



PDF Form Filling Tool

Version 3.6

User Manual

Contact: pdfsupport@pdf-tools.com

Owner: **PDF Tools AG**
Kasernenstrasse 1
8184 Bachenbülach
Switzerland
www.pdf-tools.com

Copyright © 2000-2012

January 10, 2012

Table of Contents

1	Introduction	3
2	Parameters	4
2.1	General Options (All Editions).....	4
2.2	Security Related Options (All Editions).....	4
2.3	Pass Form Data (All Editions)	4
2.4	Add a Text Field (All Editions).....	4
2.5	Delete a Form Field (All Editions)	5
2.6	Use a Control File (FF).....	5
2.7	Flatten the Form Fields (FF)	5
3	Examples	6
3.1	List the Fields of a PDF File (Option -l)	6
3.2	Set some Field Values	6
3.3	Add a New Form Field.....	6
3.4	Delete a Form Field	6
3.5	Read Encrypted Input Document (Option -pw)	6
3.6	Encryption Output File (Options -user -owner -perm).....	6
3.7	Read Field Assignments from a Control File	7
	Example 1	7
	Example 2	7
3.8	Importing Data from an Excel Table	8
3.9	Set Font Size Limit for Autosize	9
	Example	9
3.10	Use a Font Template to Add Form Fields with a Barcode	10
4	List of Return Codes.....	11
	Appendix A: Security	12

1 Introduction

The PDF Form Filling and Flattening Tool is a command line tool that can create, edit, fill in, and delete form fields in a PDF document.

The creation and editing of form fields is managed through a control file. The control file can also be used to indicate what content should be entered in each of the form fields.

The following functionality is available in the PDF Form Filling Tool PDF Form Filling and Flattening Tool:

- List all form fields in a PDF document
- Fill-in the form fields with text content
- Add a new form field
- Delete a form field
- Pass the form commands in a control file
- Flatten the form fields (convert form field entries to standard text content)
- Decrypt the input PDF files the command line
- Specify user and owner passwords to encrypt the output PDF files and set access permissions

The PDF Form Filling and Flattening Tool is provided in the following editions:

- Standard (as part of the Command Line Suite CLS): Listing and filling of forms with command line data. This executable is named *pdform*.
- Professional (FF): As above, plus input from control file, plus flattening of text fields. This executable is named *pdformp*.

2 Parameters

PDFFORM is controlled via command line parameters. The general syntax is:

```
pdform [options] input-file output-file [data-assignments]
```

2.1 General Options (All Editions)

-l	list fields of input file
----	---------------------------

2.2 Security Related Options (All Editions)

(Note that in the standard version of 2.x, security is not supported. In version 3.0 and above, security is supported in both editions)

-pw thepassword	Specify password of input (form) PDF
-user theuserpw	The user password to be set in the output PDF
-owner theownerpw	The owner password of the output PDF
-permit psca	The protection flags (p: do not allow printing; s: do not allow selection of text or graphics; c: do not allow changes; a: do not allow changes of annotations)

2.3 Pass Form Data (All Editions)

Form data is passed on the command line using the following syntax:

```
FIELDNAME=Fielddata
```

These name/data pairs must represent a single command line argument. If name or data contain blank spaces or other special characters, they must be enclosed in quotes. Note that this syntax actually depends on the shell program you are using. CMD.EXE (the standard NT command shell) is fairly limited. UNIX command shells are much more flexible. If your platform is Win32, you may want to use an alternate shell (e.g. MKS Korn Shell, or use the Win32 API from a VB or C program).

2.4 Add a Text Field (All Editions)

PDFFORM is capable to dynamically add a new text field. The command syntax is:

```
+FIELDNAME@1,2,3,4,5,6,7=value
```

The numeric parameters 1 through 7 represent

- 1: the page number, on which the field is to be created
- 2: the (left) x coordinate of the field
- 3: the (bottom) y coordinate of the field
- 4: the right x coordinate of the field

January 10, 2012

- 5: the top (y) coordinate of the field
6. the font number (0: Helvetica; 1: Helvetica Bold; Helvetica Oblique; Helvetica Bold Oblique; Courier, Courier Bold, Courier Oblique, Courier Bold Oblique, Times Roman; Times Bold; Times Italic; Times Bold Italic; Zapf Dingbats; Symbol)
7. the font size in points

2.5 Delete a Form Field (All Editions)

To delete a form field, specify a 'minus' (-) in front of the field name:

-FIELDNAME

2.6 Use a Control File (FF)

To redirect input to a file, use the '@' sign; e. g.

@control.txt

Each line of the control file represents a field command as described above. To specify an explicit line break for a multi-line text field, put a backslash character as last character on the line. The following line will then be interpreted as continued field data, and a line break is inserted in place of the backslash.

2.7 Flatten the Form Fields (FF)

A form field is an annotation and not part of the actual page. The option `-flatten` deletes the form fields and writes the corresponding content of the field onto the page content. This process is irreversible.

3 Examples

3.1 List the Fields of a PDF File (Option -l)

List all form fields in the document.

```
pdform -l input.pdf
```

3.2 Set some Field Values

Set new content for the fields F1 and F2.

```
pdform input.pdf output.pdf F1="Data of field F1" F2="field 2 data"
```

3.3 Add a New Form Field

Add the new form field N at page 1, at position 20 points from the left border and 600 points from the bottom, a width of 120 points and a height of 12.5 points. The font is Courier, size is 10.5.

```
pdform i.pdf o.pdf +N@1,20,600,120,612.5,4,10.5="Data of new form field"
```

3.4 Delete a Form Field

Delete the field F1.

```
pdform input.pdf output.pdf -F1
```

3.5 Read Encrypted Input Document (Option -pw)

Specify either the user or the owner password to read an encrypted PDF input document.

```
pdform -pw password input.pdf output.pdf
```

3.6 Encryption Output File (Options -user -owner -perm)

Specify a user and owner password and access permissions to encrypt the output files. The following command sets the user password "upw" and the owner password "opw".

```
pdform -user upw -owner opw -permit psca input.pdf output.pdf
```

Password and permission flags can also be passed in the control file.

For the permissions flags see Appendix A: Security

3.7 Read Field Assignments from a Control File

(Professional edition only)

Data can be read from a text control file. This is specially useful for PDF documents with many form fields. The syntax to use a control file is as following:

```
pdformp input.pdf output.pdf @control.txt
```

Each line in the control file corresponds to one parameter.

Example 1

Assume the PDF document *input.pdf* has the following two form fields:

```
pdformp -l input.pdf
field1 1 [133,664/276,30] : field 1
field2 1 [133,620/276,30] : field 2
```

Create a text field (*control.txt*) with the following content.

```
field1=Hello World
field2=Bye Bye
```

Note that field contents containing blanks do not need to be written in "brackets" when using a control file.

Apply the control file to fill the fields with new values:

```
pdformp input.pdf output.pdf @control.txt
```

Example 2

```
# sample control file (comment sign is # at beginning of line)
# each line contains a field command (data assignment,
# add or delete). Recursion is not implemented (use of @ sign).
#
# This first assignment sets the data of a multi-line text field
# with an explicit line break between street and town
address=10 Downing Street\
London
#
# This next statement removes the field F
-F
# etc.
```

3.8 Importing Data from an Excel Table

It is possible to use Excel to create a control file that can be used as input for the PDF Form Filling Tool. To create an Excel table, please take a look at the following example:

Each line starts with an operator, followed by a series of operands. The number of operands depends on the operator. A line that starts with a # is a comment.

The operators are:

/ copies the selected page range (non-zero based).

+ adds a new form field.

= fills data into a form field.

A table as shown below can be created to do form field processing.

	A	B	C	D	E	F	G	H	I	J
1	# Operator /	From Page (Default: 1)	To Page (Default: Last)							
2	# Operator +	Name	Page Number	Left X	Bottom Y	Right X	Top Y	Font Number	Font Size	Value
3	# Operator =	Name	Value							
4	/	2	2							
			pw password for input file							
			user set user password of output file							
			owner set owner password of output file							
			perm set security flags for output file							
5	=	F1	l list fields of input file							
6	=	F2	data in field with UNICODE name							
7	+	F3	1	20	800	120	820	0	14	v3
8	/	1	1							
9	=	F1	data for page 2							
10	/	1	1							
11	/	3	3							
12										
13										

An MS-Excel table needs to be saved as either CSV (*.csv) or as Unicode text (*.txt). These two formats can be interpreted by the PDF Form Filling Tool.

(Note, the following control files cannot be used directly, i.e. copied from the PDF and pasted to create a control file. Instead, create the files from a table using the "Save As" function.)

A CSV-saved document looks like this:

```
# Operator /;From Page (Default: 1);To Page (Default: Last);;;;;;
# Operator +;Name;Page Number;Left X;Bottom Y;Right X;Top Y;Font
Number;Font Size;Value
# Operator =;Name;Value;;;;;
/;2;2;;;;;
=;F1;"pw          password for input file
user          set user password of output file
owner        set owner password of output file
perm         set security flags for output file
l            list fields of input file";;;;;;
=;F2;data in field with UNICODE name;;;;;
+;F3;1;20;800;120;820;0;14;v3
/;1;1;;;;;
```

January 10, 2012

```
=;F1;data for page 2;;;;;;;;;
/;1;1;;;;;;;;;
/;3;3;;;;;;;;;
```

The Unicode text file like this:

```
# Operator /;From Page (Default: 1);To Page (Default: Last);;;;;;;;;;
# Operator +;Name;Page Number;Left X;Bottom Y;Right X;Top Y;Font
Number;Font Size;Value
# Operator =;Name;Value;;;;;;;;;
/ 2 2
=F1 "pw          password for input file
user          ""set user password of output file""
owner         set owner password of output file
perm          set security flags for output file
l             list fields of input file"
= abc
+ F31 20 30 12045 0 14 v3
- T
/ 1 1
- F3
=F1 "this is the data for the second page"
= more data
/ 1 1
/ 3 3
```

If the CSV file is called *form-f3.pdf* and the Unicode file *form-f4-uni.pdf*, then they could be processed with the following commands:

```
pdformp -flatten form.pdf output-csv.pdf :form-f3.csv
```

```
pdformp -flatten form.pdf output-uni.pdf @form-f4-uni.txt
```

3.9 Set Font Size Limit for Autosize

Use this switch to limit the minimum and maximum font size for forum fields that use the font size "autosize".

Example

The following settings applies a minimum font size of 8 points and a maximum font size of 16 points for autosize form fields:

```
pdformp -fs 8 16 input.pdf output.pdf
```

3.10 Use a Font Template to Add Form Fields with a Barcode

```
# How to use a font template to add a form field with a bar code
# This file must be in UNICODE text format.
# 1) prepare the font template
# fnt2pdf -c winansi -e barcode.ttf bar.pdf
# 2) use this control file with pdformp.exe
++ bar.pdf
+bar 1 10 10 110 30 NameOfBarCodeFont 12
```

4 List of Return Codes

Below, all return codes are listed. Note that on some (Unix-) systems, the codes are returned modulo 256.

0		Success
2		the annotations cannot be copied
22		the file is encrypted, password incorrect
1001	NotPDF	the file does not start with %PDF
1002	Trailer	the trailer of the PDF file could not be found
1003	Xref	the XRef table could not be found as defined in trailer these two errors indicate that the PDF file has been corrupted as sometimes happens when copied in ASCII mode by FTP
1004	NullRef	an object reference could not be resolved (object missing in file)
1005	BadParamValue	an illegal parameter value was specified in a method
1006	ObjRead	a particular PDF object could not be read from the file
1007	AlreadyWritten	a particular PDF object was attempted to write twice
1008	BadCallSequence	a particular function was called in an inappropriate context
1009	Internal	an unexpected situation was encountered that could not be handled
1010	UnexpectedVal	an unexpected value was encountered in a PDF object
1011	IO	an input/output error was encountered
1012	InvalidHandle	the handle specified is not valid
1013	Duplicate	an attempt to create a duplicate object is made
1014	IllegalFont	an invalid font name was specified
1015	NoSuchPage	an invalid page number was specified
1016	NotFound	requested information not found for specified criteria
1017	Failed	generic error
1018	Encrypted	input file is encrypted (password protected)
1019	InvalidPassword	the password supplied is not correct

Appendix A: Security

Encrypting a PDF file is only useful in combination with security flags. These flags define what actions the user may or may not perform. Here is a list of the permission flags:

"p": do not print the document from Acrobat

"c": changing the document is denied in Acrobat

"s": selection and copying of text and graphics is denied

"a": adding or changing annotations or form fields is denied

The following flags are defined for 128 bit encryption (PDF 1.4, Acrobat 5.0):

"i": disable editing of form fields

"e": disable extraction of text and graphics

"d": disable document assembly

"q": disable high quality printing

The flag "5" can be used in combination with one of the "old" flags to force 128 bit encryption without setting any of the i, e, d, or q flags. Note that using any of these Acrobat 5 related flags will produce a file that cannot be opened with older versions of Acrobat.

To set permission flags an owner password must be set. The owner password is required to change the permission flags or change a password.