



# PDF Form Filling Tool

(Executable for Win32 and UNIX Systems)

## User's Manual

### Version 3.0

---

Contact: [pdfsupport@pdf-tools.com](mailto:pdfsupport@pdf-tools.com)

Owner: **PDF Tools AG**  
Geerenstrasse 33  
CH-8185 Winkel  
Switzerland  
[www.pdf-tools.com](http://www.pdf-tools.com)

## Table of Contents

<b>1</b>	<b>Introduction .....</b>	<b>3</b>
<b>2</b>	<b>Parameters.....</b>	<b>3</b>
2.1	General options (all editions) .....	3
2.2	Security related options (all editions).....	3
2.3	Pass form data (all editions) .....	3
2.4	Add a text field (all editions).....	4
2.5	Delete a form field (all editions) .....	4
2.6	Use a control file (FF) .....	4
<b>3</b>	<b>Examples.....</b>	<b>5</b>
3.1	List the fields of a PDF file (Option -l) .....	5
3.2	Set some field values .....	5
3.3	Add a new form field .....	5
3.4	Delete a form field .....	5
3.5	Read encrypted input document (Option -pw).....	5
3.6	Encryption output file (Options -user -owner -perm).....	5
3.7	Read field assignments from a control file.....	5
3.8	Sample control file .....	6
<b>4</b>	<b>List of return codes .....</b>	<b>7</b>
	<b>Appendix A: Security .....</b>	<b>8</b>

## 1 Introduction

PDFFORM is a command line tool, that can process PDF documents with form fields:

- Listing of form fields
- Filling in data
- Flattening of text fields (Professional Edition only)

The PDFFORM 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
- Professional (FF): As above, plus input from control file, plus flattening of text fields

## 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
- 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.dat
```

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.

## 3 Examples

### 3.1 *List the fields of a PDF file (Option -l)*

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

### 3.2 *Set some field values*

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

### 3.3 *Add a new form field*

```
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*

```
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.

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

For the permissions flags see Appendix A: Security

### 3.7 *Read field assignments from a control file*

(Professional edition only)

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

### **3.8    *Sample control file***

```
# 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.
```

## 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.