

**NAME**

**pdftk** - manipulate PDF files

**SYNOPSIS**

**pdftk -h | --help**

**pdftk** [*handle=*]pdf\_input\_file ... | - | **PROMPT** [*input\_pw* [*handle=*]password ... | **PROMPT**]  
 [cat | shuffle | burst | generate\_fdf | fill\_form | background | multibackground | stamp | multistamp |  
 dump\_data | dump\_data\_utf8 | dump\_data\_fields | dump\_data\_fields\_utf8 | update\_info |  
 update\_info\_utf8 | attach\_files | unpack\_files arguments] [output\_filename | - | **PROMPT**]  
 [encrypt\_40bit | encrypt\_128bit] [allow\_permissions] [owner\_pw password | **PROMPT**]  
 [user\_pw password | **PROMPT**] [flatten] [compress | uncompress]  
 [keep\_first\_id | keep\_final\_id | drop\_xfa] [dont\_ask | do\_ask] [verbose]

**DESCRIPTION**

If PDF is electronic paper, then **pdftk** is an electronic staple-remover, hole-punch, binder, secret-decoder-ring, and X-Ray glasses. It is a simple tool that can be used to

- o merge PDF documents or collate PDF page scans
- o split PDF pages into a new document
- o rotate PDF documents or pages
- o decrypt input as necessary (password required)
- o encrypt output as desired
- o fill PDF forms with X/FDF data and/or flatten forms
- o generate FDF data stencils from PDF forms
- o apply a background watermark or a foreground stamp
- o report PDF metrics such as metadata and bookmarks
- o update PDF metadata
- o attach files to PDF pages or the PDF document
- o unpack PDF attachments
- o burst a PDF document into single pages
- o uncompress and re-compress page streams
- o repair corrupted PDF (where possible)

**Input Arguments**

[*handle=*]pdf\_input\_file ... | - | **PROMPT**

A list of the PDF input files. If you plan to combine these PDFs (without using handles) then list the files in the order you want them combined. Use - to pass a single PDF into pdftk via stdin. Use **PROMPT** to make **pdftk** ask the user interactively for an input file name.

Every input can be associated with a *handle*, a single upper-case letter, e.g.

A=input1.pdf B=input2.pdf

Handles are often omitted. They are useful when specifying PDF passwords or page ranges (see below).

### **input\_pw** [*handle=*]password ... | **PROMPT**

Input PDF owner passwords, if necessary, are associated with files by using their handles, e.g.

A=input1.pdf B=input2.pdf input\_pw A=foo B=bar

Without handles, passwords are associated with input files by order.

Most pdftk features require that encrypted input PDF are accompanied by the *owner* password. If the input PDF has no owner password, then the user password must be given, instead.

When running in **do\_ask** mode (see below), **pdftk** will prompt you for a password if the supplied password is incorrect or none was given. Use **PROMPT** to explicitly make **pdftk** interactively ask the user for a password.

## Operations

Available operations are: **cat**, **shuffle**, **burst**, **generate\_fdf**, **fill\_form**, **background**, **multibackground**, **stamp**, **multistamp**, **dump\_data**, **dump\_data\_utf8**, **dump\_data\_fields**, **dump\_data\_fields\_utf8**, **update\_info**, **update\_info\_utf8**, **attach\_files**, **unpack\_files**. Some operations take additional arguments, as described below.

When no operation is specified, then **pdftk** runs in “filter” mode. Filter mode takes only one PDF input and creates a new PDF after applying all of the output options, like encryption and compression.

### **cat** [*handle*][*begin*-[*end*[*qualifier*]]][*rotation*] ...

Concatenates pages from input PDFs to create a new PDF. Page order in the new PDF is specified by the order of the given page ranges.

The *handle* identifies one of the input PDF files. If the handle is omitted from the page range, then the pages are taken from the first input PDF. The *begin* and *end* page numbers are one-based references to pages in the corresponding input PDF. A single page can be referenced by specifying a single number as the *begin* argument and omitting the *end*. The keyword **end** may be used to reference the final page. The handle may be used alone to represent the entire document; e.g., “B” is the same as “B1-end”. The *end* specifier may be less than *begin*, e.g. **1-3** yields pages 1, 2 and 3, in that order, and **3-1** yields pages 3, 2 and 1, in that order. The *qualifier* can be **even** or **odd**. The **even** qualifier causes **pdftk** to use only the even-numbered

PDF pages, so **6-1even** yields pages 6, 4 and 2. The **odd** qualifier works similarly. The *rotation* can be **N**, **S**, **E**, **W**, **L**, **R** or **D**. The page rotation setting causes **pdftk** to rotate pages and documents as follows (in degrees): **N: 0**, **E: 90**, **S: 180**, **W: 270**, **L: -90**, **R: +90**, **D: +180**. The **L**, **R**, and **D** make relative adjustments to a page's rotation. If no arguments are passed to **cat**, then **pdftk** combines all input PDFs in the order they were given to create the output.

Examples without handles:

**6-1**            reverse pages in range  
**5 1 20**       take single pages  
**1-endE**       rotate entire document 90 degrees  
**5-25oddW**    take odd pages in range, rotate 90 degrees

Examples with handles; say *A=Pa in1.pdf* *B=Pa in2.pdf*, then:

**B**                use all of *in2.pdf*  
**AW**             rotate entire *in1.pdf* document 90 degrees  
**A72**            take a single page from *in1.pdf*  
**A A**            concatenate *in1.pdf* with *in1.pdf*  
**A1-21**          take a range from *in1.pdf*  
**AW BW BD**     concatenate rotated documents  
**Bend-1odd**     take all odd pages from *in2.pdf* in reverse order  
**A2-30evenL**   take even pages from range, rotate 90 degrees left  
**AevenW AoddE** rotate even pages of *in1.pdf* west and odd pages east  
**A1-21 Beven A72** assemble pages from both *in1.pdf* and *in2.pdf*

**shuffle** [*handle*][*begin*-[*end*[*qualifier*]]][*rotation*] ...

Collates pages from input PDFs to create a new PDF. Works like the **cat** operation except that it takes one page at a time from each page range to assemble the output PDF. If one range runs out of pages, it continues with the remaining ranges. Ranges can use all of the features described above for **cat**, like reverse page ranges, multiple ranges from a single PDF, and page rotation. This feature was designed to help collate PDF pages after scanning paper documents.

**burst**    Splits a single input PDF document into individual pages. Also creates a report named *doc\_data.txt* which is the same as the output from **dump\_data**. If the **output** section (see below) is omitted, then the individual pages are named as *pg\_%04d.pdf*, e.g. *pg\_0001.pdf*, *pg\_0002.pdf*, etc. To name these pages yourself, supply a printf(3) format string in the **output** section. For example, specify **output page\_%02d.pdf** if you want the pages to be named *page\_01.pdf*, *page\_02.pdf*, etc.

**generate\_fdf**

Reads a single input PDF file and generates an FDF file out of it, suitable for **fill\_form**. If no

output file name is specified, **pdftk** asks for one interactively.

**fill\_form** *FDF\_data\_filename* | *XFDF\_data\_filename* | - | **PROMPT**

Fills the single input PDF's form fields with the data from an FDF file, a XFDF file or stdin. Enter the data filename after *fill\_form*, or use - to pass the data via stdin; for example

```
pdftk form.pdf fill_form data.fdf output filled.pdf
```

After filling a form, the form fields remain interactive unless you also use the **flatten** output option, which merges the form fields with the PDF pages. You can use **flatten** alone, too, but only on a single PDF, e.g.

```
pdftk form.pdf fill_form data.fdf output out.pdf flatten
pdftk filled.pdf output out.pdf flatten.
```

If the input FDF file includes Rich Text formatted data in addition to plain text, then the Rich Text data is packed into the form fields *as well as* the plain text. **pdftk** also sets a flag that cues Acrobat/Reader to generate new field appearances based on the Rich Text data. That way, when the user opens the PDF, the viewer will create the Rich Text fields on the spot. If the user's PDF viewer does not support Rich Text, then the user will see the plain text data instead. If you flatten this form before Acrobat has a chance to create (and save) new field appearances, then the plain text field data is what you'll see.

**background** *background\_PDF\_filename* | - | **PROMPT**

Applies a PDF watermark to the background of a single input PDF. **pdftk** only uses the first page from the background PDF and applies it to every page of the input PDF. This page is scaled and rotated as needed to fit the input page. You can use - to pass a background PDF into **pdftk** via stdin. If the input PDF does not have a transparent background (such as a PDF created from page scans) then the resulting background won't be visible. Use the **stamp** operation instead.

**multibackground** *background* | - | **PROMPT**

Same as the **background** operation, but applies each page of the background PDF to the corresponding page of the input PDF. If the input PDF has more pages than the stamp PDF, then the final stamp page is repeated across these remaining pages in the input PDF.

**stamp** *stamp* | - | **PROMPT**

This behaves just like the **background** operation except it overlays the stamp PDF page *on top* of the input PDF document's pages. This works best if the stamp PDF page has a transparent background.

**multistamp** *stamp* | - | **PROMPT**

Same as the **stamp** operation, but applies each page of the background PDF to the corresponding page of the input PDF. If the input PDF has more pages than the stamp PDF, then the final stamp page is repeated across these remaining pages in the input PDF.

**dump\_data**

Reads a single input PDF file and reports various statistics, metadata, bookmarks (a.k.a outlines), and page labels to the given output filename or (if no output is given) to stdout. Non-ASCII characters are encoded as XML numerical entities. Does not create a new PDF file.

**dump\_data\_utf8**

Same as **dump\_data** except that the output is encoded as UTF-8.

**dump\_data\_fields**

Reads a single input PDF file and reports form field statistics to the given output filename or (if no output is given) to stdout. Non-ASCII characters are encoded as XML numerical entities. Does not create a new PDF file.

**dump\_data\_fields\_utf8**

Same as **dump\_data\_fields** except that the output is encoded as UTF-8.

**update\_info** *info* | - | **PROMPT**

Changes the metadata stored in a single PDF file's Info dictionary to match the input data file. The input data file uses the same syntax as the output from **dump\_data**. Non-ASCII characters should be encoded as XML numerical entities. This does not change the metadata stored in the PDF's XMP stream, if it has one. For example:

```
pdftk in.pdf update_info in.info output out.pdf
```

**update\_info\_utf8** *info* | - | **PROMPT**

Same as **update\_info** except that the input is encoded as UTF-8.

**attach\_files** *attachments* | **PROMPT** [*to\_page\_number* | **PROMPT**]

Packs arbitrary files into a PDF using PDF's file attachment features. More than one attachment may be listed after **attach\_files**. Attachments are added at the document level unless the optional **to\_page** option is given, in which case the files are attached to the given page number. The first page is 1, the final page is **end**. For example:

```
pdftk in.pdf attach_files tab.html img.jpg to_page 6 output out.pdf
```

**unpack\_files**

Copies all of the attachments from the input PDF into the current folder or to an output directory given in **output** For example:

```
pdftk report.pdf unpack_files output ~/atts/
```

**Output Options****output** *filename* | - | **PROMPT**

Sets the filename of the output PDF file. The output filename may not be the same as an input filename. Use - to output to stdout. When using the **dump\_data** operation, use **output** to set the name of the output data file. When using the **unpack\_files** operation, use **output** to set the name of an output directory. When using the **burst** operation, you can use **output** to control the resulting PDF page filenames (see above).

**encrypt\_40bit** | **encrypt\_128bit**

If an output PDF user or owner password is given, output PDF encryption strength defaults to 128 bits. This can be overridden by specifying **encrypt\_40bit**.

**allow** *permissions*

Permissions are applied to the output PDF file only if an encryption strength is specified or an owner or user password is given. If permissions are not specified, they default to “none”, which means all of the following features are disabled. The *permissions* section may include one or more of the following features: **Printing** (top quality printing), **DegradedPrinting** (lower quality printing), **ModifyContents** (also allows Assembly), **Assembly CopyContents** (also allows ScreenReaders), **ScreenReaders ModifyAnnotations** (also allows FillIn), **FillIn AllFeatures** (allows the user to perform all of the above, and top quality printing).

**owner\_pw** *password* | **PROMPT****user\_pw** *password* | **PROMPT**

If an encryption strength is given but no passwords are supplied, then the owner and user passwords remain empty, which means that the resulting PDF may be opened and its security parameters altered by anybody.

**compress** | **uncompress**

These are only useful when you want to edit PDF code in a text editor. Remove PDF page stream compression by applying the **uncompress** filter. Use the **compress** filter to restore compression.

**flatten** Use this option to merge an input PDF’s interactive form fields (and their data) with the PDF’s

pages. Only one input PDF may be given. Sometimes used with the **fill\_form** operation.

**keep\_first\_id | keep\_final\_id**

When combining pages from multiple PDFs, use one of these options to copy the document ID from either the first or final input document into the new output PDF. Otherwise **pdftk** creates a new document ID for the output PDF. When no operation is given, **pdftk** always uses the ID from the (single) input PDF.

**drop\_xfa**

If the input PDF is a form created using Acrobat 7 or Adobe Designer, then it probably has XFA data. Filling such a form using **pdftk** yields a PDF with data that fails to display in Acrobat 7 (and 6?). The workaround is to remove the form's XFA data, either before you fill the form using **pdftk** or at the time you fill the form. Using this option causes **pdftk** to omit the XFA data from the output PDF form.

This option is only useful when running **pdftk** on a single input PDF. When assembling a PDF from multiple inputs, any XFA data in the input is automatically omitted.

**dont\_ask | do\_ask**

Depending on compile-time settings, **pdftk** might prompt you for further input when it encounters a problem, such as a bad password. Override this default behavior by adding **dont\_ask** or **do\_ask**. When running in **dont\_ask** mode, **pdftk** will overwrite output files without notice.

**verbose**

By default, **pdftk** runs quietly. Append **verbose** to the end and it will speak up.

**EXAMPLES**

Collate scanned pages

```
pdftk A=even.pdf B=odd.pdf shuffle A B output collated.pdf
```

or if *odd.pdf* is in reverse order:

```
pdftk A=even.pdf B=odd.pdf shuffle A Bend-1 output collated.pdf
```

Decrypt a PDF:

```
pdftk secured.pdf input_pw foopass output unsecured.pdf
```

Encrypt a PDF using 128-bit strength, withhold all permissions:

```
pdftk in.pdf output out.pdf owner_pw foopass
```

Same as above, except a user password is also used in the output PDF:

```
pdftk in.pdf output out.pdf owner_pw foo user_pw baz
```

Same as above, except printing is allowed (once the PDF is open):

```
pdftk in.pdf output out.pdf owner_pw foo user_pw baz allow Printing
```

Join *in1.pdf* and *in2.pdf* into *out.pdf*:

```
pdftk in1.pdf in2.pdf cat output out.pdf
```

or (using handles):

```
pdftk A=in1.pdf B=in2.pdf cat A B output out.pdf
```

or (using wildcards):

```
pdftk in?.pdf cat output out.pdf
```

Remove page 13 from *in.pdf* to create *out.pdf*:

```
pdftk in.pdf cat 1-12 14-end output out.pdf
```

Apply 40-bit encryption to output, revoking all permissions (the default). Set the owner password to “foo”:

```
pdftk 1.pdf 2.pdf cat output out.pdf encrypt_40bit owner_pw foo
```

Join two files, one of which requires the password “foopass”. The output is not encrypted.

```
pdftk A=secured.pdf other.pdf input_pw A=foopass cat output out.pdf
```

Uncompress PDF page streams for editing the PDF in a text editor:

```
pdftk doc.pdf output uncompressed.pdf uncompress
```

Repair a PDF’s corrupted XREF table and stream lengths, if possible:

```
pdf broken.pdf output fixed.pdf
```

Burst a single PDF document into pages and dump its data to *doc\_data.txt*:

```
pdftk in.pdf burst
```

Burst a single PDF document into encrypted pages. Allow low-quality printing:

```
pdftk in.pdf burst owner_pw foopass allow DegradedPrinting
```

Write a report on PDF document metadata and bookmarks to *report.txt*:

```
pdftk in.pdf dump_data output report.txt
```

Rotate the first PDF page 90 degrees clockwise

```
pdftk in.pdf cat 1E 2-end output out.pdf
```



Rotate an entire PDF document 180 degrees

```
pdftk in.pdf cat 1-endS output out.pdf
```

## SEE ALSO

*PDFtk's homepage*: <http://www.pdftk.com>

Sid Steward, *PDF Hacks*, O'Reilly, 2004.

## AUTHOR

Sid Steward <[sid.steward@pdflabs.com](mailto:sid.steward@pdflabs.com)> maintains **pdftk**. Please email him with questions or bug reports. Include pdftk in the subject line to ensure successful delivery. Thank you.