

pdfjam

David Firth*

Reuben Thomas[†]

Markus Kurtz[‡]

v4.3 – 2026-04-02

Contents

1	Overview	2
1.1	Major Features	2
1.2	Strengths, Weaknesses and Alternatives	3
2	Installation	3
3	Comprehensive list of options	4
3.1	Change operation mode	4
3.2	Define where to put files	5
3.3	Configure LaTeX run	5
3.4	Set PDF metadata	6
3.5	Set paper format	6
3.6	Adjust page	7
3.7	Pdfpages settings	9
3.7.1	N-up pages	9
3.7.2	Shift pages	11
3.7.3	Scale pages	12
3.7.4	Mirror pages	14
3.7.5	Create signatures	15
3.7.6	Add blank pages	16
3.7.7	Duplicate pages	17
3.7.8	Run LaTeX commands	19
3.7.9	Add hyperlinks	20
3.8	Graphicx options	21

*Initial author. <http://warwick.ac.uk/dfirth>

[†]Previous maintainer. <https://rrt.sc3d.org>

[‡]Current maintainer. <https://mgkurtz.de>

4 Usage examples	26
4.1 Batch 2-upping of documents	26
4.2 Merging pages from 2 documents	26
4.3 A 4-up document with frames	27
4.4 Convert a ‘US letter’ document to A4	27
4.5 Handouts from presentation slides	27
4.6 Trimming pages; and piped output	27
4.7 Output pages suitable for binding	28
4.8 Input file with nonstandard name	28
4.9 Rotate every 2nd page	28
5 Configuration	28
6 Troubleshooting	28
7 Reporting bugs	29
8 Developing	29
9 The History of <i>pdfjam</i>	29

Website: <https://github.com/pdfjam/pdfjam>

License: GPL 2

Pdfjam assembles a list of documents (pdf, ps, eps, jpg or png) into a single pdf. It allows to rotate, scale, n-up, reorder and combine pages freely using the power of the pdfpages and graphicx LaTeX packages.

1 Overview

1.1 Major Features

Pdfjam is for you, if you want to create ready to print pdf files from any number of source files using a command line interface.

A pretty basic call to `pdfjam` looks like

```
pdfjam --outfile output.pdf input.pdf 2,4-6,9-last input-image.jpg
```

Some options (including example values) include

- `--nup 2x2` for putting 4 input pages on each output page
- `--paper a4` for setting ISO A4 page size
- `--angle 90` for rotating to the left
- `--scale 0.7` for scaling by 70 %
- `--booklet true` to arrange the input as a booklet
- `--signature 24` to arrange as a book consisting of separate signatures, made up of 24 pages each
- `--batch` Create one output file for each input file

See the comprehensive list of options for more and advanced options.

1.2 Strengths, Weaknesses and Alternatives

The many options for dealing with pdfs as graphics plus the support for various image formats turn *pdfjam* into a powerful tool among pdf manipulating programs.

The *pdfjam* script builds upon *LaTeX* and needs a Unix shell. This will be fine for some, but if you do not have both installed, some other tool may suit your needs better.

The LaTeX backend of *pdfjam* deals with pdf files as vector images only. Thus all interactive features such as hyperlinks or bookmarks get lost. If you need to keep them, or want a tool that manipulates any specific pdf properties, best consider one of the below.

All listed alternatives allow combining various input pdfs, selecting pages and rotating them.

- PSUtils: Python-based, features a subset of pdfjam's features using a much smaller technology stack
- PDFtk-Java: Java-based, supports encryption and decryption, as well as working with pdf forms and more
 - PDFtk: also Java-based, the ancestor of the above
 - pdfchain: a GUI for PDFtk
- qpdf: similar features as PDFtk
- mutool: part of *mupdf*, some of PDFtk's features, supports epub files
- PDFsam: Java-based GUI program
- pypdf: self-contained Python package
 - pdfly: CLI for a subset of pypdf
- ghostscript: Has been around for ages, comes with many features, high and low level

Useful tools while working with *pdfjam* contain **pdfinfo** from poppler-utils to inspect pdf metadata and **pdfcrop** to trim any whitespace border around a pdf. You may also want to take a look at **inkscape** for editing pdf files or converting or cropping them via its command line interface.

2 Installation

Currently *pdfjam* works for Unix-like operating systems (Linux, Mac OS X, FreeBSD, etc.) only. Under Windows, you may try a Linux subsystem, Cygwin or Git Bash.

Some options to install *pdfjam*

1. Install via your operating system's repository.
2. If you installed a TeX Live distribution not managed by your operating system, use the TeX Live cockpit.

3. Download or clone the pdfjam sources. Then run `l3build install --full` in the pdfjam directory to install locally. Also see the developing section for working with, rebuilding and testing the code.
4. Download a packaged release, it contains a file `bin/pdfjam` which you need to make available on your `PATH` and `man/pdfjam.1` which you should put in your `MANPATH`.

If you choose one of the later options, make sure your system contains a working installation of LaTeX and the pdfpages LaTeX package.

For the `--keepinfo` option, you need `pdftinfo` available by installing `poppler-utils`.

3 Comprehensive list of options

3.1 Change operation mode

-h, -help

Print help message.

-V, -version

Print the version number.

-configpath

Print the `configpath` variable.

-q, -quiet

Suppress verbose commentary on progress.

-batch

Run pdfjam sequentially on each input file in turn, and produce a separate output file for each input.

-checkfiles *respectively* -no-checkfiles

Use `file` utility to determine file type.

-vanilla

Suppress the reading of any pdfjam configuration files.

`-enc` encoding

Command-line encoding as understood by `iconv`.

3.2 Define where to put files

`-o` output, `-outfile` output, `-output` output

File or directory name for output(s).

`-suffix` string

Suffix for output, when file name is not given explicitly.

`-tidy` *respectively* `-no-tidy`

Clean temporary build directory. On by default.

`-builddir` directory

Set build directory.

3.3 Configure LaTeX run

`-latex` engine

Absolute path to LaTeX engine to be used.

`-latexopts` latex-command-line-options

Provide arguments to use when calling the LaTeX engine.

`-preamble` tex

Append code to the LaTeX preamble.

`-runs` number-of-runs

Run latex N times, for each output document made.

3.4 Set PDF metadata

`-keepinfo` *respectively* `-no-keepinfo`

Preserve Title, Author, Subject and Keywords (from the last input PDF) in the output PDF file.

`-pdftitle` string

Set Title of the output PDF file.

`-pdfauthor` string

Set Author of the output PDF file.

`-pdfsubject` string

Set Subject of the output PDF file.

`-pdfkeywords` string

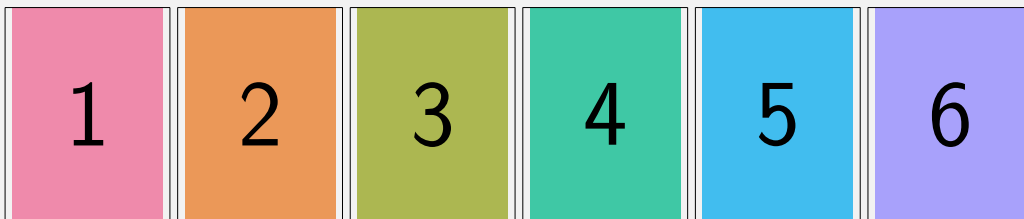
Set Keywords of the output PDF file.

3.5 Set paper format

`-paper` paper-name

Paper format.

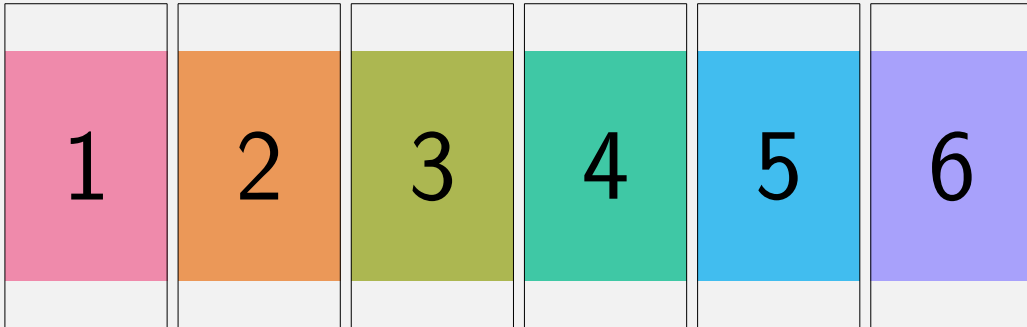
`pdfjam --paper letter a4.pdf 1-6`



`-papersize width,height`

Specify a paper size as width × height.

`pdfjam --papersize 15cm,30cm a4.pdf 1-6`

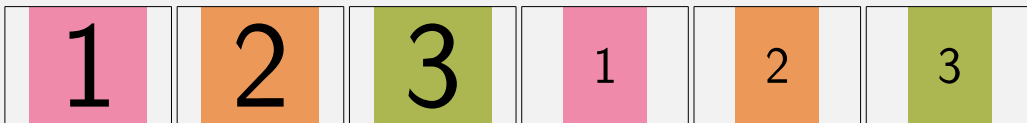


3.6 Adjust page

`-landscape respectively -no-landscape`

Exchange width and height of paper.

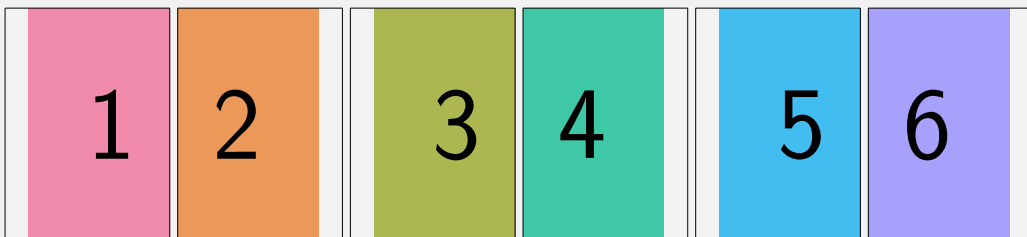
`pdfjam --landscape square.pdf a4.pdf 1-3`



`-twoside respectively -no-twoside`

Specify twoside document class option.

`pdfjam --twoside --offset '3cm 0' a4.pdf 1-6`



`-otheredge` *respectively* `-no-otheredge`

Rotate every odd page by 180 degrees.

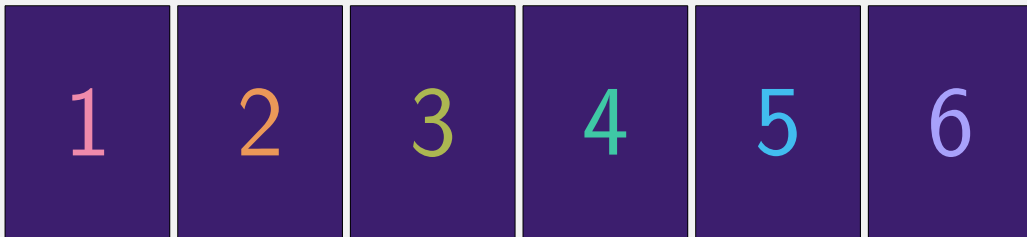
`pdfjam --otheredge a4.pdf 1-6`



`-pagecolor` `rgb`

Background color.

`pdfjam --pagecolor 60,30,110 a4text.pdf 1-6`



3.7 Pdftables settings

3.7.1 N-up pages

`--nup XxY`

Put multiple logical pages onto each sheet of paper.

`pdftables --nup 3x3 a4.pdf`

`pdftables --nup 4x2 --landscape a4.pdf`

`pdftables --nup 3x2 --papersize 10cm,10cm a4.pdf`

`-column` bool

Use column-major layout, where successive pages are arranged in columns down the paper.

```
pdfjam --column true --nup 3x3 a4.pdf
```

1	4	7	10	13	16	19	21	23
2	5	8	11	14	17	20	22	24
3	6	9	12	15	18			

`-columnstrict` bool

For column-major layout only: Do not balance the columns on the last page.

```
pdfjam --columnstrict true --column true --nup 3x3 a4.pdf
```

1	4	7	10	13	16	19	22	
2	5	8	11	14	17	20	23	
3	6	9	12	15	18	21	24	

3.7.2 Shift pages

`--delta 'horizontal-space vertical-space'`

Put some horizontal and vertical space between the logical pages.

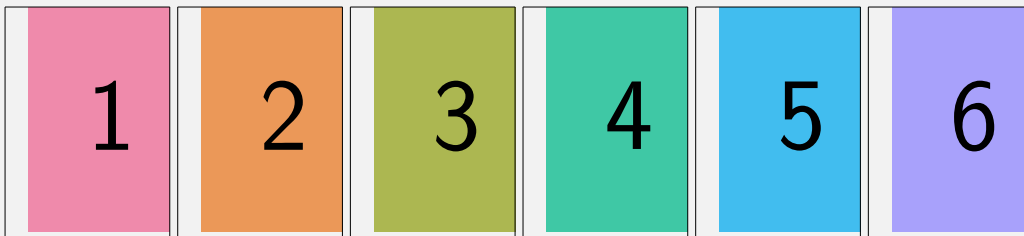
```
pdfjam --delta '1cm 2mm' --nup 3x3 a4.pdf
```



`--offset 'horizontal-displacement vertical-displacement'`

Displace the origin of the inserted pages.

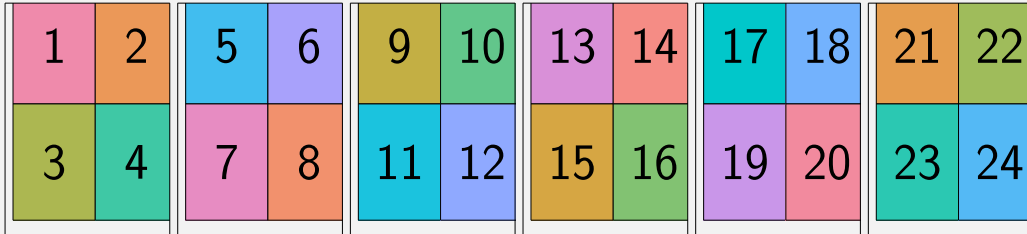
```
pdfjam --offset '3cm 1cm' a4.pdf 1-6
```



`--frame` bool

Put a frame around each logical page.

```
pdfjam --frame true --nup 2x2 --offset '1cm 2cm' a4.pdf
```

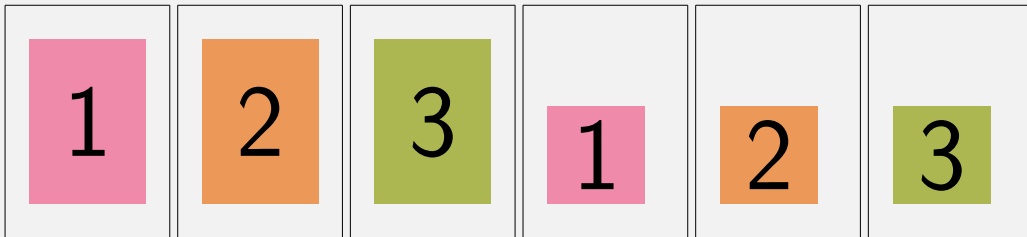


3.7.3 Scale pages

`--noautoscale` bool

Suppress automatic scaling of pages.

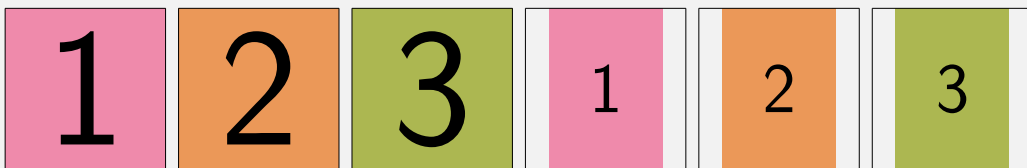
```
pdfjam --noautoscale true a5.pdf square.pdf 1-3
```



`--fitpaper` bool

Adjust the paper size to the size of the inserted document.

```
pdfjam --fitpaper true square.pdf a4.pdf 1-3
```



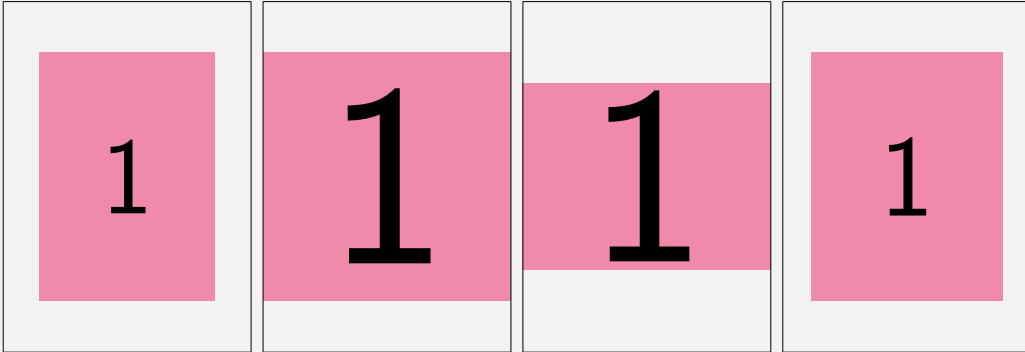
`--turn` bool

Tell PDF viewer to display landscape pages in landscape orientation. On by default.

`-pagetemplate` pagenumber

Declare page to be used as a template. All other pages are scaled such that they match within its size.

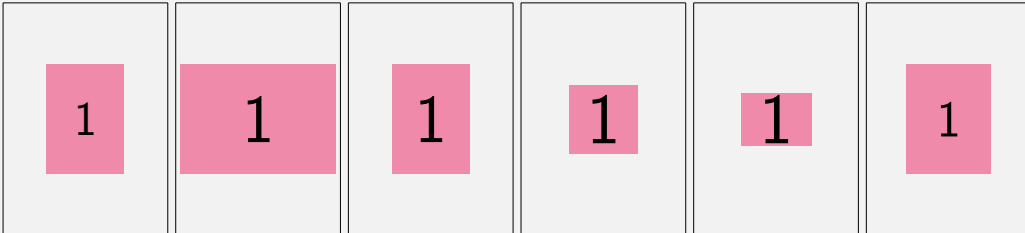
```
pdfjam --pagetemplate 2 a4+square.pdf - beamer.pdf letter.pdf 1
```



`-templatesize` width,height

Specify size of page template. All pages are scaled such that they match within this size.

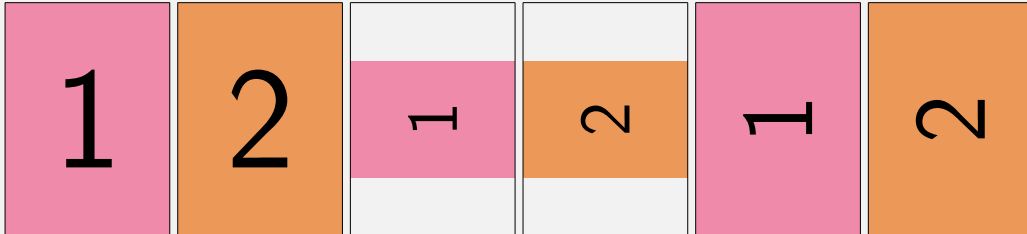
```
pdfjam --templatesize '30cm,20cm' \  
a4.pdf a4landscape.pdf a5.pdf square.pdf beamer.pdf letter.pdf 1
```



`--rotateoversize` bool

Rotate oversized pages.

```
pdfjam --rotateoversize true a5.pdf a4.pdf a4landscape.pdf 1-2
```

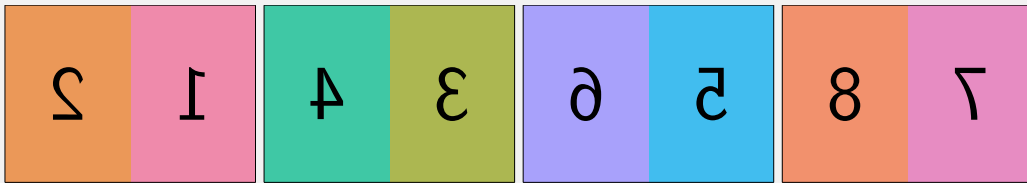


3.7.4 Mirror pages

`--reflect` bool

Reflect output pages.

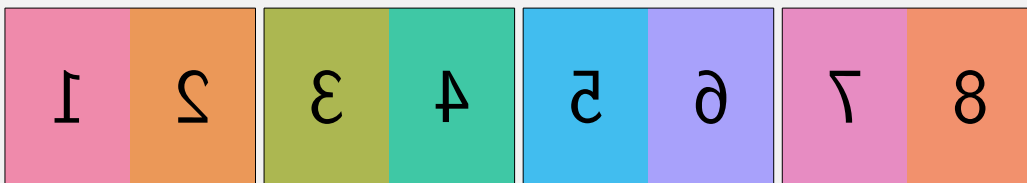
```
pdfjam --reflect true --nup 2x1 --landscape a4.pdf 1-8
```



`--reflect*` bool

Reflect input pages.

```
pdfjam --reflect\* true --nup 2x1 --landscape a4.pdf 1-8
```

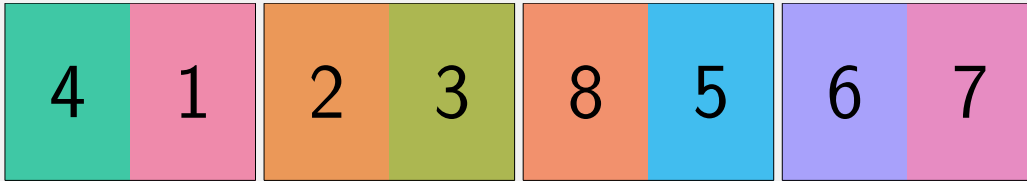


3.7.5 Create signatures

`--signature` multiple-of-4

Create booklets by rearranging pages into signatures of N (the argument) pages each.

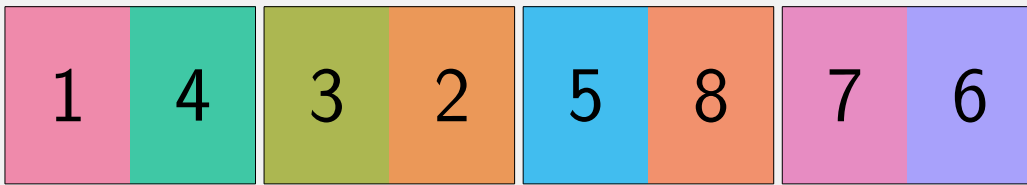
```
pdfjam --signature 4 --nup 2x1 --landscape a4.pdf 1-8
```



`--signature*` multiple-of-4

Similar to signature, but for right-edge binding.

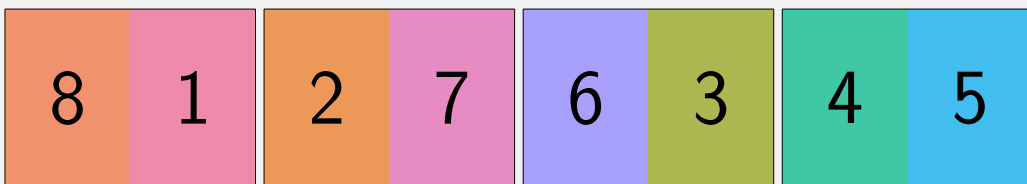
```
pdfjam --signature\* 4 --nup 2x1 --landscape a4.pdf 1-8
```



`--booklet` bool

Same as signature with signature size chosen such that all pages fit into one signature.

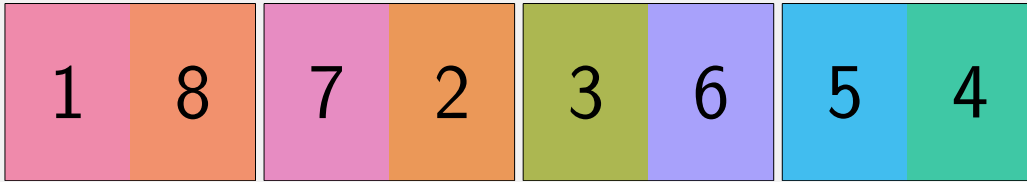
```
pdfjam --booklet true --nup 2x1 --landscape a4.pdf 1-8
```



`--booklet*` bool

Similar to `booklet`, but for right-edge binding.

```
pdfjam --booklet\* true --nup 2x1 --landscape a4.pdf 1-8
```



`--flip-other-edge` bool

For signatures/booklets: set duplex binding edge perpendicular (instead of parallel) to signature binding edge. Has same result as `otheredge`.

```
pdfjam --flip-other-edge true --booklet true --nup 2x1 \  
--landscape a4.pdf 1-8
```

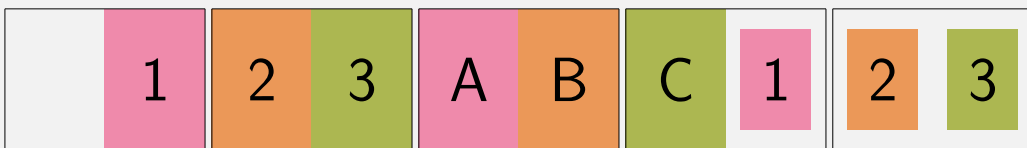


3.7.6 Add blank pages

`--openright` bool

Put an empty page before the very first logical page.

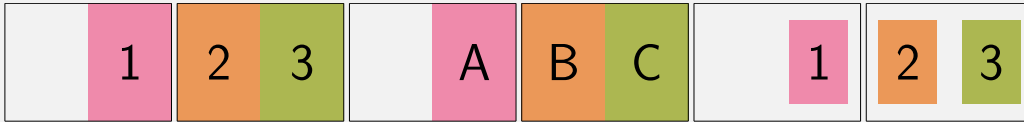
```
pdfjam --openright true --landscape --nup 2x1 \  
a4.pdf a4abc.pdf a5.pdf 1-3
```



`--openrighteach` bool

Put an empty page before the first logical page of each file.

```
pdfjam --openrighteach true --landscape --nup 2x1 \  
a4.pdf a4abc.pdf a5.pdf 1-3
```

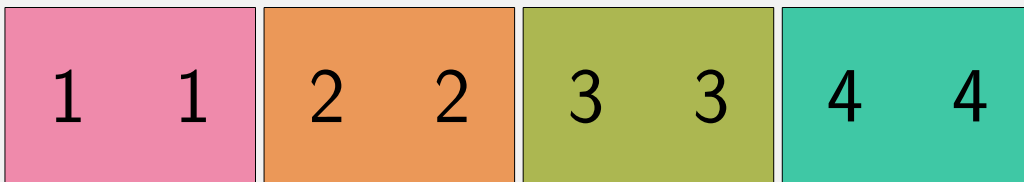


3.7.7 Duplicate pages

`--doublepages` bool

Insert every page twice.

```
pdfjam --doublepages true --landscape --nup 2x1 a4.pdf 1-4
```



`--doublepagetwist` bool

Insert every page twice: once upside down and once normally.

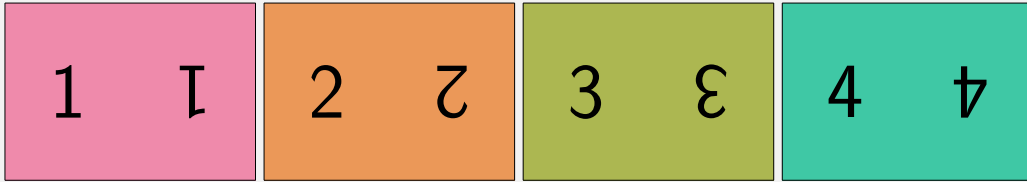
```
pdfjam --doublepagetwist true --landscape --nup 2x1 a4.pdf 1-4
```



`--doublepagestwistodd` bool

Insert every page twice: once normally and once upside down.

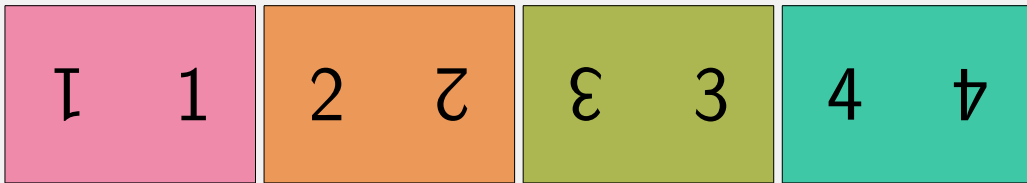
```
pdfjam --doublepagestwistodd true --landscape --nup 2x1 a4.pdf 1-4
```



`--doublepagestwist*` bool

Insert every page twice: for odd pages, the first copy is upside down; for even pages, the second copy.

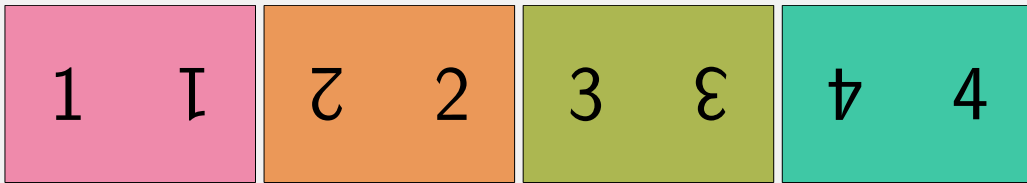
```
pdfjam --doublepagestwist\* true --landscape --nup 2x1 a4.pdf 1-4
```



`--doublepagestwistodd*` bool

Insert every page twice: for odd pages, the second copy is upside down; for even pages, the first copy.

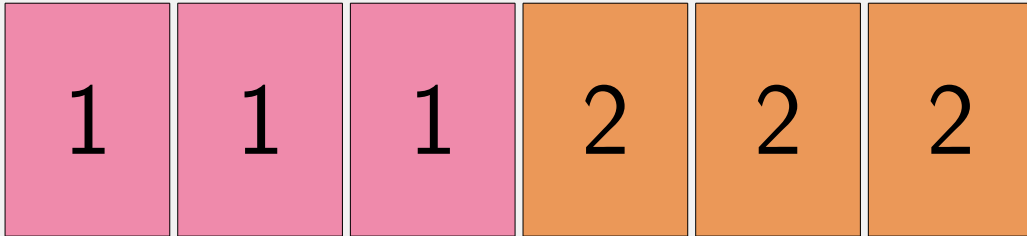
```
pdfjam --doublepagestwistodd\* true --landscape --nup 2x1 a4.pdf 1-4
```



`-duplicatepages number`

Insert every page multiple times.

```
pdfjam --duplicatepages 3 a4.pdf 1-2
```

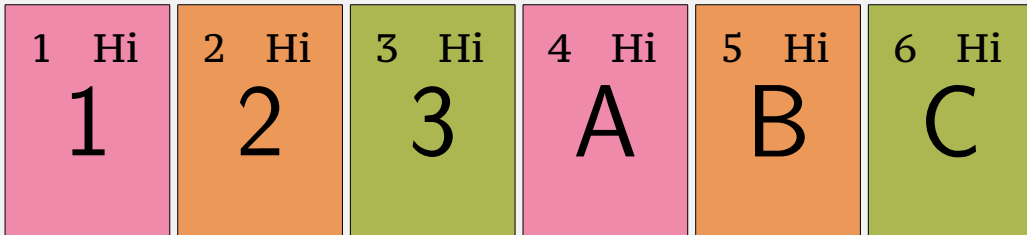


3.7.8 Run LaTeX commands

`-pagecommand tex`

Declare LaTeX commands, which are executed on each sheet of paper. Default: `\thispagestyle{empty}`.

```
pdfjam --pagecommand '\section{Hi}\thispagestyle{empty}' \  
a4.pdf a4abc.pdf 1-3
```



`-pagecommand* tex`

Declare LaTeX commands, which are executed on the very first page only.

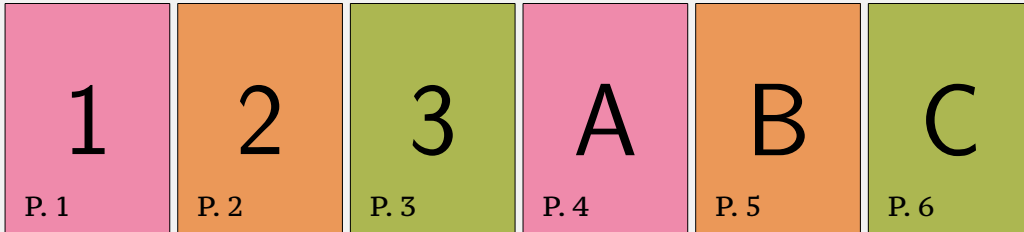
```
pdfjam --pagecommand\* '\section{:D}' a4.pdf a4abc.pdf 1-3
```



`-picturecommand tex`

Similar to `pagecommand`, but executed within a picture environment with base point at the lower left page corner.

```
pdfjam --picturecommand '\put(72,72){P. \thepage}' \  
a4.pdf a4abc.pdf 1-3
```



`-picturecommand* tex`

Similar to `picturecommand`, but for very first page only.

```
pdfjam --picturecommand* '\put(72,72){:D}' a4.pdf a4abc.pdf 1-3
```



3.7.9 Add hyperlinks

`-link bool`

Each inserted page becomes the target of the hyperlink `<filename>.<pagenumber>`.

`-linkname name`

For link option only: Change the link base name from `<filename>` to `name`.

`-threadname name`

For thread option only: Change the thread name from `<filename>` to `name`.

`--linkfit` (Fit | FitH <top> | FitV <left> | FitB | FitBH <top> | FitBV <left> | Region)

For link option only: Specify, how the viewer displays a linked page.

`--addtoc` pagenumber,section,level,heading,label

Add an entry to the table of contents.

`--addtolist` pagenumber,type,heading,label

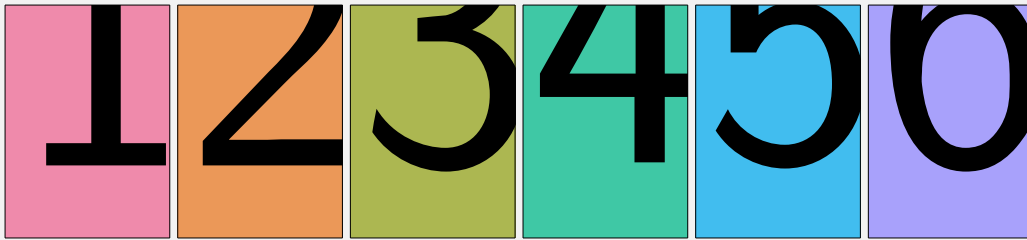
Add an entry to the list of figures, the list of tables, or any other list.

3.8 Graphicx options

`--viewport` 'left bottom right top'

Consider image to consist of given rectangle only.

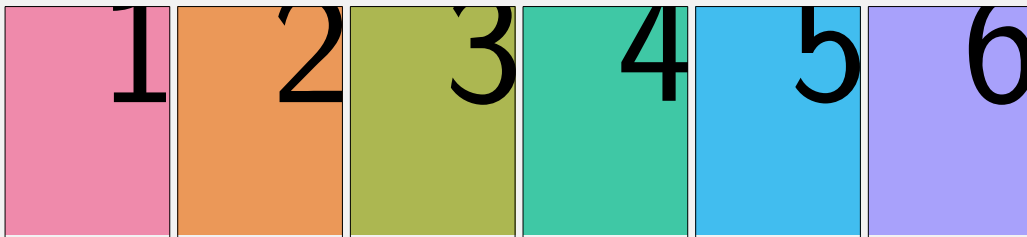
```
pdfjam --viewport '8cm 8cm 12cm 16cm' a4.pdf 1-6
```



`--trim` 'left bottom right top'

Similar to viewport, but here the four lengths specify the amount to remove or add to each side.

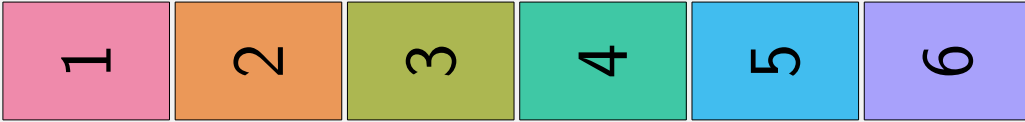
```
pdfjam --trim '0 4cm 8cm 16cm' a4.pdf 1-6
```



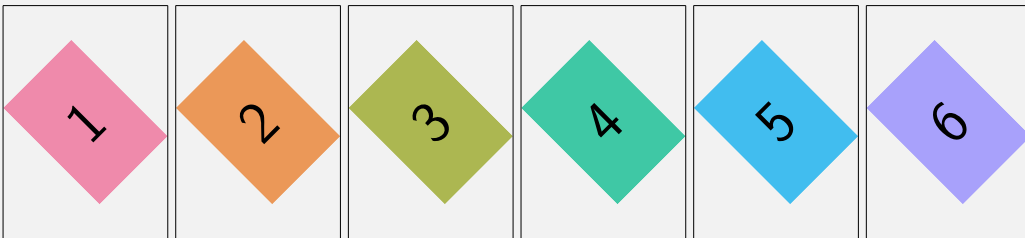
-angle angle

Rotation angle (counterclockwise).

`pdfjam --landscape --angle 90 a4.pdf 1-6`



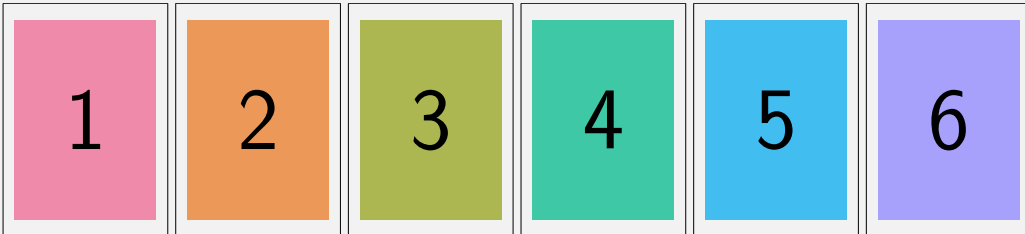
`pdfjam --angle 45 a4.pdf 1-6`



-width width

Required width. The graphic is scaled to this width.

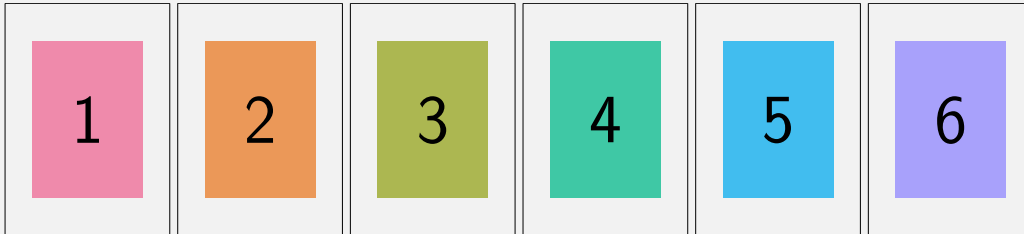
`pdfjam --width 18cm a4.pdf 1-6`



`--height height`

Required height. The graphic is scaled to this height.

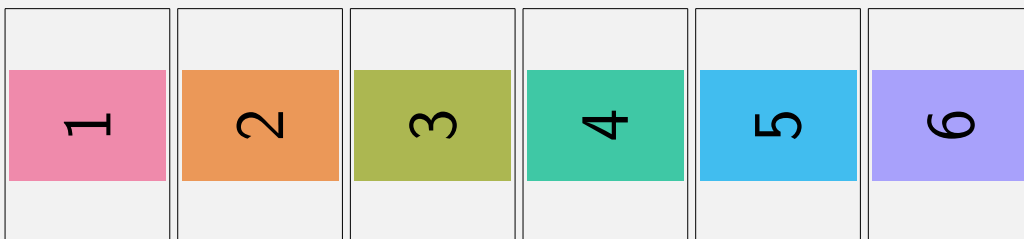
```
pdfjam --height 20cm a4.pdf 1-6
```



```
pdfjam --angle 90 --height 20cm --noautoscale true a4.pdf 1-6
```



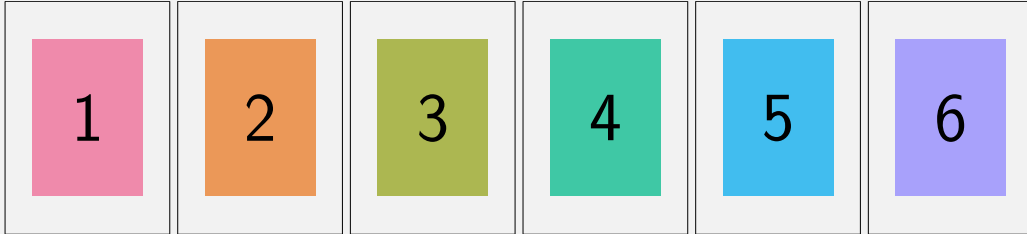
```
pdfjam --height 20cm --angle 90 --noautoscale true a4.pdf 1-6
```



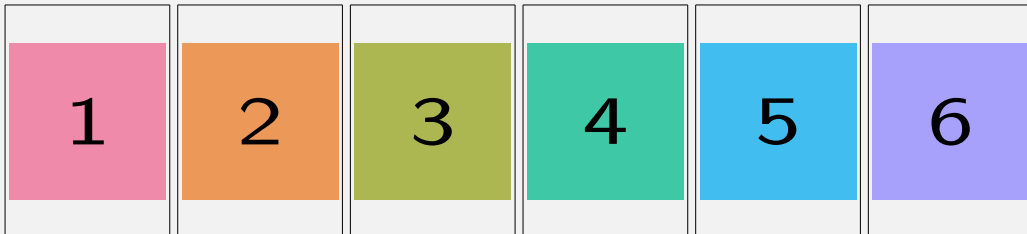
`--keepaspectratio` bool

Do not distort figure if both width and height are given.

```
pdfjam --keepaspectratio true --width 20cm --height 20cm a4.pdf 1-6
```



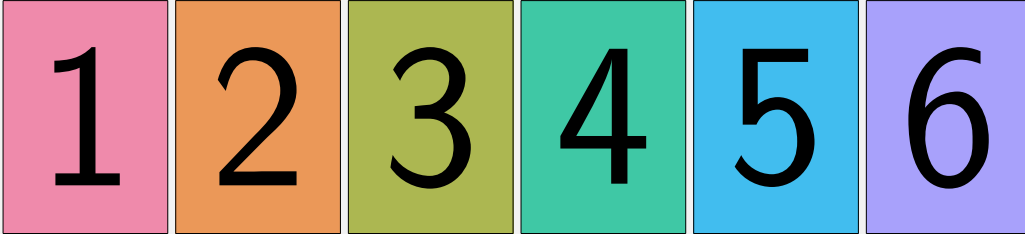
```
pdfjam --keepaspectratio false --width 20cm --height 20cm a4.pdf 1-6
```



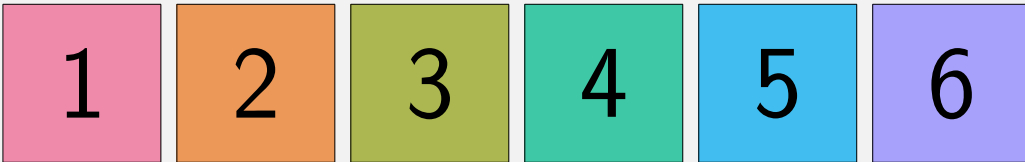
-scale float

Scale factor. Applied after automatic scaling.

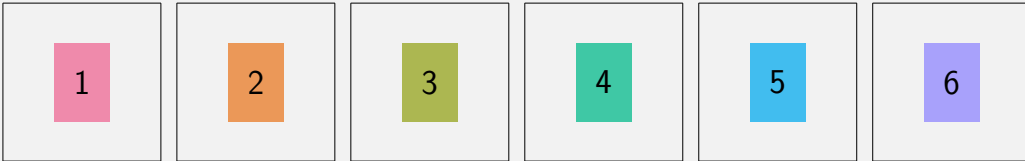
```
pdfjam --scale 2 a4.pdf 1-6
```



```
pdfjam --scale 0.5 --papersize 10cm,10cm --noautoscale true \  
a4.pdf 1-6
```



```
pdfjam --scale 0.5 --papersize 10cm,10cm a4.pdf 1-6
```



-clip bool

Clip the graphic to the viewport.

```
pdfjam --clip true --viewport '8cm 8cm 12cm 16cm' a4.pdf 1-6
```



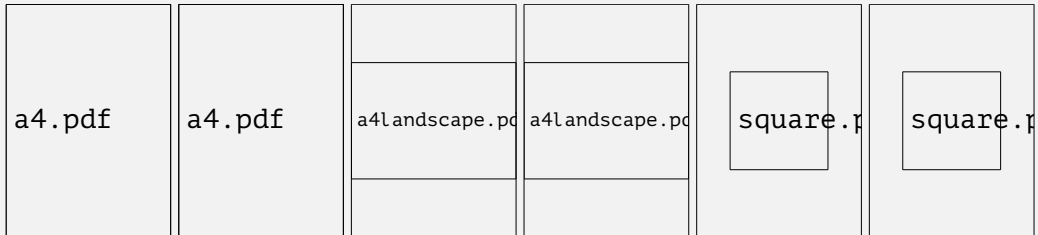
`-pagebox` (mediabox | cropbox | bleedbox | trimbox | artbox)

Specify which PDF bounding box specification to read.

`-draft` bool

Switch to draft mode.

```
pdfjam --draft true a4.pdf a4landscape.pdf square.pdf 1-2
```



4 Usage examples

4.1 Batch 2-upping of documents

Consider converting each of two documents to a side-by-side “2-up” format. Since we want the two documents to be processed separately, we’ll use the `--batch` option:

```
pdfjam --batch --nup 2x1 --suffix 2up --landscape file1.pdf file2.pdf
```

This will produce new files `file1-2up.pdf` and `file2-2up.pdf` in the current working directory.

4.2 Merging pages from 2 documents

Suppose we want a single new document which puts together selected pages from two different files:

```
pdfjam file1.pdf '{},2-' file2.pdf '10,3-6' --outfile ../myNewFile.pdf
```

The new file `myNewFile.pdf`, in the parent directory of the current one, contains an empty page, followed by all pages of `file1.pdf` except the first, followed by pages 10, 3, 4, 5 and 6 from `file2.pdf`.

The resulting PDF page size will be whatever is the default paper size for you at your site. If instead you want to preserve the page size of (the first included page from) `file1.pdf`, use the option `--fitpaper true`.

All pages in an output file from pdfjam will have the same size and orientation. For joining together PDF files while preserving *different* page sizes and orientations, pdfjam is not the tool to use.

4.3 A 4-up document with frames

To make a portrait-oriented 4-up file from the pages of three input files, with a thin-line frame around the input pages:

```
pdfjam file1.pdf file2.pdf file3.pdf --frame true --nup 2x2 \  
  --suffix 4up --outfile ~/Documents
```

Here a *directory* was specified at `--outfile`: the resultant file in this case will be `~/Documents/file3-4up.pdf`. (Note that **if there's a writeable file with that name already, it will be overwritten**: no check is made, and no warning given.)

4.4 Convert a 'US letter' document to A4

Suppose we have a document made up of 'US letter' size pages, and we want to convert it to A4:

```
pdfjam 'my US letter file.pdf' --a4paper --outfile 'my A4 file.pdf'
```

4.5 Handouts from presentation slides

A useful application of `pdfjam` is for producing a handout from a file of presentation slides. For slides made with the standard 4:3 aspect ratio a nice 6-up handout on A4 paper can be made by

```
pdfjam --nup 2x3 --frame true --noautoscale false --delta '0.2cm 0.3cm' \  
  --scale 0.95 myslides.pdf --outfile myhandout.pdf
```

The `--delta` option here comes from the `pdfpages` package; the `--scale` option is passed to LaTeX's `\includegraphics` command.

Slides made by LaTeX's `beamer` package, using the `handout` class option, work especially nicely with this! The example wrapper scripts `pdfjam-slides3up` and `pdfjam-slides6up`, in the `pdfjam-extras` repository, are for 3-up and 6-up handouts, respectively.

4.6 Trimming pages; and piped output

Suppose we want to trim the pages of our input file prior to n-upping. This can be done by using a pipe:

```
pdfjam myfile.pdf --trim '1cm 2cm 1cm 2cm' --clip true \  
  --outfile /dev/stdout \  
  | pdfjam --nup 2x1 --frame true --outfile myoutput.pdf
```

The `--trim` option specifies an amount to trim from the left, bottom, right and top sides respectively; to work as intended here it needs also `--clip true`. These (i.e., `trim` and `clip`) are in fact options to LaTeX's `\includegraphics` command (in the standard `graphics` package).

Thanks go to Christophe Lange and Christian Lohmaier for suggesting an example on this.

4.7 Output pages suitable for binding

To offset the content of double-sided printed pages so that they are suitable for binding with a Heftstreifen, use the `--twoside` option:

```
pdfjam --twoside myfile.pdf --offset '1cm 0cm' --suffix 'offset'
```

4.8 Input file with nonstandard name

To use PDF input files whose names do not end in `.pdf`, you will need to use the `--checkfiles` option. This depends on the availability of the `file` utility, with support for the options `-Lb`; this can be checked by trying

```
file -Lb 'my PDF file'
```

where `'my PDF file'` is the name of a PDF file on your system. The result should be something like `'PDF document, version 1.4'` (possibly with a different version number).

With `'file -Lb'` available, we can use PDF files whose names lack the usual `.pdf` extension. For example,

```
pdfjam --nup 2x1 --checkfiles 'my PDF file'
```

will result in a file named `'my PDF file-2x1.pdf'` in the current working directory.

4.9 Rotate every 2nd page

If you want to print a landscape-oriented PDF document on both sides of the paper, using a duplex printer that does not have `'tumble'` capability, make a new version with every second page rotated for printing:

```
pdfjam --landscape --doublepagestwistodd true my-landscape-document.pdf
```

5 Configuration

You can configure `pdfjam` to your needs by putting a file called `pdfjam.conf` in any directory specified by `pdfjam --configpath` or a `.pdfjam.conf` in your home directory. See the `pdfjam.conf` coming with your installation or online for a list of available configuration variables.

For setting the default paper size, also consider configuring (or installing) `libpaper`.

6 Troubleshooting

- The script runs but the output doesn't look the way it should. Why?

Most likely either your `pdfTeX` or your `pdftex` installation is an old version. You could check also that `pdftex.def`, typically to be found in

.../texmf/tex/latex/graphics/, is up to date. If the problem persists even with up-to-date versions of pdfTeX, pdftex.def and pdfpages, then please do report it.

- What can I do to solve a “Too many open files” error?

This error has been reported to occur sometimes, when dealing with large numbers of documents/pages. A suggested solution, if this happens, is to include additionally (in the call to pdfjam):

```
--preamble '\let\mypdfximage\pdfximage
\def\pdfximage{\immediate\mypdfximage}'
```

- Minor details of the font got missing in the output. Why?

Sometimes font information (such as ligatures) is lost from the output of pdfjam. It seems that a fairly simple fix when this happens is to add the option `--preamble '\pdfinclusioncopyfonts=1'` in your call to pdfjam.

- My `--preamble '\usepackage[...]{geometry}'` causes failure. What shall I do?

The `geometry` package gets already loaded by pdfjam, so just use `\geometry{}` for any further settings.

- My `--preamble '\usepackage[...]{color}'` does not work. What to do?

Either do not use in conjunction with `--pagecolor` or use with `--pagecolor`, but then do not load the `color` package again.

7 Reporting bugs

Please report any bugs found in pdfjam on GitHub.

8 Developing

This project uses l3build for building (`l3build unpack`), testing (`l3build check`) and installing (`l3build install`). See the first few lines of `build.lua` for more information.

9 The History of pdfjam

The first versions of PDFjam were made by David Firth with the first release of the `pdfnup` script dating back to 2002-04-04 and the name PDFjam debuting with the 1.0 release on 2004-05-07. Several releases followed until version 2.08 on 2010-11-14.

Nine years later — on 2019-11-14 — David Firth published an overhauled version 3.02, now called pdfjam (in lower letters). The next November, maintenance went over to Reuben Thomas where it remained until November 2024, when Markus Kurtz took over.

What follows is David Firth’s account of pdfjam’s history:

This all grew originally from a script named `pdfnup`. That was later joined, in a published package called ‘PDFjam’, by two further scripts `pdfjoin` and `pdf90`.

At version 2.00, everything was unified through a single script `pdfjam`, with many more options. Along with `pdfjam` various ‘wrapper’ scripts — i.e., other scripts that use `pdfjam` in different ways — were provided, mainly as examples.

From version 3.02, the extra ‘wrapper’ scripts are removed from the package, mainly because they are hard to maintain: different users want different things, and `pdfjam` itself provides all the options in any case. So I have broken out the wrapper scripts into a separate repository, *unsupported* — so that people can still see and use/adapt them if they want. And maybe even someone else will want to take on the task of improving and maintaining some of them, who knows? The wrapper scripts (**no longer maintained**) can now be found at <https://github.com/pdfjam/pdfjam-extras>.

For historic release notes, see the `README.md` as of `pdfjam v3.04`.

Releases from version 1.0 up to version 2.08 are still available at <https://pdfjam.github.io/pdfjam>. For any releases after that, see <https://github.com/pdfjam/pdfjam/releases>.