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: February 1, 2010

TABLE OF CONTENTS

1. Introduction

1.1 An Introduction to PDFTron PDF2Image

PDFTron's **PDF2Image** is an easy-to-use, stand-alone command-line application that provides users with an efficient means of converting PDF documents to various raster image file formats. PDF2Image can currently export to PNG, PNG8, JPEG, TIFF, BMP, and RAW, while providing a wide range of options to control the output image size and quality.

Like other PDFTron products, PDF2Image does not rely on any other third-party software. PDF2Image can be used in server environments or as a batch conversion process. Please see http://www.pdftron.com/pdf2image for more information.

1.1.1 Key Functions

- Convert PDF to PNG, PNG8, JPEG, TIFF, BMP, and RAW.
- Support for all versions of the PDF standard, including PDF 1.8 and Acrobat 9 documents (including ISO PDF 32000).
- Full support for encrypted documents (40 and 128 bit RC4 and 128 bit AES).
- Support for Unicode and all PDF font formats.
- Batch conversion.
- Options to fine tune image rendering and smoothing.
- Options to render specific page subsets and page ranges.
- Option to rasterize different page regions.
- Options to control annotation and forms rendering.
- Options to control target compression algorithm and quality.
- Support for rasterizing high resolution images as tiles or stripes with user defined clip regions.
- Files with broken cross reference tables are automatically repaired.
- Configuration file for frequently used options.

1.1.2 Common Use Case Scenarios

- Server-based, on-demand conversion of PDF documents to raster images.
- Batch processing of PDF collections with same rasterization options. PDF2Image is particularly useful in assembling product catalogues and brochures.
- Thumbnail generation for PDF documents.

1.1.3 Operating Systems Supported

- Windows 7, 2008, Vista, XP, 2003, 2000, NT, 98
- Mac OSX
- Linux

1.1.4 System Requirements

- At least 10 MB of free disk space.
- Memory requirement is heavily dependent on the nature of the document being converted into an image file.

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/pdfnet 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.

2. Installing and Uninstalling PDF2Image

2.1 PDF2Image Installation

PDF2Image Command-line Application is supplied as a download from a distributor or directly from www.pdftron.com. The release is packaged as a .zip file. (pdf2image.zip). To install the software, simply unzip the archive in the desired location and make sure to preserve the directory (folder) structure during this process. In order to register the software, copy the license file provided to you into the "pdf2image" folder.

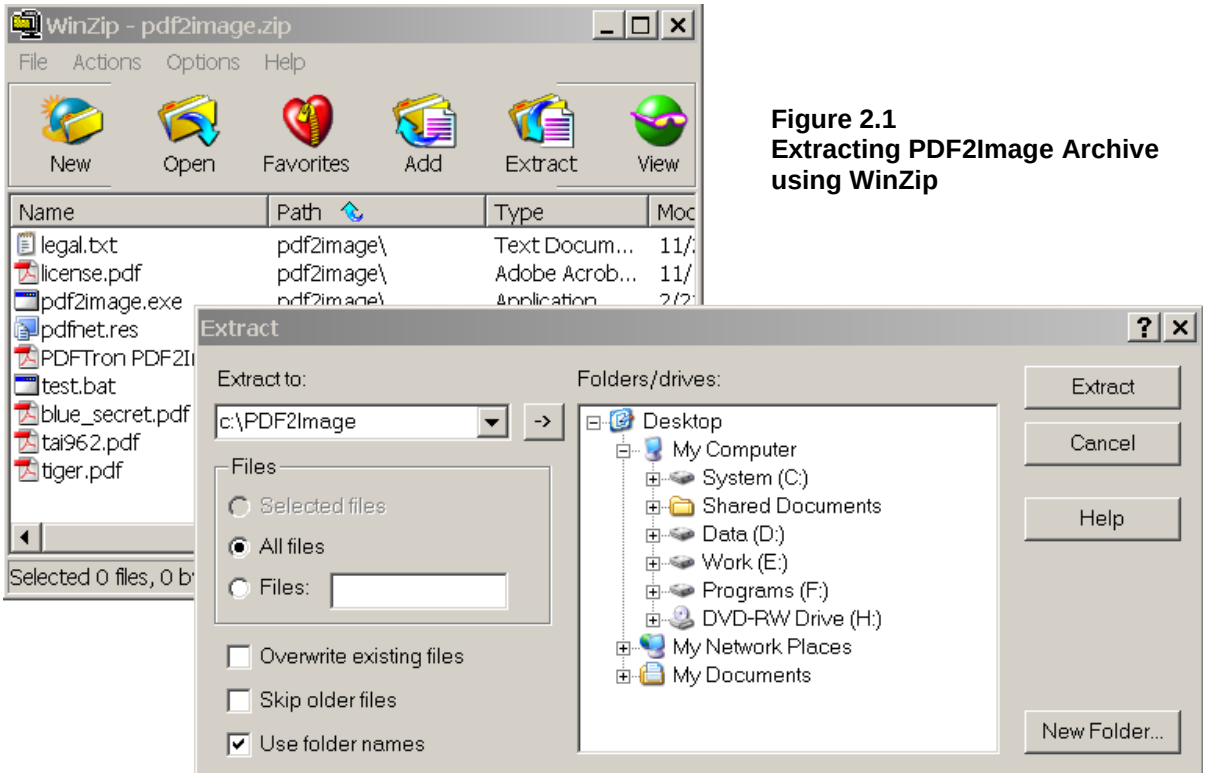

Figure 2.1
Extracting PDF2Image Archive using WinZip

2.2 Demo Version Installation

If you wish to evaluate the product, you can download the demo version of the product without any serial number or license key.

To do this, go to PDFTron's **Downloads** page at www.pdftron.com/downloads.html. Click on the appropriate product version/name, which will bring you to the product and the appropriate link for the demo download. Simply Download the zip file (pdf2image.zip) and extract the archive in the desired location, while making sure to preserve the directory (folder) structure when extracting the archive. This will provide you with a working copy of the application. The limitation of the evaluation version is that all pages in processed documents will have a demo stamp.

2.3 Registering PDF2Image on your Server(s)

When you are ready to switch to production, and you want to get rid of the watermarking feature of the demo version, follow these steps to register DocPub and sign up for an On-Premise PDFTron Web Services (PWS) pay-as-you-go account.

1. Go to https://api.pdftron.com
2. Click on the “Sign up” button and fill in the registration details.
3. Shortly after the registration you will receive a confirmation email with an activation link and your API ID and API Secret, which is required to connect to Cloud API from your app.
4. To activate subscription, log into your account and proceed to “Subscription” page to select your account plan.
5. Select the plan that meets your projected conversion volume and fill in your billing info.
6. Open ‘docpub.lic’ in a text editor.
7. Fill-in API_ID and API_Secret that you obtained via email (in step 3). You can also find the same information under the 'Account' section in your PDFTron Cloud account. For example:

```
#------------------------------------
# License Information:
#------------------------------------
API_ID = your_API_ID_here
API_Secret = your_API_Secret_here
#------------------------------------
# To register the Software, save the attachment in the 'docpub' folder.
# When extracting the archive, please make sure to preserve the directory (folder)
```

8. Once registered, any conversions you perform will be free from demo stamps and will be reported in your PDFTron Cloud management console (https://api.pdftron.com/v2/console).

2.4 Uninstalling PDF2Image

To remove PDF2Image from a computer, simply delete the “pdf2image” folder.

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.

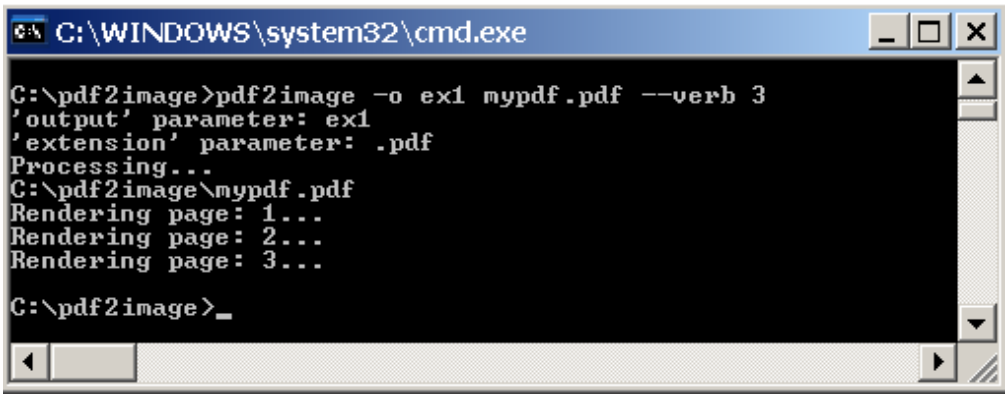

Figure 3.0 PDF2Image Command-line Application.

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: ▪ **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) The default output format is **png**.
-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 Render all even pages: -a even Render pages in the range from 3-11 and page 50: --pages 3-11,50 Render all odd pages	Specifies the list of pages to convert. By default, all pages are converted.

	and all pages in the range from 100 to the last page: -a odd,100-	
-b or --box	-b media	Specifies the page box/region to rasterize. Possible values are: ▪ **media** ▪ **crop** ▪ **trim** ▪ **bleed** ▪ **art** By default, page crop region will be rasterized.
-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	Rotates all pages by a given number of degrees, counterclockwise. The allowed values are: ▪ **0** ▪ **90** ▪ **180** ▪ **270** The default value is 0.
-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)

- **raw** (raw RGB or Gray data)

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

If your output image format is TIFF, you can convert PDF to a single, multi-page TIFF document instead of a separate file for every page using the '--multipage' option.

For example:

```
pdf2image --multipage -f tif --verb 3 mypdf.pdf
```

3.3.5 How do I open a password protected PDF?

PDF2Image will, without user intervention, decrypt and convert documents secured with a master/owner password. If the document is secured using a user (or file open) password, PDF2Image will prompt you to enter the password.

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

```
pdf2image -p secret -f png secured.pdf
```

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:

```
pdf2image -f tiff --gray in.pdf
```

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:

```
pdf2image -a 1,3,10 in.pdf
```

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.

All even pages can be selected using the ‘e’ (or ‘even’) string. For example, the following line renders all even pages:

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

Similarly odd pages can be selected using the ‘o’ (or ‘odd’) string. The following line renders all odd pages in the document and every page in the range from 100 to the last page:

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

3.3.8 How do I specify the resolution of the output image?

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?’

By default, PDF2Image uses resolution of 92 Dots Per Inch (DPI), which is the typical screen resolution. Smaller DPI numbers result in smaller images (e.g. suitable for use as thumbnails), while larger DPI numbers generate larger images (e.g. suitable for high-quality output).

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.

3.3.9 How do I specify dimensions of the output image in pixels?

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 in 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.

- The trim box defines the intended dimensions of the finished page after trimming. It may be smaller than the media box to allow for production related content, such as printing instructions, cut marks, or color bars. The default value is the page's crop box.
- The art box defines the extent of the page's meaningful content (including potential white space) as intended by the page's creator. 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:

```
pdf2image --box media in.pdf
```

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

To render the PDF as a monochrome (1 bit per pixel) image compressed using G4 CCITT, simply add the option '--mono' within the command-line string. For example,

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

3.3.17 Does PDF2Image have any dependencies on third party components/software?

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

3.4 General Usage Examples

Example 1. The simplest command line: Convert PDF to PNG.

Notes:

- This command line heavily relies on PDF2Image defaults. The default output image format is PNG and all images will be rendered at 92 Dots Per Inch (DPI).
- The '–o' (or –output) parameter is used to specify the output folder. If this option was not specified, all images would be stored in the current working folder.

```
pdf2image -o ex1 test/tiger.pdf
```

Example 2. Convert PDF to JPEG at 300 DPI and higher compression.

Notes:

- The '-d' (or –dpi) parameter is used to specify the output image resolution.
- The '-f' (or –format) parameter specifies that the output format is JPEG.
- The '--verb' option instructs pdf2image to output more feedback in the console window.

```
pdf2image --output ex2 -d 300 -f jpg --verb 3 --quality 60 test/tiger.pdf
```

Example 3. Convert a password protected file to a TIFF file of given pixel dimensions.

Notes:

- In this example, the '--hres' and '--vres' parameters are used instead of the '–dpi' parameter to specify absolute dimensions of the target image.

```
pdf2image -p secret -o ex3 --hres 1000 --vres 1000 -f tif --verb 3
test/blue_secret.pdf
```

Example 4. Convert PDF to multi-page TIFF.

To convert a PDF file to a multi-page TIFF, you can use the following lines:

```
pdf2image -o OUT2 --multipage -f tif --gray --verb 4 D:\
pdf2image --subfolders -o OUT2 --multipage -f tif --verb 3 "D:\My PDF"
```

3.5 Batch Processing and the Use of Wildcards

PDF2Image supports processing of multiple input documents in the same run. For example, it is possible to specify multiple PDF folders and PDF2Image will automatically process all PDF documents matching a given file extension. For example, the following command-line will process all PDF documents in folders ‘test1’ and ‘test2’

```
c:\>pdf2image -o c:/output_folder c:/test1 c:/test2
```

Wildcard characters can also be used to process multiple input files.

For example, if a directory contains the following PDF documents:

```
C:\test1 >dir
 Directory of C:\test1
01/04/2007  03:35 PM    <DIR>          .
01/04/2007  03:35 PM    <DIR>          ..
05/21/2004  02:27 PM            A1.pdf
05/03/2005  09:38 AM            A2.pdf
05/20/2003  08:46 AM            B1.pdf
05/15/2003  12:50 PM            B2.pdf
```

To process all PDF documents in this folder, you could specify:

```
c:\>pdf2image -o c:/output_folder c:/test1/*.pdf
```

To process all PDF documents staring with ‘A’, you could specify:

```
pdf2image -o c:/output_folder c:/test1/A*.pdf
```

Or to process all PDF documents ending with ‘1’, you could specify:

```
pdf2image -o c:/output_folder c:/test1/*1.pdf
```

You can use either of the two standard wildcards — the question mark (?) and the asterisk (*) — to specify filename and path arguments on the command line.

The wildcards are expanded in the same manner as operating system commands. (Please refer to your operating system user's guide if you are unfamiliar with wildcards). Enclosing an argument in double quotation marks (" ") suppresses the wildcard expansion. Within quoted arguments, you can represent quotation marks literally by preceding the double-quotation-mark character with a backslash (****). If no matches are found for the wildcard argument, the argument is passed literally.

3.6 Exit Codes

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.

The following table lists possible exit codes and their description:

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.

All codes other then '0' indicate that there was an error during the conversion process.

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

```
@echo off rem convert all PDF files in 'data' folder
pdf2image ./data
if errorlevel 1 goto passwd
if errorlevel 2 goto inputerr
if errorlevel 3 goto othererror
if errorlevel 0 goto exit

:passwd
echo Document is protected. Need a valid password to open the document.
goto exit

:inputerr
echo No input files specified.
goto exit

:othererror
echo An error encountered during processing.
goto exit

:exit
```

4. Support

4.1 Reporting Problems

If you encounter a problem or question regarding PDFTron PDF2Image, which is not addressed on PDFTron's website, please submit a problem report to PDFTron's Support group at http://www.pdftron.com/reportproblem.html.

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

- Contact details
- Product and Version of the product
- Detailed description of problem
- Problem file(s)
- Whether you have an AMS (Annual Maintenance Subscription)
- Any other information that may be related

4.2 Contact Information

To contact PDFTron directly, please use the contact information below:

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

Web site: www.pdftron.com

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