDF2Image assumes that all pages are numbered sequentially starting from page 1.

To specify a range of pages, use dash character between numbers. For example:

```
pdf2image -a 1,10-20,50- in.pdf
```

will render the first page, pages in the range from 10 to 20 and all pages starting with page 50 to the last page in the document.

```
pdf2image --pages even in.pdf
```

```
pdf2image --pages odd,100- in.pdf
```

Using PDF2Image output image resolution can be specified explicitly (using the ‘-d’ or ‘--dpi’ option) or implicitly (using the ‘--hres’ and ‘--vres’ parameters). In this section, we cover the use of the ‘-dpi’ parameter. For more information on the ‘--hres’ and ‘--vres’ parameters, see ‘How do I specify dimensions of the output image in pixels?’

For example, to convert a PDF document to a multi-page TIF at 300 DPI (Dots Per Inch), use the following line:

```
pdf2image -f tif --multipage --dpi 300 in.pdf
```

Depending on the dimensions of the input page, high DPI/resolution rasterization requires lots of memory. For example, rasterization of a single A4 page (8x11) at 1000 DPI will require more than 350MB of memory. If PDF2Image fails to allocate enough memory, you can render the image in stripes or tiles, as described in ‘How do I render high-resolution images’, or by trying to decrease DPI value.

A 'typical' range of acceptable DPI values is between 1 and 1000 DPI. PDF2Image can rasterize images beyond 1000 or 2000 DPI using tiled or striped rendering.

To specify absolute dimensions of the output image in pixels, use the '--hres' and '--vres' parameters.

When these parameters are specified, PDF2Image will automatically determine the DPI (Dots Per Inch) ratio required to match the pixel dimensions of the output image.

For example, to generate 100 by 100 pixels thumbnails for a given PDF, you can use the following line:

```
pdf2image -f jpg --hres 100 --vres 100 in.pdf
```

Because the input PDF page may not perfectly fit the absolute pixel size of the output image, PDF2Image will also center the page and preserve the aspect ratio during rendering.

To generate images that are proportional in their size to the input PDF pages, simply omit one of the parameters (either `--hres` or `--vres`). For example,

```
pdf2image -f jpg --hres 100 in.pdf
```

will convert all PDF pages to images that are 100 pixels wide, with height proportional to the dimensions of the input page.

Similarly, the following line will create images with fixed height (100 pixels) and variable width (to preserve the aspect ratio).

```
pdf2image -f jpg --vres 100 in.pdf
```

3.3.10 How do I render only a subset of a given page?

Using PDF2Image you can rasterize a subset of a page using the '--clip' parameter. The parameter accepts a list of four numbers, separated using commas, giving the coordinates of a pair of diagonally opposite corners. Typically, the list takes the form: *llx, lly, urx, ury* specifying the lower-left x, lower-left y, upper-right x, and upper-right y coordinates of the rectangle, in that order. The other two corners of the rectangle are then assumed to have coordinates (*llx, ury*) and (*urx, lly*). All coordinates need to be expressed in points (a basic unit of PDF 'user' coordinate system). One PDF point is 1/72 of an inch and is approximately the same as a point (unit commonly used in the printing industry).

The '--clip' parameter is not only useful for cropping pages, but it can be also used to speed up the rendering process and to reduce memory consumption (see 'How to I render very large images?' for details).

PDF2Image also supports clipping to predefined page regions, such as page media, crop, trim, bleed, and art box. For more information on clipping to predefined regions, see 'How can I show/hide crop marks or the trim region?'

3.3.11 How do I render very large images?

Depending on the dimensions of the input page, high DPI/resolution rasterization requires lots of memory. For example, rasterization of a single A4 page (8x11) at 1000 DPI will require more than 350MB of memory. If PDF2Image fails to allocate enough memory (a single contiguous block of memory), you can render the image in stripes or tiles by repeatedly rendering different regions of the page using the '--clip' parameter (also see 'How do I render only a subset of a given page?').

For example, if the input page has a media box 0,0,595,842, you could render the page at 2000 DPI (Dots Per Inch) in four stripes (using 210.5 point increments along the Y axis) as follows:

```

pdf2image --dpi 2000 --clip 0,0,595,210.5 --prefix t01 Test/tiger.pdf
pdf2image --dpi 2000 --clip 0,210.5,595,421 --prefix t02 Test/tiger.pdf
pdf2image --dpi 2000 --clip 0,421,595,631.5 --prefix t03 Test/tiger.pdf
pdf2image --dpi 2000 --clip 0,631.5,595,842 --prefix t04 Test/tiger.pdf

```

Rendering of the same image in a single pass would require more than 1.4 GB in memory.

3.3.12 How do I batch convert files?

PDF2Image supports batch conversion of many PDF files in a single pass. To convert all PDF files in a given folder(s) you can use the following syntax:

```
pdf2image myfolder1
```

The '--subfolders' option can be used to recursively process all subfolders. For example, the following line will convert all documents in 'myfolder1' and 'myfolder2' as well as all subfolders:

```
pdf2image --subfolders myfolder1 myfolder2
```

By default, PDF2Image will convert all files with the extension '.pdf'. To select different files based on the extension use the '--extension' parameter. For example, to convert all PDF documents with a custom extension '.blob', you could use the following line:

```
pdf2image --extension .blob --subfolders myfolder1
```

3.3.13 How do I specify compression ratio for JPEG format?

The JPEG image format offers a lossy type of compression and the option to trade between the loss in image quality and compression ratio. To fine-tune JPEG compression quality, use the '--quality' parameter as illustrated in the following sample:

```
pdf2image --quality 80 -f jpg Test/tiger.pdf
```

Compression quality is a number in the range from 1 to 100. Lower numbers usually result in better compression at the expense of image quality. The default is 80.

3.3.14 How can I rotate pages?

By default, PDF2Image respects page rotation attribute. Image rotation can be modified using the '-r' (or --rotate) option. For example, the following line rotates all pages 90 degrees counterclockwise:

```
pdf2image --rotate 90 Test/tiger.pdf
```

Similarly, the following line rotates the page 270 degrees counterclockwise (or 90 degrees clockwise):

```
pdf2image --rotate 270 Test/tiger.pdf
```

3.3.15 How can I show/hide crop marks or the trim region?

A PDF page can define as many as five separate boundaries to control various aspects of the imaging process:

- The media box defines the boundaries of the physical medium on which the page is to be printed. It may include any extended area surrounding the finished page for bleed, printing marks, or other such purposes. It may also include areas close to the edges of the medium that cannot be marked because of physical limitations of the output device. Content falling outside this boundary can safely be discarded without affecting the meaning of the PDF file.
- The crop box defines the region to which the contents of the page are to be clipped (cropped) when displayed or printed. Unlike the other boxes, the crop box has no defined meaning in terms of physical page geometry or intended use; it merely imposes clipping on the page contents. The default value is the page's media box.
- The bleed box defines the region to which the contents of the page should be clipped when output in a production environment. This may include any extra bleed area needed to accommodate the physical limitations of cutting, folding, and trimming equipment. The default value is the page's crop box.

- By default, PDF2Image uses the page crop box as a default clip region. Different page regions can be selected as the default clip region using the `-b` (or `--box`) parameter. For example, the following line will instruct PDF2Image to use the media box for rasterization:

3.3.16 How do I render PDF as CCITT Group 4 FAX TIFF or monochrome PNG?

```
pdf2image --mono -f TIFF --verb 10 --dpi 300 in.pdf
```

PDF2Image is a completely stand alone application and does not include any dependencies on third-party components or software.

To provide additional feedback, PDF2Image returns exit codes after completing processing. The exit codes can be used to provide user feedback, for logging etc. This is particularly important for applications running in an unattended environment.

Exit Code	Description
0	All files converted successfully.
1	Document is secured. Need a valid password to open the document.
2	Error opening the input file(s).
3	An unknown exception encountered.

The following illustrates a sample Windows batch script that processes exit codes:

4.1 Reporting Problems

When submitting a problem you will be asked to provide the following information:

- ## 4.2 Contact Information

Tel: 1-604-730-8989
Fax: 1-604-676-2477

Email Contacts:

General Business Inquiries: info@pdftron.com
Sales & Licensing: sales@pdftron.com
Product Support: support@pdftron.com
Professional Services: services@pdftron.com
Website related questions: webmaster@pdftron.com
Press & News: press@pdftron.com