

Coherent PDF Command Line Toolkit

User Manual
Version 2.2 (March 2017)

Coherent Graphics Ltd

For bug reports, feature requests and comments, email
contact@coherentgraphics.co.uk

©2017 Coherent Graphics Limited. All rights reserved. ISBN 978-0957671140

Adobe, Acrobat, Adobe PDF, Adobe Reader and PostScript are registered trademarks of Adobe Systems Incorporated. Windows, Powerpoint and Excel are registered trademarks of Microsoft Corporation.

Contents

1	Basic Usage	1
1.1	Input and Output Files	1
1.2	Input Ranges	2
1.3	Working with Encrypted Documents	3
1.4	Standard Input and Standard Output	3
1.5	Doing Several Things at Once with AND	3
1.6	Units	4
1.7	Setting the Producer and Creator	4
1.8	PDF Version Numbers	5
1.9	File IDs	5
1.10	Linearization	5
1.11	Object Streams	6
1.12	Malformed Files	6
1.13	Error Handling	7
1.14	Control Files	7
1.15	String Arguments	7
1.16	Text Encodings	8
1.17	Font Embedding	8
2	Merging and Splitting	9
2.1	Merging	9
2.2	Splitting	9
2.3	Splitting on Bookmarks	10
2.4	Encrypting with Split and Split Bookmarks	10
3	Pages	11
3.1	Page Sizes	11
3.2	Scale Pages	12
3.3	Shift Page Contents	12
3.4	Rotating Pages	13
3.5	Flipping Pages	13
3.6	Boxes and Cropping	13

4	Encryption and Decryption	15
4.1	Introduction	15
4.2	Encrypting a Document	16
4.3	Decrypting a Document	16
5	Compression	17
5.1	Decompressing a Document	17
5.2	Compressing a Document	17
5.3	Squeezing a Document	17
6	Bookmarks	19
6.1	List Bookmarks	19
6.2	Remove Bookmarks	20
6.3	Add Bookmarks	20
7	Presentations	21
8	Watermarks and Stamps	23
8.1	Add a Watermark or Logo	24
8.2	Stamp Text, Dates and Times.	24
8.2.1	Page Numbers	24
8.2.2	Date and Time Formats	25
8.2.3	Bates Numbers	25
8.2.4	Position	25
8.2.5	Font and Size	26
8.2.6	Colors	27
8.2.7	Outline Text	27
8.2.8	Multi-line Text	27
8.2.9	Special Characters	28
8.3	Stamping Graphics	28

Typographical Conventions

Command lines to be typed are shown in **typewriter font** in a box. For example:

```
cpdf in.pdf -o out.pdf
```

When describing the general form of a command, rather than a particular example, square brackets [] are used to enclose optional parts, and angled braces <> to enclose general descriptions which may be substituted for particular instances. For example,

```
cpdf <operation> in.pdf [<range>] -o out.pdf
```

describes a command line which requires an operation and, optionally, a range. An exception is that we use `in.pdf` and `out.pdf` instead of `<input file>` and `<output file>` to reduce verbosity. Under Microsoft Windows, type `cpdf.exe` instead of `cpdf`.

Chapter 1

Basic Usage

<code>-o</code>	<code>-idir <directory></code>	<code>-recrypt</code>
<code>-stdout</code>	<code>-stdin</code>	<code>-stdin-user <password></code>
<code>-stdin-owner <password></code>	<code>-producer <text></code>	<code>-creator <text></code>
<code>-change-id</code>	<code>-l</code>	<code>-cpdfin <filename></code>
<code>-keep-l</code>	<code>-no-preserve-objstm</code>	<code>-create-objstm</code>
<code>-control <filename></code>	<code>-args <filename></code>	<code>-utf8</code>
<code>-stripped</code>	<code>-raw</code>	<code>-no-embed-font</code>

The Coherent PDF tools provide a wide range of facilities for modifying PDF files created by other means. There is a single command-line program `cpdf` (`cpdf.exe` under Microsoft Windows). The rest of this manual describes the options that may be given to this program.

1.1 Input and Output Files

The typical pattern for usage is

```
cpdf [<operation>] <input file(s)> -o <output file>
```

and the simplest concrete example, assuming the existence of a file `in.pdf` is:

```
cpdf in.pdf -o out.pdf
```

which copies `in.pdf` to `out.pdf`. The input and output may be the same file. Of course, we should like to do more interesting things to the PDF file than that!

Files on the command line are distinguished from other input by their containing a period. If an input file does not contain a period, it should be preceded by `-i`. For example:

```
cpdf -i in -o out.pdf
```

A whole directory of files may be added (where a command supports multiple files) by using the `-idir` option:

```
cpdf -merge -idir myfiles -o out.pdf
```

The files in the directory `myfiles` are considered in alphabetical order. They must all be PDF files. If the names of the files are numeric, leading zeroes will be required for the order to be correct (e.g `001.pdf`, `002.pdf` etc).

1.2 Input Ranges

An *input range* may be specified after each input file. This is treated differently by each operation. For instance

```
cpdf in.pdf 2-5 out.pdf
```

extracts pages two, three, four and five from `in.pdf`, writing the result to `out.pdf`, assuming that `in.pdf` contains at least five pages. Here are the rules for building input ranges:

- A dash (-) defines ranges, e.g. 1-5 or 6-3.
- A comma (,) allows one to specify several ranges, e.g. 1-2,4-5.
- The word **end** represents the last page number.
- The words **odd** and **even** can be used in place of or at the end of a page range to restrict to just the odd or even pages.
- The words **portrait** and **landscape** can be used in place of or at the end of a page range to restrict to just those pages which are portrait or landscape. Note that the meaning of “portrait” and “landscape” does not take account of any viewing rotation in place (use **-upright** first, if required). A page with equal width and height is considered neither portrait nor landscape.
- The word **reverse** is the same as **end-1**.
- The word **all** is the same as **1-end**.
- A range must contain no spaces.
- A tilde (~) defines a page number counting from the end of the document rather than the beginning. Page **~1** is the last page, **~2** the penultimate page etc.

For example:

```
cpdf in.pdf 1,2,7-end -o out.pdf
```

Remove pages three, four, five and six from a document.

```
cpdf in.pdf 1-16odd -o out.pdf
```

Extract the odd pages 1,3,...,13,15.

```
cpdf in.pdf landscape -rotate 90 -o out.pdf
```

Rotate all landscape pages by ninety degrees.

```
cpdf in.pdf 1,all -o out.pdf
```

Duplicate the front page of a document, perhaps as a fax cover sheet.

```
cpdf in.pdf 3- 1 -o out.pdf
```

Extract the last three pages of a document, in order.

1.3 Working with Encrypted Documents

In order to perform many operations, encrypted input PDF files must be decrypted. Some require the owner password, some either the user or owner passwords. Either password is supplied by writing `user=<password>` or `owner=<password>` following each input file requiring it (before or after any range). The document will *not* be re-encrypted upon writing. For example:

```
cpdf in.pdf user=charles -info
cpdf in.pdf owner=fred reverse -o out.pdf
```

To re-encrypt the file with its existing encryption upon writing, which is required if only the user password was supplied, but allowed in any case, add the `-recrypt` option:

```
cpdf in.pdf user=fred reverse -recrypt -o out.pdf
```

The password required (owner or user) depends upon the operation being performed. Separate facilities are provided to decrypt and encrypt files (See Section 4).

1.4 Standard Input and Standard Output

Thus far, we have assumed that the input PDF will be read from a file on disk, and the output written similarly. Often it's useful to be able to read input from `stdin` (Standard Input) or write output to `stdout` (Standard Output) instead. The typical use is to join several programs together into a *pipe*, passing data from one to the next without the use of intermediate files. Use `-stdin` to read from standard input, and `-stdout` to write to standard input, either to pipe data between multiple programs, or multiple invocations of the same program. For example, this sequence of commands (all typed on one line)

```
cpdf in.pdf reverse -stdout |
cpdf -stdin 1-5 -stdout |
cpdf -stdin reverse -o out.pdf
```

extracts the last five pages of `in.pdf` in the correct order, writing them to `out.pdf`. It does this by reversing the input, taking the first five pages and then reversing the result.

To supply passwords for a file from `-stdin`, use `-stdin-owner <password>` and/or `-stdin-user <password>`.

Using `-stdout` on the final command in the pipeline to output the PDF to screen is not recommended, since PDF files often contain compressed sections which are not screen-readable.

Several `cpdf` operations write to standard output by default (for example, listing fonts). A useful feature of the command line (not specific to `cpdf`) is the ability to redirect this output to a file. This is achieved with the `>` operator:

```
cpdf -info in.pdf > file.txt
```

Use the `-info` operation (See Section ??), redirecting the output to `file.txt`.

1.5 Doing Several Things at Once with AND

The keyword `AND` can be used to string together several commands in one. The advantage compared with using pipes is that the file need not be repeatedly parsed and written out, saving time.

To use `AND`, simply leave off the output specifier (e.g `-o`) of one command, and the input specifier (e.g filename) of the next. For instance:

```
cpdf -merge in.pdf in2.pdf AND -add-text "Label"
AND -merge in3.pdf -o out.pdf
```

Merge `in.pdf` and `in2.pdf` together, add text to both pages, append `in3.pdf` and write to `out.pdf`.

To specify the range for each section, use `-range`:

```
cpdf -merge in.pdf in2.pdf AND -range 2-4 -add-text "Label"
AND -merge in3.pdf -o out.pdf
```

1.6 Units

When measurements are given to `cpdf`, they are in points (1 point = 1/72 inch). They may optionally be followed by some letters to change the measurement. The following are supported:

<code>pt</code>	Points (72 points per inch). The default.
<code>cm</code>	Centimeters
<code>mm</code>	Millimeters
<code>in</code>	Inches

For example, one may write `14mm` or `21.6in`. In addition, the following letters stand, in some operations (`-scale-page`, `-scale-to-fit`, `-scale-contents`, `-shift`, `-mediabox`, `-crop`) for various page dimensions:

<code>PW</code>	Page width
<code>PH</code>	Page height
<code>PMINX</code>	Page minimum x coordinate
<code>PMINY</code>	Page minimum y coordinate
<code>PMAXX</code>	Page maximum x coordinate
<code>PMAXY</code>	Page maximum y coordinate
<code>CW</code>	Crop box width
<code>CH</code>	Crop box height
<code>CMINX</code>	Crop box minimum x coordinate
<code>CMINY</code>	Crop box minimum y coordinate
<code>CMAXX</code>	Crop box maximum x coordinate
<code>CMAXY</code>	Crop box maximum y coordinate

For example, we may write `PMINX PMINY` to stand for the coordinate of the lower left corner of the page.

Simple arithmetic may be performed using the words `add`, `sub`, `mul` and `div` to stand for addition, subtraction, multiplication and division. For example, one may write `14in sub 30pt` or `PMINX mul 2`

1.7 Setting the Producer and Creator

The `-producer` and `-creator` options may be added to any `cpdf` command line to set the producer and/or creator of the PDF file. If the file was converted from another format, the *creator* is the program producing the original, the *producer* the program converting it to PDF.

```
cpdf -merge in.pdf in2.pdf -producer MyMerger -o out.pdf
```

Merge `in.pdf` and `in2.pdf`, setting the producer to `MyMerger` and writing the output to `out.pdf`.

1.8 PDF Version Numbers

When an operation which uses a part of the PDF standard which was introduced in a later version than that of the input file, the PDF version in the output file is set to the later version (most PDF viewers will try to load any PDF file, even if it is marked with a later version number). However, this automatic version changing may be suppressed with the `-keep-version` flag.

Here is a list of Acrobat versions together with the maximum PDF version they are intended to support:

PDF 1.2	Acrobat 3.0
PDF 1.3	Acrobat 4.0
PDF 1.4	Acrobat 5.0
PDF 1.5	Acrobat 6.0
PDF 1.6	Acrobat 7.0
PDF 1.7	Acrobat 8.0, 9.0, 10.0

If you wish to manually alter the PDF version of a file, use the `-set-version` option described in Section ??.

1.9 File IDs

PDF files contain an ID (consisting of two parts), used by some workflow systems to uniquely identify a file. To change the ID, behavior, use the `-change-id` operation. This will create a new ID for the output file.

```
cpdf -change-id in.pdf -o out.pdf
```

Write `in.pdf` to `out.pdf`, changing the ID.

1.10 Linearization

Linearized PDF is a version of the PDF format in which the data is held in a special manner to allow content to be fetched only when needed. This means viewing a multipage PDF over a slow connection is more responsive. By default, `cpdf` does not linearize output files. To make it do so, add the `-l` option to the command line, in addition to any other command being used. For example:

```
cpdf -change-id in.pdf -o out.pdf
cpdf -l in.pdf -o out.pdf
```

Linearize the file `in.pdf`, writing to `out.pdf`.

This requires the existence of the external program `cpdfnlin` which is provided with commercial versions of `cpdf`. This must be installed as described in the installation documentation provided with your copy of `cpdf`. If you are unable to install `cpdfnlin`, you must use `-cpdfnlin` to let `cpdf` know where to find it:

```
cpdf.exe -cpdflin "C:\\cpdflin.exe" -l in.pdf -o out.pdf
```

Linearize the file `in.pdf`, writing to `out.pdf`.

In extremis, you may place `cpdflin` and its resources in the current working directory, though this is not recommended. For further help, refer to the installation instructions for your copy of `cpdf`.

To keep the existing linearization status of a file (produce linearized output if the input is linearized and the reverse), use `-keep-l` instead of `-l`.

1.11 Object Streams

PDF 1.5 introduced a new mechanism for storing objects to save space: object streams. by default, `cpdf` will preserve object streams in input files, creating no more. To prevent the retention of existing object streams, use `-no-preserve-objstm`:

```
cpdf -no-preserve-objstm in.pdf -o out.pdf
```

Write the file `in.pdf` to `out.pdf`, removing any object streams.

To create new object streams if none exist, or augment the existing ones, use `-create-objstm`:

```
cpdf -create-objstm in.pdf -o out.pdf
```

Write the file `in.pdf` to `out.pdf`, preserving any existing object streams, and creating any new ones for new objects which have been added.

To create wholly new object streams, use both options together:

```
cpdf -create-objstm -no-preserve-objstm in.pdf -o out.pdf
```

Write the file `in.pdf` to `out.pdf` with wholly new object streams.

Files written with object streams will be set to PDF 1.5 or higher, unless `-keep-version` is used (see above).

1.12 Malformed Files

There are many malformed PDF files in existence, including many produced by otherwise-reputable applications. `cpdf` attempts to correct these problems silently.

Grossly malformed files will be reconstructed. The reconstruction progress is shown on `stderr` (Standard Error):

```
./cpdf in.pdf -o out.pdf
couldn't lex object number
Attempting to reconstruct the malformed pdf in.pdf...
Read 5530 objects
Malformed PDF reconstruction succeeded!
```

Sometimes files can be technically well-formed but use inefficient PDF constructs. If you are sure the input files you are using are impeccably formed, the `-fast` option added to the command line (or, if using `AND`, to each section of the command line). This will use certain shortcuts which speed up processing, but would fail on badly-produced files.

The `-fast` option may be used with:

```
Chapter 3
-rotate-contents -upright -vflip -hflip
-shift -scale -scale-to-fit -scale-contents
```

```
Chapter 8
-add-text
-stamp-on -stamp-under -combine-pages
```

If problems occur, refrain from using `-fast`.

1.13 Error Handling

When `cpdf` encounters an error, it exits with code 2. An error message is displayed on `stderr` (Standard Error). In normal usage, this means it's displayed on the screen. When a bad or inappropriate password is given, the exit code is 1.

1.14 Control Files

```
Chapter 3
cpdf -control <filename>
cpdf -args <filename>
```

Some operating systems have a limit on the length of a command line. To circumvent this, or simply for reasons of flexibility, a control file may be specified from which arguments are drawn. This file does not support the full syntax of the command line. Commands are separated by whitespace, quotation marks may be used if an argument contains a space, and the sequence `\` may be used to introduce a genuine quotation mark in such an argument.

Several `-control` arguments may be specified, and may be mixed in with conventional command-line arguments. The commands in each control file are considered in the order in which they are given, after all conventional arguments have been processed. It is recommended to use `-args` in all new applications. However, `-control` will be supported for legacy applications.

To avoid interference between `-control` and `AND`, a new mechanism has been added. Using `-args` in place of `-control` will perform direct textual substitution of the file into the command line, prior to any other processing.

1.15 String Arguments

Command lines are handled differently on each operating system. Some characters are reserved with special meanings, even when they occur inside quoted string arguments. To avoid this problem, `cpdf` performs processing on string arguments as they are read.

A backslash is used to indicate that a character which would otherwise be treated specially by the command line interpreter is to be treated literally. For example, Unix-like systems attribute a special meaning to the exclamation mark, so the command line

```
cpdf -add-text "Hello!" in.pdf -o out.pdf
```

would fail. We must escape the exclamation mark with a backslash:

```
cpdf -add-text "Hello\!" in.pdf -o out.pdf
```

It follows that backslashes intended to be taken literally must themselves be escaped (i.e. written `\\`).

1.16 Text Encodings

Some `cpdf` commands write text to standard output, or read text from the command line or configuration files. These are:

```
-info
-list-bookmarks
-set-author et al.
-list-annotations
```

There are three options to control how the text is interpreted:

```
-utf8
-stripped
-raw
```

Add `-utf8` to use Unicode UTF8, `-stripped` to convert to 7 bit ASCII by dropping any high characters, or `-raw` to perform no processing. The default is `-stripped`.

1.17 Font Embedding

Use the `-no-embed-font` to avoid embedding the Standard 14 Font metrics when adding text with `-add-text`.

Chapter 2

Merging and Splitting

```
cpdf -merge in1.pdf [<range>] in2.pdf [<range>] [<more names/ranges>]
[-retain-numbering] [-remove-duplicate-fonts] -o out.pdf
cpdf -split in.pdf -o <format> [-chunk <chunksize>]
cpdf -split-bookmarks <level> in.pdf -o <format>
```

2.1 Merging

The `-merge` operation allow the merging of several files into one. Ranges can be used to select only a subset of pages from each input file in the output. The output file consists of the concatenation of all the input pages in the order specified on the command line. Actually, the `-merge` can be omitted, since this is the default operation of `cpdf`.

```
cpdf -merge a.pdf 1 b.pdf 2-end -o out.pdf
```

Take page one of `a.pdf` and all but the first page of `b.pdf`, merge them and produce `out.pdf`.

Merge maintains bookmarks, named destinations, and name dictionaries.

Forms and other objects which cannot be merged are retained if they are from the document which first exhibits that feature.

The `-retain-numbering` option keeps the PDF page numbering labels of each document intact, rather than renumbering the output pages from 1.

The `-remove-duplicate-fonts` ensures that fonts used in more than one of the inputs only appear once in the output.

2.2 Splitting

The `-split` operation splits a PDF file into a number of parts which are written to file, their names being generated from a *format*. The optional `-chunk` option allows the number of pages written to each output file to be set.

```
cpdf -split a.pdf -o out%%.pdf
```

Split **a.pdf** to the files **out001.pdf**, **out002.pdf** etc.

```
cpdf -split a.pdf 1 even -chunk 10 -o dir/out%%.pdf
```

Split the even pages of **a.pdf** to the files **out001.pdf**, **out002.pdf** etc. with at most ten pages in each file. The directory (folder) **dir** must exist.

If the output format does not provide enough numbers for the files generated, the result is unspecified. The following format operators may be used:

%, %, %% etc.	Sequence number padded to the number of percent signs
@F	Original filename without extension
@N	Sequence number without padding zeroes
@S	Start page of this chunk
@E	End page of this chunk
@B	Bookmark name at this page

2.3 Splitting on Bookmarks

The **-split-bookmarks <level>** operation splits a PDF file into a number of parts, according to the page ranges implied by the document's bookmarks. These parts are then written to file with names generated from the given format.

Level 0 denotes the top-level bookmarks, level 1 the next level (sub-bookmarks) and so on. So **-split-bookmarks 1** creates breaks on level 0 and level 1 boundaries.

```
cpdf -split-bookmarks 0 a.pdf -o out%%.pdf
```

Split **a.pdf** to the files **out001.pdf**, **out002.pdf** on bookmark boundaries.

Now, there may be many bookmarks on a single page (for instance, if paragraphs are bookmarked or there are two subsections on one page). The splits calculated by **-split-bookmarks** ensure that each page appears in only one of the output files. It is possible to use the **@** operators above, including operator **@B** which expands to the text of the bookmark:

```
cpdf -split-bookmarks 0 a.pdf -o @B.pdf
```

Split **a.pdf** on bookmark boundaries, using the bookmark text as the filename.

The bookmark text used for a name is converted from unicode to 7 bit ASCII, and the following characters are removed, in addition to any character with ASCII code less than 32:

```
/ ? < > \ % : * | " ^ + =
```

2.4 Encrypting with Split and Split Bookmarks

The encryption parameters described in Chapter 4 may be added to the command line to encrypt each split PDF. Similarly, the **-recrypt** switch described in 1 may be given to re-encrypt each file with the existing encryption of the source PDF.

Chapter 3

Pages

```
cpdf -scale-page "<scale x> <scale y>" in.pdf [<range>] -o out.pdf
cpdf -scale-to-fit "<x size> <y size>" [-scale-to-fit-scale <scale>]
in.pdf [<range>] -o out.pdf
cpdf -scale-contents [<scale>] [<position>] in.pdf [<range>] -o out.pdf
cpdf -shift "<shift x> <shift y>" in.pdf [<range>] -o out.pdf
cpdf -rotate <angle> in.pdf [<range>] -o out.pdf
cpdf -rotateby <angle> in.pdf [<range>] -o out.pdf
cpdf -rotate-contents <angle> in.pdf [<range>] -o out.pdf
cpdf -upright in.pdf [<range>] -o out.pdf
cpdf -hflip in.pdf [<range>] -o out.pdf
cpdf -vflip in.pdf [<range>] -o out.pdf
cpdf -mediabox "<x> <y> <w> <h>" in.pdf [<range>] -o out.pdf
cpdf -crop "<x> <y> <w> <h>" in.pdf [<range>] -o out.pdf
cpdf -remove-crop in.pdf [<range>] -o out.pdf
cpdf -frombox <boxname> -tobox <boxname> [-mediabox-if-missing]
in.pdf [<range>] -o out.pdf
```

3.1 Page Sizes

Any time when a page size is required, instead of writing, for instance "210mm 197mm" one can instead write `a4portrait`. Here is a list of supported page sizes:

a0portrait	a1portrait	a2portrait
a3portrait	a4portrait	a5portrait
a6portrait	a7portrait	a8portrait
a9portrait	a10portrait	
a0landscape	a1landscape	a2landscape
a3landscape	a4landscape	a5landscape
a6landscape	a7landscape	a8landscape
a9landscape	a10landscape	
usletterportrait	usletterlandscape	
uslegalportrait	uslegallandscape	

3.2 Scale Pages

The `-scale-page` operation scales each page in the range by the X and Y factors given. This scales both the page contents, and the page size itself. It also scales any Crop Box and other boxes (Art Box, Trim Box etc). As with several of these commands, remember to take into account any page rotation when considering what the X and Y axes relate to.

```
cpdf -scale-page "2 2" in.pdf -o out.pdf
```

Convert an A4 page to A3, for instance.

The `-scale-to-fit` operation scales each page in the range to fit a given page size, preserving aspect ratio and centering the result.

```
cpdf -scale-to-fit "297mm 210mm" in.pdf -o out.pdf cpdf -scale-to-fit a4portrait in.pdf -o out.pdf
```

Scale a file's pages to fit A4 portrait.

The scale can optionally be set to a percentage of the available area, instead of filling it.

```
cpdf -scale-to-fit a4portrait -scale-to-fit-scale 0.9 in.pdf -o out.pdf
```

Scale a file's pages to fit A4 portrait, scaling the page 90% of its possible size.

The `-scale-contents` operation scales the contents about the center of the crop box (or, if absent, the media box), leaving the page dimensions (boxes) unchanged.

```
cpdf -scale-contents 0.5 in.pdf -o out.pdf
```

Scale a file's contents on all pages to 50% of its original dimensions.

To scale about a point other than the center, one can use the positioning commands described in Section 8.2.4. For example:

```
cpdf -scale-contents 0.5 -topright 20 in.pdf -o out.pdf
```

Scale a file's contents on all pages to 50% of its original dimensions about a point 20pts from its top right corner.

3.3 Shift Page Contents

The `-shift` operation shifts the contents of each page in the range by X points horizontally and Y points vertically.

```
cpdf -shift "50 0" in.pdf even -o out.pdf
```

Shift pages to the right by 50 points (for instance, to increase the binding margin).

3.4 Rotating Pages

There are two ways of rotating pages: (1) setting a value in the PDF file which asks the viewer (e.g. Acrobat) to rotate the page on-the-fly when viewing it (use **-rotate** or **-rotateby**) and (2) actually rotating the page contents and/or the page dimensions (use **-upright** afterwards or **-rotate-contents** to just rotate the page contents).

The possible values for **-rotate** and **-rotate-by** are 0, 90, 180 and 270, all interpreted as being clockwise. Any value may be used for **-rotate-contents**.

The **-rotate** operation sets the viewing rotation of the selected pages to the absolute value given.

```
cpdf -rotate 90 in.pdf -o out.pdf
```

Set the rotation of all the pages in the input file to ninety degrees clockwise.

The **-rotateby** operation changes the viewing rotation of all the given pages by the relative value given.

```
cpdf -rotateby 90 in.pdf -o out.pdf
```

Rotate all the pages in the input file by ninety degrees clockwise.

The **-rotate-contents** operation rotates the contents and dimensions of the page by the given relative value.

```
cpdf -rotate-contents 90 in.pdf -o out.pdf
```

Rotate all the page contents in the input file by ninety degrees clockwise. Does not change the page dimensions.

The **-upright** operation does whatever combination of **-rotate** and **-rotate-contents** is required to change the rotation of the document to zero without altering its appearance. In addition, it makes sure the media box has its origin at (0,0), changing other boxes to compensate.

3.5 Flipping Pages

The **-hflip** and **-vflip** operations flip the contents of the chosen pages horizontally or vertically. No account is taken of the current page rotation when considering what "horizontally" and "vertically" mean, so you may like to use **-upright** first.

```
cpdf -hflip in.pdf even -o out.pdf
```

Flip the even pages in **in.pdf** horizontally.

```
cpdf -vflip in.pdf -o out.pdf
```

Flip all the pages in **in.pdf** vertically.

3.6 Boxes and Cropping

All PDF files contain a *media box* for each page, giving the dimensions of the paper. To change these dimensions (without altering the page contents in any way), use the **-mediabox** option.

```
cpdf -mediabox "0pt 0pt 500pt 500pt" in.pdf -o out.pdf
```

Set the media box to 500 points square.

The four numbers are minimum x, minimum y, width, height. x coordinates increase to the right, y coordinates increase upwards. PDF file can also optionally contain a *crop box* for each page, defining to what extent the page is cropped before being displayed or printed. A crop box can be set, changed and removed, without affecting the underlying media box. To set or change the crop box use `-crop`. To remove any existing crop box, use `-remove-crop`.

```
cpdf -crop "0pt 0pt 200mm 200mm" in.pdf -o out.pdf
```

Crop pages to the bottom left 200-millimeter square of the page.

```
cpdf -remove-crop in.pdf -o out.pdf
```

Remove cropping.

Note that the crop box is only obeyed in some viewers.

```
cpdf -frombox <boxname> -tobox <boxname> [-mediabox-if-missing]  
in.pdf [<range>] -o out.pdf
```

Copy the contents of one box to another.

This operation copies the contents of one box (Media box, Crop box, Trim box etc.) to another. If `-mediabox-if-missing` is added, the media box will be substituted when the 'from' box is not set for a given page. For example

```
cpdf -frombox /TrimBox -tobox /CropBox in.pdf -o out.pdf
```

copies the Trim Box of each page to the Crop Box of each page. The possible boxes are `/MediaBox`, `/CropBox`, `/BleedBox`, `/TrimBox`, `/ArtBox`.

Chapter 4

Encryption and Decryption

```
cpdf -encrypt <method> <owner> <user>  
[-no-encrypt-metadata] <permissions> in.pdf -o out.pdf  
cpdf -decrypt in.pdf owner=<owner password> -o out.pdf
```

4.1 Introduction

PDF files can be encrypted using various types of encryption and attaching various permissions describing what someone can do with a particular document (for instance, printing it or extracting content). There are two types of person:

The **User** can do to the document what is allowed in the permissions.

The **Owner** can do anything, including altering the permissions or removing encryption entirely.

There are five kinds of encryption:

- 40-bit encryption (method **40bit**) in Acrobat 3 (PDF 1.1) and above
- 128-bit encryption (method **128bit**) in Acrobat 5 (PDF 1.4) and above
- 128-bit AES encryption (method **AES**) in Acrobat 7 (PDF 1.6) and above
- 256-bit AES encryption (method **AES256**) in Acrobat 9 (PDF 1.7) – *this is deprecated – do not use for new documents*
- 256-bit AES encryption (method **AES256ISO**) in PDF 2.0

All encryption supports these kinds of permissions:

-no-edit	Cannot change the document
-no-print	Cannot print the document
-no-copy	Cannot select or copy text or graphics
-no-annot	Cannot add or change form fields or annotations

In addition, 128-bit encryption (Acrobat 5 and above) and AES encryption supports these:

<code>-no-forms</code>	Cannot edit form fields
<code>-no-extract</code>	Cannot extract text or graphics
<code>-no-assemble</code>	Cannot merge files etc.
<code>-no-hq-print</code>	Cannot print high-quality

Add these flags to the command line to prevent each operation.

4.2 Encrypting a Document

To encrypt a document, the owner and user passwords must be given (here, **fred** and **charles** respectively):

```
cpdf -encrypt 40bit fred charles -no-print in.pdf -o out.pdf
cpdf -encrypt 128bit fred charles -no-extract in.pdf -o out.pdf
cpdf -encrypt AES fred "" -no-edit -no-copy in.pdf -o out.pdf
```

A blank user password is common. In this event, PDF viewers will typically not prompt for a password for when opening the file or for operations allowable with the user password.

```
cpdf -encrypt AES256 fred "" -no-forms in.pdf -o out.pdf
```

In addition, the usual method can be used to give the existing owner password, if the document is already encrypted.

When using AES encryption, the option is available to refrain from encrypting the metadata. Add `-no-encrypt-metadata` to the command line.

4.3 Decrypting a Document

To decrypt a document, the owner password is provided.

```
cpdf -decrypt in.pdf owner=fred -o out.pdf
```

The user password cannot decrypt a file.

Chapter 5

Compression

```
cpdf -decompress in.pdf -o out.pdf
cpdf -compress in.pdf -o out.pdf
cpdf -squeeze in.pdf [-squeeze-log-to <filename>] -o out.pdf
```

cpdf provides basic facilities for decompressing and compressing PDF streams.

5.1 Decompressing a Document

To decompress the streams in a PDF file, for instance to manually inspect the PDF, use:

```
cpdf -decompress in.pdf -o out.pdf
```

If `cpdf` finds a compression type it can't cope with, the stream is left compressed. When using `-decompress`, object streams are not compressed.

5.2 Compressing a Document

To compress the streams in a PDF file, use: `cpdf -compress in.pdf -o out.pdf`

`cpdf` compresses any streams which have no compression using the **FlateDecode** method, with the exception of Metadata streams, which are left uncompressed.

5.3 Squeezing a Document

To *squeeze* a PDF file, reducing its size by an average of about twenty percent (though sometimes not at all), use:

```
cpdf -squeeze in.pdf -o out.pdf
```

Adding `-squeeze` to the command line when using another operation will *squeeze* the file or files upon output.

The `-squeeze` operation writes some information about the squeezing process to standard output. The squeezing process involves several processes which losslessly attempt to reduce the file size. It is slow, so should not be used without thought.

```
$ ./cpdf -squeeze in.pdf -o out.pdf
Beginning squeeze: 123847 objects
Squeezing... Down to 114860 objects
Squeezing... Down to 114842 objects
Squeezing page data
Recompressing document
```

The `-squeeze-log-to <filename>` option writes the log to the given file instead of to standard output.

Chapter 6

Bookmarks

```
cpdf -list-bookmarks [-utf8 | -raw] in.pdf
cpdf -remove-bookmarks in.pdf -o out.pdf
cpdf -add-bookmarks <bookmark file> in.pdf -o out.pdf
```

PDF Bookmarks (properly called the *document outline*) represent a tree of references to parts of the file, typically displayed at the side of the screen. The user can click on one to move to the specified place. `cpdf` provides facilities to list, add, and remove bookmarks. The format used by the list and add operations is the same, so you can feed the output of one into the other, for instance to copy bookmarks.

6.1 List Bookmarks

The `-list-bookmarks` operation prints (to standard output) the bookmarks in a file. The first column gives the level of the tree at which a particular bookmark is. Then the text of the bookmark in quotes, then the page number which the bookmark points to, then (optionally) the word "open" if the bookmark should have its children (at the level immediately below) visible when the file is loaded. For example, upon executing `cpdf -list-bookmarks doc.pdf`

the result might be:

```
0 "Part 1" 1 open
1 "Part 1A" 2
1 "Part 1B" 3
0 "Part 2" 4
1 "Part 2a" 5
```

If the page number is 0, it indicates that clicking on that entry doesn't move to a page.

By default, `cpdf` converts unicode to ASCII text, dropping characters outside the ASCII range. To prevent this, and return unicode UTF8 output, add the `-utf8` option to the command. To prevent any processing, use the `-raw` option.

6.2 Remove Bookmarks

The `-remove-bookmarks` operation removes all bookmarks from the file.

```
cpdf -remove-bookmarks in.pdf -o out.pdf
```

6.3 Add Bookmarks

The `-add-bookmarks` operation adds bookmarks as specified by a *bookmarks file*, a text file in ASCII or UTF8 encoding and in the same format as that produced by the `-list-bookmarks` option. If there are any bookmarks in the input PDF already, they are discarded. For example, if the file `bookmarks.txt` contains the output from `-list-bookmarks` above, then the command

```
cpdf -add-bookmarks bookmarks.txt in.pdf -o out.pdf
```

adds the bookmarks to the input file, writing to `out.pdf`. An error will be given if the bookmarks file is not in the correct form (in particular, the numbers in the first column which specify the level must form a proper tree with no entry being more than one greater than the last).

Chapter 7

Presentations

```
cpdf -presentation in.pdf [<range>] -o out.pdf  
[-trans < transition-name>] [-duration < float> ]  
[-vertical] [-outward] [-direction < int >]  
[-effect-duration < float >]
```

The PDF file format, starting at Version 1.1, provides for simple slide-show presentations in the manner of Microsoft Powerpoint. These can be played in Acrobat and possibly other PDF viewers, typically started by entering full-screen mode. The **-presentation** operation allows such a presentation to be built from any PDF file.

The **-trans** option chooses the transition style. When a page range is used, it is the transition *from* each page named which is altered. The following transition styles are available:

Split Two lines sweep across the screen, revealing the new page. By default the lines are horizontal. Vertical lines are selected by using the **-vertical** option.

Blinds Multiple lines sweep across the screen, revealing the new page. By default the lines are horizontal. Vertical lines are selected by using the **-vertical** option.

Box A rectangular box sweeps inward from the edges of the page. Use **-outward** to make it sweep from the center to the edges.

Wipe A single line sweeps across the screen from one edge to the other in a direction specified by the **-direction** option.

Dissolve The old page dissolves gradually to reveal the new one.

Glitter The same as **Dissolve** but the effect sweeps across the page in the direction specified by the **-direction** option.

To remove a transition style currently applied to the selected pages, omit the **-trans** option.

The **-effect-duration** option specifies the length of time in seconds for the transition itself. The default value is one second.

The `-duration` option specifies the maximum time in seconds that the page is displayed before the presentation automatically advances. The default, in the absence of the `-duration` option, is for no automatic advancement.

The `-direction` option (for **Wipe** and **Glitter** styles only) specifies the direction of the effect. The following values are valid:

0 Left to right

90 Bottom to top (**Wipe** only)

180 Right to left (**Wipe** only)

270 Top to bottom

315 Top-left to bottom-right (**Glitter** only)

For example:

```
cpdf -presentation in.pdf 2-end -trans Split -duration 10 -o out.pdf
```

The **Split** style, with vertical lines, and each slide staying ten seconds unless manually advanced. The first page (being a title) does not move on automatically, and has no transition effect.

To use different options on different page ranges, run `cpdf` multiple times on the file using a different page range each time.

Chapter 8

Watermarks and Stamps

```
cpdf -stamp-on source.pdf
[-scale-stamp-to-fit] [<positioning command>] [-relative-to-cropbox]
in.pdf [<range>] -o out.pdf!

cpdf -stamp-under source.pdf
[-scale-stamp-to-fit] [<positioning command>] [-relative-to-cropbox]
in.pdf [<range>] -o out.pdf

cpdf -combine-pages over.pdf under.pdf -o out.pdf

cpdf ([-add-text <text-format>| -add-rectangle
<size>])
[-font <fontname>]
[-font-size <size-in-points>]
[-color <color>]
[-line-spacing <number >]
[-outline]
[-linewidth <number>]
[-underneath]
[-relative-to-cropbox]
[-prerotate]
[-bates <number>]
[-bates-at-range <number>]
[-bates-pad-to <number>]
[-opacity <number>]
[-midline]
[-topline]
in.pdf [<range>] -o out.pdf
```

See also positioning commands below.

```
cpdf -remove-text in.pdf [<range>] -o out.pdf
```

8.1 Add a Watermark or Logo

The `-stamp-on` and `-stamp-under` operations stamp the first page of a source PDF onto or under each page in the given range of the input file. For example,

```
cpdf -stamp-on logo.pdf in.pdf odd -o out.pdf
```

stamps the file `logo.pdf` onto the odd pages of `in.pdf`, writing to `out.pdf`. A watermark should go underneath each page:

```
cpdf -stamp-under topsecret.pdf in.pdf -o out.pdf
```

The position commands in Section 8.2.4 can be used to locate the stamp more precisely (they are calculated relative to the crop box of the stamp). Or, preprocess the stamp with `-shift` first.

The `-scale-stamp-to-fit` option can be added to scale the stamp to fit the page before applying it. The use of positioning commands together with `-scale-stamp-to-fit` is not recommended.

The `-combine-pages` operation takes two PDF files and stamps each page of one over each page of the other. The length of the output is the same as the length of the “under” file. For instance:

```
cpdf -combine-pages over.pdf under.pdf -o out.pdf
```

Page attributes (such as the display rotation) are taken from the “under” file. For best results, remove any rotation differences in the two files using `-upright` first.

The `-relative-to-cropbox` option takes the positioning command to be relative to the crop box of each page rather than the media box.

8.2 Stamp Text, Dates and Times.

The `-add-text` operation allows text, dates and times to be stamped over one or more pages of the input at a given position and using a given font, font size and color.

```
cpdf -add-text "Copyright 2014 ACME Corp." in.pdf -o out.pdf
```

The default is black 12pt Times New Roman text in the top left of each page. The text can be placed underneath rather than over the page by adding the `-underneath` option.

Text previously added by `cpdf` may be removed by the `-remove-text` operation.

```
cpdf -remove-text in.pdf -o out.pdf
```

8.2.1 Page Numbers

There are various special codes to include the page number in the text:

<code>%Page</code>	Page number in arabic notation (1, 2, 3...)
<code>%roman</code>	Page number in lower-case roman notation (i, ii, iii...)
<code>%Roman</code>	Page number in upper-case roman notation (I, II, III...)
<code>%EndPage</code>	Last page of document in arabic notation
<code>%Label</code>	The page label of the page
<code>%EndLabel</code>	The page label of the last page
<code>%filename</code>	The full file name of the input document

For example, the format `"Page %Page of %EndPage"` might become `"Page 5 of 17"`.

NB: In some circumstances (e.g in batch files) on Microsoft Windows, `%` is a special character, and must be escaped (written as `%%`). Consult your local documentation for details.

8.2.2 Date and Time Formats

%a Abbreviated weekday name (Sun, Mon etc.)
%A Full weekday name (Sunday, Monday etc.)
%b Abbreviated month name (Jan, Feb etc.)
%B Full month name (January, February etc.)
%d Day of the month (01–31)
%e Day of the month (1–31)
%H Hour in 24-hour clock (00–23)
%I Hour in 12-hour clock (01–12)
%j Day of the year (001–366)
%m Month of the year (01–12)
%M Minute of the hour (00–59)
%p "a.m" or "p.m"
%S Second of the minute (00–61)
%T Same as %H:%M:%S
%u Weekday (1–7, 1 = Monday)
%w Weekday (0–6, 0 = Monday)
%Y Year (0000–9999)
%% The % character.

8.2.3 Bates Numbers

Unique page identifiers can be specified by putting **%Bates** in the format. The starting point can be set with the **-bates** option. For example:

```
cpdf -add-text "Page ID: %Bates" -bates 23745 in.pdf -o
```

To specify that bates numbering begins at the first page of the range, use **-bates-at-range** instead. This option must be specified after the range is specified. To pad the bates number up to a given number of leading zeros, use **-bates-pad-to** in addition to either **-bates** or **-bates-at-range**.

8.2.4 Position

The position of the text may be specified either in absolute terms:

```
-pos-center "200 200"
```

Position the center of the baseline text at (200pt, 200pt)

```
-pos-left "200 200"
```

Position the left of the baseline of the text at (200pt, 200pt)

```
-pos-right "200 200"
```

Position the right of the baseline of the text at (200pt, 200pt)

Positions relative to certain common points can be set:

<code>-top 10</code>	Center of baseline 10 pts down from the top center
<code>-topleft 10</code>	Left of baseline 10 pts down and in from top left
<code>-topright 10</code>	Right of baseline 10 pts down and left from top right
<code>-left 10</code>	Left of baseline 10 pts in from center left
<code>-bottomleft 10</code>	Left of baseline 10 pts in and up from bottom left
<code>-bottom 10</code>	Center of baseline 10 pts up from bottom center
<code>-bottomright 10</code>	Right of baseline 10 pts up and in from bottom right
<code>-right 10</code>	Right of baseline 10 pts in from the center right
<code>-diagonal</code>	Diagonal, bottom left to top right, centered on page
<code>-reverse-diagonal</code>	Diagonal, top left to bottom right, centered on page
<code>-center</code>	Centered on page

No attempt is made to take account of the page rotation when interpreting the position, so `-prerotate` must be added to the command line if the file contains pages with a non-zero viewing rotation. This is equivalent to pre-processing the document with `-upright`.

The `-relative-to-cropbox` modifier can be added to the command line to make these measurements relative to the crop box instead of the media box.

The default position is equivalent to `-topleft 100`.

The `-midline` option may be added to specify that the positioning commands above are to be considered relative to the midline of the text, rather than its baseline. Similarly, the `-topline` option may be used to specify that the position is taken relative to the top of the text.

8.2.5 Font and Size

The font may be set with the `-font` option. The 14 Standard PDF fonts are available:

Times-Roman
 Times-Bold
 Times-Italic
 Times-BoldItalic
 Helvetica
 Helvetica-Bold
 Helvetica-Oblique
 Helvetica-BoldOblique
 Courier
 Courier-Bold
 Courier-Oblique
 Courier-BoldOblique
 Symbol
 ZapfDingbats

For example, page numbers in Times Italic can be achieved by:

```
cpdf -add-text "-%Page-" -font "Times-Italic" in.pdf -o out.pdf
```

See Section ?? for how to use other fonts.

The font size can be altered with the `-font-size` option, which specifies the size in points:

```
cpdf -add-text "-%Page-" -font-size 36 in.pdf -o out.pdf!
```


8.2.6 Colors

The `-color` option takes an RGB color, where red, green and blue components range between 0 and 1. The following values are predefined:

Color	R, G, B
white	1, 1, 1
black	0, 0, 0
red	1, 0, 0
green	0, 1, 0
blue	0, 0, 1

```
cpdf -add-text "Hullo" -color "red" in.pdf -o out.pdf
cpdf -add-text "Hullo" -color "0.5 0.5 0.5" in.pdf -o out.pdf
```

Partly-transparent text may be specified using the `-opacity` option. Wholly opaque is 1 and wholly transparent is 0. For example:

```
cpdf -add-text "DRAFT" -color "red" -opacity 0.3 -o out.pdf
```

8.2.7 Outline Text

The `-outline` option sets outline text. The line width (default 1pt) may be set with the `-linewidth` option. For example, to stamp documents as drafts:

```
cpdf -add-text "DRAFT" -diagonal -outline in.pdf -o out.pdf
```

8.2.8 Multi-line Text

The code `\n` can be included in the text string to move to the next line. In this case, the vertical position refers to the baseline of the first line of text (if the position is at the top, top left or top right of the page) or the baseline of the last line of text (if the position is at the bottom, bottom left or bottom right).

```
cpdf -add-text "Specification\n%Page of %EndPage"
-topright 10 in.pdf -o out.pdf
```

The `-midline` option may be used to make these vertical positions relative to the midline of a line of text rather than the baseline, as usual.

The `-line-spacing` option can be used to increase or decrease the line spacing, where a spacing of 1 is the standard.

```
cpdf -add-text "Specification\n%Page of %EndPage"
-topright 10 -line-spacing 1.5 in.pdf -o out.pdf
```

Justification of multiple lines is handled by the `-justify-left`, `-justify-right` and `-justify-center` options. The defaults are left justification for positions relative to the left hand side of the page, right justification for those relative to the right, and center justification for positions relative to the center of the page. For example:

```
cpdf -add-text "Long line\nShort" -justify-right
in.pdf -o out.pdf
```

8.2.9 Special Characters

If your command line allows for the inclusion of unicode characters, the input text will be considered as UTF8 by `cpdf`. Special characters which exist in the PDF WinAnsiEncoding Latin 1 code (such as many accented characters) will be reproduced in the PDF. This does not mean, however, that every special character can be reproduced. You must experiment.

For compatibility with previous versions of `cpdf`, special characters may be introduced manually with a backslash followed by the three-digit octal code of the character in the PDF WinAnsiEncoding Latin 1 Code. The full table is included in Appendix D of the Adobe PDF Reference Manual, which is available at http://www.adobe.com/devnet/pdf/pdf_reference.html.

For example, a German sharp s (ß) may be introduced by `\337`.

8.3 Stamping Graphics

A rectangle may be placed on one or more pages by using the `-add-rectangle <size>` command. Most of the options discussed above for text placement apply in the same way. For example:

```
cpdf -add-rectangle "200 300" -pos-right 30 -color red -outline
in.pdf -o out.pdf
```

This can be used to blank out or highlight part of the document. The following positioning options work as you would expect: `-topleft`, `-top`, `-topright`, `-right`, `-bottomright`, `-bottom`, `-bottomleft`, `-left`, `-center`. When using the option `-pos-left "x y"`, the point (x, y) refers to the bottom-left of the rectangle. When using the option `-pos-right "x y"`, the point (x, y) refers to the bottom-right of the rectangle. When using the option `-pos-center "x y"`, the point (x, y) refers to the center of the rectangle. The options `-diagonal` and `-reverse-diagonal` have no meaning.