

The pdfcol package

Heiko Oberdiek
<heiko.oberdiek at gmail.com>

2007/12/12 v1.2

Abstract

Since version 1.40 pdfTeX supports color stacks. The driver file `pdfTeX.def` for package `color` defines and uses a main color stack since version v0.04b. Package `pdfcol` is intended for package writers. It defines macros for setting and maintaining new color stacks.

Contents

1	Documentation	2
1.1	Requirements	2
1.2	Interface	2
2	Implementation	3
2.1	Reload check and package identification	3
2.2	Catcodes	4
2.3	Check requirements	5
2.3.1	Check package <code>luacolor</code>	5
2.3.2	Check PDF mode	5
2.3.3	Check version of pdfTeX	6
2.3.4	Check <code>pdfTeX.def</code>	6
2.4	Enabled interface macros	6
2.5	Disabled interface macros	8
3	Test	9
3.1	Catcode checks for loading	9
3.2	Very simple test	10
3.3	Detection of package <code>luacolor</code>	11
4	Installation	11
4.1	Download	11
4.2	Bundle installation	12
4.3	Package installation	12
4.4	Refresh file name databases	12
4.5	Some details for the interested	12
5	History	13
	[2007/09/09 v1.0]	13
	[2007/12/09 v1.1]	13
	[2007/12/12 v1.2]	13
6	Index	13

1 Documentation

Version 1.40 of pdfTeX adds new primitives `\pdfcolorstackinit` and `\pdfcolorstack`. Now color stacks can be defined and used. A main color stack is maintained by the driver file `pdfTeX.def` similar to dvips or dvipdfm. However the number of color stacks is not limited to one in pdfTeX. Thus further color problems can now be solved, such as footnotes across pages or text that is set in parallel columns (e.g. packages `parallel` or `parcolumn`). Unlike the main color stack, the support by additional color stacks cannot be done in a transparent manner.

This package `pdfcol` provides an easier interface to additional color stacks without the need to use the low level primitives.

1.1 Requirements

- pdfTeX 1.40 or greater.
- pdfTeX in PDF mode. (I don't know a DVI driver that support several color stacks.)
- `pdfTeX.def` 2007/01/02 v0.04b.

Package `pdfcol` checks the requirements and sets switch `\ifpdfcolAvailable` accordingly.

1.2 Interface

`\ifpdfcolAvailable`

If the requirements of section 1.1 are met the switch `\ifpdfcolAvailable` behaves as `\iftrue`. Otherwise the other interface macros in this section will be disabled with a message. Also the first use of such a macro will print a message. The messages are print to the `.log` file only if pdfTeX is not used in PDF mode.

`\pdfcolErrorNoStacks`

The first call of `\pdfcolErrorNoStacks` prints an error message, if color stacks are not available.

`\pdfcolInitStack` $\{\langle name \rangle\}$

A new color stack is initialized by `\pdfcolInitStack`. The $\langle name \rangle$ is used for indentifying the stack. It usually consists of letters and digits. (The name must survive a `\csname`.)

The intension of the macro is the definition of an additional color stack. Thus the stack is not page bounded like the main color stack. Black (`0 g 0 G`) is used as initial color value. And colors are written with modifier `direct` that means without setting the current transfer matrix and changing the current point (see documentation of pdfTeX for `\pdfliteral direct{\dots}`).

`\pdfcolIfStackExists` $\{\langle name \rangle\}$ $\{\langle then \rangle\}$ $\{\langle else \rangle\}$

Macro `\pdfcolIfStackExists` checks whether color stack $\langle name \rangle$ exists. In case of success argument $\langle then \rangle$ is executed and $\langle else \rangle$ otherwise.

`\pdfcolSwitchStack {⟨name⟩}`

Macro `\pdfcolSwitchStack` switches the color stack. The color macros of package `color` (or `xcolor`) now uses the new color stack with name `⟨name⟩`.

`\pdfcolSetCurrentColor`

Macro `\pdfcolSetCurrentColor` replaces the topmost entry of the stack by the current color (`\current@color`).

`\pdfcolSetCurrent {⟨name⟩}`

Macro `\pdfcolSetCurrent` sets the color that is read in the top-most entry of color stack `⟨name⟩`. If `⟨name⟩` is empty, the default color stack is used.

2 Implementation

1 (*package)

2.1 Reload check and package identification

Reload check, especially if the package is not used with L^AT_EX.

```
2 \begingroup\catcode61\catcode48\catcode32=10\relax%
3 \catcode13=5 % ^~M
4 \endlinechar=13 %
5 \catcode35=6 % #
6 \catcode39=12 % '
7 \catcode44=12 % ,
8 \catcode45=12 % -
9 \catcode46=12 % .
10 \catcode58=12 % :
11 \catcode64=11 % @
12 \catcode123=1 % {
13 \catcode125=2 % }
14 \expandafter\let\expandafter\x\csname ver@pdfcol.sty\endcsname
15 \ifx\x\relax % plain-TeX, first loading
16 \else
17   \def\empty{}%
18   \ifx\x\empty % LaTeX, first loading,
19     % variable is initialized, but \ProvidesPackage not yet seen
20   \else
21     \expandafter\ifx\csname PackageInfo\endcsname\relax
22       \def\x#1#2{%
23         \immediate\write-1{Package #1 Info: #2.}%
24       }%
25     \else
26       \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
27     \fi
28     \x{pdfcol}{The package is already loaded}%
29   \aftergroup\endinput
30 \fi
31 \fi
32 \endgroup%
```

Package identification:

```
33 \begingroup\catcode61\catcode48\catcode32=10\relax%
34 \catcode13=5 % ^~M
35 \endlinechar=13 %
36 \catcode35=6 % #
37 \catcode39=12 % '
38 \catcode40=12 % (
```

```

39 \catcode41=12 % )
40 \catcode44=12 % ,
41 \catcode45=12 % -
42 \catcode46=12 % .
43 \catcode47=12 % /
44 \catcode58=12 % :
45 \catcode64=11 % @
46 \catcode91=12 % [
47 \catcode93=12 % ]
48 \catcode123=1 % {
49 \catcode125=2 % }
50 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
51   \def\x#1#2#3[#4]{\endgroup
52     \immediate\write-1{Package: #3 #4}%
53     \xdef#1{#4}%
54   }%
55 \else
56   \def\x#1#2[#3]{\endgroup
57     #2[{#3}]%
58     \ifx#1\@undefined
59       \xdef#1{#3}%
60     \fi
61     \ifx#1\relax
62       \xdef#1{#3}%
63     \fi
64   }%
65 \fi
66 \expandafter\x\csname ver@pdfcol.sty\endcsname
67 \ProvidesPackage{pdfcol}%
68 [2007/12/12 v1.2 Maintaining additional color stacks for pdfTeX (HO)]%

```

2.2 Catcodes

```

69 \begingroup\catcode61\catcode48\catcode32=10\relax%
70 \catcode13=5 % ^^M
71 \endlinechar=13 %
72 \catcode123=1 % {
73 \catcode125=2 % }
74 \catcode64=11 % @
75 \def\x{\endgroup
76   \expandafter\edef\csname PDFCOL@AtEnd\endcsname{%
77     \endlinechar=\the\endlinechar\relax
78     \catcode13=\the\catcode13\relax
79     \catcode32=\the\catcode32\relax
80     \catcode35=\the\catcode35\relax
81     \catcode61=\the\catcode61\relax
82     \catcode64=\the\catcode64\relax
83     \catcode123=\the\catcode123\relax
84     \catcode125=\the\catcode125\relax
85   }%
86 }%
87 \x\catcode61\catcode48\catcode32=10\relax%
88 \catcode13=5 % ^^M
89 \endlinechar=13 %
90 \catcode35=6 % #
91 \catcode64=11 % @
92 \catcode123=1 % {
93 \catcode125=2 % }
94 \def\TMP@EnsureCode#1#2{%
95   \edef\PDFCOL@AtEnd{%
96     \PDFCOL@AtEnd
97     \catcode#1=\the\catcode#1\relax

```

```

98 }%
99 \catcode#1=#2\relax
100 }
101 \TMP@EnsureCode{39}{12}% '
102 \TMP@EnsureCode{40}{12}% (
103 \TMP@EnsureCode{41}{12}% )
104 \TMP@EnsureCode{43}{12}% +
105 \TMP@EnsureCode{44}{12}% ,
106 \TMP@EnsureCode{46}{12}% .
107 \TMP@EnsureCode{47}{12}% /
108 \TMP@EnsureCode{91}{12}% [
109 \TMP@EnsureCode{93}{12}% ]
110 \TMP@EnsureCode{96}{12}% '
111 \edef\PDFCOL@AtEnd{\PDFCOL@AtEnd\noexpand\endinput}

```

2.3 Check requirements

\PDFCOL@RequirePackage

```

112 \begingroup\expandafter\expandafter\expandafter\endgroup
113 \expandafter\ifx\csname RequirePackage\endcsname\relax
114 \def\PDFCOL@RequirePackage#1[#2]{\input #1.sty\relax}%
115 \else
116 \def\PDFCOL@RequirePackage#1[#2]{%
117 \RequirePackage{#1}[#2]}%
118 }%
119 \fi

120 \PDFCOL@RequirePackage{ltxcmds}[2010/03/01]

```

ifpdfcolAvailable

```

121 \ltx@newif\ifpdfcolAvailable
122 \pdfcolAvailabletrue

```

2.3.1 Check package luacolor

```

123 \ltx@newif\ifPDFCOL@luacolor
124 \begingroup\expandafter\expandafter\expandafter\endgroup
125 \expandafter\ifx\csname ver@luacolor.sty\endcsname\relax
126 \PDFCOL@luacolorfalse
127 \else
128 \PDFCOL@luacolortrue
129 \fi

```

2.3.2 Check PDF mode

```

130 \PDFCOL@RequirePackage{infwarerr}[2007/09/09]
131 \PDFCOL@RequirePackage{ifpdf}[2007/09/09]
132 \ifcase\ifpdf\ifPDFCOL@luacolor 1\fi\else 1\fi 0 %
133 \def\PDFCOL@Message{%
134 \PackageWarningNoLine{pdfcol}%
135 }%
136 \else
137 \pdfcolAvailablefalse
138 \def\PDFCOL@Message{%
139 \@PackageInfoNoLine{pdfcol}%
140 }%
141 \PDFCOL@Message{%
142 Interface disabled because of %
143 \ifPDFCOL@luacolor
144 package 'luacolor'%
145 \else
146 missing PDF mode of pdfTeX%
147 \fi
148 }%

```

```
149 \fi
```

2.3.3 Check version of pdfTeX

```
150 \ifpdfcolAvailable
151   \begingroup\expandafter\expandafter\expandafter\endgroup
152   \expandafter\ifx\csname pdfcolorstack\endcsname\relax
153     \pdfcolAvailablefalse
154     \PDFCOL@Message{%
155       Interface disabled because of too old pdfTeX.\MessageBreak
156       Required is version 1.40+ for \string\pdfcolorstack
157     }%
158   \fi
159 \fi
160 \ifpdfcolAvailable
161   \begingroup\expandafter\expandafter\expandafter\endgroup
162   \expandafter\ifx\csname pdfcolorstack\endcsname\relax
163     \pdfcolAvailablefalse
164     \PDFCOL@Message{%
165       Interface disabled because of too old pdfTeX.\MessageBreak
166       Required is version 1.40+ for \string\pdfcolorstackinit
167     }%
168   \fi
169 \fi
```

2.3.4 Check pdftex.def

```
170 \ifpdfcolAvailable
171   \begingroup\expandafter\expandafter\expandafter\endgroup
172   \expandafter\ifx\csname @pdfcolorstack\endcsname\relax
173     Try to load package color if it is not yet loaded (LATEX case).
174     \begingroup\expandafter\expandafter\expandafter\endgroup
175     \expandafter\ifx\csname ver@color.sty\endcsname\relax
176       \begingroup\expandafter\expandafter\expandafter\endgroup
177       \expandafter\ifx\csname documentclass\endcsname\relax
178         \else
179           \RequirePackage[pdftex]{color}\relax
180         \fi
181       \fi
182     \begingroup\expandafter\expandafter\expandafter\endgroup
183     \expandafter\ifx\csname @pdfcolorstack\endcsname\relax
184       \pdfcolAvailablefalse
185       \PDFCOL@Message{%
186         Interface disabled because ‘pdftex.def’\MessageBreak
187         is not loaded or it is too old.\MessageBreak
188         Required is version 0.04b or greater%
189       }%
190     \fi
191   \fi
192 \let\pdfcolAvailabletrue\relax
193 \let\pdfcolAvailablefalse\relax
```

2.4 Enabled interface macros

```
194 \ifpdfcolAvailable
195   \pdfcolErrorNoStacks
196   \let\pdfcolErrorNoStacks\relax
197   \pdfcol@Value
198   \expandafter\ifx\csname pdfcol@Value\endcsname\relax
199     \def\pdfcol@Value{0 g 0 G}%
200   \fi
```

```

\pdfcol@LiteralModifier
199 \expandafter\ifx\csname pdfcol@LiteralModifier\endcsname\relax
200 \def\pdfcol@LiteralModifier{direct}%
201 \fi

\pdfcolInitStack
202 \def\pdfcolInitStack#1{%
203 \expandafter\ifx\csname pdfcol@Stack@#1\endcsname\relax
204 \global\expandafter\chardef\csname pdfcol@Stack@#1\endcsname=%
205 \pdfcolorstackinit\pdfcol@LiteralModifier{\pdfcol@Value}%
206 \relax
207 \@PackageInfo{pdfcol}{%
208 New color stack ‘#1’ = \number\csname pdfcol@Stack@#1\endcsname
209 }%
210 \else
211 \@PackageError{pdfcol}{%
212 Stack ‘#1’ is already defined%
213 }\@ehc
214 \fi
215 }%

\pdfcolIfStackExists
216 \def\pdfcolIfStackExists#1{%
217 \expandafter\ifx\csname pdfcol@Stack@#1\endcsname\relax
218 \expandafter\@secondoftwo
219 \else
220 \expandafter\@firstoftwo
221 \fi
222 }%

\@firstoftwo
223 \expandafter\ifx\csname @firstoftwo\endcsname\relax
224 \long\def\@firstoftwo#1#2{#1}%
225 \fi

\@secondoftwo
226 \expandafter\ifx\csname @secondoftwo\endcsname\relax
227 \long\def\@secondoftwo#1#2{#2}%
228 \fi

\pdfcolSwitchStack
229 \def\pdfcolSwitchStack#1{%
230 \pdfcolIfStackExists{#1}{%
231 \expandafter\let\expandafter\@pdfcolorstack
232 \csname pdfcol@Stack@#1\endcsname
233 }{%
234 \pdfcol@ErrorNoStack{#1}%
235 }%
236 }%

\pdfcolSetCurrentColor
237 \def\pdfcolSetCurrentColor{%
238 \pdfcolorstack\@pdfcolorstack set{\current@color}%
239 }%

\pdfcolSetCurrent
240 \def\pdfcolSetCurrent#1{%
241 \ifx\#1\%
242 \pdfcolorstack\@pdfcolorstack current\relax
243 \else

```

```

244     \pdfcolIfStackExists{#1}{%
245     \pdfcolorstack\csname pdfcol@Stack@#1\endcsname current\relax
246     }{%
247     \pdfcol@ErrorNoStack{#1}%
248     }%
249     \fi
250 }%

```

\pdfcol@ErrorNoStack

```

251 \def\pdfcol@ErrorNoStack#1{%
252   \@PackageError{pdfcol}{Stack ‘#1’ does not exists}\@ehc
253 }%

```

2.5 Disabled interface macros

```

254 \else

```

\pdfcolErrorNoStacks

```

255 \def\pdfcolErrorNoStacks{%
256   \@PackageError{pdfcol}{%
257     Color stacks are not available%
258   }{%
259     Update pdfTeX (1.40) and ‘pdftex.def’ (0.04b) %
260     if necessary.\MessageBreak
261     Ensure that ‘pdftex.def’ is loaded %
262     (package ‘color’ or ‘xcolor’).\MessageBreak
263     Further messages can be found in TeX’s %
264     protocol file ‘\jobname.log’.\MessageBreak
265     \MessageBreak
266     \@ehc
267   }%
268   \global\let\pdfcolErrorNoStacks\relax
269 }%

```

\PDFCOL@Disabled

```

270 \def\PDFCOL@Disabled{%
271   \PDFCOL@Message{%
272     pdfTeX’s color stacks are not available%
273   }%
274   \global\let\PDFCOL@Disabled\relax
275 }%

```

\pdfcolInitStack

```

276 \def\pdfcolInitStack#1{%
277   \PDFCOL@Disabled
278 }%

```

\pdfcolIfStackExists

```

279 \long\def\pdfcolIfStackExists#1#2#3{#3}%

```

\pdfcolSwitchStack

```

280 \def\pdfcolSwitchStack#1{%
281   \PDFCOL@Disabled
282 }%

```

\pdfcolSetCurrentColor

```

283 \def\pdfcolSetCurrentColor{%
284   \PDFCOL@Disabled
285 }%

```


\pdfcolSetCurrent

```
286 \def\pdfcolSetCurrent#1{%
287   \PDFCOL@Disabled
288 }%

289 \fi

290 \PDFCOL@AtEnd%
291 \end{package}
```

3 Test

3.1 Catcode checks for loading

```
292 \test1

293 \catcode'\{=1 %
294 \catcode'\}=2 %
295 \catcode'\#=6 %
296 \catcode'\@=11 %
297 \expandafter\ifx\csname count@\endcsname\relax
298   \countdef\count@=255 %
299 \fi
300 \expandafter\ifx\csname @gobble\endcsname\relax
301   \long\def\@gobble#1{}%
302 \fi
303 \expandafter\ifx\csname @firstofone\endcsname\relax
304   \long\def\@firstofone#1{#1}%
305 \fi
306 \expandafter\ifx\csname loop\endcsname\relax
307   \expandafter\@firstofone
308 \else
309   \expandafter\@gobble
310 \fi
311 {%
312   \def\loop#1\repeat{%
313     \def\body{#1}%
314     \iterate
315   }%
316   \def\iterate{%
317     \body
318     \let\next\iterate
319   \else
320     \let\next\relax
321   \fi
322   \next
323 }%
324 \let\repeat=\fi
325 }%
326 \def\RestoreCatcodes{}
327 \count@=0 %
328 \loop
329   \edef\RestoreCatcodes{%
330     \RestoreCatcodes
331     \catcode\the\count@=\the\catcode\count@\relax
332   }%
333 \ifnum\count@<255 %
334   \advance\count@ 1 %
335 \repeat
336
337 \def\RangeCatcodeInvalid#1#2{%
338   \count@=#1\relax
339   \loop
```

```

340     \catcode\count@=15 %
341     \ifnum\count@<#2\relax
342     \advance\count@ 1 %
343     \repeat
344 }
345 \def\RangeCatcodeCheck#1#2#3{%
346     \count@=#1\relax
347     \loop
348     \ifnum#3=\catcode\count@
349     \else
350     \errmessage{%
351         Character \the\count@\space
352         with wrong catcode \the\catcode\count@\space
353         instead of \number#3%
354     }%
355     \fi
356     \ifnum\count@<#2\relax
357     \advance\count@ 1 %
358     \repeat
359 }
360 \def\space{ }
361 \expandafter\ifx\csname LoadCommand\endcsname\relax
362     \def\LoadCommand{\input pdfcol.sty\relax}%
363 \fi
364 \def\Test{%
365     \RangeCatcodeInvalid{0}{47}%
366     \RangeCatcodeInvalid{58}{64}%
367     \RangeCatcodeInvalid{91}{96}%
368     \RangeCatcodeInvalid{123}{255}%
369     \catcode'\@=12 %
370     \catcode'\=0 %
371     \catcode'\%=14 %
372     \LoadCommand
373     \RangeCatcodeCheck{0}{36}{15}%
374     \RangeCatcodeCheck{37}{37}{14}%
375     \RangeCatcodeCheck{38}{47}{15}%
376     \RangeCatcodeCheck{48}{57}{12}%
377     \RangeCatcodeCheck{58}{63}{15}%
378     \RangeCatcodeCheck{64}{64}{12}%
379     \RangeCatcodeCheck{65}{90}{11}%
380     \RangeCatcodeCheck{91}{91}{15}%
381     \RangeCatcodeCheck{92}{92}{0}%
382     \RangeCatcodeCheck{93}{96}{15}%
383     \RangeCatcodeCheck{97}{122}{11}%
384     \RangeCatcodeCheck{123}{255}{15}%
385     \RestoreCatcodes
386 }
387 \Test
388 \csname @@end\endcsname
389 \end
390 </test1>

```

3.2 Very simple test

```

391 <{*test2 | test3}
392 \NeedsTeXFormat{LaTeX2e}
393 \nofiles
394 \documentclass{article}
395 \usepackage{pdfcol}[2007/12/12]
396 \usepackage{qstest}
397 \IncludeTests{*}
398 \LogTests{log}{*}{*}
399 \begin{document}

```

```

400 \begin{qstest}{pdfcol}{}%
401 \makeatletter
402 \test2
403 \Expect*{\ifpdfcolAvailable true\else false\fi}{false}%
404 \test2
405 \test3
406 \Expect*{\ifpdfcolAvailable true\else false\fi}{true}%
407 \Expect*{\number\@pdfcolorstack}{0}%
408 \test3
409 \setbox0=\hbox{%
410 \pdfcolInitStack{test}%
411 \test3
412 \Expect*{\number\pdfcol@Stack@test}{1}%
413 \Expect*{\number\@pdfcolorstack}{0}%
414 \test3
415 \pdfcolSwitchStack{test}%
416 \test3
417 \Expect*{\number\@pdfcolorstack}{1}%
418 \test3
419 \pdfcolSetCurrent{test}%
420 \pdfcolSetCurrent{}%
421 }%
422 \Expect*{\the\wd0}{0.0pt}%
423 \test3
424 \Expect*{\number\@pdfcolorstack}{0}%
425 \Expect*{\number\pdfcol@Stack@test}{1}%
426 \Expect*{\pdfcolIfStackExists{test}{true}{false}}{true}%
427 \test3
428 \Expect*{\pdfcolIfStackExists{dummy}{true}{false}}{false}%
429 \end{qstest}%
430 \end{document}
431 \test2 | \test3

```

3.3 Detection of package luacolor

```

432 \test4
433 \NeedsTeXFormat{LaTeX2e}
434 \documentclass{article}
435 \usepackage{luacolor}
436 \usepackage{pdfcol}
437 \makeatletter
438 \ifpdfcolAvailable
439 \latexerror{Detection of package luacolor failed}%
440 \fi
441 \csname @@end\endcsname
442 \test4

```

4 Installation

4.1 Download

Package. This package is available on CTAN¹:

[CTAN:macros/latex/contrib/oberdiek/pdfcol.dtx](http://ctan.org/ctan/CTAN:macros/latex/contrib/oberdiek/pdfcol.dtx) The source file.

[CTAN:macros/latex/contrib/oberdiek/pdfcol.pdf](http://ctan.org/ctan/CTAN:macros/latex/contrib/oberdiek/pdfcol.pdf) Documentation.

Bundle. All the packages of the bundle ‘oberdiek’ are also available in a TDS compliant ZIP archive. There the packages are already unpacked and the documentation files are generated. The files and directories obey the TDS standard.

[CTAN:install/macros/latex/contrib/oberdiek.tds.zip](http://ctan.org/ctan/CTAN:install/macros/latex/contrib/oberdiek.tds.zip)

¹<http://ftp.ctan.org/tex-archive/>

TDS refers to the standard “A Directory Structure for \TeX Files” ([CTAN:tds/tds.pdf](#)). Directories with `texmf` in their name are usually organized this way.

4.2 Bundle installation

Unpacking. Unpack the `oberdiek.tds.zip` in the TDS tree (also known as `texmf` tree) of your choice. Example (linux):

```
unzip oberdiek.tds.zip -d ~/texmf
```

Script installation. Check the directory `TDS:scripts/oberdiek/` for scripts that need further installation steps. Package `attachfile2` comes with the Perl script `pdfatfi.pl` that should be installed in such a way that it can be called as `pdfatfi`. Example (linux):

```
chmod +x scripts/oberdiek/pdfatfi.pl
cp scripts/oberdiek/pdfatfi.pl /usr/local/bin/
```

4.3 Package installation

Unpacking. The `.dtx` file is a self-extracting `docstrip` archive. The files are extracted by running the `.dtx` through plain \TeX :

```
tex pdfcol.dtx
```

TDS. Now the different files must be moved into the different directories in your installation TDS tree (also known as `texmf` tree):

```
pdfcol.sty          → tex/generic/oberdiek/pdfcol.sty
pdfcol.pdf          → doc/latex/oberdiek/pdfcol.pdf
test/pdfcol-test1.tex → doc/latex/oberdiek/test/pdfcol-test1.tex
test/pdfcol-test2.tex → doc/latex/oberdiek/test/pdfcol-test2.tex
test/pdfcol-test3.tex → doc/latex/oberdiek/test/pdfcol-test3.tex
test/pdfcol-test4.tex → doc/latex/oberdiek/test/pdfcol-test4.tex
pdfcol.dtx          → source/latex/oberdiek/pdfcol.dtx
```

If you have a `docstrip.cfg` that configures and enables `docstrip`’s TDS installing feature, then some files can already be in the right place, see the documentation of `docstrip`.

4.4 Refresh file name databases

If your \TeX distribution (te \TeX , mi \TeX , ...) relies on file name databases, you must refresh these. For example, te \TeX users run `texhash` or `mktextlsr`.

4.5 Some details for the interested

Attached source. The PDF documentation on CTAN also includes the `.dtx` source file. It can be extracted by AcrobatReader 6 or higher. Another option is `pdftk`, e.g. unpack the file into the current directory:

```
pdftk pdfcol.pdf unpack_files output .
```

Unpacking with \LaTeX . The `.dtx` chooses its action depending on the format:
plain \TeX : Run `docstrip` and extract the files.

\LaTeX : Generate the documentation.

If you insist on using \LaTeX for `docstrip` (really, `docstrip` does not need \LaTeX), then inform the autodetect routine about your intention:

```
latex \let\install=y\input{pdfcol.dtx}
```

Do not forget to quote the argument according to the demands of your shell.

\PassOptionsToClass{a4paper}{article}

```
pdflatex pdfcol.dtx
makeindex -s gind.ist pdfcol.idx
pdflatex pdfcol.dtx
makeindex -s gind.ist pdfcol.idx
pdflatex pdfcol.dtx
```

[2007/09/09 v1.0]

- [2007/12/09 v1.1]**

- [2007/12/12 v1.2]

- Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; plain numbers refer to the code lines where the entry is used.

13

E	
\empty	17, 18
\end	389, 429, 430
\endcsname	14, 21, 50, 66, 76, 113, 125, 152, 162, 172, 174, 176, 182, 196, 199, 203, 204, 208, 217, 223, 226, 232, 245, 297, 300, 303, 306, 361, 388, 441
\endinput	29, 111
\endlinechar	4, 35, 71, 77, 89
\errmessage	350
\Expect	403, 406, 407, 412, 413, 417, 422, 424, 425, 426, 428
H	
\hbox	409
I	
\ifcase	132
\ifnum	333, 341, 348, 356
\ifpdf	132
\ifPDFCOL@luacolor	123, 132, 143
\ifpdfcolAvailable	2, 121, 121, 150, 160, 170, 194, 403, 406, 438
\ifx	15, 18, 21, 50, 58, 61, 113, 125, 152, 162, 172, 174, 176, 182, 196, 199, 203, 217, 223, 226, 241, 297, 300, 303, 306, 361
\immediate	23, 52
\IncludeTests	397
\input	114, 362
\iterate	314, 316, 318
J	
\jobname	264
L	
\LoadCommand	362, 372
\LogTests	398
\loop	312, 328, 339, 347
\ltx@newif	121, 123
M	
\makeatletter	401, 437
\MessageBreak	155, 165, 185, 186, 260, 262, 264, 265
N	
\NeedsTeXFormat	392, 433
\next	318, 320, 322
\nofiles	393
\number	208, 353, 407, 412, 413, 417, 424, 425
P	
\PackageInfo	26
\PDFCOL@AtEnd	95, 96, 111, 290
\PDFCOL@Disabled	270, 277, 281, 284, 287
\pdfcol@ErrorNoStack	234, 247, 251
\pdfcol@LiteralModifier	199, 205
\PDFCOL@luacolorfalse	126
\PDFCOL@luacolortrue	128
\PDFCOL@Message	133, 138, 141, 154, 164, 184, 271
\PDFCOL@RequirePackage	112, 120, 130, 131
\pdfcol@Stack@test	412, 425
\pdfcol@Value	196, 205
\pdfcolAvailablefalse	137, 153, 163, 183, 193
\pdfcolAvailabletrue	122, 192
\pdfcolErrorNoStacks	2, 195, 255
\pdfcolIfStackExists	2, 216, 230, 244, 279, 426, 428
\pdfcolInitStack	2, 202, 276, 410
\pdfcolorstack	156, 238, 242, 245
\pdfcolorstackinit	166, 205
\pdfcolSetCurrent	3, 240, 286, 419, 420
\pdfcolSetCurrentColor	3, 237, 283
\pdfcolSwitchStack	3, 229, 280, 415
\ProvidesPackage	19, 67
R	
\RangeCatcodeCheck	345, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383, 384
\RangeCatcodeInvalid	337, 365, 366, 367, 368
\repeat	312, 324, 335, 343, 358
\RequirePackage	117, 178
\RestoreCatcodes	326, 329, 330, 385
S	
\setbox	409
\space	351, 352, 360
T	
\Test	364, 387
\the	77, 78, 79, 80, 81, 82, 83, 84, 97, 331, 351, 352, 422
\TMP@EnsureCode	94, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110
U	
\usepackage	395, 396, 435, 436
W	
\wd	422
\write	23, 52
X	
\x	14, 15, 18, 22, 26, 28, 51, 56, 66, 75, 87