

# **Coherent PDF Command Line Toolkit**

User Manual  
Version 1.8 (October 2014)



**Coherent Graphics Ltd**

For bug reports, feature requests and comments, email  
[contact@coherentgraphics.co.uk](mailto:contact@coherentgraphics.co.uk)

©2013 Coherent Graphics Limited. All rights reserved.

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

Contents	iii
<b>1 Basic Usage</b>	<b>1</b>
1.1 Input and Output Files . . . . .	1
1.2 Input Ranges . . . . .	1
1.3 Decryption . . . . .	3
1.4 Standard Input and Standard Output . . . . .	3
1.5 Doing several things at once with AND . . . . .	4
1.6 Units . . . . .	4
1.7 PDF Version Numbers . . . . .	4
1.8 File IDs . . . . .	5
1.9 Linearization . . . . .	5
1.10 Object Streams . . . . .	5
1.11 Malformed Files . . . . .	6
1.12 Error Handling . . . . .	7
1.13 Control Files . . . . .	7
1.14 String Arguments . . . . .	7
1.15 Text Encodings . . . . .	7
<b>2 Merging and Splitting</b>	<b>9</b>
2.1 Merging . . . . .	9
2.2 Splitting . . . . .	9
2.3 Splitting on Bookmarks . . . . .	10
<b>3 Pages</b>	<b>11</b>
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 . . . . .	14
<b>4 Encryption and Decryption</b>	<b>15</b>
4.1 Introduction . . . . .	15
4.2 Encrypting a Document . . . . .	16
4.3 Decrypting a Document . . . . .	16
<b>5 Compression</b>	<b>17</b>
5.1 Decompressing a Document . . . . .	17

5.2	Compressing a Document . . . . .	17
5.3	Squeezing a Document . . . . .	17
<b>6</b>	<b>Bookmarks</b>	<b>19</b>
6.1	List Bookmarks . . . . .	19
6.2	Remove Bookmarks . . . . .	20
6.3	Add Bookmarks . . . . .	20
<b>7</b>	<b>Presentations</b>	<b>21</b>
<b>8</b>	<b>Watermarks and Stamps</b>	<b>23</b>
8.1	Add a Watermark or Logo . . . . .	23
8.2	Stamp Text, Dates and Times. . . . .	24
<b>9</b>	<b>Multipage Facilities</b>	<b>29</b>
9.1	Two-up . . . . .	29
9.2	Inserting Blank Pages . . . . .	29
<b>10</b>	<b>Annotations</b>	<b>31</b>
10.1	List Annotations . . . . .	31
10.2	Copy Annotations . . . . .	31
10.3	Remove Annotations . . . . .	31
<b>11</b>	<b>Document Information and Metadata</b>	<b>33</b>
11.1	Listing Fonts . . . . .	33
11.2	Reading Document Information . . . . .	34
11.3	Setting Document Information . . . . .	35
11.4	Upon Opening a Document . . . . .	36
11.5	Metadata . . . . .	37
11.6	Page Labels . . . . .	37
<b>12</b>	<b>File Attachments</b>	<b>39</b>
12.1	Adding Attachments . . . . .	39
12.2	Listing Attachments . . . . .	39
12.3	Removing Attachments . . . . .	40
<b>13</b>	<b>Working with Images</b>	<b>41</b>
13.1	Detecting Low-resolution Images . . . . .	41
<b>14</b>	<b>Fonts</b>	<b>43</b>
14.1	Copying Fonts . . . . .	43
14.2	Removing Fonts . . . . .	43
14.3	Listing Missing Fonts . . . . .	44
<b>15</b>	<b>Miscellaneous</b>	<b>45</b>
15.1	Draft Documents . . . . .	45
15.2	Blackening Text, Lines and Fills . . . . .	45
15.3	Hairline Removal . . . . .	46
15.4	Garbage Collection . . . . .	46
15.5	Change PDF Version Number . . . . .	46
15.6	Copy ID . . . . .	46

<i>CONTENTS</i>	v
15.7 Remove ID . . . . .	47
<b>A Dates</b>	<b>49</b>
<b>Index</b>	<b>51</b>



# 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`.





# 1 Basic Usage

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

## 1. BASIC USAGE

---

```
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 Decryption

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=fred reverse -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 §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 output, 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 §11.2), redirecting the output to `file.txt`.

## 1. BASIC USAGE

---

### 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). 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

### 1.7 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 §15.5.

## 1.8 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.

## 1.9 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 -l in.pdf -o out.pdf
```

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

This requires the existence of the external program `cpdfwin` which is provided with `cpdf`. For Unix and Mac OS X, and for Windows under Cygwin or Mingw, it suffices to place the `cpdfwin` executable in the same folder as `cpdf`. On Windows with `command.exe`, you must use `-cpdfwin` to let `cpdf` know where to find it:

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

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

## 1.10 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`:

## 1. BASIC USAGE

---

```
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).

Object streams are always removed prior to linearization.

### 1.11 Malformed Files

There are many malformed PDF files in existence, including many produced by otherwise-reputable applications. `cpdf` attempts to correct these problems silently, but sometimes this can be very slow.

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:

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

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

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

## 1.12 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.

## 1.13 Control Files

```
cpdf -control <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.

## 1.14 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 `\\`).

## 1.15 Text Encodings

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

## 1. BASIC USAGE

---

```
-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`.



## 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.

## 2. MERGING AND SPLITTING

---

```
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:

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

## 3 Pages

```
cpdf -scale-page "<scale x> <scale y>" in.pdf [<range>] -o out.pdf
cpdf -scale-to-fit "<x size> <y size>" 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:

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

### 3. PAGES

---

#### 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, see the positioning commands described in §8.2.

#### 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. Doesn't 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.

### 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`.

## 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:

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

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

## 4. ENCRYPTION AND DECRYPTION

---

### 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.



## 5 Compression

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

`cpdf` provides basic facilities for decompressing and compressing PDF streams, and a more advanced PDF *squeezer*.

### 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.

### 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
```

## 5. COMPRESSION

---

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
```

## 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` option 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).

## 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.

## 7. PRESENTATIONS

---

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.

## 8 Watermarks and Stamps

```
cpdf -stamp-on source.pdf in.pdf [<range>] -o out.pdf
cpdf -stamp-under source.pdf in.pdf [<range>] -o out.pdf
cpdf -combine-pages over.pdf under.pdf -o out.pdf

cpdf -add-text <text-format>
    [-font <fontname>]
    [-font-size <size-in-points>]
    [-color <color>]
    [-line-spacing <number>]
    [-outline]
    [-linewidth <number>]
    [-underneath]
    [-relative-to-cropbox]
    [-prerotate]
    [-bates <number>]
    [-opacity <number>]
    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 §8.2 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.

## 8. WATERMARKS AND STAMPS

---

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.

### 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 2013 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
```

### 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.

### Date and Time Formats

<code>%a</code>	Abbreviated weekday name (Sun, Mon etc.)
<code>%A</code>	Full weekday name (Sunday, Monday etc.)
<code>%b</code>	Abbreviated month name (Jan, Feb etc.)
<code>%B</code>	Full month name (January, February etc.)
<code>%d</code>	Day of the month (01–31)
<code>%e</code>	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.

### 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 out.pdf
```

### 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:

## 8. WATERMARKS AND STAMPS

---

<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, bottom right to top left, 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.

### 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 §14.1 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
```

### 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
```

### 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
```

### 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"
-toprigh 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.

## 8. WATERMARKS AND STAMPS

---

```
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
```

### 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](http://www.adobe.com/devnet/pdf/pdf_reference.html).

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

## 9 Multipage Facilities

```
cpdf -twoup-stack in.pdf -o out.pdf
cpdf -twoup in.pdf -o out.pdf
cpdf -pad-before in.pdf [<range>] -o out.pdf
cpdf -pad-after in.pdf [<range>] -o out.pdf
cpdf -pad-every [<integer>] in.pdf -o out.pdf
cpdf -pad-multiple [<integer>] in.pdf -o out.pdf
```

### 9.1 Two-up

This facility puts multiple logical pages on a single physical page.

The `-twoup-stack` operation puts two logical pages on each physical page, rotating them 90 degrees to do so. The new mediabox is thus larger.

The `-twoup` operation does the same, but scales the new sides down so that the media box is unchanged.

### 9.2 Inserting Blank Pages

Sometimes, for instance to get a printing arrangement right, it's useful to be able to insert blank pages into a PDF file. `cpdf` can add blank pages before a given page or pages, or after. The pages in question are specified by a range in the usual way:

```
cpdf -pad-before in.pdf 1 -o out.pdf
```

Add a blank page before page 1 (i.e. at the beginning of the document.)

```
cpdf -pad-after in.pdf 2,16,38,84,121,147 -o out.pdf
```

Add a blank page after pages 2, 16, 38, 84, 121 and 147 (for instance, to add a clean page between chapters of a document.)

The dimensions of the padded page are derived from the boxes (media box, crop box etc.) of the page after or before which the padding is to be applied.

The `-pad-every n` operation places a blank page after every `n` pages, excluding any last one. For example...

## 9. MULTIPAGE FACILITIES

---

```
cpdf -pad-every 3 in.pdf -o out.pdf
```

Add a blank page after every three pages

... on a 9 page document adds a blank page after pages 3 and 6.

The `-pad-multiple n` operation adds blank pages so the document has a multiple of `n` pages. For example:

```
cpdf -pad-multiple 8 in.pdf -o out.pdf
```

Add blank pages to `in.pdf` so it has a multiple of 8 pages.

## 10 Annotations

```
cpdf -list-annotations in.pdf [<range>]
cpdf -copy-annotations from.pdf to.pdf [<range>] -o out.pdf
cpdf -remove-annotations in.pdf [<range>] -o out.pdf
```

### 10.1 List Annotations

The `-list-annotations` operation prints the textual content of any annotations on the selected pages to standard output. Each annotation is preceded by a separation marker consisting of a carriage return, a line of 80 dashes, and another carriage return.

```
cpdf -list-annotations in.pdf > annots.txt
Print annotations from in.pdf, redirecting output to annots.txt.
```

### 10.2 Copy Annotations

The `-copy-annotations` operation copies the annotations in the given page range from one file (the file specified immediately after the option) to another pre-existing PDF. The range is specified after this pre-existing PDF. The result is then written an output file, specified in the usual way.

```
cpdf -copy-annotations from.pdf to.pdf 1-10 -o result.pdf
Copy annotations from the first ten pages of from.pdf onto the PDF file to.pdf, writing the
result to results.pdf.
```

### 10.3 Remove Annotations

The `-remove-annotations` operation removes all annotations from the given page range.

```
cpdf -remove-annotations in.pdf 1 -o out.pdf
Remove annotations from the first page of a file only.
```





# 11 Document Information and Metadata

```
cpdf -list-fonts in.pdf
cpdf -info [-raw | -utf8] in.pdf
cpdf -page-info in.pdf
cpdf -pages in.pdf
cpdf -set-title <title of document> in.pdf -o out.pdf
(Also -set-author etc. See §11.3.)
cpdf -set-page-layout <layout> in.pdf -o out.pdf
cpdf -set-page-mode <mode> in.pdf -o out.pdf
cpdf -hide-toolbar <true | false> in.pdf -o out.pdf
    -hide-menubar
    -hide-window-ui
    -fit-window
    -center-window
    -display-doc-title
cpdf -set-metadata <metadata-file> in.pdf -o out.pdf
cpdf -remove-metadata in.pdf -o out.pdf
cpdf -print-metadata in.pdf -o out.pdf
cpdf -add-page-labels in.pdf -o out.pdf
    [-label-style <style>]
    [-label-prefix <string>]
    [-label-startval <integer>]

cpdf -remove-page-labels in.pdf -o out.pdf
```

## 11.1 Listing Fonts

The `-list-fonts` operation prints the fonts in the document, one-per-line to standard output. For example:

```
1 /F245 /Type0 /ClearGothic-Bold /Identity-H
1 /F247 /Type0 /ClearGothicSerialLight /Identity-H
1 /F248 /Type1 /Times-Roman /WinAnsiEncoding
1 /F250 /Type0 /ClearGothic-RegularItalic /Identity-H
```

## 11. DOCUMENT INFORMATION AND METADATA

---

```
2 /F13 /Type0 /Cleargothic-Bold /Identity-H
2 /F16 /Type0 /Arial-ItalicMT /Identity-H
2 /F21 /Type0 /ArialMT /Identity-H
2 /F58 /Type1 /Times-Roman /WinAnsiEncoding
2 /F59 /Type0 /ClearGothicSerialLight /Identity-H
2 /F61 /Type0 /Cleargothic-BoldItalic /Identity-H
2 /F68 /Type0 /Cleargothic-RegularItalic /Identity-H
3 /F47 /Type0 /Cleargothic-Bold /Identity-H
3 /F49 /Type0 /ClearGothicSerialLight /Identity-H
3 /F50 /Type1 /Times-Roman /WinAnsiEncoding
3 /F52 /Type0 /Cleargothic-BoldItalic /Identity-H
3 /F54 /Type0 /TimesNewRomanPS-BoldItalicMT /Identity-H
3 /F57 /Type0 /Cleargothic-RegularItalic /Identity-H
4 /F449 /Type0 /Cleargothic-Bold /Identity-H
4 /F451 /Type0 /ClearGothicSerialLight /Identity-H
4 /F452 /Type1 /Times-Roman /WinAnsiEncoding
```

The first column gives the page number, the second the internal unique font name, the third the type of font (Type1, TrueType etc), the fourth the PDF font name, the fifth the PDF font encoding.

### 11.2 Reading Document Information

The `-info` option prints entries from the document information dictionary to standard output.

```
$cpdf -info pdf_reference.pdf
Encryption: 40bit
Linearized: true
Permissions: No edit
Version: 1.6
Pages: 1310
Title: PDF Reference, version 1.7
Author: Adobe Systems Incorporated
Subject: Adobe Portable Document Format (PDF)
Keywords:
Creator: FrameMaker 7.2
Producer: Acrobat Distiller 7.0.5 (Windows)
Created: D:20061017081020Z
Modified: D:20061118211043-02'30'
```

The details of the format for creation and modification dates can be found in Appendix A.

By default, `cpdf` strips to ASCII, discarding character codes in excess of 127. In order to preserve the original unicode, add the `-utf8` option. To disable all postprocessing of the string, add `-raw`.

The `-page-info` option prints the media box and other boxes page-by-page to standard output:

```
$cpdf -page-info 14psfonts.pdf
Page 1:
MediaBox: 0.000000 0.000000 600.000000 450.000000
CropBox: 200.000000 200.000000 500.000000 500.000000
BleedBox:
TrimBox:
ArtBox:
Rotation: 0
```

Note that the format for boxes is minimum x, minimum y, maximum x, maximum y.

The `-pages` operation prints the number of pages in the file.

```
cpdf -pages Archos.pdf
8
```

## 11.3 Setting Document Information

The *document information dictionary* in a PDF file specifies various pieces of information about a PDF. These can be consulted in a PDF viewer (for instance, Acrobat).

Here is a summary of the commands for setting entries in the document information dictionary:

Information	Example command-line fragment
Title	<code>cpdf -set-title "Discourses"</code>
Author	<code>cpdf -set-author "Joe Smith"</code>
Subject	<code>cpdf -set-subject "Behavior"</code>
Keywords	<code>cpdf -set-keywords "Ape Primate"</code>
Creator	<code>cpdf -set-creator "Original Program"</code>
Producer	<code>cpdf -set-producer "Distilling Program"</code>
Creation Date	<code>cpdf -set-create "D:19970915110347-08'00' "</code>
Modification Date	<code>cpdf -set-modify "D:19970915110347-08'00' "</code>
Mark as Trapped	<code>cpdf -set-trapped</code>
Mark as Untrapped	<code>cpdf -set-untrapped</code>

(The details of the format for creation and modification dates can be found in Appendix A. Using the date `"now"` uses the time and date at which the command is executed.)

For example, to set the title, the full command line would be

## 11. DOCUMENT INFORMATION AND METADATA

---

```
cpdf -set-title "A Night in London" in.pdf -o out.pdf
```

The text string is considered to be in UTF8 format, unless the `-raw` option is added—in which case, it is unprocessed.

### 11.4 Upon Opening a Document

#### Page Layout

The `-set-page-layout` option specifies the page layout to be used when a document is opened in, for instance, Acrobat. The possible (case-sensitive) values are:

<code>SinglePage</code>	Display one page at a time
<code>OneColumn</code>	Display the pages in one column
<code>TwoColumnLeft</code>	Display the pages in two columns, odd numbered pages on the left
<code>TwoColumnRight</code>	Display the pages in two columns, even numbered pages on the left
<code>TwoPageLeft</code>	(PDF 1.5 and above) Display the pages two at a time, odd numbered pages on the left
<code>TwoPageRight</code>	(PDF 1.5 and above) Display the pages two at a time, even numbered pages on the left

For instance:

```
cpdf -set-page-layout TwoColumnRight in.pdf -o out.pdf
```

#### Page Mode

The *page mode* in a PDF file defines how a viewer should display the document when first opened. The possible (case-sensitive) values are:

<code>UseNone</code>	Neither document outline nor thumbnail images visible
<code>UseOutlines</code>	Document outline (bookmarks) visible
<code>UseThumbs</code>	Thumbnail images visible
<code>FullScreen</code>	Full-screen mode (no menu bar, window controls, or anything but the document visible)
<code>UseOC</code>	(PDF 1.5 and above) Optional content group panel visible
<code>UseAttachments</code>	(PDF 1.5 and above) Attachments panel visible

For instance:

```
cpdf -set-page-mode FullScreen in.pdf -o out.pdf
```

### Display Options

<code>-hide-toolbar</code>	Hide the viewer's toolbar
<code>-hide-menubar</code>	Document outline (bookmarks) visible
<code>-hide-window-ui</code>	Hide the viewer's scroll bars
<code>-fit-window</code>	Resize the document's windows to fit size of first page
<code>-center-window</code>	Position the document window in the center of the screen
<code>-display-doc-title</code>	Display the document title instead of the file name in the title bar

For instance:

```
cpdf -hide-toolbar true in.pdf -o out.pdf
```

## 11.5 Metadata

PDF files can contain a piece of arbitrary metadata, often in XML format. This is typically stored in an uncompressed stream, so that other applications can read it without having to decode the whole PDF. To set the metadata:

```
cpdf -set-metadata data.xml in.pdf -o out.pdf
```

To remove any metadata:

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

To print the current metadata to standard output:

```
cpdf -print-metadata in.pdf
```

## 11.6 Page Labels

It is possible to add *page labels* to a document. These are not the printed on the page, but may be displayed alongside thumbnails or in print dialogue boxes by PDF readers. We use `-add-page-labels` to do this, by default with decimal arabic numbers (1,2,3...). We can add `-label-style` to choose what type of labels to add from these kinds:

## 11. DOCUMENT INFORMATION AND METADATA

---

DecimalArabic	1,2,3,4,5...
LowercaseRoman	i,ii,iii,iv,v...
UppercaseRoman	I,II,III,IV,V...
LowercaseLetters	a,b,c,...,z,aa,bb...
UppercaseLetters	A,B,C,...,Z,AA,BB...
NoLabelPrefixOnly	No number, but a prefix will be used if defined.

We can use `-label-prefix` to add a textual prefix to each label. Consider a file with twenty pages and no current page labels (a PDF reader will assume 1,2,3... if there are none). We will add the following page labels:

i, ii, iii, iv, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, A-0, A-1, A-2, A-3, A-4, A-5

Here are the commands, in order:

```
cpdf -add-page-labels in.pdf 1-4 -label-style LowercaseRoman
-o out.pdf

cpdf -add-page-labels out.pdf 5-14 -o out.pdf

cpdf -add-page-labels out.pdf 15-20 -label-prefix "A-"
-label-startval 0 -o out.pdf
```

By default the labels begin at page number 1 for each range. To override this, we can use `-label-startval` (we used 0 in the final command), where we want the numbers to begin at zero rather than one.

Page labels may be removed altogether by using `-remove-page-labels` command.

## 12 File Attachments

```
cpdf -attach-file <filename> [-to-page <page number>] in.pdf -o out.pdf
cpdf -list-attached-files in.pdf
cpdf -remove-files in.pdf -o out.pdf
```

PDF supports adding attachments (files of any kind, including other PDFs) to an existing file. The `cpdf` tool supports adding and removing *document-level attachments* — that is, ones which are associated with the document as a whole rather than with an individual page, and also *page-level attachments*, associated with a particular page.

### 12.1 Adding Attachments

To add an attachment, use the `-attach-file` option. For instance,

```
cpdf -attach-file sheet.xls in.pdf -o out.pdf
```

attaches the Excel spreadsheet `sheet.xls` to the input file. If the file already has attachments, the new file is added to their number. You can specify multiple files to be attached by using `-attach-file` multiple times. They will be attached in the given order.

The `-to-page` option can be used to specify that the files will be attached to the given page, rather than at the document level. The `-to-page` option may be specified at most once.

### 12.2 Listing Attachments

To list all document- and page-level attachments, use the `-list-attached-files` operation. The page number and filename of each attachment is given, page 0 representing a document-level attachment.

```
$cpdf -list-attached-files 14psfonts.pdf
0 utility.ml
0 utility.mli
4 notes.xls
```

## 12. FILE ATTACHMENTS

---

### 12.3 Removing Attachments

To remove all document-level and page-level attachments from a file, use the `-remove-files` operation:

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



## 13 Working with Images

```
cpdf -image-resolution <minimum resolution> in.pdf [<range>]
```

### 13.1 Detecting Low-resolution Images

To list all images in the given range of pages which fall below a given resolution (in dots-per-inch), use the `-image-resolution` function:

```
cpdf -image-resolution 300 in.pdf [<range>]
```

```
2, /Im5, 531, 684, 149.935297, 150.138267
2, /Im6, 184, 164, 149.999988, 150.458710
2, /Im7, 171, 156, 149.999996, 150.579145
2, /Im9, 65, 91, 149.999986, 151.071856
2, /Im10, 94, 60, 149.999990, 152.284285
2, /Im15, 184, 139, 149.960011, 150.672060
4, /Im29, 53, 48, 149.970749, 151.616446
```

The format is *page number, image name, x pixels, y pixels, x resolution, y resolution*. The resolutions refer to the image's effective resolution at point of use (taking account of scaling, rotation etc).



# 14 Fonts

```
cpdf -copy-font fromfile.pdf -copy-font-page <int>
      -copy-font-name <name> in.pdf [<range>] -o out.pdf
cpdf -remove-fonts in.pdf -o out.pdf
cpdf -missing-fonts in.pdf
```

## 14.1 Copying Fonts

In order to use a font other than the standard 14 with `-add-text`, it must be added to the file. The font source PDF is given, together with the font's resource name on a given page, and that font is copied to all the pages in the input file's range, and then written to the output file.

The font is named in the output file with its basefont name, so it can be easily used with `-add-text`.

For example, if the file `fromfile.pdf` has a font `/GHLIGA+c128` with the name `/F10` on page 1 (this information can be found with `-list-fonts`), the following would copy the font to the file `in.pdf` on all pages, writing the output to `out.pdf`:

```
cpdf -copy-font fromfile.pdf -copy-font-name /F10
      -copy-font-page 1 in.pdf -o out.pdf
```

Text in this font can then be added by giving `-font /GHLIGA+c128`. Be aware that due to the vagaries of PDF font handling concerning which characters are present in the source font, not all characters may be available, or the encoding (mapping from input codes to glyphs) may be non-obvious.

## 14.2 Removing Fonts

To remove embedded fonts from a document, use `-remove-fonts`. PDF readers will substitute local fonts for the missing fonts. The use of this function is only recommended when file size is the sole consideration.

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

## 14. FONTS

---

### 14.3 Listing Missing Fonts

The `-missing-fonts` operation lists any unembedded fonts in the document, one per line.

```
cpdf -missing-fonts in.pdf
```

The format is

```
Page number, Name, Subtype, Basefont, Encoding
```

## 15 Miscellaneous

```
cpdf -draft [-boxes] in.pdf [<range>] -o out.pdf
cpdf -blacktext in.pdf [<range>] -o out.pdf
cpdf -blacklines in.pdf [<range>] -o out.pdf
cpdf -blackfills in.pdf [<range>] -o out.pdf
cpdf -thinlines <minimum thickness> in.pdf [<range>] -o out.pdf
cpdf -clean in.pdf -o out.pdf
cpdf -set-version <version number> in.pdf -o out.pdf
cpdf -copy-id-from source.pdf in.pdf -o out.pdf
cpdf -remove-id in.pdf -o out.pdf
```

### 15.1 Draft Documents

The `-draft` option removes bitmap (photographic) images from a file, so that it can be printed with less ink. Optionally, the `-boxes` option can be added, filling the spaces left blank with a crossed box denoting where the image was. This is not guaranteed to be fully visible in all cases (the bitmap may have been partially covered by vector objects or clipped in the original). For example:

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

### 15.2 Blackening Text, Lines and Fills

Sometimes PDF output from an application (for instance, a web browser) has text in colors which would not print well on a grayscale printer. The `-blacktext` operation blackens all text on the given pages so it will be readable when printed.

This will not work on text which has been converted to outlines, nor on text which is part of a form.

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

The `-blacklines` operation blackens all lines on the given pages.

## 15. MISCELLANEOUS

---

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

The `-blackfills` operation blackens all fills on the given pages.

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

### 15.3 Hairline Removal

Quite often, applications will use very thin lines, or even the value of 0, which in PDF means “The thinnest possible line on the output device”. This might be fine for on-screen work, but when printed on a high resolution device, such as by a commercial printer, they may be too faint, or disappear altogether. The `-thinlines` option prevents this by changing all lines thinner than `<minimal thickness>` to the given thickness. For example:

```
cpdf -thinlines 0.2mm in.pdf [<range>] -o out.pdf
```

Thicken all lines less than 0.2mm to that value.

### 15.4 Garbage Collection

Sometimes incremental updates to a file by an application, or bad applications can leave data in a PDF file which is no longer used. This function removes that unneeded data.

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

### 15.5 Change PDF Version Number

To change the pdf version number, use the `-set-version` operation, giving the part of the version number after the decimal point. For example:

```
cpdf -set-version 4 in.pdf -o out.pdf
```

Change file to PDF 1.4.

This does not alter any of the actual data in the file — just the supposed version number.

### 15.6 Copy ID

The `-copy-id-from` operation copies the ID from the given file to the input, writing to the output.

```
cpdf -copy-id-from source.pdf in.pdf -o out.pdf
```

Copy the id from `source.pdf` to the contents of `in.pdf`, writing to `out.pdf`.

If there is no ID in the source file, the existing ID is retained.

## 15.7 Remove ID

The `-remove-id` operation removes the ID from a document.

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

Remove the ID from `in.pdf`, writing to `out.pdf`.





# A Dates

Dates in PDF are specified according to the following format:

D : YYYYYMMDDHHmmSSOHH' mm'

where:

- YYYY is the year;
- MM is the month;
- DD is the day (01-31);
- HH is the hour (00-23);
- mm is the minute (00-59);
- SS is the second (00-59);
- O is the relationship of local time to Universal Time (UT), denoted by '+', '-' or 'Z';
- HH is the absolute value of the offset from UT in hours (00-23);
- mm is the absolute value of the offset from UT in minutes (00-59).

A contiguous prefix of the parts above can be used instead, for lower accuracy dates. For example:

D : 2013 (2013)

D : 20130103 (3rd March 2013)

D : 201301031854-08' 00' (3rd March 2013, 6:54PM, US Pacific Standard Time)



# Index

- annotations
  - copying, 31
  - listing, 31
  - removing, 31
- attachments, 39
  - adding, 39
  - listing, 39
  - removing, 40
- bates numbers, 25
- blacken fills, 46
- blacken lines, 45
- blacken text, 45
- blank pages
  - inserting, 29
- bookmarks, 19
  - adding, 20
  - listing, 19
  - removing, 20
- color, 26
- compressing, 17
- control file, 6
- copy ID, 46
- crop pages, 14
- date, 24
- dates
  - defined, 49
- decompressing, 17
- decryption, 2, 15
- document information, 33
- document outline, 19
- draft, 45
- encryption, 15
- error handling, 6
- flip pages, 13
- font, 26
- fonts
  - listing, 33
- garbage collection, 46
- hairline removal, 46
- input files, 1
- input range, 1
- linearization, 5
- media box, 14
- merging, 9
- metadata, 33, 37
- outline text, 27
- output files, 1
- owner password, 2
- page
  - layout, 36
  - mode, 36
  - numbers, 24
  - range, 2
- page size, 11
- password, 2
- presentations, 21
- range, 1
- remove ID, 47
- removing text, 24
- reversing, 2
- rotate
  - contents, 13
  - pages, 13
- scale pages, 12
- shift page contents, 12
- splitting, 9
  - on bookmarks, 10

## INDEX

---

squeeze, 17  
stamp text, 24  
stamps, 23  
standard input, 3  
standard output, 3  
  
text encodings, 7  
time, 24  
two-up, 29  
  
units, 4  
user password, 2  
  
version number, 4, 46  
  
watermarks, 23