



# PDCat COM DLL

(for Win32)

## 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 Overview .....</b>	<b>3</b>
<b>2 Introduction to pdcat .....</b>	<b>4</b>
2.1 Parameters .....	4
2.1.1 <i>General options</i> .....	4
2.1.2 <i>Input file related options</i> .....	5
2.2 Error Handling .....	7
2.2.1 <i>List of all warnings and error messages</i> .....	7
2.2.2 <i>List of return codes</i> .....	8
2.3 Examples .....	9
2.3.1 <i>Add a background logo to a PDF file</i> .....	9
2.3.2 <i>Concatenate two PDF files and merge hyperlinks from existing file</i> ....	9
2.3.3 <i>Add a link to acrobat.pdf that points to the first page of license.pdf</i>	10
2.3.4 <i>Extract the second and third page of acrobat.pdf</i> .....	10
<b>3 Introduction to pdsplit .....</b>	<b>11</b>
3.1 Parameters .....	11
3.1.1 <i>Options</i> .....	11
3.1.2 <i>Link Format</i> .....	11
3.1.3 <i>Output File Format</i> .....	12
3.2 Error Handling .....	12
3.3 Examples .....	12
<b>4 The COM DLL Interface .....</b>	<b>14</b>

## 1 Overview

The PDCat COM DLL is a wrapper to the two command line tools pdcat and pdsplit, which are part of the PDF Command Line Tool Suite.

With PDCAT COM DLL you can:

- Merge the pages of PDF files into one large file
- Create bookmarks for each input file
- Add named destinations for each page
- Set the document information (author etc.)
- Set the open action
- Rotate pages
- Copy a specified range of pages
- Add link annotations from a text file
- Add logos and watermarks from a PDF document in the foreground or background
- Delete text annotations
- Substitute text strings in action links
- Clip and Crop pages
- Split a PDF document into parts with a specified number of pages
- Extract all or individual chapters according to bookmarks
- Restrict the processing to a specific bookmark level

## 2 Introduction to pdcat

pdcat is a command line tool for concatenating PDF files. It also features a variety of modification or enhancement functions like adding a background or logo to each page of an input file, adding hyperlinks and bookmarks, and so on.

### 2.1 Parameters

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

```
pdcat { general option } { [ input file related options ] input-file } output-file
```

#### 2.1.1 General options

<b>-r</b>	Overwrite output file.
<b>-a filename</b>	Merge with link annotations from specified file.
<b>-attach index.pdx</b>	Attach the specified search index.
<b>-b</b>	Do not generate any bookmarks.
<b>-b1</b>	Do not generate the "Page" bookmarks.
<b>-c "New-Title"</b>	put the specified text into the Title attribute of the resulting PDF file
<b>-c @fn</b>	Read document attributes from the specified file. If <fn> is omitted, standard input is read. Other attributes than "Title" can be set by specifying the key including the leading slash on a separate line, followed by a "=" sign and the value for this key (e. g. /Author=William Shakespeare).
<b>-sub "New-Subject"</b>	Put the specified text into the Subject attribute of the resulting PDF file.
<b>-co</b>	Copy the OpenAction entry from the input file's Catalog.
<b>-oT</b>	Set the page mode to "thumb nails".
<b>-oO</b>	Set the page mode to "outlines".
<b>-oP</b>	Set the page mode to "page only".
<b>-pf</b>	Preserve font names (when merging subsetted fonts from different files; this is an issue with pre-4.05 viewers that will not display text correctly).
<b>-l filename</b>	Add link annotations as defined in the specified file.
<b>-bs:1</b>	Set the border style of hyperlinks added via "-l"

<b>-annot</b>	Do not copy any text annotations to output.
<b>-scale 1.0</b>	Scale (reduce/enlarge) each page by the specified factor.
<b>-logo filename</b>	Use the content of the first page of the specified PDF file as a logo that is put on (top of) each page of the generated file.
<b>-bglogo filename</b>	Same as <b>-logo</b> , but the logo is put below the page content that is merged from the input files. Please note that non-transparent pages in input files will hide the logo (or a part of it).
<b>-bglogo +"&lt;matrix&gt;" fn</b>	The additional +" <b>&lt;matrix&gt;</b> " parameter results in the application of a "cm" operator to the logo. The logo can thus be scaled, shifted and/or rotated. The matrix must consist of 6 numbers. Please refer to the Adobe PDF Specification document for details.  Please note that the matrix parameter must constitute a single command line parameter. When using a UNIX shell, you can use single or double quotes to achieve this. The Windows command shell only accepts double quotes.
<b>-bglogo:1-3,5 filename</b>	Put the logo only on selected pages. The specification of these pages is a comma separated list of numbers or number ranges. It can be combined with the matrix feature above, and is also valid for the <b>-logo</b> option.
<b>-@controlfile.txt</b>	Expands the content of the control file. Every argument in the control file needs a separate line, except for options with value, these stand on the same line.
<b>-s string1 string2</b>	Substitute occurrences of string "string1" by "string2" in file- and web- links. This option can only be set once.
<b>-sp string1 string2</b>	Substitute occurrences of partial string "string1" by "string2" in file- and web- links. This option can be set multiple times. The first match is applied.

## 2.1.2 Input file related options

<b>-mPre</b>	Add a named destination mark for each page of the next input file(s) using the specified prefix (to avoid name clashes when combining several input files).
<b>-m</b>	Do not add a named destination mark for each page using a prefix; use this option to reset <b>-mPre</b> from a previous input file.
<b>-n</b>	Do not copy bookmarks of next input file.

<b>-i "Text"</b>	Use the specified text as the title for the bookmark generated for the next input file, if this input file does not contain a Title attribute.
<b>-I "Text"</b>	Always use the specified text as the title for the bookmark generated for the next input file (even if it contains a Title attribute).
<b>+I</b>	Add the bookmarks of the next file at level 0.
<b>-R 90</b>	Set the page rotate attribute of the pages of the next file (or files) to the specified value (0, 90, 180 or 270 are valid arguments). Any existing individual Rotate values are replaced.
<b>-R -90</b>	Decrease the Rotate attribute of the pages of the next file by the specified value. To increase the value, specify the corresponding value - 360 degrees (e. g. -270 for +90).
<b>-p:2-5</b>	Restrict the number of pages to be copied from the next input file to the specified range of pages.

pdcat prints warnings and errors to standard error.

The following conditions are errors which are signaled by a non-zero exit code:

**"output file exists", exit code = 1007**

This error occurs when the option "-r" has not been specified, and pdcat would have to overwrite an existing file

**"cannot create <output-file>", exit code = 1007**

**"error closing output, error = %errno", exit code = 1007**

The following condition results in a warning message which is printed to standard error:

**"<input> not found, skipped" (no program exit)**

Other warnings can be signaled which are due to problems with PDF input. Errors that are detected during opening the input file will result in the "<input> not found, skipped" warning (e.g. if the file is not a PDF file). Errors which occur during copying result in a warning that is written to standard error. This class of errors is due to problems encountered in the structure of the PDF file. The resulting file may or may not contain pages from this input file.

## **2.2 Error Handling**

### **2.2.1 List of all warnings and error messages**

#### **2.2.1.1 Warnings:**

**Illegal syntax <string>**

Syntax of input file is not correct

**Bookmark skipped, missing parent level <string>**

The bookmark level of the parent is unknown.

**Illegal bookmark syntax <string>**

The bookmark syntax in the control file is incorrect.

**page number reference lost, id = <string>**

**page reference lost, id = <string>**

**page number ref lost, id = <string>**

**ConvertNames - error 1**

**input file already has annotations**

The option -a is used, but the file already has annotations.

**Cannot open <string>**

The input file cannot be opened.

The logo file cannot be opened.

**Invalid password for <string>**

The password to open the logo file is incorrect.

**array or reference expected for annotation**

While processing annotations, an array or reference was expected.

**Cannot open control file <string>**

The control file using the option -@<filename> cannot be opened

**option <string> not recognized**

The option is not recognized.

**<string> is encrypted, password incorrect - skipped.**

The input file is encrypted and the password is incorrect.

The file is skipped in the processing.

**Error copying file, error=<number>**

**<string> not found, skipped.**

The input file does not exist or the PDF file does not start with %PDF.

The option -e is not used.

**option -c not compatible with -t, ignored**

Collision: Both commands set the info text for the output file.

**cannot open file <string>**

### 2.2.1.2 Abort Messages:

**cannot open <string> for input**

input file cannot be opened

**output file exist**

The output file exists and the option -r is not selected.

**cannot create <string>**

The output file cannot be created.

**<string> is encrypted, password incorrect - abort.**

The input file is encrypted and the password is incorrect.

The processing is aborted because the option -e is set.

**<string> not found, abort.**

The input file does not exist or the PDF file does not start with %PDF.

The processing is aborted because the option -e is set.

**Error closing output, error=<number>**

The output file cannot be closed.

### 2.2.2 List of return codes

Below, all return codes are listed. Note that on some (Unix-) systems, the codes are returned modulo 256. Using the option -e, pdcat will break on the first non-zero return code, otherwise it will proceed on warnings and the final return code may be zero, even though there was a processing problem.

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

## 2.3 Examples

### 2.3.1 Add a background logo to a PDF file

```
C:\> pdcat -r -b -bglogo +"1 0 0 1 5 8" background.pdf input.pdf output.pdf
```

This command will put the first page of background.pdf as a watermark behind the text (or other content) of each page of input.pdf and produce output.pdf.

The option `-r` will avoid that pdcat prints an error if a file "output.pdf" already exists.

The option `-b` will avoid that bookmarks will be added.

The matrix "1 0 0 1 5 8" will result in a shift of 5 points to the right and 8 points up when placing the logo.

To do this only on page 2, the command line is

```
C:\> pdcat -r -b -bglogo:2 +"1 0 0 1 5 8" background.pdf input.pdf output.pdf
```

### 2.3.2 Concatenate two PDF files and merge hyperlinks from existing file

(Replace existing output file, do not copy book marks. Input files may not contain any annotations or hyperlinks yet. If they do, produce a link definition file and use option `-l`)

```
C:\> pdcat -r -b -a linksrc.pdf input.pdf output.pdf
```

**input.pdf..**

### **2.3.3 Add a link to acrobat.pdf that points to the first page of license.pdf**

```
C:\> echo Page 1 > link.tmp
C:\> echo 125 447 285 598 "license.pdf" 1 >> link.tmp
C:\> echo .bs 22900 >> link.tmp
C:\> echo 400 447 285 200 "http://www.adobe.com" >> link.tmp
C:\> echo Bookmarks >> link.tmp
C:\> echo 0 Adobe "http://www.adobe.com" >> link.tmp
C:\> echo 1 "Get Reader" "http://www.adobe.com/download" >> link.tmp
C:\> pdcat -b -l link.tmp -box:1 acrobat.pdf output.pdf
acrobat.pdf..
```

If input files contain named destinations, they will be transferred to the output file. Note that there can be name conflicts. In that case, the first destination will be active.

If input files contain forms, they will be transferred to the output file. However, the forms dictionary is not constructed in the output file. It is possible that fields with identical names are merged into one output file. The appearance of these fields will remain unchanged, but if you load the resulting file into Acrobat and enter data into one of these fields, Acrobat will apply your entry to all fields with that name.

See in the appendix for sample link files.

### **2.3.4 Extract the second and third page of acrobat.pdf**

```
C:\> pdcat -b -t= -p:2-3 acrobat.pdf output.pdf
acrobat.pdf..
```

(-b: no extra bookmarks, -t: copy info object as is, -p: only specified range of pages)

Note that pdcat is not efficient to extract a few pages from a large document, as it is optimized to concatenate files. You should use "pdselect" for this.

For further examples see the document:

<http://www.pdf-tools.com/public/downloads/manuals/cle.pdf>

### 3 Introduction to pdsplit

PDSPLIT is a tool to split multi-page PDF files into smaller PDF files – either page by page, or based on the bookmarks contained in the PDF file.

Cross page navigation links are converted to cross file links. This can be used for example to implement a page view counter in a web application by formatting the cross page links into suitable servlet requests.

#### 3.1 Parameters

pdsplit obtains parameters from the command line.

The general syntax is:

```
pdsplit { general option } input-file [ link-format [ out-filename-format ] ]
```

##### 3.1.1 Options

<b>-v</b>	Verbose mode
<b>-b</b>	Extract pages according to the bookmarks of the file; the output files will be named according to the bookmark titles
<b>-b:title=filename</b>	Extract a specific chapter according to the specified bookmark title and store it in the given file. The option can be used for several title/filename pairs – use the colon (":") to separate (e. g. -b: title1=1.pdf:title2=2.pdf)

##### 3.1.2 Link Format

The parameter following the input file is a formatting specification that defines how cross page links shall be translated into cross file links.

The default value for this depends on the "-b" setting. If -b is not set, the file is split page by page, and output file references will be generated according to the default format "xx%2.2d.pdf". This is a standard C sprintf format specification. Note that you can only use number formats, because the argument passed is the page number.

When -b is turned on, output file references will be created according to the format "xx-%s.pdf"; here, the bookmark title is passed as an argument to sprintf.

The "%s" place holder in the format string will be expanded to the bookmark title. It is also possible to specify a %d (or e.g. %3.3d) format. In this case, a sequence number is placed instead. Note that "%s" must occur unaltered – it is not possible to specify e. g. "%3s".

### 3.1.3 Output File Format

The output file format is by default the same as the reference format. The same rules apply as outlined above.

## 3.2 Error Handling

The command line version of pdsplit has the following return codes:

- 0 normal successful completion
- 1 missing arguments
- 2 output file(s) cannot be created
- 3 no files selected (-b: option)
- 4 no bookmark titles in file (option -b)
- 5 other errors (e. g. unreadable input file, ...)

The COM interface returns these errors as:

pdSuccess  
pdErrCannotCreateOutput  
pdErrNoFilesSelected  
pdErrNoBookmarks

Other errors related to processing of the input file are reported as with PDCAT.

## 3.3 Examples

The pdsplit program splits large PDF files into smaller pieces – either one page files, or according to the bookmarks stored in the file. This is controlled by the option “-b” (meaning do split according to bookmarks).

The program can also selectively extract a particular chapter, if the corresponding bookmark title is specified (e. g. -b: 'Chapter one'=one.pdf).

This command will create files xx01.pdf, xx02.pdf, etc:

**pdsplit big.pdf**

When you specify a second argument, the links and file names will be generated according to given the format, in this case small001.pdf, small002.pdf, etc.

```
pdsplit big.pdf small%3.3d.pdf
```

Note that you must specify a “printf” conforming format. The argument supplied to the format is always an integer number (the page number). Therefore, you can use ‘d’ (for decimal), ‘x’ (for hexadecimal), ‘u’ (unsigned decimal), or ‘o’ (octal). If you specify ‘s’ (char string), pdsplit will crash, because the page number will not be a valid string address.

By using pdsplit as shown below, you can change the links between the resulting PDF files to web links that invoke a servlet method on the web server, which can do some on the fly processing while serving the desired PDF page.

```
pdsplit big.pdf http://myweb/servlet/big?page=%d big-%d.pdf
```

It is also possible to change links to relative web links:

```
pdsplit big.pdf http:big?page=%d big-%d.pdf
```

```
pdsplit big.pdf http:/servlet/big?page=%d big-%d.pdf
```

```
pdsplit -b big.pdf chapter-%s.pdf
```

If the file “big.pdf” contains a bookmark tree pointing to the individual chapters of the file, pdsplit will extract all sections into separate files. The name of the files being created will carry the prefix “chapter-” followed by the bookmark text, and have the extension “.pdf” in the example above.

For further examples see the document:

<http://www.pdf-tools.com/public/downloads/manuals/cls.pdf>

## 4 The COM DLL Interface

Before the COM DLL can be used, it must be registered using REGSVR32: copy the DLL into the installation directory (where you would like it to reside when being used, e. g. C:\WINNT). In the command shell, call REGSVR32.EXE and specify the name of the DLL as a parameter.

The COM interface works very much like the command line interface, except that the parameter list must be constructed via the API. It is not possible to pass it as one single string. The COM API looks thus as follows:

```
Dim obj As New PDAPPLib.PDCat  
obj.AddParameter "-r"  
obj.AddParameter ...  
obj.Execute
```

Note that you must add the PDAPPLib type library to the references of the Visual Basic project to make this work. Alternatively, you can use the CreateObject method:

```
Dim obj As Object  
Set obj = CreateObject("PDApp.PDCat")  
....
```

For integration into COM based applications on Windows, a special DLL containing the functionality of the PDCAT executable is available. This is the pdapp.dll, which is a separate product and not included in the Command Line Suite.

Declaration:

```
Dim obj As New PDAPPLib.PDCat
```

Parameter passing:

```
obj.AddParameter "-r"  
obj.AddParameter "input.pdf"  
...  
obj.AddParameter "output.pdf"
```

The AddParameter method is used to compose the arguments just as they would be specified on the command line for the PDCAT executable (but without the restriction of the maximum length imposed by the command shell).

Execution (concatenation of the input files into the output file):

```
Dim res As PDErrorType, osErr As Long  
res = obj.Execute()  
If res = pdOSError Then osErr = obj.GetErrorCode
```

The Execute method returns a result code as defined in the IDL file (and the PDApp.h file).

If an error is returned from the operating system, the corresponding error number can be retrieved using the GetErrorCode method. This typically happens when there is a problem with file input/output.

The pdsplit class works in the same way.