Coherent PDF Command Line Toolkit

User Manual

Version 1.4 (January 2009)

Coherent Graphics Ltd

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

Contents

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 by using the `-idir` option:

```
cpdf -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` represent the odd and even pages.
- The word `reverse` is the same as `end-1`.
- The word `all` is the same as `1-end`.
- A range must contain no spaces.

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 even,odd -o out.pdf
```

Prepare a document for manual duplexing.

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

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

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 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.

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`.

1.5 Joining several commmands with AND

The keyword `AND` can be used to string together several commands in one. The advantage compared to using pipes is that the file doesn't need to 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:

pt	Points (72 points per inch). The default.
cm	Centimeters
mm	Millimeters
in	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

If you wish to manually alter the PDF version of a file, use the `-set-version` option described in §14.7.

1.8 File IDs

PDF files contain an ID (consisting of two parts), used by some workflow systems to uniquely identify a file. By default, `cpdf` operations leave the file ID unchanged, even when multiple files are created. To override this behavior, add the `-change-id` option to the command line. This will create a new ID for each output file.

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. For example:

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

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

1.10 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. If you are sure the input files you are using are well-formed, the `-fast` option added to the command line (or, if using `AND`, to each section of the command line). Currently this affects only `-add-text`.

1.11 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.

1.12 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 arguements. The commands in each control file are considered in the order in which they are given, after all conventional arguements have been processed.

1.13 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 \\).

2 Merging and Splitting

```
cpdf -merge in1.pdf [<range>] in2.pdf [<range>] ...
     [-retain-numbering] [-remove-duplicate-fonts] -o out.pdf

cpdf -split in.pdf [<range>] -o <format> [-chunk <chunksize>]

cpdf -split-at-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. The range of pages taken can be selected in the usual fashion.

```
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

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.

```
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, and an additional 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 is converted from unicode to ASCII, and the following characters are removed, in addition to any character with ASCII code less than 32 or more than 126:

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

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-to-fit-best "<x size> <y size>" in.pdf [<range>] -o out.pdf
cpdf -scale-to-fit-minus "<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 -crop "<x> <y> <w> <h>" in.pdf [<range>] -o out.pdf
cpdf -remove-crop in.pdf [<range>] -o out.pdf
cpdf -mediabox "<x> <y> <w> <h>" 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
a0landscape       a1landscape        a2landscape
a3landscape       a4landscape        a5landscape
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-to-fit-best` and `-scale-to-fit-minus` are similar, but will rotate a page by $90°$ or $-90°$ respectively on any page where doing so would maximise the scale.

The scale can optionally be set to a percentage of the available area, instead of filling it, when using any of the three operations.

```
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, leaving the page dimensions unchanged. The position of the scaled content in the output can be specified using the same postional operators used when adding text (See §8.2).

```
cpdf -scale-contents 0.5 -topleft 10mm in.pdf -o out.pdf
```

Scale a file's contents on all pages to 50% of its original dimensions, placing the result 10mm from the top left of the page.

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

Set the media box to 500 points square.

The four numbers are minimum x, minimum y, maximum x, maximum y. 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 "0 0 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.

4 Encryption and Decryption

```
cpdf -encrypt <method> <owner> <user>
     [-no-encrypt-metadata] <permissons> 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.

There are three 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, and AES encryption (method `AES`) in Acrobat 7 (PDF 1.6) and above.

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.

4.2 Encrypting a Document

To encrypt a document, the user and owner 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 in.pdf -o out.pdf
cpdf -encrypt AES fred "" 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. A blank user password is permissible. In this event, PDF viewers will typically not prompt for a password for operations allowable with the user password.

When using AES encryption, the option is available not to encrypt 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
```

5 Compression

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

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.

6 Bookmarks

```
cpdf -list-bookmarks [-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.

Bookmark text is either plain text, or encoded as UTF16BE unicode. By default, `cpdf` converts unicode to ASCII text. To prevent this, and return the original unicode, add the `-raw` option to the command.

6.2 Remove Bookmarks

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

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

6.3 Add Bookmarks

The `-add-bookmarks` file adds bookmarks as specified by a *bookmarks file*, a text file 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.

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 Logos, 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>] [-shorter-side]
              [-outline] [-linewidth <number>]
              in.pdf [<range>] -o out.pdf
```

See also positioning commmands 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 `-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` and `-mediabox` 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 2009 ACME Corp." in.pdf -o out.pdf
```

The default is black 12pt Times New Roman text in the top left of each page. 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:

`%Page`	Page number in arabic notation (1, 2, 3...)
`%roman`	Page number in lower-case roman notation (i, ii, iii...)
`%Roman`	Page number in upper-case roman notation (I, II, III...)
`%EndPage`	Last page of document in arabic notation
`\n`	starts a new line

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 (preceeded by a `\`).

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.

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:

`-top 10`	Center of baseline 10 pts down from the top center
`-topleft 10`	Left of baseline 10 pts down and in from top left
`-topright 10`	Right of baseline 10 pts down and left from top right
`-left 10`	Left of baseline 10 pts in from center left
`-bottomleft 10`	Left of baseline 10 pts in and up from bottom left
`-bottom 10`	Center of baseline 10 pts up from bottom center
`-bottomright 10`	Right of baseline 10 pts up and in from bottom right
`-right 10`	Right of baseline 10 pts in from the center right

No attempt is made to take account of the page rotation, so you might like to use `-upright` (see §3.4) first.

The `-shorter-side` modifier can be used to indicate that all the positions above are relative to the shorter side of the page, any rotation required being automatic. In other words, `top, topleft, topright` are either on the top or left, depending upon which is the shorter side, and `bottom, bottomleft, bottomright` are either on the bottom or right similarly.

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 §14.9 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
```

Text can be added in outline style by adding `-outline` to the command. The `-linewidth` option can be used to specify the width of the outline, 1pt being the default.

```
cpdf -add-text "-%Page-" -font-size 36
     -outline -linewidth 2.5 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
```

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 `-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
```

Special Characters

Special characters are introduced 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.

For example, a german sharp s (ß) may be introduced by `\373`.

9 Multipage Facilities

```
cpdf -twoup-stack 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
```

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.

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 media box 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…

```
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.

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] 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
```

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
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
Encrypted: 40bit
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. Add the `-raw` option to print the original unicode.

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:
```

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	`cpdf -set-title "Discourses"`
Author	`cpdf -set-author "Joe Smith"`
Subject	`cpdf -set-subject "Behavior"`
Keywords	`cpdf -set-keywords "Ape Primate"`
Creator	`cpdf -set-creator "Original Program"`
Producer	`cpdf -set-producer "Distilling Program"`
Creation Date	`cpdf -set-create "D:19970915110347-08'00'"`
Modification Date	`cpdf -set-modify "D:19970915110347-08'00'"`
Mark as Trapped	`cpdf -set-trapped`
Mark as Untrapped	`cpdf -set-untrapped`

(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

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

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:

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

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

For instance:

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

Display Options

`-hide-toolbar`	Hide the viewer's toolbar
`-hide-menubar`	Document outline (bookmarks) visible
`-hide-window-ui`	Hide the viewer's scroll bars
`-fit-window`	Resize the document's windows to fit size of first page
`-center-window`	Position the document window in the center of the screen
`-display-doc-title`	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
```

12 File Attachments

```
cpdf -attach-file <filename> in.pdf -o out.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 *top-level attachments* — that is, ones which are associated with the document as a whole rather than with an individual 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.

12.2 Removing Attachments

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

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

13 Extracting Images

```
cpdf -extract-images in.pdf [<range>] -o <string>
```

The Tools can extract images from PDF files to JPEG, JPEG2000, JBIG2 and PNM (Portable Any Map) files. Images which are already in JPEG/JPEG2000/JBIG2 format in the PDF are written in those formats, unaltered. All other images are decoded and written as PNM files (unless the decoding method is unknown). For example,

```
   cpdf -extract-images in.pdf 2-6 -o img
```

might generate `img1.jpg`, `img2.pnm`, `img3.jpg` etc. from the images on pages two to six.

14 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 -copy-font fromfile.pdf -copy-font-page <int>
     -copy-font-name <name> in.pdf [<range>] -o out.pdf
```

14.1 Draft Documents

The `-draft` option removes bitmap (photographic) images from a file, so that it can be printed with less ink. Optionally, the 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 be have been partially covered by vector objects or clipped in the original). For example:

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

14.2 Blackening Text

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
```

14.3 Blackening Lines

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

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

14.4 Blackening Fills

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

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

14.5 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.

14.6 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
```

14.7 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 page data in the file — just the supposed version number.

14.8 Copy ID

The `-copy-id-from` option 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 operation fails.

14.9 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 it's 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.

A Dates

Dates in PDF are specified according to the following format:

D:YYYYMMDDHHmmSSOHH'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:2009 (2009)
D:20090103 (3rd March 2009)
D:200901031854-08'00' (3rd March 2009, 6:54PM, US Pacific Standard Time)