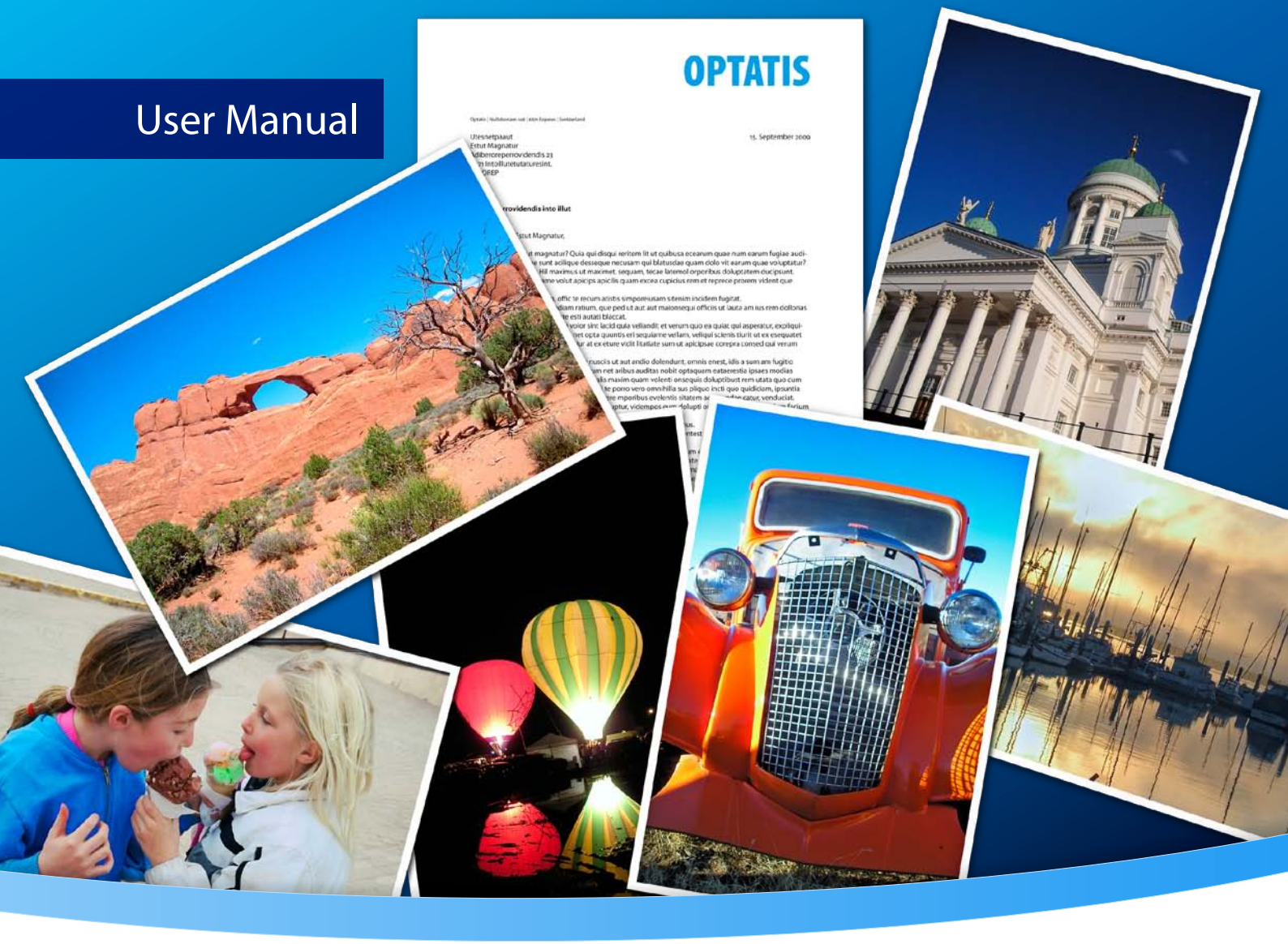
User Manual



3-Heights®
PDF to Image Converter Shell

Rendering Engine 2.0

PDF/A conform

Version 6.27.2

pdftools

Contents

1 Introduction 4
1.1 Description 4
1.2 Functions 4
1.2.1 Features 4
1.2.2 Formats 4
1.3 Operating systems 5

2 Installation 6
2.1 Windows 6
2.1.1 How to set the environment variable “Path” 6
2.2 Linux and macOS 7
2.2.1 Linux 7
2.3 Uninstall 8
2.4 Note about the evaluation license 8
2.5 Special directories 8
2.5.1 Directory for temporary files 8
2.5.2 Cache directory 8
2.5.3 Font directories 9

3 License management 10

4 User guide 11
4.1 Getting started 11
4.2 Specify image type, file name, and output folder 11
4.3 Single-page or multi-page images 12
4.4 Reducing the file size 12
4.4.1 Dimensions 13
4.4.2 Resolution 13
4.4.3 Bits per pixel 13
4.4.4 Format/Compression type 14
4.4.5 Image content, dithering 15
4.5 Options for best results on a printer 15
4.6 Color profiles 15
4.6.1 Default color profiles 16
4.6.2 Get other color profiles 16
4.7 Fonts 16
4.7.1 Font cache 16
4.7.2 Microsoft core fonts on Linux or macOS 17
4.7.3 Font configuration file fonts.ini 17
4.8 Dithering 18
4.8.1 Remarks 18
4.8.2 Color images 19
4.8.3 Bitonal images 20
4.8.4 Guidelines 22

5 Interface reference ... 24
5.1 Supported codecs ... 24
5.2 Rendering options ... 25
5.2.1 `-1` Create one image file per PDF page ... 25
5.2.2 `-b` Set the bits per pixel ... 25
5.2.3 `-c` Compression type of TIFF images ... 25
5.2.4 `-cms` Set the color management engine ... 26
5.2.5 `-cn` Center mode ... 27
5.2.6 `-cs` Set the color space ... 27
5.2.7 `-d` Set the resolution in DPI ... 28
5.2.8 `-dx` Set the X resolution in DPI ... 28
5.2.9 `-dy` Set the Y resolution in DPI ... 28
5.2.10 `-f` Fit page mode ... 28
5.2.11 `-fax` Convert to Class F ... 29
5.2.12 `-fo` Bit fill order ... 29
5.2.13 `-fs` Filter size ... 29
5.2.14 `-g` Gray color space ... 30
5.2.15 `-h` Dithering mode ... 30
5.2.16 `-i` Indexed color space ... 30
5.2.17 `-m` Rendering mode ... 31
5.2.18 `-oc` Disable black point compensation (BPC) ... 31
5.2.19 `-of` Fit rectangle paths to pixel grid ... 31
5.2.20 `-oh` Disable hinting ... 31
5.2.21 `-ohs` Manual character size limit for hinting ... 31
5.2.22 `-op` Set print mode ... 32
5.2.23 `-oq` Disable high quality rendering ... 32
5.2.24 `-p` Read an encrypted PDF file ... 32
5.2.25 `-pbt` Set page box type ... 32
5.2.26 `-pi` Print page information ... 33
5.2.27 `-pg` Set page range ... 33
5.2.28 `-pgs` Set of pages ... 33
5.2.29 `-q` Set image quality ... 34
5.2.30 `-ri` Ignore page rotate attribute ... 34
5.2.31 `-rl` Rotate pages to landscape ... 34
5.2.32 `-rp` Rotate pages to portrait ... 34
5.2.33 `-s` Set width and height of image in points ... 34
5.2.34 `-sa` Set width and height in pixels and preserve ratio ... 35
5.2.35 `-so` Page offset in points ... 35
5.2.36 `-sp` Set width and height of image in pixel ... 35
5.2.37 `-t` Set threshold when dithering is disabled ... 35
5.3 General options ... 36
5.3.1 `-v` Verbose mode ... 36
5.3.2 `-lk` Set license key ... 36
5.4 Return codes ... 36

6 Troubleshooting ... 37
6.1 Output ... 37
6.1.1 Generated files are large size ... 37
6.1.2 Images are too low quality ... 37
6.1.3 Image does not contain the whole content ... 37
6.1.4 Colors are gone ... 37
6.2 Font and text issues ... 37

6.2.1 Handle non-embedded fonts 37
Font replacement strategy 37

7 Version history 39
7.1 Changes in versions 6.19–6.27 39
7.2 Changes in versions 6.13–6.18 39
7.3 Changes in versions 6.1–6.12 39
7.4 Changes in version 5 39
7.5 Changes in version 4.12 39
7.6 Changes in version 4.11 39
7.7 Changes in version 4.10 40
7.8 Changes in version 4.9 40
7.9 Changes in version 4.8 40

8 Licensing, copyright, and contact 41

1 Introduction

1.1 Description

The 3-Heights® PDF to Image Converter Shell converts PDF documents into single-page or multi-page raster images such as TIFF or JPEG. It can also convert PDF files into rasterized PDF.

Its areas of use include the web, TIFF-based DMS solutions, archiving and workflow systems, and the protection of PDF documents. The converter is characterized by its high speed and outstanding quality.

1.2 Functions

The 3-Heights® PDF to Image Converter Shell merges pages from different input files to form one or more files. Color space and image size are defined automatically during the process. The converter supports scaled and un-scaled conversions and a variety of image formats such as PNG, TIFF, JBIG2, or JPEG2000.

1.2.1 Features

PDF to image

- Create single-page and multi-page image files and rasterized PDF documents
- Convert individual pages
- Convert PDF files to CCITT fax files
- Define page dimensions in points or pixels
- Set rotation (Force portrait or landscape or inherit rotation from original document)
- Set resolution (DPI)
- Add dithering (Floyd Steinberg, Halftone Block, Halftone Continuous, Atkinson)
- Set image filters
- Set color depth
- Set color space
- Set TIFF file compression
- Set the quality of lossy image compression
- Set bit filling order for fax files
- Add watermark images

PDF to PDF image

- Raster PDF content (image)
- Keep or remove links, outlines or viewer preferences in PDF output document

1.2.2 Formats

Input formats

- PDF 1.x (PDF 1.0, …, PDF 1.7)
- PDF 2.0

- PDF/A-1, PDF/A-2, PDF/A-3

Output formats

- TIFF (Tagged Image File Format)
- JPEG (Joint Photographic Expert Group)
- PNG (Portable Network Graphics)
- GIF (Graphics Interchange Format)
- BMP (Window Bitmap)
- EPS (Encapsulated PostScript)
- JBIG2 (Joint Bi-level Image Experts Group)
- JPEG2000
- Extended JPEG2000
- PBM (Portable Bitmap File Format)

Conformance

Standards:

- ISO 32000-1 (PDF 1.7)
- ISO 32000-2 (PDF 2.0)
- ISO 19005-1 (PDF/A-1)
- ISO 19005-2 (PDF/A-2)
- ISO 19005-3 (PDF/A-3)

1.3 Operating systems

The 3-Heights® PDF to Image Converter Shell is available for the following operating systems:

- Windows Client 7+ | x86 and x64
- Windows Server 2008, 2008 R2, 2012, 2012 R2, 2016, 2019, 2022 | x86 and x64
- Linux:
  - Red Hat, CentOS, Oracle Linux 7+ | x64
  - Fedora 29+ | x64
  - Debian 8+ | x64
  - Other: Linux kernel 2.6+, GCC toolset 4.8+ | x64
- macOS 10.10+ | x64

'+' indicates the minimum supported version.

2 Installation

2.1 Windows

The 3-Heights® PDF to Image Converter Shell comes as a ZIP archive or as an MSI installer.

To install the software, proceed as follows:

1. You need administrator rights to install this software.
2. Log in to your download account at https://www.pdf-tools.com. Select the product "PDF to Image Converter Shell". If you have no active downloads available or cannot log in, please contact pdfsales@pdf-tools.com for assistance.
   You can find different versions of the product available. Download the version that is selected by default. You can select a different version.
   There is an MSI (`*.msi`) package and a ZIP (`*.zip`) archive available. The MSI (Microsoft Installer) package provides an installation routine that installs and uninstalls the product for you. The ZIP archive allows you to select and install everything manually.
   There is a 32 and a 64-bit version of the product available. While the 32-bit version runs on both 32 and 64-bit platforms, the 64-bit version runs on 64-bit platforms only. The MSI installs the 64-bit version, whereas the ZIP archive contains both the 32-bit and the 64-bit version of the product. Therefore, on 32-bit systems, the ZIP archive must be used.
3. If you select an MSI package, start it and follow the steps in the installation routine.
4. If you are using the ZIP archive, uU nzip the archive to a local folder, e.g. `C:\Program Files\PDF Tools AG\`.
   This creates the following subdirectories:

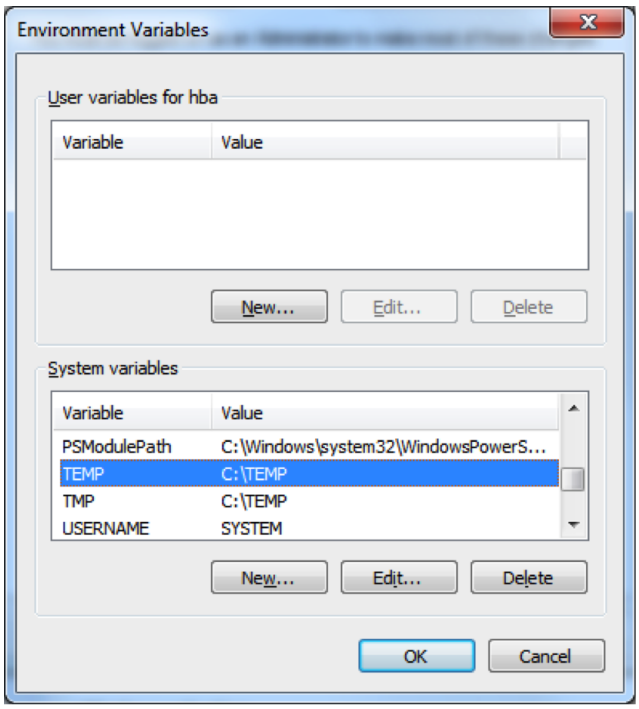
Subdirectory	Description
`bin`	Runtime executable binaries
`doc`	Documentation

5. (Optional) To easily use the 3-Heights® PDF to Image Converter Shell from a shell, the directory needs to be included in the "Path" environment variable.
6. (Optional) Register your license key using the License management.
7. Ensure the cache directory exists as described in Special directories.
8. Make sure your platform meets the requirements regarding color spaces and fonts described in Color profiles and Fonts, respectively.

2.1.1 How to set the environment variable "Path"

To set the environment variable "Path" in Windows, go to Start → Control Panel (classic view) → System → Advanced → Environment Variables.

Select "Path" and "Edit", then add the directory where `pdf2imgR2.exe` is located to the "Path" variable. If the environment variable "Path" does not exist, create it.



2.2 Linux and macOS

This section describes installation steps required on Linux or macOS.

Here is an overview of the files that come with the 3-Heights® PDF to Image Converter Shell:

File description

Name	Description
`bin/x64/pdf2imgR2`	Main executable
`doc/*.*`	Documentation

2.2.1 Linux

1. Unpack the archive in an installation directory, e.g. `/opt/pdf-tools.com/`
2. Verify that the GNU shared libraries required by the product are available on your system:

```
ldd pdf2imgR2
```

   If the previous step reports any missing libraries, you have two options:
   a. Download an archive that is linked to a different version of the GNU shared libraries and verify whether they are available on your system. Use any version whose requirements are met. Note that this option is not available for all platforms.
   b. Use your system's package manager to install the missing libraries. It usually suffices to install the package `libstdc++6`.
3. Create a link to the executable from one of the standard executable directories, e.g.

```
ln -s /opt/pdf-tools.com/bin/x64/pdf2imgR2 /usr/bin
```

4. Optionally, register your license key using the license manager.
5. Ensure the cache directory exists as described in Special directories.
6. Make sure your platform meets the requirements regarding color spaces and fonts described in Color profiles and Fonts, respectively.

2.3 Uninstall

If you have used the MSI for the installation, go to Start → 3-Heights® PDF to Image Converter Shell… → Uninstall …

If you have used the ZIP file for the installation, undo all the steps done during installation.

2.4 Note about the evaluation license

With the evaluation license, the 3-Heights® PDF to Image Converter Shell automatically adds a watermark to the output files.

2.5 Special directories

2.5.1 Directory for temporary files

This directory for temporary files is used for data specific to one instance of a program. The data is not shared between different invocations and is deleted after termination of the program.

The directory is determined as follows. The product checks for the existence of environment variables in the following order and uses the first path found:

Windows

1. The path specified by the `%TMP%` environment variable
2. The path specified by the `%TEMP%` environment variable
3. The path specified by the `%USERPROFILE%` environment variable
4. The Windows directory

Linux and macOS

1. The path specified by the `$PDFTMPDIR` environment variable
2. The path specified by the `$TMP` environment variable
3. The `/tmp` directory

2.5.2 Cache directory

The cache directory is used for data that is persistent and shared between different invocations of a program. The actual caches are created in subdirectories. The content of this directory can safely be deleted to clean all caches.

This directory should be writable by the application; otherwise, caches cannot be created or updated and performance degrades significantly.

Windows

- If the user has a profile:
  `%LOCAL_APPDATA%\PDF Tools AG\Caches`
- If the user has no profile:
  `<TempDirectory>\PDF Tools AG\Caches`

Linux and macOS

- If the user has a home directory:
  `~/.pdf-tools/Caches`
- If the user has no home directory:
  `<TempDirectory>/pdf-tools/Caches`

where `<TempDirectory>` refers to the Directory for temporary files.

2.5.3 Font directories

The location of the font directories depends on the operating system. Font directories are traversed recursively in the order as specified below.

If two fonts with the same name are found, the latter one takes precedence, i.e. user fonts always take precedence over system fonts.

Windows

1. `%SystemRoot%\Fonts`
2. User fonts listed in the registry key `\HKEY_CURRENT_USER\Software\Microsoft\Windows NT\CurrentVersion\Fonts`. This includes user specific fonts from `C:\Users\<user>\AppData\Local\Microsoft\Windows\Fonts` and app specific fonts from `C:\Program Files\WindowsApps`
3. `Fonts` directory, which must be a direct subdirectory of where `pdf2imgR2.exe` resides.

macOS

1. `/System/Library/Fonts`
2. `/Library/Fonts`

Linux

1. `/usr/share/fonts`
2. `/usr/local/share/fonts`
3. `~/.fonts`
4. `$PDFFONTDIR` or `/usr/lib/X11/fonts/Type1`

3 License management

The 3-Heights® PDF to Image Converter Shell requires a valid license in order to run correctly. If no license key is set or the license is not valid, then the executable will fail and the return code is set to 10.

More information about license management is available in the license key technote.

4 User guide

The 3-Heights® PDF to Image Converter Shell provides the executable `pdf2imgR2`

`pdf2imgR2` is used to convert PDF files to raster images, such as TIFF, JPEG, etc. The functionality of this executable is documented in this manual.

All switches are described in the usage of the tool. (Type `pdfimgR2` with no arguments to list the usage).

4.1 Getting started

The simplest command requires one parameter: The name of the PDF input file. When no output file name is specified, the output file is named as the input file, and the image type TIFF is selected.

Example: The following command

```
pdf2imgR2 input.pdf
```

creates an uncompressed TIFF file. It inherits the name of the input file and is named:

```
input.tif
```

If the PDF file has more than one page, the generated TIFF is a multi-page TIFF. Keep in mind that TIFF supports multi-paging, whereas most other image formats are single-page formats.

4.2 Specify image type, file name, and output folder

The name and type of the output file can be specified using a second parameter. Here is a list of supported extensions and the corresponding file type:

File formats

Extension	File format
`.tif, .tiff`	Tagged Image File Format
`.jpg, .jpe, .jpeg`	Joint Photographic Expert Group (JPEG)
`.png`	Portable Network Graphics
`.gif`	Graphics Interchange Format
`.bmp`	Window Bitmap
`.jb2`	Joint Bi-level Image Experts Group
`.jp2`	JPEG2000
`.jpx`	Extended JPEG2000

File formats

`.pbm,.pgm,.pnm,.ppm`	Portable Bitmap File Format
`.eps`	Encapsulated PostScript (Output only)

Example: The following command creates a JPEG image in the current working folder

```
pdf2imgR2 input.pdf output.jpg
```

Example: The output directory can simply be added in front of the output file name

```
pdf2imgR2 input.pdf myfolder\output.jpg
```

or absolute

```
pdf2imgR2 input.pdf C:\myfolder\output.jpg
```

Example: Quotes must be used for paths or file names that contain blanks

```
pdf2imgR2 "My File.pdf" "My Documents\output.jpg"
```

4.3 Single-page or multi-page images

By default, the PDF to Image Converter generates multi-page images if the TIFF format is selected and the input PDF has more than one pages. To create one TIFF file per PDF page, use the `-1` switch. This option should also be used for all other formats that do not support multi-paging, if the input PDF document has more than one page.

When doing so, the output files can be named with wildcards.

Example: Create images consisting of 1 page per image, add the page number to the file name as four digit number:

```
pdf2imgR2 -1 input.pdf output%04d.png
```

Image files created this way are named `output0001.png`, `output0002.png`, `output0003.png`, etc.

4.4 Reducing the file size

There are different ways to reduce the file size of an image. You need to be aware that from a certain point on, a smaller file size results in a poorer visual quality.

The main factors on which the file size of an image depends are:

- Dimensions in pixel (width and height)
- Bits per pixel
- Compression type
- The content of the image (influenced by dithering)

4.4.1 Dimensions

Reducing the dimensions and therefore, the amount of the total pixels reduces also the file size. Obviously, a 1024x768 pixel image has a larger file size than an equivalent 600x480 image.

Example: Set the dimensions in pixels.

```
-sp 600 480 -f
```

Example: Set the dimension in points.

```
-s 600 480 -f
```

If the dimensions are set in points, the dimensions in pixel are computed depending on the resolution.

4.4.2 Resolution

The resolution in dots per inch (DPI) lets you specify how detailed the image is. The default value is 150 DPI, which generates an image that looks sharp when not aumented in zoom. A larger value generates a more detailed image, but also increases the file size, because it requires more pixels. On the other hand, a lower resolution generates a file with a smaller file size, but the image is also of lower visual quality.

Example: Setting the resolution value to 75 DPI instead of 150 DPI reduces the file size to about one quarter.

```
-d 75
```

4.4.3 Bits per pixel

Using 1-bit (black/white) or 8-bit grayscale instead of 24-bit true color reduces the file size. Keep in mind that not all formats support all color depths.

8-bit grayscale images are a third as large in size as 24-bit color images. With 1-bit images that use dithering, the size heavily depends on the content. It can be as small as 1% of the 8-bit image.

Example: Create a grayscale image.

```
-b 8
```

Example: Create a bi-level image with Atkinson dithering.

```
-b 1 -h 6 -oq
```

4.4.4 Format/Compression type

The 3-Heights® PDF to Image Converter Shell supports various image formats. For most formats, the compression is given. For example, a PNG image is always Flate-compressed, a JPEG image is always JPEG-compressed. However, for TIFF, the compression type is selectable.

Images formats that are supported by most Internet browsers are JPEG, GIF and PNG.

There are two fundamentally different types of compression: Lossless and lossy.

Lossless compression The transformation from the original to the compressed state of the image does not change the content. Thus, the transformation is reversible and the original image can be regained from the compression state.

Lossless compression is normally used for artificial images or scanned text. It is applied to the following types of images: GIF, PNG, BMP, JPEG2000 if quality is set to 100, JBIG2, and TIFF compressed with G3, G4, LZW or Flate.

Lossy compression The compression algorithm alters the content of the image in a way that it compresses better. Thus, a lossy compressed image cannot be reverted back to its original state. It also means multiple applications of lossy compression to the same image alter the image every time and thereby reduce the quality every time. The degree to which the image may be altered to improve the compression rate is controlled by a quality index ranging from 1 to 100 and normally defaulted at 75.

Lossy algorithms usually provide a better compression rate, at the cost of visual quality. Lossy compression is normally used for photographs.

It is applied to the following types of images: JPEG, and JPEG2000 if quality is less than 100.

There are various compression types supported for the TIFF image format. These are:

CCITT Group 3, Group 3-2D CCITT Group 3 is the predecessor to CCITT Group 4. It is a simpler algorithm that normally results in a lower compression ratio.

CCITT Group 4 CCITT Group 4 is the standard compression for bi-level TIFF images (i.e. facsimile).

LZW LZW (Lempel-Ziv-Welch) compression is a lossless compression algorithm for images.

See the copyright laws of your country prior to using this compression algorithm.

JPEG TIFF allows images to be compressed with JPEG, which is a lossy compression algorithm. JPEG provides a high compression ratio for 8 and 24 bit images. It is best suited for TIFFs containing photographs and little or no text.

ZIP (Flate) ZIP is a lossless compression algorithm. It is useful for the compression of large images with no loss in quality.

Flate compression (also used by the ZIP format) and JPEG compression can be used for color or grayscale images. CCITT Group 3, 3-2D and 4, as well as Flate can be used for black and white images.

Example: Apply Flate compression to a TIFF image.

```
-z
```

4.4.5 Image content, dithering

The content of the image itself has a direct impact on how well it compresses. It seems quite obvious that a plain white image compresses much better than a page filling photograph.

Dithering is an algorithm that arranges the pixels of an image in a way that it creates a visual effect of colors that do not exist in the available colors of the image, such as different grays in a 1-bit black and white image. This complex arrangement of pixels, however, does not compress well and increase the file size. Disabling dithering reduces the file size. In the 3-Heights® PDF to Image Converter Shell, dithering is also implemented for color images.

Example: Disable dithering (e.g. for scanned text).

```
-h 0
```

For more information, see Dithering.

4.5 Options for best results on a printer

Regular laser printers use a resolution of 600-1200 DPI. For best results on printers, choose a resolution of the same value. In this case, it is also important to generate black and white image and thus set the bits per pixel to 1. The command looks like this:

Example: Create a bitonal, 1200 DPI, CCITT G4 compressed TIFF.

```
pdf2imgR2 -b 1 -d 1200 -g4 input.pdf output.tif
```

An A4 black and white image with a resolution of 1200dpi is about 1MB in size.

> **Note:** that using such a high resolution in combination with 8bit grayscale or 24 bit color images generates huge files (several hundreds of Megabytes uncompressed, and around 10 Megabytes using JPEG compression).

DPI values larger than 2400 take a lot of CPU power and memory. You should not use values above 2400 for A4 paper size PDF documents.

4.6 Color profiles

A PDF document may contain graphical objects using various different color spaces and the output file of 3-Heights® PDF to Image Converter Shell may yet use another color space. Therefore often colors have to be converted between different color spaces.

For calibrated color spaces (such color spaces with an associated ICC color profile), the color conversion is well defined. For the conversion of uncalibrated device color spaces (DeviceGray, DeviceRGB, DeviceCMYK), however, the 3-Heights® PDF to Image Converter Shell requires apropriate color profiles. Therefore, it is important that the profiles are available and that they describe the colors of the device your input documents are intended for.

> **Note:** When setting an alternative color management system such as Neugebauer, no color profiles are required.

If no color profiles are available, default profiles for both RGB and CMYK are generated on the fly by the 3-Heights® PDF to Image Converter Shell.

4.6.1 Default color profiles

If no particular color profiles are set, default profiles are used. For device RGB colors, a color profile named `"sRGB Color Space Profile.icm"` and for device CMYK, a profile named `"USWebCoatedSWOP.icc"` are searched for in the following directories:

Windows

1. `%SystemRoot%\System32\spool\drivers\color`
2. directory `Icc`, which must be a direct subdirectory of where the `pdf2imgR2.exe` resides.

Linux and macOS

1. `$PDF_ICC_PATH` if the environment variable is defined
2. the current working directory

4.6.2 Get other color profiles

Most systems have pre-installed color profiles available. For example, on Windows at `%SystemRoot%\system32\spool\drivers\color\`. Color profiles can also be downloaded from the links provided in the directory `bin\Icc\` or from the following websites:

- https://www.pdf-tools.com/public/downloads/resources/colorprofiles.zip
- https://www.color.org/srgbprofiles.html

4.7 Fonts

PDF documents may contain both embedded and non-embedded fonts. When rendering non-embedded fonts, the best result can be achieved, if the font is available on the system. Therefore, it is important to make sure the Font directories contain all fonts required.

For more information on how to deal with font issues, please refer to Font and text issues.

Note that on Windows, when a font is installed, it is by default installed only for a particular user. It is important to either install fonts for all users, or make sure the 3-Heights® PDF to Image Converter Shell is run under that user and the user profile is loaded.

On Linux and macOS, it is recommended to install the Liberation fonts, Google Noto CJK fonts, and the OpenSymbol font. On Debian based systems, the packates are called `fonts-liberation2`, `fonts-noto-cjk`, and `fonts-opensymbol`.

4.7.1 Font cache

A cache of all fonts in all Font directories is created. If fonts are added or removed from the font directories, the cache is updated automatically.

In order to achieve optimal performance, make sure that the cache directory is writable for the 3-Heights® PDF to Image Converter Shell. Otherwise, the font cache cannot be updated and the font directories have to be scanned on each program startup.

The font cache is created in the subdirectory `<CacheDirectory>/Installed Fonts` of the Cache directory.

4.7.2 Microsoft core fonts on Linux or macOS

Many PDF documents use Microsoft core fonts like Arial, Times New Roman, and other fonts commonly used on Windows. Therefore, it is recommended to install these fonts to your default font directories. Many Linux distributions offer an installable package for these "Microsoft TrueType core fonts". For instance, on Debian based systems, the package is called `ttf-mscorefonts-installer`.

Alternatively, you can download the fonts from here:

https://corefonts.sourceforge.net/

Microsoft has an FAQ on the subject, that covers licensing related questions as well:

https://docs.microsoft.com/en-us/typography/fonts/font-faq

4.7.3 Font configuration file fonts.ini

The font configuration file is optional. It can be used to control the mapping of fonts used in the PDF to fonts preinstalled on the system.

The file `fonts.ini` must reside at the following location , which is platform dependent:

Windows: In a directory named `Fonts`, which must be a direct subdirectory of where `pdf2imgR2.exe` resides.

Unix: The `fonts.ini` file is searched in the following locations

1. If the environment variable `PDFFONTDIR` is defined: `$PDFFONTDIR/fonts.ini`
2. `~/.pdf-tools/fonts/fonts.ini`
3. `/etc/opt/pdf-tools/fonts/fonts.ini`

It consists of two sections: `[fonts]` and `[replace]`. Both sections are used to map fonts in the PDF to fonts in the installed font collection on the operating system. This comes into play when the font in the PDF document does not have an embedded font program, or the embedded font is not usable.

The mapping only works if the font types of the specified fonts are matching; for example, if the font in the PDF is a symbolic font, such as "Symbol" or "ZapfDingbats", the mapped font must be symbolic too.

The section `[fonts]` is only considered if the font-matcher does not find an appropriate font among the existing installed fonts. It is suggested to only use this section.

The section `[replace]` is stronger and applied before the font-matcher. This means a font will be replaced as defined, even if the correctly installed font is available on the system.

Syntax: The syntax of the mapping file is as follows

```
[fonts]
PDF_font_1=installed_font_1{,font_style}
PDF_font_2=installed_font_2{,font_style}
[replace]
PDF_font_n=installed_font_n{,font_style}
```

`PDF_font_*` is the name of the font in the PDF.

This name can be found in one of the following ways:

- Use any tool that can list fonts. Such as 3-Heights® PDF Extract or 3-Heights® PDF Optimizer. Ignore possible prefixes of font subsets. A subset prefix consists of 6 characters followed by the plus sign. For example "KHFOKE+MonotypeCorsiva", in this case only use "MonotypeCorsiva" as font name in the mapping file.

- Open the document with Adobe Acrobat, use the "MarkUp Text Tool", mark the text of which you would like to know the font name, right-click it, select "Properties..."

`installed_font_*` is the font family name of the installed font.

To retrieve this name, find the font in the Windows' font directory and open it by double-clicking. The first line in the property window displays the font family name (this may vary depending on the operating system). The font family name does not include font styles; so an example of a font family name is "Arial", but not "Arial Italic".

`font_style` is an optional style that is added coma-separated after the font family name.

The style is always one word. Examples of font styles are "Italic", "Bold", "BoldItalic". Omit the font style, if it is "Regular" or "Normal".

Remove blanks from all font names, i.e. in both the `PDF_font_*` and the `installed_font_*`.

Example:

```
[fonts]
Ryumin-Light=MSMincho
GothicBBB-Medium=MSGothic
[replace]
ArialIta=Arial,BoldItalic
```

4.8 Dithering

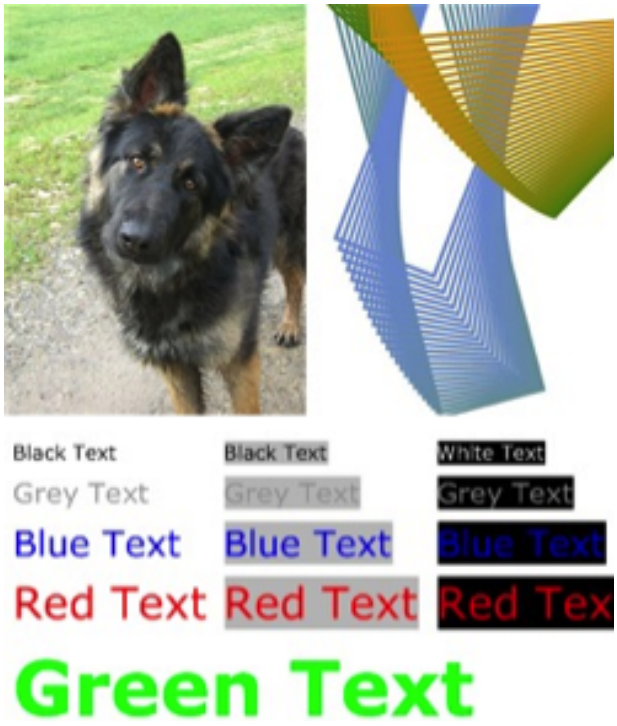
Dithering is a common means used in images to simulate colors that are not available as actual colors. Its use is best observed in image with a low color depth, where colors or shades of gray need to simulated with other colors (e.g. only black/white pixels).

4.8.1 Remarks

1. All images below have quite a low resolution. As a result, the effects of the different dithering types become more obvious. The higher the resolution and the large the number of colors, the higher the quality of the image.
2. The rendering filter and current zoom level of the PDF viewing application may have an additional impact on how the images below are displayed.

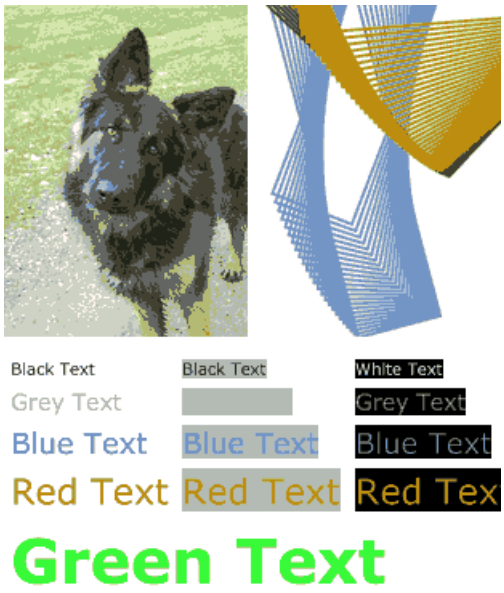
4.8.2 Color images

Color space	RGB (24 bit)
Dithering	None
File size as PNG	129 kB
+	Highest quality
-	Highest file size

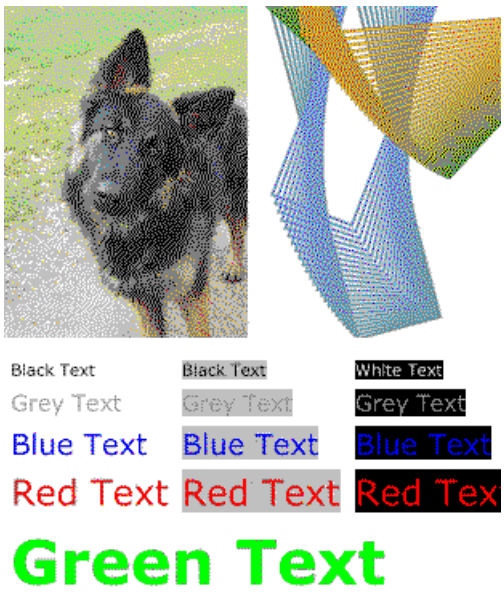


A 24 bit RGB color image can have up to 16.7 millions of different colors. Dithering does not need to be applied since all required colors exist and none need to be simulated.

Color space	16 colors (4 bit)
Dithering	None
File size as PNG	16 kB
+	Small file size
+	Works well for images with a small number of colors (artificial images, text)
-	Does not work well for images with lots of colors photographic images) - parts of the image can become plain-colored and details get lost



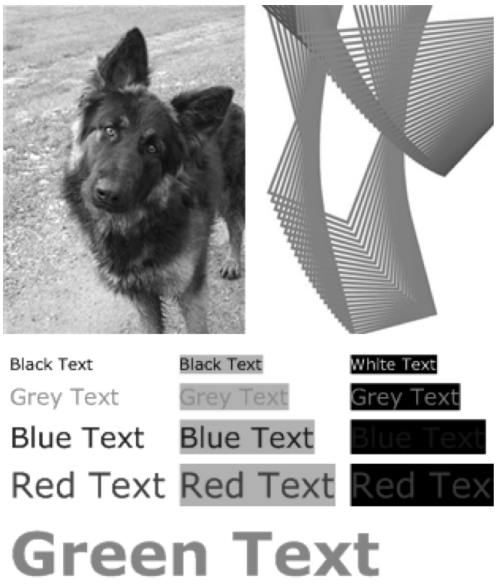
Color space	16 colors (4 bit)
Dithering	Floyd-Steinberg
File size as PNG	18 kB
+	Renders details better
+	Usually better overall quality, especially in photographic images than without dithering
-	Sometimes generates unwanted artifacts (striking pixels)
-	Larger file size then without dithering



4.8.3 Bitonal images

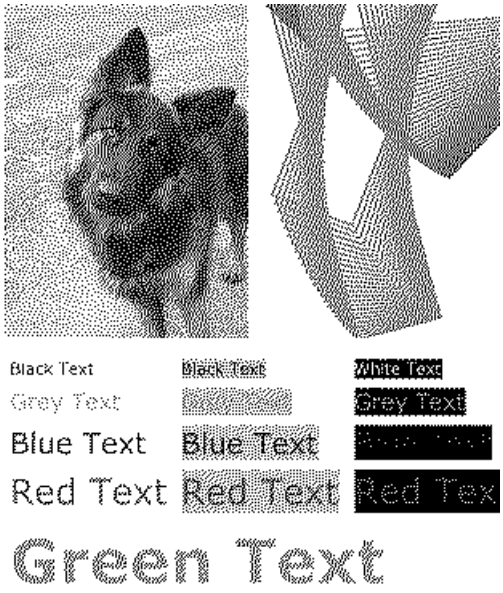
(The 8 bit image just acts as reference.)

Color space	Grayscale (8 bit)
Dithering	None
File size as PNG	46 kB

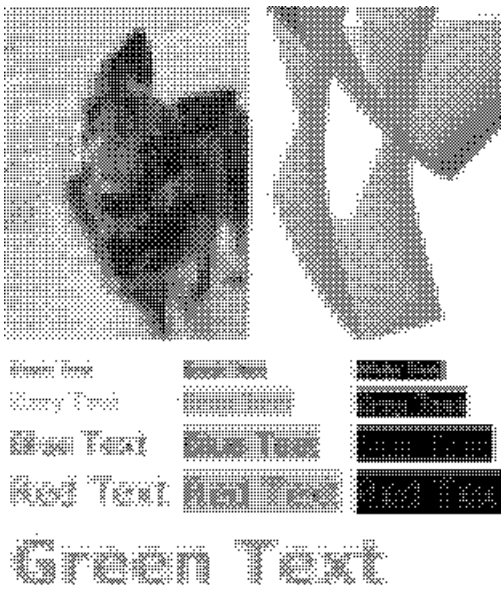


Color space	Grayscale (1 bit)
Dithering	None
File size as PNG	2.6 kB
+	Smallest file size
+	Works well for documents with high contrast (black text on white background)
+	Does not generate artifacts
-	Details get lost, because shades of gray are not approximated, but converted to either black or white (in fact images or part of them can become completely black or white)

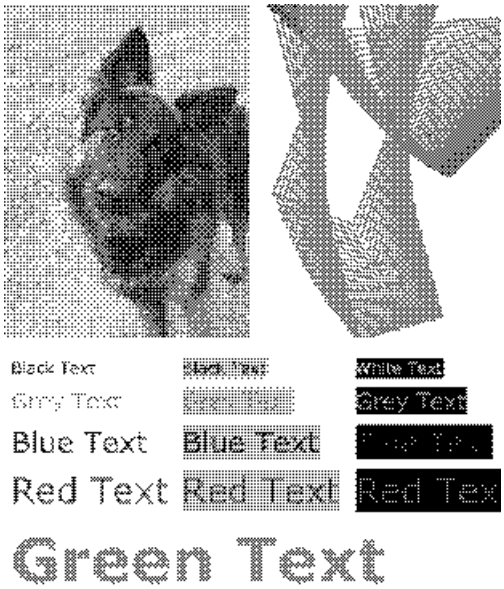
Color space	Grayscale (1 bit)
Dithering	Floyd-Steinberg
File size as PNG	9 kB
+	Generally higher quality, specially of photographic images
+	Can approximate any shade of gray
-	Larger file size than without dithering
-	Generates artifacts (e.g. a very bright gray paper is approximated by far-spread single black pixels)
-	Not well suited for text, unless the color of the text must be reflected



Color space	Grayscale (1 bit)
Dithering	Halftone
File size as PNG	4 kB
+	Small file size
+	Approximates shades of gray
-	Not well suited for text or artificial images



Color space	Grayscale (1 bit)
Dithering	Pattern
File size as PNG	5 kB
+	Works acceptable for all types of content (text, photographic images, artificial images)
-	Is not excellent in any type of content



4.8.4 Guidelines

As seen in the examples above, different types of dithering behave different for different types of content. Below are some suggestions on the dithering type that works best for a give type of content:

Text, OCR No dithering

Artificial images with few colors and no bright colors No dithering

Artificial images with many colors Test which dithering type yields the best result

Photographic images Floyd-Steinberg

Mixed content Test which dithering type yields the best result

Mixed content, high-resolution For resolutions above 300 DPI, Floyd-Steinberg almost always yields the best result (exception: for pure black text on white background, use no dithering)

Keep in mind that dithering should only be applied for images with a low color depth, such as black and white (1 bit). Dithering for images with a color depth of 8 bit or higher (256 colors or grayscale) has little to no visual impact.

5 Interface reference

5.1 Supported codecs

The following table lists the capabilities that the different codecs that are supported by the 3-Heights® PDF to Image Converter Shell. Other capabilities not listed here are not supported.

Codec capabilities

Codec	Bits per Pixel	Gray	Indexed	Quality	Compression
TIFF	1,2,3,4,8,24[1]	Yes	Yes	Yes	Raw, Flate, LZW(default), JPEG, Group3, Group3_2D, Group4
JPEG	8, 24	Yes	No	Yes	JPEG (lossy only)
BMP	1, 2, 4, 8, 24[1]	Yes	Yes	No	Raw
GIF	2-8	Yes	Yes	No	LZW
PNG	1-8, 24	Yes	Yes	No	Flate
JBIG2	1	Yes	No	No	JBIG2 (lossless only)
JPEG2000	8, 24	Yes	Yes	Yes	JPEG2000 (lossless: Q = 100)[2]
PBM	1-8, 24	Yes	No	No	Raw
EPS	1, 2, 4, 8, 24[1]	Yes	No	No	Raw

Codec The **Co**mpression/**Dec**ompression type.

Bits per pixel The supported values for bits per pixel. 1 = bitonal, 8 = 256 colors/grayscales, 24 = True Color

Gray This format supports grayscale.

Indexed This format supports indexed colors.

Quality This format supports the setting of a quality parameter.

Compression Supported compression types.

[1] For palette creation: The number of palette entries is equal to 2^BitsPerPixel where BitsPerPixel is smaller or equal to 8. This means it is possible to create a 3 bits per pixel TIFF or BMP, but the palette size is equal as for 4 bits. However the 3 bits per pixel image will compress better than the 4 bits per pixel image.

[2] To create lossless JPEG2000 images, set the quality parameter to 100. For values <100, a lossy compression algorithm is applied.

5.2 Rendering options

5.2.1 -1 Create one image file per PDF page

Create one image file per PDF page `-1`

By default, the 3-Heights® PDF to Image Converter Shell generates multi-page images if the TIFF format is selected (unless the input PDF document has only one page). To create one TIFF file per PDF page, simply use the `-1` option. This option should also be used for all formats that do not support multi-paging if the input PDF document has more than one page.

Example: To generate one TIFF image for every page of the PDF input file, the `-1` switch can be used:

```
pdf2imgR2 -1 input.pdf
```

This generates a series of one-page TIFF images like this:

```
input_1.tif, input_2.tif, input_3.tif, etc.
```

Example: To specify the output name and directory, use a command like this:

```
pdf2imgR2 -1 input.pdf ../output_%d.tiff
```

The output files are then named

```
output_1.tiff, output_2.tiff, output_3.tiff, etc.
```

5.2.2 -b Set the bits per pixel

Set the bits per pixel `-b ‹n›`

The argument for this parameter depends on the image format (see Codecs table).

For JPEG, this option has two allowed values: 8 for grayscale and 24 for TrueColor.

Example: The following command creates a grayscale JPEG image.

```
pdf2imgR2 -b 8 input.pdf output.jpg
```

When using 1 bit per pixel, it is suggested to disable anti-aliasing (-oq option) and set a suitable dithering algorithm (-h option).

5.2.3 -c Compression type of TIFF images

Compression type of TIFF images `-c ‹compression›`

Compression options can only be set for TIFF images. For all other image types, the compression is defined through the image format (.gif, .jpg, etc.).

The default compression is `l` (LZW).

> **Note:** Not all image viewer support all compression types for TIFF.

Compression table

Compression	Description
`f`	**Flate compression (ZIP)** is a lossless compression algorithm. It is useful for the compression of large images with no loss in quality.
`g3`	**CCITT Fax Group 3** is the predecessor to CCITT Fax Group 4. It is a simpler algorithm that normally results in a weaker compression level.
`g3_2D`	**CCITT Fax Group 3 2D** is a 2-dimensional version of the CCITT Group 3 Huffman encoding algorithm.
`g4`	**CCITT Fax Group 4** is the standard compression for bi-level images (i.e. facsimile).
`j`	**Joint Photographic Expert Group (JPEG)** is a lossy compression algorithm. JPEG provides a high level compression of 8 and 24 bit images. It is best suited for pictures, but not for text images. The `-q` option can be used to set an image quality.
`j6`	**TIFF embedded JPEG (6)** is an older version of JPEG. Certain (older) image software may support this compression, but not the newer version of JPEG (e.g. Photoshop 8). Allowed in `pdf2img` only.
`l`	**Lempel-Ziv-Welch (LZW)** is a lossless compression algorithm for images. Please consult the copyright laws of your country prior to using this compression algorithm.
`raw`	**No compression**

5.2.4 `-cms` Set the color management engine

Set the color management engine `-cms ‹engine›`

The transformation of colors from one color space to another is performed using a color management engine.

Supported engines are:

`none` The algorithms specified in the PDF reference are used. This results in the maximum possible contrast.

`neugebauer` The Neugebauer algorithm efficiently converts CMYK to RGB. It does not need any color profiles. The results look similar to conversion using color profiles.

`lcms` (default): Use ICC color profiles. Default profiles are used for all unmanaged device color spaces as described in Color profiles.

`‹FileName›` When providing a file name, a configurable version of the Neugebauer algorithm is applied. The coefficients can be defined in the text file. The default Neugebauer coefficients are listed below (Red, Green, Blue; Color):

```
1.000000, 1.000000, 1.000000; White
0.000000, 0.682353, 0.937255; C
0.925490, 0.000000, 0.549020; M
1.000000, 0.949020, 0.000000; Y
0.137255, 0.121569, 0.125490; K
0.180392, 0.188235, 0.572549; CM
0.000000, 0.650980, 0.313725; CY
0.000000, 0.054902, 0.137255; CK
0.929412, 0.109804, 0.141176; MY
0.137255, 0.000000, 0.000000; MK
0.105882, 0.098039, 0.000000; YK
0.211765, 0.211765, 0.223529; CMY
0.000000, 0.000000, 0.003922; CMK
0.000000, 0.070588, 0.000000; CYK
0.133333, 0.000000, 0.000000; MYK
0.000000, 0.000000, 0.000000; CMYK
```

The Neugebauer algorithm mixes the colors based on the amount of color and the corresponding weighted coefficient. Altering the values for a pure color specifically changes the result for this pure color.

The color transition remains smooth.

Example: The following command selects the Neugebauer color management engine

```
pdf2imgR2 -cms neugebauer input.pdf output.jpg
```

5.2.5 -cn Center mode

Center mode `-cn`

Center the PDF. This option is useful in combination with setting page dimensions.

5.2.6 -cs Set the color space

Set the color space `-cs ‹n›`

This options sets the color space. Supported values are:

Color space

Value	Description
0	Grayscale
2	RGB
4	CMYK
7	Indexed

Color space	
9	CMYK with K only

5.2.7 -d Set the resolution in DPI

Set the resolution in DPI `-d ‹dpi›`

The default resolution is set to 150 DPI (dots per inch).

This switch is redundant to the specialized options `-dx` and `-dy`, meaning the last option set on the command takes precedence.

Example: To set the DPI value to 100, use the option `-d` like this:

```
pdf2imgR2 -d 100 input.pdf
```

5.2.8 -dx Set the X resolution in DPI

Set the X resolution in DPI `-dx ‹dpi›`

Set the resolution only for the X-axis. The default resolution is set to 150 DPI.

Example: Set the resolution in X to 72 DPI.

```
pdf2imgR2 -dx 72 input.pdf
```

5.2.9 -dy Set the Y resolution in DPI

Set the Y resolution in DPI `-dy ‹dpi›`

Set the resolution only for the Y-axis. The default resolution is set to 150 DPI.

Example: Set the resolution in Y to 72 DPI.

```
pdf2imgR2 -dy 72 input.pdf
```

5.2.10 -f Fit page mode

Fit page mode `-f`

Make the PDF fit the page (in either width or height). This option is useful in combination with setting page dimensions.

Example: Convert a PDF to a 800 by 600 pixel TIFF image and scale the page of the PDF to fit the page of the image.

```
pdf2imgR2 -s 800 600 -f input.pdf output.tiff
```

5.2.11 -fax Convert to Class F

Convert to Class F `-fax ‹class F›`

This options creates a Class F TIFF, which is used by fax machines. There are two types:

Class F TIFF

Value	Description	Equal to these settings
s	Standard	`-rp -sp 1728 0 -dx 204 -dy 98 -c g3`
h	High resolution	`-rp -sp 1728 0 -dx 204 -dy 196 -c g3`

Example: Convert a PDF to a standard Class F TIFF.

```
pdf2imgR2 -fax s input.pdf output.tif
```

5.2.12 -fo Bit fill order

Bit fill order `-fo ‹n›`

Set the fill order of bits used in fax compressions.

Bit fill order

Value	Description
1 (default)	Most significant bit (MSB) first.
2	Least significant bit (LSB) first.

5.2.13 -fs Filter size

Filter size `-fs ‹n›`

This setting is used to enable and parameterize super-sampling, a technique to initially render the image at a higher resolution and then sample it down to the target resolution. As a result of that process, the final image appears smoother, i.e. anti-aliased.

Applying super-sampling improves the image quality when rendering at low target resolutions (72 DPI or less); the higher the target resolution, the less the visual impact.

This setting requires memory and CPU time quadratically to the ratio, therefore only small values, such as 2 or 3 should be used.

If a too high value (in combination with the original image size) is set, it is ignored.

5.2.14 -g Gray color space

[Deprecated] Gray color space `-g`

No longer supported: Use the `-cs` option instead.

5.2.15 -h Dithering mode

Dithering mode `-h ‹mode›`

Set the dithering mode. Allowed values are:

Dithering mode

Value	Description
0	No dithering
1	(Default) Floyd-Steinberg
2	Halftone block
3	Halftone continuous
6	Atkinson dithering is very fast and produces images that can be compressed really well with reasonably good image quality.

Dithering provides a better image quality, especially for 1 bit images, at the cost of a larger file size.

Example: Disable dithering for a bitonal image.

```
pdf2imgR2 -h 0 -b 1 input.pdf output.tif
```

5.2.16 -i Indexed color space

[Deprecated] Indexed color space `-i`

Use the option `-cs` instead.

5.2.17 -m Rendering mode

Rendering mode `-m ‹n›`

Parameter:

`‹n›`

- **0** Draft mode
- **1** Quality mode

Set the rendering quality.

5.2.18 -oc Disable black point compensation (BPC)

Disable black point compensation (BPC) `-oc`

This option disables the use of black point compensation (BPC).

5.2.19 -of Fit rectangle paths to pixel grid

Fit rectangle paths to pixel grid `-of`

This option enables fitting of clipping paths that describe exactly one rectangle to the pixel grid.

5.2.20 -oh Disable hinting

Disable hinting `-oh`

In the context of text rendering, **hinting** refers to the process of distorting glyph outlines with the goal of producing a clearer and more easily readable rendered image. For example, horizontal and vertical lines may be moved slightly such that they fall on the pixel grid.

Normally, the rules for performing hinting and a threshold for switching off hinting at low resolutions are contained in a font program. If this option is specified, then hinting is switched off entirely.

5.2.21 -ohs Manual character size limit for hinting

Manual character size limit for hinting `-ohs ‹size›`

This option allows to override the threshold for switching off hinting at low resolutions.

Hinting is only enabled if the size (vertically or horizontally) of a character to be rendered exceeds the specified `‹size›` in pixels.

See also -oh.

5.2.22 -op Set print mode

Set print mode `-op`

This switch enables the print mode.

5.2.23 -oq Disable high quality rendering

Disable high quality rendering `-oq`

This option disables anti-aliasing.

5.2.24 -p Read an encrypted PDF file

Read an encrypted PDF file `-p ‹password›`

When the input PDF file is encrypted and has a user password set (the password to open the PDF), the password can be provided with the `-p` option.

Example: If the user password were `userpwd`, then the command to read and process the encrypted PDF would look like this:

```
pdf2imgR2 -p userpwd input.pdf
```

When a PDF is encrypted and the user password is not provided or is incorrect, the 3-Heights® PDF to Image Converter Shell cannot decrypt and read the file. Instead, it generates the error message: `Password wasn't correct.`

5.2.25 -pbt Set page box type

Set page box type `-pbt ‹type›`

Set the box type that is used for rendering the page.

The default page box type is `crop`.

Page box table

Page box type	Description
`art`	ArtBox
`bleed`	BleedBox
`crop`	CropBox
`media`	MediaBox

Page box table

`trim`	TrimBox

5.2.26 `-pi` Print page information

Print page information `-pi`

Print page information of all converted pages to standard output. Printed information includes page number, page size in user units (1/72 inch), dimensions of output image in pixels, and DPI of output image.

5.2.27 `-pg` Set page range

Set page range `-pg ‹first› ‹last›`

With this switch, the page range to be converted can be set. Note that the output image format must support multi-page. TIFF and JBIG2 support multi-page.

Example: Convert only pages 1 through 3.

```
pdf2imgR2 -pg 1 3 input.pdf output.tif
```

5.2.28 `-pgs` Set of pages

Set of pages `-pgs ‹page_set›`

A set of pages can be defined using single pages, ranges of pages, and comma-separated combinations thereof.

Example: Convert pages 1, 2-4, 6 and 10.

```
pdf2imgR2 -pgs 1,2-4,6,10 input.pdf output.tif
```

Negative numbers can be used to denote a page counting from the back, e.g. '-1' represents the last page of the file.

Example: Convert the last two pages of the file.

```
pdf2imgR2 -pgs -2--1 input.pdf output.tif
```

If a range is given starting from a larger number leading to a smaller number, then the pages are written in reverse.

Example: Convert pages 4, 3 and 2.

```
pdf2imgR2 -pgs 4-2 input.pdf output.tif
```

Note that all of the above ways of adressing can be combined:

Example: Supposing a 6-page file, this example would generate pages 3, 2, 1 and 6 in this order.

```
pdf2imgR2 -pgs (3--6,-1 input.pdf output.tif
```

5.2.29 -q Set image quality

Set image quality `-q ‹n›`

Set the image quality of lossy image compressions (such as JPEG). Default: 80. If a compression algorithm supports both lossy and lossless compression, a quality value of 100 applies lossless compression.

Example: Lossy compression with a quality index of 50

```
pdf2imgR2 -q 50 input.pdf output.jpg
```

Example: Lossless compression

```
pdf2imgR2 -q 100 input.pdf output.jp2
```

5.2.30 -ri Ignore page rotate attribute

Ignore page rotate attribute `-ri`

A PDF document can have a page rotation attribute that describes if the PDF is to be rotated when displayed (for example, when a 90° rotated portrait is displayed as landscape). The PDF to Image Converter by default respects this attribute and rotates pages automatically. Using the `-ri` switch, the page rotation attribute is ignored.

5.2.31 -rl Rotate pages to landscape

Rotate pages to landscape `-rl`

This option rotates all pages to landscape.

5.2.32 -rp Rotate pages to portrait

Rotate pages to portrait `-rp`

This option rotates all pages to portrait.

5.2.33 -s Set width and height of image in points

Set width and height of image in points `-s ‹w› ‹h›`

When not specified, the image has the same dimensions as the input PDF (specifically, its CropBox). To set the dimensions manually, use this option, where ‹w› is the width and ‹h›, the height. If either of the dimensions is set to 0, the value is computed proportionally based on the other value.

Example: The following command generates an image that is 400x300 points.

```
pdf2imgR2 -s 400 300 input.pdf
```

5.2.34 -sa Set width and height in pixels and preserve ratio

Set width and height in pixels and preserve ratio -sa ‹w› ‹h›

The -sa option has the same effect as -sp, but the aspect ratio is preserved. This means that you can specify the maximum size allowed, the image is then scaled to fit one of the dimensions (for example, a 400 by 400 points PDF is converted with the -sa 600 800 option. The ratio of the input file is preserved, which is 1:1. The maximum that fits in 600 by 800 pixels is therefore a 600 by 600 pixel image).

5.2.35 -so Page offset in points

Page offset in points -so ‹x› ‹y›

Set the offset of the image in points. The default offset is the CropBox's offset.

5.2.36 -sp Set width and height of image in pixel

Set width and height of image in pixel -sp ‹w› ‹h›

This option can be used to set the dimensions of the image in pixels manually. If either of the dimensions is set to 0, the value is calculated proportionally based on the other value.

Example: The following command creates a file with a width of 1024 pixels, the height is calculated proportionally.

```
pdf2imgR2  -sp 1024 0 input.pdf
```

5.2.37 -t Set threshold when dithering is disabled

Set threshold when dithering is disabled -t ‹threshold›

When producing bitonal images, e.g. with the -b 1, -g3 or -fax option, then the dithering mode is set with the -h switch. If "no dithering" is selected (-h 0), then the conversion of color or grayscale images is done by applying a threshold. All pixels with brightness above the threshold are converted to white and all others to black.

The -t switch can be used to set a custom threshold. The allowed range is 0 (all white) to 255 (all black). The default value is 181.

5.3 General options

5.3.1 -v Verbose mode

Verbose mode `-v`

This option turns on the verbose mode.

5.3.2 -lk Set license key

Set license key `-lk ‹key›`

Pass a license key to the application at runtime, instead of using one that is installed on the system.

```
pdf2imgR2 -lk X-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX-XXXXX ...
```

This is required in an OEM scenario only.

5.4 Return codes

All return codes other than `0` indicate an error in the processing.

Return codes

Value	Description
0	Success.
1	Couldn't open input file.
2	PDF output file could not be created.
4	PDF input file is encrypted and password is missing or incorrect.
5	Extraction error either due to corrupt input PDF or failure when storing an extracted file.
5	Rendering error.
10	License error, e.g. invalid license key.

6 Troubleshooting

6.1 Output

6.1.1 Generated files are large size

See User guide Reducing the file size.

6.1.2 Images are too low quality

Increase the resolution to increase to pixel mass. This is done using the `-d` option.

For lossy compression algorithms such as JPEG, increase the quality parameter, e.g. `-q 85`.

6.1.3 Image does not contain the whole content

This can happen when the `-s` option is used to set dimensions that have a different ratio than the original dimensions. To automatically make the page fit the new dimensions, use the `-f` option.

Alternatively, ensure the page dimensions of the image are large enough to hold the complete page.

6.1.4 Colors are gone

The `-b` option allows you to set the bits per pixel. JPEG 8 bit is always grayscale, since indexed colors are not supported for this format. For TIFF and GIF, the indexed colors need to be enabled if 8 bit is selected. This is done with the `-cs 7` option.

6.2 Font and text issues

1. For issues with text using non-embedded fonts:
   1. Ensure the required fonts are available on the system (see Fonts).
   2. See Handle non-embedded fonts.

6.2.1 Handle non-embedded fonts

Font replacement strategy

This section describes how the rendering engine handles fonts. It is rather technical and it is not required to be understood to use the software.

The following steps are performed sequentially when searching for a font. If a font is found, the search is stopped; otherwise, the next step is performed.

1. If the font is not embedded:

   a. If the font name appears in the `[replace]` section in the configuration file `fonts.ini`, the name is replaced and looked up in the installed font collection.
   b. If it is a standard font[3] it is replaced by the equivalent TrueType font name and it is looked up in the installed font collection.
   c. If the font name appears in the `[fonts]` section in the configuration file `fonts.ini`, the name is replaced and looked up in the installed font collection.
   d. If the font has "Italic" or "Bold" in its name, the font without these styles is looked up in the installed font collection.
2. If a font name is looked up in the installed font collection, then the name comparison is performed as follows:
   a. PostScript name.
   b. TrueType name without blanks (a missing style is interpreted as "Regular" or "Normal").
   c. TrueType name without modifications.
3. If a font from the installed font collection matches the metrics of the font, the installed font is used.
4. If the font is a CID font using a specific character collection, e.g. "Japan1", an installed font that contains the required code pages is used.
5. If the font is a non-symbolic simple font, a font program with the font metrics required is created dynamically.

[3] e.g. Times-Roman, Helvetica, Courier

7 Version history

7.1 Changes in versions 6.19–6.27

- **Update** license agreement to version 2.9

7.2 Changes in versions 6.13–6.18

No functional changes.

7.3 Changes in versions 6.1–6.12

- **Improved** search algorithm for installed fonts: User fonts under Windows are now also taken into account.

Shell `pdf2imgR2`

- **New** option `-pbt` to select the box which is used for rendering the page.

7.4 Changes in version 5

- **New** additional supported operating system: Windows Server 2019.

Shell `pdf2imgR2`

No functional changes.

7.5 Changes in version 4.12

- **Improved** the rendering quality of small text using Type3 fonts (bitmap fonts).
- **Improved** the quality of low-pass filtered images, if the default Gauss algorithm is used. Note that for creating bitonal output images, the option `-ofb` is recommended.
- **Improved** the rendering speed for documents that have resources with circular references.
- **New** HTTP proxy setting in the GUI license manager.

7.6 Changes in version 4.11

- **New** support for reading PDF 2.0 documents.

- **Improved** search in installed font collection to also find fonts by other names than TrueType or PostScript names.

Shell `pdf2imgR2`

- **Improved** option `-pgs` to also address pages from the back of the document as well as reordering pages freely.

7.7 Changes in version 4.10

- **Improved** rendering engine R2 performance when using Type 3 fonts.
- **Improved** robustness against corrupt input PDF documents.
- **Improved** annotation appearance generation for polyline, squiggly, and stamp annotations.
- **Removed** the font `ZapfDingbats.ttf` from the product kit as it is not required anymore.

7.8 Changes in version 4.9

- **Improved** support for and robustness against corrupt input PDF documents.
- **Improved** repair of embedded font programs that are corrupt.
- **New** support for OpenType font collections in installed font collection.
- **Improved** metadata generation for standard PDF properties.

7.9 Changes in version 4.8

- **Improved** creation of annotation appearances to use less memory and processing time.
- **Added** repair functionality for TrueType font programs whose glyphs are not ordered correctly.

8 Licensing, copyright, and contact

Pdftools (PDF Tools AG) is a world leader in PDF software, delivering reliable PDF products to international customers in all market segments.

Pdftools provides server-based software products designed specifically for developers, integrators, consultants, customizing specialists, and IT departments. Thousands of companies worldwide use our products directly and hundreds of thousands of users benefit from the technology indirectly via a global network of OEM partners. The tools can be easily embedded into application programs and are available for a multitude of operating system platforms.

Licensing and copyright The 3-Heights® PDF to Image Converter Shell is copyrighted. This user manual is also copyright protected; It may be copied and distributed provided that it remains unchanged including the copyright notice.

Contact

PDF Tools AG
Brown-Boveri-Strasse 5
8050 Zürich
Switzerland
https://www.pdf-tools.com
pdfsales@pdf-tools.com