

# The pdfescape package

Heiko Oberdiek  
<oberdiek@uni-freiburg.de>

2007/10/27 v1.7

## Abstract

This package implements pdf<sub>TEX</sub>'s escape features (`\pdfescapehex`, `\pdfunescapehex`, `\pdfescapename`, `\pdfescapestring`) using  $\text{T}_{\text{E}}\text{X}$  or  $\varepsilon\text{-T}_{\text{E}}\text{X}$ .

## Contents

<b>1</b>	<b>Documentation</b>	<b>2</b>
1.1	Additional unescape macros	2
1.2	Sanitizing macro	2
<b>2</b>	<b>Implementation</b>	<b>3</b>
2.1	Reload check and package identification	3
2.2	Catcodes	4
2.3	Sanitizing	4
2.3.1	Space characters	5
2.3.2	Space normalization	5
2.4	<code>\EdefUnescapeName</code>	5
2.5	<code>\EdefUnescapeString</code>	7
2.6	User macros (pdf <sub>TEX</sub> analogues)	10
2.7	Help macros	11
2.7.1	Characters	11
2.7.2	Switch for $\varepsilon\text{-T}_{\text{E}}\text{X}$	11
2.8	Conversions	11
2.8.1	Conversion to hex string	11
2.8.2	Character code to octal number	12
2.8.3	Unpack hex string	13
2.8.4	Conversion to PDF name	14
2.8.5	Conversion to PDF string	15
<b>3</b>	<b>Test</b>	<b>15</b>
3.1	Catcode checks for loading	15
3.2	Macro tests	16
3.3	Test with <code>\pdfescape...</code> commands	16
3.4	Test without <code>\pdfescape...</code> , with $\varepsilon\text{-T}_{\text{E}}\text{X}$	16
3.5	Test without <code>\pdfescape...</code> and $\varepsilon\text{-T}_{\text{E}}\text{X}$	16
3.6	Check/ensure test preconditions	17
3.6.1	Check <code>\pdfescape...</code>	17
3.6.2	Check $\varepsilon\text{-T}_{\text{E}}\text{X}$	17
3.7	Common part	17
<b>4</b>	<b>Installation</b>	<b>23</b>
4.1	Download	23
4.2	Bundle installation	23
4.3	Package installation	24
4.4	Refresh file name databases	24
4.5	Some details for the interested	24

<b>5 History</b>	<b>25</b>
[2007/02/21 v1.0]	25
[2007/02/25 v1.1]	25
[2007/03/20 v1.2]	25
[2007/04/11 v1.3]	25
[2007/04/21 v1.4]	25
[2007/08/27 v1.5]	25
[2007/09/09 v1.6]	25
[2007/10/27 v1.7]	25
<b>6 Index</b>	<b>25</b>

# 1 Documentation

<code>\EdefEscapeHex {⟨cmd⟩} {⟨string⟩}</code> <code>\EdefUnescapeHex {⟨cmd⟩} {⟨string⟩}</code> <code>\EdefEscapeName {⟨cmd⟩} {⟨string⟩}</code> <code>\EdefEscapeString {⟨cmd⟩} {⟨string⟩}</code>
--

These commands converts  $\langle string \rangle$  and stores the result in macro  $\langle cmd \rangle$ . The conversion result is the same as the conversion of the corresponding pdfTeX's primitives. Note that the argument  $\langle string \rangle$  is expanded before the conversion.

For example, if pdfTeX  $\geq 1.30$  is present, then `\EdefEscapeHex` becomes to:

```
\def\EdefEscapeHex#1#2{%
  \edef#1{\pdfescapehex{#2}}%
}
```

The package provides implementations for the case that pdfTeX is not present (or too old). Even  $\varepsilon$ -TeX can be missing, however it is used if it is detected.

**Babel.** The input strings may contain shorthand characters of package `babel`.

## 1.1 Additional unescape macros

<code>\EdefUnescapeName {⟨cmd⟩} {⟨string⟩}</code>
---

Sequences of a hash sign with two hexadecimal digits are converted to the corresponding character (PDF-1.2). A hash sign that is not followed by two hexadecimal digits is left unchanged. The catcodes in the result string follow TeX's conventions. The space has catcode 10 (space) and the other characters have catcode 12 (other).

<code>\EdefUnescapeString {⟨cmd⟩} {⟨string⟩}</code>
---

Macro  $\langle cmd \rangle$  stores the unescaped string in  $\langle string \rangle$ . All the rules for literal strings are implemented, see PDF specification. The catcodes in the result string follow TeX's conventions.

## 1.2 Sanitizing macro

<code>\EdefSanitize {⟨cmd⟩} {⟨string⟩}</code>
---

Argument  $\langle string \rangle$  is expanded, converted to a string of tokens with catcode 12 (other) and space tokens, and stored in macro  $\langle cmd \rangle$ .

## 2 Implementation

```
1 <*package>
```

### 2.1 Reload check and package identification

Reload check, especially if the package is not used with L<sup>A</sup>T<sub>E</sub>X.

```
2 \begingroup
3 \catcode44 12 % ,
4 \catcode45 12 % -
5 \catcode46 12 % .
6 \catcode58 12 % :
7 \catcode64 11 % @
8 \expandafter\let\expandafter\x\csname ver@pdfescape.sty\endcsname
9 \ifcase 0%
10 \ifx\x\relax % plain
11 \else
12 \ifx\x\empty % LaTeX
13 \else
14 1%
15 \fi
16 \fi
17 \else
18 \expandafter\ifx\csname PackageInfo\endcsname\relax
19 \def\x#1#2{%
20 \immediate\write-1{Package #1 Info: #2.}%
21 }%
22 \else
23 \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
24 \fi
25 \x{pdfescape}{The package is already loaded}%
26 \endgroup
27 \expandafter\endinput
28 \fi
29 \endgroup
```

Package identification:

```
30 \begingroup
31 \catcode40 12 % (
32 \catcode41 12 % )
33 \catcode44 12 % ,
34 \catcode45 12 % -
35 \catcode46 12 % .
36 \catcode47 12 % /
37 \catcode58 12 % :
38 \catcode64 11 % @
39 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
40 \def\x#1#2#3[#4]{\endgroup
41 \immediate\write-1{Package: #3 #4}%
42 \xdef#1{#4}%
43 }%
44 \else
45 \def\x#1#2[#3]{\endgroup
46 #2[#{#3}]%
47 \ifx#1\relax
48 \xdef#1{#3}%
49 \fi
50 }%
51 \fi
52 \expandafter\x\csname ver@pdfescape.sty\endcsname
53 \ProvidesPackage{pdfescape}%
54 [2007/10/27 v1.7 Provides hex, PDF name and string conversions (H0)]
```

## 2.2 Catcodes

```

55 \expandafter\edef\csname PE@AtEnd\endcsname{%
56   \catcode64 \the\catcode64\relax
57 }
58 \catcode64 11 % @
59 \def\TMP@EnsureCode#1#2#3{%
60   \edef\PE@AtEnd{%
61     \PE@AtEnd
62     #1#2 \the#1#2\relax
63   }%
64   #1#2 #3\relax
65 }
66 \TMP@EnsureCode\catcode{0}-{12}% ^^@
67 \TMP@EnsureCode\catcode{34}-{12}% "
68 \TMP@EnsureCode\catcode{39}-{12}% '
69 \TMP@EnsureCode\catcode{42}-{12}% *
70 \TMP@EnsureCode\catcode{45}-{12}% -
71 \TMP@EnsureCode\catcode{46}-{12}% .
72 \TMP@EnsureCode\catcode{60}-{12}% <
73 \TMP@EnsureCode\catcode{61}-{12}% =
74 \TMP@EnsureCode\catcode{62}-{12}% >
75 \TMP@EnsureCode\catcode{94}-{7}% ^
76 \TMP@EnsureCode\catcode{96}-{12}% ‘
77 \TMP@EnsureCode\uccode{34}-{0}% "
78 \TMP@EnsureCode\uccode{48}-{0}% 0
79 \TMP@EnsureCode\uccode{61}-{0}% =

```

## 2.3 Sanitizing

`\EdefSanitize` Macro `\EdefSanitize` takes #2, entirely converts it to token with catcode 12 (other) and stores the result in macro #1.

```

80 \begingroup\expandafter\expandafter\expandafter\endgroup
81 \expandafter\ifx\csname detokenize\endcsname\relax
82   \long\def\EdefSanitize#1#2{%
83     \begingroup
84       \csname @safe@activetrue\endcsname
85       \edef#1{#2}%
86       \PE@onelevel@sanitize#1%
87     \expandafter\endgroup
88     \expandafter\def\expandafter#1\expandafter{#1}%
89   }%
90 \begingroup\expandafter\expandafter\expandafter\endgroup
91 \expandafter\ifx\csname @onelevel@sanitize\endcsname\relax
92   \def\PE@onelevel@sanitize#1{%
93     \edef#1{\expandafter\PE@strip@prefix\meaning#1}%
94   }%
95   \def\PE@strip@prefix#1>{}%
96 \else
97   \let\PE@onelevel@sanitize\@onelevel@sanitize
98 \fi
99 \else
100  \long\def\EdefSanitize#1#2{%
101    \begingroup
102      \csname @safe@activetrue\endcsname
103      \edef#1{#2}%
104    \expandafter\endgroup
105    \expandafter\def\expandafter#1\expandafter{%
106      \detokenize\expandafter{#1}%
107    }%
108  }%
109  \def\PE@onelevel@sanitize#1{%
110    \edef#1{\detokenize\expandafter{#1}}%

```

```

111 }%
112 \fi

```

`\PE@sanitize` Macro `\PE@sanitize` is only defined for compatibility with version 1.4. Its use is deprecated.

```

113 \let\PE@sanitize\EdefSanitize

```

### 2.3.1 Space characters

`\PE@space@other`

```

114 \begingroup
115 \catcode'\ =12\relax%
116 \def\x{\endgroup\def\PE@space@other{ }}\x\relax

```

`\PE@space@space`

```

117 \def\PE@space@space{ }

```

### 2.3.2 Space normalization

`\PE@SanitizeSpaceOther`

```

118 \def\PE@SanitizeSpaceOther#1{%
119 \edef#1{\expandafter\PE@SpaceToOther#1 \relax}%
120 }

```

`\PE@SpaceToOther`

```

121 \def\PE@SpaceToOther#1 #2\relax{%
122 #1%
123 \ifx\\#2\\%
124 \else
125 \PE@space@other
126 \@ReturnAfterFi{%
127 \PE@SpaceToOther#2\relax
128 }%
129 \fi
130 }

```

`\@ReturnAfterFi`

```

131 \long\def\@ReturnAfterFi#1\fi{\fi#1}

```

## 2.4 \EdefUnescapeName

`\EdefUnescapeName`

```

132 \def\EdefUnescapeName#1#2{%
133 \EdefSanitize#1{#2}%
134 \PE@SanitizeSpaceOther#1%
135 \PE@UnescapeName#1%
136 \PE@onelevel@sanitize#1%
137 }

```

`\PE@UnescapeName`

```

138 \begingroup
139 \catcode'\$=6 % hash
140 \catcode'\#=12 % other
141 \gdef\PE@UnescapeName$1{%
142 \begingroup
143 \PE@InitUccodeHexDigit
144 \def\PE@result{}%
145 \expandafter\PE@DeName$1#\relax\relax
146 \expandafter\endgroup
147 \expandafter\def\expandafter$1\expandafter{\PE@result}%

```

```

148 }%
149 \gdef\PE@DeName$1#$2$3{%
150   \ifx\relax$2%
151     \edef\PE@result{\PE@result$1}%
152     \let\PE@next\relax
153   \else
154     \ifx\relax$3%
155       % wrong escape sequence in input
156       \edef\PE@result{\PE@result$1#}%
157       \let\PE@next\relax
158     \else
159       \uppercase{%
160         \def\PE@testA{$2}%
161         \def\PE@testB{$3}%
162       }%
163       \ifcase\ifcase\expandafter\PE@TestUcHexDigit\PE@testA
164         \ifcase\expandafter\PE@TestUcHexDigit\PE@testB
165         \z@
166       \else
167         \@ne
168       \fi
169     \else
170       \@ne
171     \fi
172     \uccode\z@="\PE@testA\PE@testB\relax
173     \uppercase{%
174       \def\PE@temp{^~@}%
175     }%
176     \uccode\z@=\z@
177     \edef\PE@result{\PE@result$1\PE@temp}%
178     \let\PE@next\PE@DeName
179   \else
180     % wrong escape sequence in input
181     \edef\PE@result{\PE@result$1#}%
182     \def\PE@next{\PE@DeName$2$3}%
183   \fi
184 \fi
185 \fi
186 \PE@next
187 }%
188 \endgroup

```

\PE@InitUccodeHexDigit

```

189 \def\PE@InitUccodeHexDigit{%
190   \uccode'a='A\relax
191   \uccode'b='B\relax
192   \uccode'c='C\relax
193   \uccode'd='D\relax
194   \uccode'e='E\relax
195   \uccode'f='F\relax
196   \uccode'A=\z@
197   \uccode'B=\z@
198   \uccode'C=\z@
199   \uccode'D=\z@
200   \uccode'E=\z@
201   \uccode'F=\z@
202   \uccode'0=\z@
203   \uccode'1=\z@
204   \uccode'2=\z@
205   \uccode'3=\z@
206   \uccode'4=\z@
207   \uccode'5=\z@
208   \uccode'6=\z@

```

```

209 \uccode'7=\z@
210 \uccode'8=\z@
211 \uccode'9=\z@
212 }

```

\PE@TestUcHexDigit

```

213 \def\PE@TestUcHexDigit#1{%
214 \ifnum'#1<48 % 0
215 \@ne
216 \else
217 \ifnum'#1>70 % F
218 \@ne
219 \else
220 \ifnum'#1>57 % 9
221 \ifnum'#1<65 % A
222 \@ne
223 \else
224 \z@
225 \fi
226 \else
227 \z@
228 \fi
229 \fi
230 \fi
231 }

```

## 2.5 \EdefUnescapeString

\EdefUnescapeString

```

232 \def\EdefUnescapeString#1#2{%
233 \EdefSanitize#1{#2}%
234 \PE@SanitizeSpaceOther#1%
235 \PE@NormalizeLineEnd#1%
236 \PE@UnescapeString#1%
237 \PE@onelevel@sanitize#1%
238 }

239 \begingroup
240 \uccode'\8=10 % lf
241 \uccode'\9=13 % cr
242 \def\x#1#2{\endgroup

```

\PE@NormalizeLineEnd

```

243 \def\PE@NormalizeLineEnd##1{%
244 \def\PE@result{}%
245 \expandafter\PE@@NormalizeLineEnd##1#2\relax
246 \let##1\PE@result
247 }%

```

\PE@@NormalizeLineEnd

```

248 \def\PE@@NormalizeLineEnd##1#2##2{%
249 \ifx\relax##2%
250 \edef\PE@result{\PE@result##1}%
251 \let\PE@next\relax
252 \else
253 \edef\PE@result{\PE@result##1#1}%
254 \ifx#1##2% lf
255 \let\PE@next\PE@@NormalizeLineEnd
256 \else
257 \def\PE@next{\PE@@NormalizeLineEnd##2}%
258 \fi
259 \fi

```

```

260     \PE@next
261 }%
262 }%
263 \uppercase{%
264   \x 89%
265 }

266 \begingroup
267   \catcode'\|=0 %
268   \catcode'\|=12 %

```

\PE@UnescapeString

```

269 |gdef|PE@UnescapeString#1{%
270   |begingroup
271     |def|PE@result{}%
272     |expandafter|PE@DeString#1\|relax
273   |expandafter|endgroup
274   |expandafter|def|expandafter#1|expandafter{|PE@result}%
275 }%

```

\PE@DeString

```

276 |gdef|PE@DeString#1\#2{%
277   |ifx|relax#2%
278     |edef|PE@result{|PE@result#1}%
279     |let|PE@next|relax
280   |else
281     |if n#2%
282       |uccode|z@=10 %
283     |else|if r#2%
284       |uccode|z@=13 %
285     |else|if t#2%
286       |uccode|z@=9 %
287     |else|if b#2%
288       |uccode|z@=8 %
289     |else|if f#2%
290       |uccode|z@=12 %
291     |else
292       |uccode|z@=|z@
293     |fi|fi|fi|fi|fi
294     |ifnum|uccode|z@>|z@
295       |uppercase{%
296         |edef|PE@temp{^^@}%
297       }%
298       |edef|PE@result{|PE@result#1|PE@temp}%
299       |let|PE@next|PE@DeString
300     |else
301       |if\#2% backslash
302         |edef|PE@result{|PE@result#1}%
303         |let|PE@next|PE@CheckEndBackslash
304       |else
305         |ifnum'#2=10 % linefeed
306         |edef|PE@result{|PE@result#1}%
307         |let|PE@next|PE@DeString
308       |else
309         |ifcase|PE@TestOctDigit#2%
310         |edef|PE@result{|PE@result#1}%
311         |def|PE@next{|PE@OctI#2}%
312       |else
313         |edef|PE@result{|PE@result#1#2}%
314         |let|PE@next|PE@DeString
315       |fi
316     |fi

```



```

317         |fi
318     |fi
319 |fi
320 |PE@next
321 }%

\PE@CheckEndBackslash

322 |gdef|PE@CheckEndBackslash#1{%
323     |ifx|relax#1%
324     |else
325         |edef|PE@result{|PE@result\}%
326         |expandafter|PE@DeString|expandafter#1%
327     |fi
328 }%

329 |endgroup

\PE@TestOctDigit

330 \def\PE@TestOctDigit#1{%
331     \ifnum'#1<48 % 0
332         \@ne
333     \else
334         \ifnum'#1>55 % 7
335             \@ne
336         \else
337             \z@
338         \fi
339     \fi
340 }

\PE@OctI

341 \def\PE@OctI#1#2{%
342     \ifcase\PE@TestOctDigit#2%
343         \def\PE@next{\PE@OctII{#1#2}}%
344     \else
345         \def\PE@next{\PE@OctAll#1#2}%
346     \fi
347     \PE@next
348 }

\PE@OctII

349 \def\PE@OctII#1#2{%
350     \ifcase\PE@TestOctDigit#2%
351         \def\PE@next{\PE@OctAll{#1#2}}%
352     \else
353         \def\PE@next{\PE@OctAll{#1}#2}%
354     \fi
355     \PE@next
356 }

\PE@OctAll

357 \def\PE@OctAll#1{%
358     \uccode\z@='#1\relax
359     \uppercase{%
360         \edef\PE@result{\PE@result^^@}%
361     }%
362     \PE@DeString
363 }

```

## 2.6 User macros (pdfTeX analogues)

```

364 \begingroup\expandafter\expandafter\expandafter\endgroup
365 \expandafter\ifx\csname pdfescapehex\endcsname\relax

\edefEscapeHex
366 \long\def\edefEscapeHex#1#2{%
367   \edefSanitize#1{#2}%
368   \PE@SanitizeSpaceOther#1%
369   \PE@EscapeHex#1%
370 }%

\edefUnescapeHex
371 \def\edefUnescapeHex#1#2{%
372   \edefSanitize#1{#2}%
373   \PE@UnescapeHex#1%
374 }%

\edefEscapeName
375 \long\def\edefEscapeName#1#2{%
376   \edefSanitize#1{#2}%
377   \PE@SanitizeSpaceOther#1%
378   \PE@EscapeName#1%
379 }%

\edefEscapeString
380 \long\def\edefEscapeString#1#2{%
381   \edefSanitize#1{#2}%
382   \PE@SanitizeSpaceOther#1%
383   \PE@EscapeString#1%
384 }%

385 \else

\PE@edefbabel Help macro that adds support for babel's shorthand characters.
386 \long\def\PE@edefbabel#1#2#3{%
387   \begingroup
388     \csname @save@activetrue\endcsname
389     \edef#1{#2{#3}}%
390   \expandafter\endgroup
391   \expandafter\def\expandafter#1\expandafter{#1}%
392 }%

\edefEscapeHex
393 \long\def\edefEscapeHex#1#2{%
394   \PE@edefbabel#1\pdfescapehex{#2}%
395 }%

\edefUnescapeHex
396 \def\edefUnescapeHex#1#2{%
397   \PE@edefbabel#1\pdfunescapehex{#2}%
398 }%

\edefEscapeName
399 \long\def\edefEscapeName#1#2{%
400   \PE@edefbabel#1\pdfescapename{#2}%
401 }%

\edefEscapeString
402 \long\def\edefEscapeString#1#2{%
403   \PE@edefbabel#1\pdfescapestring{#2}%
404 }%

```

```

405 \PE@AtEnd
406 \expandafter\endinput
407 \fi

```

## 2.7 Help macros

### 2.7.1 Characters

Special characters with catcode 12 (other) are created and stored in macros.

```

\PE@hash
408 \edef\PE@hash{\string#}

\PE@backslash
409 \begingroup
410 \escapechar=-1 %
411 \edef\x{\endgroup
412 \def\noexpand\PE@backslash{\string\\}%
413 }
414 \x

```

### 2.7.2 Switch for $\varepsilon$ -TeX

```

415 \newif\ifPE@etex
416 \begingroup\expandafter\expandafter\expandafter\endgroup
417 \expandafter\ifx\csname numexpr\endcsname\relax
418 \else
419 \PE@etextrue
420 \fi

```

## 2.8 Conversions

### 2.8.1 Conversion to hex string

```

\PE@EscapeHex
421 \ifPE@etex
422 \def\PE@EscapeHex#1{%
423 \edef#1{\expandafter\PE@ToHex#1\relax}%
424 }%
425 \else
426 \def\PE@EscapeHex#1{%
427 \def\PE@result{%
428 \expandafter\PE@ToHex#1\relax
429 \let#1\PE@result
430 }%
431 \fi

\PE@ToHex
432 \def\PE@ToHex#1{%
433 \ifx\relax#1%
434 \else
435 \PE@HexChar{#1}%
436 \expandafter\PE@ToHex
437 \fi
438 }%

\PE@HexChar
439 \ifPE@etex
440 \def\PE@HexChar#1{%
441 \PE@HexDigit{\numexpr\dimexpr.0625\dimexpr'#1sp\relax\relax\relax}%
442 \PE@HexDigit{%
443 \numexpr'#1-16*\dimexpr.0625\dimexpr'#1sp\relax\relax\relax

```

```

444 }%
445 }%
446 \else
447 \def\PE@HexChar#1{%
448 \dimen0='#1sp%
449 \dimen2=.0625\dimen0 %
450 \advance\dimen0-16\dimen2 %
451 \edef\PE@result{%
452 \PE@result
453 \PE@HexDigit{\dimen2 }%
454 \PE@HexDigit{\dimen0 }%
455 }%
456 }%
457 \fi

```

\PE@HexDigit

```

458 \def\PE@HexDigit#1{%
459 \expandafter\string
460 \ifcase#1%
461 0\or 1\or 2\or 3\or 4\or 5\or 6\or 7\or 8\or 9\or
462 A\or B\or C\or D\or E\or F%
463 \fi
464 }

```

## 2.8.2 Character code to octal number

\PE@OctChar

```

465 \ifPE@etex
466 \def\PE@OctChar#1{%
467 \expandafter\PE@@OctChar
468 \the\numexpr\dimexpr.015625\dimexpr'#1sp\relax\relax
469 \expandafter\relax
470 \expandafter\relax
471 \the\numexpr\dimexpr.125\dimexpr'#1sp\relax\relax\relax
472 \relax
473 #1%
474 }%
475 \def\PE@@OctChar#1\relax#2\relax#3{%
476 \PE@backslash
477 #1%
478 \the\numexpr#2-8*#1\relax
479 \the\numexpr\dimexpr'#3sp\relax-8*#2\relax
480 }%
481 \else
482 \def\PE@OctChar#1{%
483 \dimen0='#1sp%
484 \dimen2=.125\dimen0 %
485 \dimen4=.125\dimen2 %
486 \advance\dimen0-8\dimen2 %
487 \advance\dimen2-8\dimen4 %
488 \edef\PE@result{%
489 \PE@result
490 \PE@backslash
491 \number\dimen4 %
492 \number\dimen2 %
493 \number\dimen0 %
494 }%
495 }%
496 \fi

```

### 2.8.3 Unpack hex string

\PE@UnescapeHex

```
497 \def\PE@UnescapeHex#1{%
498   \begingroup
499     \PE@InitUccodeHexDigit
500     \def\PE@result{%
501       \expandafter\PE@DeHex#1\relax\relax
502     \expandafter\endgroup
503   \expandafter\def\expandafter#1\expandafter{\PE@result}%
504 }
```

\PE@DeHex

```
505 \def\PE@DeHex#1#2{%
506   \ifx#2\relax
507     \ifx#1\relax
508       \let\PE@next\relax
509     \else
510       \uppercase{%
511         \def\PE@testA{#1}%
512       }%
513       \ifcase\expandafter\PE@TestUcHexDigit\PE@testA
514         \def\PE@next{%
515           \PE@DeHex#10\relax\relax
516         }%
517       \else
518         \let\PE@next\relax
519       \fi
520     \fi
521   \else
522     \uppercase{%
523       \def\PE@testA{#1}%
524       \def\PE@testB{#2}%
525     }%
526     \ifcase\expandafter\PE@TestUcHexDigit\PE@testA
527       \ifcase\expandafter\PE@TestUcHexDigit\PE@testB
528         \uccode\z@="\PE@testA\PE@testB\relax
529         \ifnum\uccode\z@=32 %
530           \let\PE@temp\PE@space@space
531         \else
532           \uppercase{%
533             \def\PE@temp{^^@}%
534           }%
535         \fi
536         \edef\PE@result{\PE@result\PE@temp}%
537         \let\PE@next\PE@DeHex
538       \else
539         % invalid input sequence
540         \def\PE@next{%
541           \PE@DeHex#1%
542         }%
543       \fi
544     \else
545       % invalid input sequence
546       \def\PE@next{\PE@DeHex#2}%
547     \fi
548   \fi
549   \PE@next
550 }
```

### 2.8.4 Conversion to PDF name

\PE@EscapeName

```

551 \ifPE@etex
552   \def\PE@EscapeName#1{%
553     \edef#1{\expandafter\PE@EscapeNameTokens#1\relax}%
554   }%
555 \else
556   \def\PE@EscapeName#1{%
557     \def\PE@result{%
558       \expandafter\PE@EscapeNameTokens#1\relax
559       \let#1\PE@result
560     }%
561 \fi

```

```
\PE@EscapeNameTokens
```

```

562 \def\PE@EscapeNameTokens#1{%
563   \ifx\relax#1%
564   \else
565     \ifnum'#1<33 %
566       \ifcase'#1 %
567         % drop illegal zero
568       \else
569         \PE@EscapeNameAdd\PE@hash
570         \PE@HexChar#1%
571       \fi
572     \else
573       \ifnum'#1>126 %
574         \PE@EscapeNameAdd\PE@hash
575         \PE@HexChar#1%
576       \else \ifnum'#1=35 \PE@EscapeNameHashChar 23% #
577         \else\ifnum'#1=37 \PE@EscapeNameHashChar 25% %
578         \else\ifnum'#1=40 \PE@EscapeNameHashChar 28% (
579         \else\ifnum'#1=41 \PE@EscapeNameHashChar 29% )
580         \else\ifnum'#1=47 \PE@EscapeNameHashChar 2F% /
581         \else\ifnum'#1=60 \PE@EscapeNameHashChar 3C% <
582         \else\ifnum'#1=62 \PE@EscapeNameHashChar 3E% >
583         \else\ifnum'#1=91 \PE@EscapeNameHashChar 5B% [
584         \else\ifnum'#1=93 \PE@EscapeNameHashChar 5D% ]
585         \else\ifnum'#1=123 \PE@EscapeNameHashChar 7B% {
586         \else\ifnum'#1=125 \PE@EscapeNameHashChar 7D% }
587       \else
588         \PE@EscapeNameAdd{#1}%
589       \fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi
590     \fi
591   \fi
592   \expandafter\PE@EscapeNameTokens
593 \fi
594 }%
595 \def\PE@EscapeNameHashChar#1#2{%
596   \PE@EscapeNameAdd{\PE@hash\string#1\string#2}%
597 }%

```

```
\PE@EscapeNameAdd
```

```

598 \ifPE@etex
599   \def\PE@EscapeNameAdd#1#{#1}%
600 \else
601   \def\PE@EscapeNameAdd#1{%
602     \edef\PE@result{%
603       \PE@result
604       #1%
605     }%

```

```

606 }%
607 \fi

```

### 2.8.5 Conversion to PDF string

`\PE@EscapeString`

```

608 \ifPE@etex
609 \def\PE@EscapeString#1{%
610 \edef#1{\expandafter\PE@EscapeStringTokens#1\relax}%
611 }%
612 \else
613 \def\PE@EscapeString#1{%
614 \begingroup
615 \def\PE@result{}%
616 \expandafter\PE@EscapeStringTokens#1\relax
617 \expandafter\endgroup
618 \expandafter\def\expandafter#1\expandafter{\PE@result}%
619 }%
620 \fi

```

`\PE@EscapeStringTokens`

```

621 \def\PE@EscapeStringTokens#1{%
622 \ifx\relax#1%
623 \else
624 \ifnum'#1<33 %
625 \PE@OctChar#1%
626 \else
627 \ifnum'#1>126 %
628 \PE@OctChar#1%
629 \else \ifnum'#1=40 \PE@EscapeStringAdd{\string\}% (
630 \else\ifnum'#1=41 \PE@EscapeStringAdd{\string\)}% )
631 \else\ifnum'#1=92 \PE@EscapeStringAdd{\string\\}% \
632 \else
633 \PE@EscapeStringAdd{#1}%
634 \fi\fi\fi
635 \fi
636 \fi
637 \expandafter\PE@EscapeStringTokens
638 \fi
639 }%

```

`\PE@EscapeStringAdd`

```

640 \ifPE@etex
641 \def\PE@EscapeStringAdd#1{#1}%
642 \else
643 \def\PE@EscapeStringAdd#1{%
644 \edef\PE@result{%
645 \PE@result
646 #1%
647 }%
648 }%
649 \fi

650 \PE@AtEnd
651 \</package>

```

## 3 Test

### 3.1 Catcode checks for loading

```

652 \<*test1>

```

```

653 \catcode'\@=11 %
654 \def\RestoreCatcodes{}
655 \count@=0 %
656 \loop
657   \edef\RestoreCatcodes{%
658     \RestoreCatcodes
659     \catcode\the\count@=\the\catcode\count@\relax
660   }%
661 \ifnum\count@<255 %
662   \advance\count@\@ne
663 \repeat
664
665 \def\RangeCatcodeInvalid#1#2{%
666   \count@=#1\relax
667   \loop
668     \catcode\count@=15 %
669     \ifnum\count@<#2\relax
670       \advance\count@\@ne
671     \repeat
672 }
673 \def\Test{%
674   \RangeCatcodeInvalid{0}{47}%
675   \RangeCatcodeInvalid{58}{64}%
676   \RangeCatcodeInvalid{91}{96}%
677   \RangeCatcodeInvalid{123}{255}%
678   \catcode'\@=12 %
679   \catcode'\=0 %
680   \catcode'\{=1 %
681   \catcode'\}=2 %
682   \catcode'\#=6 %
683   \catcode'\[=12 %
684   \catcode'\]=12 %
685   \catcode'\%=14 %
686   \catcode'\ =10 %
687   \catcode13=5 %
688   \input pdfescape.sty\relax
689   \RestoreCatcodes
690 }
691 \Test
692 \csname @@end\endcsname
693 \end
694 </test1>

```

### 3.2 Macro tests

```

695 <*test2 | test3 | test4>
696 \NeedsTeXFormat{LaTeX2e}
697 \makeatletter

```

### 3.3 Test with \pdfescape... commands

```

698 <*test2>
699 \ProvidesFile{pdfescape-test1.tex}%
700   [2007/10/27 v1.7 Test with \string\pdfescape... commands]%
701 </test2>

```

### 3.4 Test without \pdfescape..., with $\varepsilon$ -TeX

```

702 <*test3>
703 \ProvidesFile{pdfescape-test2.tex}%
704   [2007/10/27 v1.7 Test without \string\pdfescape..., with  $\varepsilon$ -TeX]%
705 </test3>

```

### 3.5 Test without \pdfescape... and $\varepsilon$ -TeX



```

706 <*test4>
707 \ProvidesFile{pdfescape-test3.tex}%
708 [2007/10/27 v1.7 Test without \string\pdfescape... and e-TeX]%
709 </test4>

```

### 3.6 Check/ensure test preconditions

#### 3.6.1 Check \pdfescape...

```

710 <*test2>
711 \@ifundefined{pdfescapehex}{%
712   \PackageError{pdfescape-test1}{%
713     Missing \string\pdfescape... commands%
714   }{Test aborted.}%
715   \stop
716 }{}
717 </test2>

718 <*test3 | test4>
719 \let\pdfescapehex\@undefined
720 \let\pdfunescapehex\@undefined
721 \let\pdfescapename\@undefined
722 \let\pdfescapestring\@undefined
723 </test3 | test4>

```

#### 3.6.2 Check $\varepsilon$ -TeX

```

724 <*test3>
725 \@ifundefined{numexpr}{%
726   \PackageError{pdfescape-test2}{%
727     Missing \eTeX
728   }{Test aborted.}%
729   \stop
730 }{}
731 </test3>

```

Package `qstest` uses  $\varepsilon$ -TeX, thus  $\varepsilon$ -TeX's features can only be disabled later during loading of package `pdfescape`.

### 3.7 Common part

The files for testing uses the framework, provided by the new package `qstest` of David Kastrup.

```

732 \RequirePackage{qstest}
733 \IncludeTests{*}
734 \LogTests{log}{*}{*}
735
736 \newcommand*{\ExpectVar}[2]{%
737   \ifx#1#2%
738   \else
739     \begingroup
740       \@onelevel@sanitize#1%
741       \@onelevel@sanitize#2%
742       \typeout{[#1] <> [#2]}% hash-ok
743     \endgroup
744   \fi
745   \Expect*{\ifx#1#2true\else false\fi}{true}%
746 }
747
748 \makeatletter
749 \begingroup
750   \gdef\AllBytes{}%
751   \count@=0 %
752   \catcode0=12 %
753   \@whilenum\count@<256 \do{%
754     \lccode0=\count@

```

```

755 \ifnum\count@=32 %
756 \xdef\AllBytes{\AllBytes\space}%
757 \else
758 \lowercase{%
759 \xdef\AllBytes{\AllBytes^^@}%
760 }%
761 \fi
762 \advance\count@ by 1 %
763 }%
764 \endgroup
765 \newcommand*{\AllBytesHex}{%
766 000102030405060708090A0B0C0D0E0F%
767 101112131415161718191A1B1C1D1E1F%
768 202122232425262728292A2B2C2D2E2F%
769 303132333435363738393A3B3C3D3E3F%
770 404142434445464748494A4B4C4D4E4F%
771 505152535455565758595A5B5C5D5E5F%
772 606162636465666768696A6B6C6D6E6F%
773 707172737475767778797A7B7C7D7E7F%
774 808182838485868788898A8B8C8D8E8F%
775 909192939495969798999A9B9C9D9E9F%
776 A0A1A2A3A4A5A6A7A8A9AABACADAFAF%
777 B0B1B2B3B4B5B6B7B8B9ABBBBCDBEBF%
778 C0C1C2C3C4C5C6C7C8C9CACBCCDCECF%
779 D0D1D2D3D4D5D6D7D8D9DADBDCDDDEDF%
780 E0E1E2E3E4E5E6E7E8E9EAEBECEDEEEF%
781 F0F1F2F3F4F5F6F7F8F9FABFCFDFFEFF%
782 }
783 \@onelevel@sanitize\AllBytesHex
784 \expandafter\lowercase\expandafter{%
785 \expandafter\newcommand\expandafter*\expandafter\AllBytesHexLC
786 \expandafter{\AllBytesHex}%
787 }
788 \newcommand*{\AllBytesName}{%
789 \begingroup
790 \catcode'\# =12 %
791 \xdef\AllBytesName{%
792 #01#02#03#04#05#06#07#08#09#0A#0B#0C#0D#0E#0F%
793 #10#11#12#13#14#15#16#17#18#19#1A#1B#1C#1D#1E#1F%
794 #20!"#23$#25&'#28#29*+,-.#2F%
795 0123456789:;#3C=#3E?%
796 @ABCDEFGHIJKLMNO%
797 PQRSTUVWXYZ#5B\@backslashchar#5D^_%
798 'abcdefghijklmnopqrstuvwxyz%
799 pqrstuvwxyz#7B|#7D\string~#7F%
800 #80#81#82#83#84#85#86#87#88#89#8A#8B#8C#8D#8E#8F%
801 #90#91#92#93#94#95#96#97#98#99#9A#9B#9C#9D#9E#9F%
802 #A0#A1#A2#A3#A4#A5#A6#A7#A8#A9#AA#AB#AC#AD#AE#AF%
803 #B0#B1#B2#B3#B4#B5#B6#B7#B8#B9#BA#BB#BC#BD#BE#BF%
804 #C0#C1#C2#C3#C4#C5#C6#C7#C8#C9#CA#CB#CC#CD#CE#CF%
805 #D0#D1#D2#D3#D4#D5#D6#D7#D8#D9#DA#DB#DC#DD#DE#DF%
806 #E0#E1#E2#E3#E4#E5#E6#E7#E8#E9#EA#EB#EC#ED#EE#EF%
807 #F0#F1#F2#F3#F4#F5#F6#F7#F8#F9#FA#FB#FC#FD#FE#FF%
808 }%
809 \endgroup
810 \@onelevel@sanitize\AllBytesName
811
812 \newcommand*{\AllBytesString}{%
813 \begingroup
814 \def\{|}%
815 \edef\%{\@percentchar}%
816 \catcode'\|=0 %

```

```

817 \catcode'\#=12 %
818 \catcode'\~=12 %
819 \catcode'\|=12 %
820 |xdef|AllBytesString{%
821 \000\001\002\003\004\005\006\007\010\011\012\013\014\015\016\017%
822 \020\021\022\023\024\025\026\027\030\031\032\033\034\035\036\037%
823 \040!"#$%&'(\)*+,-./%
824 0123456789:;<=>?%
825 @ABCDEFGHIJKLMNO%
826 PQRSTUVWXYZ[\]^_%
827 'abcdefghijklmno%
828 pqrstuvwxyz{|}~\177%
829 \200\201\202\203\204\205\206\207\210\211\212\213\214\215\216\217%
830 \220\221\222\223\224\225\226\227\230\231\232\233\234\235\236\237%
831 \240\241\242\243\244\245\246\247\250\251\252\253\254\255\256\257%
832 \260\261\262\263\264\265\266\267\270\271\272\273\274\275\276\277%
833 \300\301\302\303\304\305\306\307\310\311\312\313\314\315\316\317%
834 \320\321\322\323\324\325\326\327\330\331\332\333\334\335\336\337%
835 \340\341\342\343\344\345\346\347\350\351\352\353\354\355\356\357%
836 \360\361\362\363\364\365\366\367\370\371\372\373\374\375\376\377%
837 }%
838 |endgroup
839 \@onelevel@sanitize\AllBytesString
840
841 <*test4>
842 \let\org@detokenize\detokenize
843 \let\detokenize\@undefined
844 \let\org@numexpr\numexpr
845 \let\numexpr\@undefined
846 </test4>
847 \RequirePackage{pdfescape}
848 <*test4>
849 \let\detokenize\org@detokenize
850 \let\numexpr\org@numexpr
851 </test4>
852
853 \begin{qstest}{all-hex}{\AllBytes, escapehex}
854 \EdefEscapeHex\x{\AllBytes}%
855 \Expect*{\x}*{\AllBytesHex}%
856 \ExpectVar\x\AllBytesHex
857 \end{qstest}
858
859 \begin{qstest}{all-unhex}{\AllBytesHex, unescapehex}
860 \EdefUnescapeHex\x{\AllBytesHex}%
861 \Expect*{\x}*{\AllBytes}%
862 \ExpectVar\x\AllBytes
863 \end{qstest}
864
865 \begin{qstest}{all-unhex-lc}{\AllBytesHexLC, unescapehex, lowercase}
866 \EdefUnescapeHex\x{\AllBytesHexLC}%
867 \Expect*{\x}*{\AllBytes}%
868 \ExpectVar\x\AllBytes
869 \end{qstest}
870
871 \begin{qstest}{unhex-incomplete}{unescapehex, incomplete}
872 \EdefUnescapeHex\x{4}%
873 \Expect*{\x}{@}%
874 \end{qstest}
875
876 \begin{qstest}{unhex-space}{unescapehex, space}
877 \EdefUnescapeHex\x{20}%
878 \Expect*{\x}{ }%

```

```

879 \ExpectVar\x\space
880 \end{qstest}
881
882 \begin{qstest}{unhex-spaces}{unescapehex, spaces}
883 \EdefUnescapeHex\x{204020204120}%
884 \def\y#1{%
885   \edef\z{#1\string @#1#1\string A#1}%
886 } \y{ }%
887 \Expect*{\x}*{\z}%
888 \ExpectVar\x\z
889 \end{qstest}
890
891 \begin{qstest}{unhex-hash}{unescapehex, hash}
892 \catcode'\#=12 %
893 \EdefUnescapeHex\x{#20}%
894 \ExpectVar\x\space
895 \end{qstest}
896
897 \begin{qstest}{unhex-invalid}{unescapehex, invalid}
898 \def\test#1#2{%
899   \EdefUnescapeHex\x{#1}%
900   \edef\y{#2}%
901   \@onelevel@sanitize\y
902   \ExpectVar\x\y
903 }%
904 \test{ }
905 \edef\x{\pdfunescapehex{4X}}%
906 \edef\y{\string @}%
907 \ifx\x\y
908 \else
909   \def~{\space}%
910   \typeout{*****}%
911   \typeout{* Your pdfTeX contains bug 777.~~~*}%
912   \typeout{* This test is redefined as dummy, *}%
913   \typeout{* because it triggers the bug.~~~*}%
914   \typeout{*****}%
915   \def\test#1#2{%
916     \fi
917 }
918 \test{X}{}%
919 \test{XY}{}%
920 \test{XYZ}{}%
921 \test{A}{~^a0}%
922 \test{AX}{~^a0}%
923 \test{XA}{~^a0}%
924 \test{XXAXX}{~^a0}%
925 \end{qstest}
926
927 \begin{qstest}{all-name}{\AllBytes, escapename}
928 \EdefEscapeName\x{\AllBytes}%
929 \Expect*{\x}*{\AllBytesName}%
930 \ExpectVar\x\AllBytesName
931 \end{qstest}
932
933 \begin{qstest}{all-string}{\AllBytes, escapestring}
934 \EdefEscapeString\x{\AllBytes}%
935 \Expect*{\x}*{\AllBytesString}%
936 \ExpectVar\x\AllBytesString
937 \end{qstest}
938
939 \begin{qstest}{uchexdigit}{unescape, uppercase hex digit}
940 \catcode'\@=11 %

```

```

941 \catcode0=12 %
942 \def\test#1#2{%
943   \uccode0=#1\relax
944   \uppercase{%
945     \def\x{^^@}%
946   }%
947   \Expect*{%
948     \ifcase\expandafter\PE@TestUcHexDigit\x
949     true%
950   \else
951     false%
952   \fi
953 }{#2}%
954 }%
955 \def\range#1#2#3{%
956   \count0=#1\relax
957   \loop
958   \ifnum\count0<#2\relax
959     \test{\count0}{#3}%
960     \advance\count0 by 1 %
961   \repeat
962 }%
963 \range{0}{47}{false}%
964 \range{48}{57}{true}%
965 \range{58}{64}{false}%
966 \range{65}{70}{true}%
967 \range{71}{255}{false}%
968 \end{qstest}
969
970 \begin{qstest}{unescapename}{unescapename}
971   \def\test#1#2{%
972     \EdefUnescapeName\x{#1}%
973     \edef\y{#2}%
974     \@onelevel@sanitize\y
975     \ExpectVar\x\y
976   }%
977   \catcode'\#=12 %
978   \catcode0=12 %
979   \test{}{}%
980   \test{x}{x}%
981   \test{xy}{xy}%
982   \test{#}{#}%
983   \test{##}{##}%
984   \test{###}{###}%
985   \test{####}{####}%
986   \test{#x}{#x}%
987   \test{#xy}{#xy}%
988   \test{#1}{#1}%
989   \test{#40}{@}%
990   \test{#400}{@0}%
991   \test{#4x0}{#4x0}%
992   \test{#ab}{^^ab}%
993   \test{#00}{^^@}%
994   \test{x#40y#40z}{x@y@z}%
995   \test{#40#40#40#40}{@@@@}%
996   \test{a#x}{a#x}%
997   \test{a#xy}{a#xy}%
998   \test{a#1}{a#1}%
999   \test{a#40}{a@}%
1000  \test{a#400}{a@0}%
1001  \test{#20}{ }%
1002  \test{a#20}{a }%

```

```

1003 \test{a#20b}{a b}%
1004 \test{a#20#20#20b}{a \space\space b}%
1005 \end{qstest}
1006
1007 \begin{qstest}{unescapestring}{unescapestring}
1008 \def\test#1#2{%
1009     \EdefUnescapeString\x{#1}%
1010     \edef\y{#2}%
1011     \@onelevel@sanitize\y
1012     \ExpectVar\x\y
1013 }%
1014 \catcode0=12 %
1015 \def\DefChar#1#2{%
1016     \begingroup
1017         \uccode0=#2\relax
1018     \uppercase{\endgroup
1019     \def#1{^^@}%
1020     }%
1021 }%
1022 \DefChar\nul{0}%
1023 \DefChar\one{1}%
1024 \DefChar\bel{8}%
1025 \DefChar\tab{9}%
1026 \DefChar\lf{10}%
1027 \DefChar\ff{12}%
1028 \DefChar\cr{13}%
1029 \DefChar\{\{92}%
1030 \test{}{}%
1031 \test{a}{a}%
1032 \test{\}\}%
1033 \test{\}\}\}%
1034 \test{\}\}\y}{\y}%
1035 \test{\000}{\nul}%
1036 \test{\b}{\bel}%
1037 \test{\t}{\tab}%
1038 \test{\n}{\lf}%
1039 \test{\f}{\ff}%
1040 \test{\r}{\cr}%
1041 \test{\{(){}%
1042 \test{\}\})}%
1043 \test{\040}{ }%
1044 \test{\100}{@}%
1045 \test{\40}{ }%
1046 \test{\1}{\one}%
1047 \test{\01}{\one}%
1048 \test{\001}{\one}%
1049 \test{\18}{\one8}%
1050 \test{\018}{\one8}%
1051 \test{\0018}{\one8}%
1052 \test{x\}{x}%
1053 \test{x\}\}\}{x\}%
1054 \test{x\}\}\y}{x\y}%
1055 \test{x\000}{x\nul}%
1056 \test{x\b}{x\bel}%
1057 \test{x\t}{x\tab}%
1058 \test{x\n}{x\lf}%
1059 \test{x\f}{x\ff}%
1060 \test{x\r}{x\cr}%
1061 \test{x\{(){x}%
1062 \test{x\}\}){x}%
1063 \test{x\040}{x }%
1064 \test{x\100}{x@}%

```

```

1065 \test{x\\40}{x }%
1066 \test{x\\1}{x\one}%
1067 \test{x\\01}{x\one}%
1068 \test{x\\001}{x\one}%
1069 \test{x\\18}{x\one8}%
1070 \test{x\\018}{x\one8}%
1071 \test{x\\0018}{x\one8}%
1072 \test{\\b\\t\\n\\f\\r\\(\\)\\000\\040}{%
1073 \bel\tab\lf\ff\cr()\\nul\space
1074 }%
1075 \test{\\lf}{}%
1076 \test{x\\lf}{x}%
1077 \test{\cr}{\lf}%
1078 \test{\cr\lf}{\lf}%
1079 \test{\lf}{\lf}%
1080 \test{\lf\cr}{\lf\lf}%
1081 \test{x\cr}{x\lf}%
1082 \test{x\cr\lf}{x\lf}%
1083 \test{x\lf}{x\lf}%
1084 \test{x\lf\cr}{x\lf\lf}%
1085 \test{x\\cr\lf y\cr}{xy\lf}%
1086 \end{qstest}
1087 \stop
1088 </test2 | test3 | test4>

```

## 4 Installation

### 4.1 Download

**Package.** This package is available on CTAN<sup>1</sup>:

[CTAN:macros/latex/contrib/oberdiek/pdfescape.dtx](#) The source file.

[CTAN:macros/latex/contrib/oberdiek/pdfescape.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:macros/latex/contrib/oberdiek/oberdiek-tds.zip](#)

*TDS* refers to the standard “A Directory Structure for T<sub>E</sub>X 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/
```

---

<sup>1</sup><http://ftp.ctan.org/tex-archive/>

### 4.3 Package installation

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

```
tex pdfescape.dtx
```

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

```
pdfescape.sty      → tex/generic/oberdiek/pdfescape.sty
pdfescape.pdf      → doc/latex/oberdiek/pdfescape.pdf
test/pdfescape-test1.tex → doc/latex/oberdiek/test/pdfescape-test1.tex
test/pdfescape-test2.tex → doc/latex/oberdiek/test/pdfescape-test2.tex
test/pdfescape-test3.tex → doc/latex/oberdiek/test/pdfescape-test3.tex
test/pdfescape-test4.tex → doc/latex/oberdiek/test/pdfescape-test4.tex
pdfescape.dtx      → source/latex/oberdiek/pdfescape.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  $\text{\TeX}$  distribution (`te $\text{\TeX}$` , `mik $\text{\TeX}$` , ...) relies on file name databases, you must refresh these. For example, `te $\text{\TeX}$`  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 pdfescape.pdf unpack_files output .
```

**Unpacking with  $\text{\LaTeX}$ .** The `.dtx` chooses its action depending on the format:

**plain- $\text{\TeX}$ :** Run `docstrip` and extract the files.

**$\text{\LaTeX}$ :** Generate the documentation.

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

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

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

**Generating the documentation.** You can use both the `.dtx` or the `.drv` to generate the documentation. The process can be configured by the configuration file `ltxdoc.cfg`. For instance, put this line into this file, if you want to have A4 as paper format:

```
\PassOptionsToClass{a4paper}{article}
```

An example follows how to generate the documentation with `pdf $\text{\LaTeX}$` :

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



## 5 History

[2007/02/21 v1.0]

- First version.

[2007/02/25 v1.1]

- Test files added.
- `\EdefUnescapeHex` supports lowercase letters.
- Fix: `\EdefEscapeName{^^@}`
- Fix: `\EdefEscapeName{\string#}`
- Fix for `\EdefUnescapeHex` in case of incomplete hex string.
- Fix: `\EdefUnescapeHex` generates space tokens with catcode 10 (space) in all cases.
- Fix: `\EdefEscapeHex` and `\EdefEscapeName` now generate tokens with catcode 12 (other) only.

[2007/03/20 v1.2]

- Fix: Wrong year in `\ProvidesPackage`.

[2007/04/11 v1.3]

- Line ends sanitized.

[2007/04/21 v1.4]

- `\EdefUnescapeName` and `\EdefUnescapeString` added.

[2007/08/27 v1.5]

- `\EdefSanitize` added (replaces `\PE@sanitize`).

[2007/09/09 v1.6]

- Fix in catcode setup.

[2007/10/27 v1.7]

- More efficient `\EdefSanitize`.

## 6 Index

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

Symbols		
<code>\#</code>	<i>140, 276, 301, 682, 790, 817, 892, 977</i>	<code>\@ReturnAfterFi</code> ..... <i>126, <u>131</u></i>
<code>\\$</code>	..... <i>139</i>	<code>\@backslashchar</code> ..... <i>797</i>
<code>\%</code>	..... <i>685, 815</i>	<code>\@ifundefined</code> ..... <i>711, 725</i>
<code>\(</code>	..... <i>629, 823</i>	<code>\@ne</code> ..... <i>167, 170,</i> <i>215, 218, 222, 332, 335, 662, 670</i>
<code>\)</code>	..... <i>630, 823</i>	<code>\@onelevel@sanitize</code> ..... <i>97, 740,</i> <i>741, 783, 810, 839, 901, 974, 1011</i>
<code>\@</code>	..... <i>653, 678, 940</i>	

<code>\@percentchar</code> . . . . .	815		
<code>\@undefined</code> 719, 720, 721, 722, 843, 845			
<code>\@whilenum</code> . . . . .	753		
<code>\[</code> . . . . .	683		
<code>\</code> 123, 268, 412, 631, 679, 819, 826, 1029, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063, 1064, 1065, 1066, 1067, 1068, 1069, 1070, 1071, 1072, 1073, 1075, 1076, 1085			
<code>\{</code> . . . . .	680		
<code>\}</code> . . . . .	325, 681		
<code>\]</code> . . . . .	684		
<code>\ </code> . . . . .	267, 272, 814, 816		
<code>\~</code> . . . . .	818		
<b>Numbers</b>			
<code>\0</code> . . . . .	821, 822, 823		
<code>\1</code> . . . . .	828		
<code>\2</code> . . . . .	829, 830, 831, 832		
<code>\3</code> . . . . .	833, 834, 835, 836		
<code>\8</code> . . . . .	240		
<code>\9</code> . . . . .	241		
<code>\_</code> . . . . .	115, 631, 686		
<b>A</b>			
<code>\advance</code> 450, 486, 487, 662, 670, 762, 960			
<code>\AllBytes</code> 750, 756, 759, 853, 854, 861, 862, 867, 868, 927, 928, 933, 934			
<code>\AllBytesHex</code> . . . . .	765, 783, 786, 855, 856, 859, 860		
<code>\AllBytesHexLC</code> . . . . .	785, 865, 866		
<code>\AllBytesName</code> . . . . .	788, 791, 810, 929, 930		
<code>\AllBytesString</code> . . . . .	812, 839, 935, 936		
<b>B</b>			
<code>\begin</code> . . . . .	853, 859, 865, 871, 876, 882, 891, 897, 927, 933, 939, 970, 1007		
<code>\bel</code> . . . . .	1024, 1036, 1056, 1073		
<b>C</b>			
<code>\catcode</code> . . . . .	3, 4, 5, 6, 7, 31, 32, 33, 34, 35, 36, 37, 38, 56, 58, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 115, 139, 140, 267, 268, 653, 659, 668, 678, 679, 680, 681, 682, 683, 684, 685, 686, 687, 752, 790, 816, 817, 818, 819, 892, 940, 941, 977, 978, 1014		
<code>\count</code> . . . . .	956, 958, 959, 960		
<code>\count@</code> 655, 659, 661, 662, 666, 668, 669, 670, 751, 753, 754, 755, 762			
<code>\cr</code> . . . . .	1028, 1040, 1060, 1073, 1077, 1078, 1080, 1081, 1082, 1084, 1085		
<code>\csname</code> . . . . .	8, 18, 39, 52, 55, 81, 84, 91, 102, 365, 388, 417, 692		
<b>D</b>			
<code>\DefChar</code> . . . . .	1015, 1022, 1023, 1024, 1025, 1026, 1027, 1028, 1029		
<code>\detokenize</code> . . . . .	106, 110, 842, 843, 849		
<code>\dimen</code> . . . . .	448, 449, 450, 453, 454, 483, 484, 485, 486, 487, 491, 492, 493		
<code>\dimexpr</code> . . . . .	441, 443, 468, 471, 479		
<code>\do</code> . . . . .	753		
<b>E</b>			
<code>\EdefEscapeHex</code> . . . . .	2, 366, 393, 854		
<code>\EdefEscapeName</code> . . . . .	375, 399, 928		
<code>\EdefEscapeString</code> . . . . .	380, 402, 934		
<code>\EdefSanitize</code> . . . . .	2, 80, 113, 133, 233, 367, 372, 376, 381		
<code>\EdefUnescapeHex</code> . . . . .	371, 396, 860, 866, 872, 877, 883, 893, 899		
<code>\EdefUnescapeName</code> . . . . .	2, 132, 972		
<code>\EdefUnescapeString</code> . . . . .	2, 232, 1009		
<code>\empty</code> . . . . .	12		
<code>\end</code> 693, 857, 863, 869, 874, 880, 889, 895, 925, 931, 937, 968, 1005, 1086			
<code>\endcsname</code> . . . . .	8, 18, 39, 52, 55, 81, 84, 91, 102, 365, 388, 417, 692		
<code>\endinput</code> . . . . .	27, 406		
<code>\escapechar</code> . . . . .	410		
<code>\eTeX</code> . . . . .	727		
<code>\Expect</code> . . . . .	745, 855, 861, 867, 873, 878, 887, 929, 935, 947		
<code>\ExpectVar</code> 736, 856, 862, 868, 879, 888, 894, 902, 930, 936, 975, 1012			
<b>F</b>			
<code>\ff</code> . . . . .	1027, 1039, 1059, 1073		
<b>G</b>			
<code>\gdef</code> . . . . .	141, 149, 750		
<b>I</b>			
<code>\ifcase</code> . . . . .	9, 163, 164, 342, 350, 460, 513, 526, 527, 566, 948		
<code>\ifnum</code> . . . . .	214, 217, 220, 221, 331, 334, 529, 565, 573, 576, 577, 578, 579, 580, 581, 582, 583, 584, 585, 586, 624, 627, 629, 630, 631, 661, 669, 755, 958		
<code>\ifPE@etex</code> . . . . .	415, 421, 439, 465, 551, 598, 608, 640		
<code>\ifx</code> . . . . .	10, 12, 18, 39, 47, 81, 91, 123, 150, 154, 249, 254, 365, 417, 433, 506, 507, 563, 622, 737, 745, 907		
<code>\immediate</code> . . . . .	20, 41		
<code>\IncludeTests</code> . . . . .	733		
<code>\input</code> . . . . .	688		
<b>L</b>			
<code>\lccode</code> . . . . .	754		
<code>\lf</code> . . . . .	1026, 1038, 1058, 1073, 1075, 1076, 1077, 1078, 1079, 1080, 1081, 1082, 1083, 1084, 1085		
<code>\LogTests</code> . . . . .	734		
<code>\loop</code> . . . . .	656, 667, 957		
<code>\lowercase</code> . . . . .	758, 784		

<b>M</b>	
\makeatletter	697, 748
\meaning	93
<b>N</b>	
\NeedsTeXFormat	696
\newcommand	736, 765, 785, 788, 812
\newif	415
\nul	1022, 1035, 1055, 1073
\number	491, 492, 493
\numexpr	441, 443, 468, 471, 478, 479, 844, 845, 850
<b>O</b>	
\one	1023, 1046, 1047, 1048, 1049, 1050, 1051, 1066, 1067, 1068, 1069, 1070, 1071
\org@detokenize	842, 849
\org@numexpr	844, 850
<b>P</b>	
\PackageError	712, 726
\PackageInfo	23
\pdfescape	700, 704, 708, 713
\pdfescapehex	394, 719
\pdfescapename	400, 721
\pdfescapestring	403, 722
\pdfunescapehex	397, 720, 905
\PE@@NormalizeLineEnd	245, 248
\PE@@OctChar	467, 475
\PE@AtEnd	60, 61, 405, 650
\PE@backslash	409, 476, 490
\PE@CheckEndBackslash	322
\PE@DeHex	501, 505
\PE@DeName	145, 149, 178, 182
\PE@DeString	276, 362
\PE@edefbabel	386, 394, 397, 400, 403
\PE@EscapeHex	369, 421
\PE@EscapeName	378, 551
\PE@EscapeNameAdd	569, 574, 588, 596, 598
\PE@EscapeNameHashChar	576, 577, 578, 579, 580, 581, 582, 583, 584, 585, 586, 595
\PE@EscapeNameTokens	553, 558, 562
\PE@EscapeString	383, 608
\PE@EscapeStringAdd	629, 630, 631, 633, 640
\PE@EscapeStringTokens	610, 616, 621
\PE@etexttrue	419
\PE@hash	408, 569, 574, 596
\PE@HexChar	435, 439, 570, 575
\PE@HexDigit	441, 442, 453, 454, 458
\PE@InitUccodeHexDigit	143, 189, 499
\PE@next	152, 157, 178, 182, 186, 251, 255, 257, 260, 343, 345, 347, 351, 353, 355, 508, 514, 518, 537, 540, 546, 549
\PE@NormalizeLineEnd	235, 243
\PE@OctAll	345, 351, 353, 357
\PE@OctChar	465, 625, 628
\PE@OctI	341
\PE@OctII	343, 349
\PE@onelevel@sanitize	86, 92, 97, 109, 136, 237
\PE@result	144, 147, 151, 156, 177, 181, 244, 246, 250, 253, 360, 427, 429, 451, 452, 488, 489, 500, 503, 536, 557, 559, 602, 603, 615, 618, 644, 645
\PE@sanitize	113
\PE@SanitizeSpaceOther	118, 134, 234, 368, 377, 382
\PE@space@other	114, 125
\PE@space@space	117, 530
\PE@SpaceToOther	119, 121
\PE@strip@prefix	93, 95
\PE@temp	174, 177, 530, 533, 536
\PE@testA	160, 163, 172, 511, 513, 523, 526, 528
\PE@testB	161, 164, 172, 524, 527, 528
\PE@TestOctDigit	330, 342, 350
\PE@TestUcHexDigit	163, 164, 213, 513, 526, 527, 948
\PE@ToHex	423, 428, 432
\PE@UnescapeHex	373, 497
\PE@UnescapeName	135, 138
\PE@UnescapeString	236, 269
\ProvidesFile	699, 703, 707
\ProvidesPackage	53
<b>R</b>	
\range	955, 963, 964, 965, 966, 967
\RangeCatcodeInvalid	665, 674, 675, 676, 677
\repeat	663, 671, 961
\RequirePackage	732, 847
\RestoreCatcodes	654, 657, 658, 689
<b>S</b>	
\space	756, 879, 894, 909, 1004, 1073
\stop	715, 729, 1087
<b>T</b>	
\tab	1025, 1037, 1057, 1073
\Test	673, 691
\test	898, 915, 918, 919, 920, 921, 922, 923, 924, 942, 959, 971, 979, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000, 1001, 1002, 1003, 1004, 1008, 1030, 1031, 1032, 1033, 1034, 1035, 1036, 1037, 1038, 1039, 1040, 1041, 1042, 1043, 1044, 1045, 1046, 1047, 1048, 1049, 1050, 1051, 1052, 1053, 1054, 1055, 1056, 1057, 1058, 1059, 1060, 1061, 1062, 1063, 1064, 1065, 1066, 1067, 1068, 1069, 1070, 1071, 1072, 1075, 1076, 1077, 1078, 1079, 1080, 1081, 1082, 1083, 1084, 1085
\the	56, 62, 468, 471, 478, 479, 659

<code>\TMP@EnsureCode</code>	59, 66, 67, 68, 69,	23, 25, 40, 45, 52, 116, 242, 264,
	70, 71, 72, 73, 74, 75, 76, 77, 78, 79	411, 414, 854, 855, 856, 860,
<code>\typeout</code>	.. 742, 910, 911, 912, 913, 914	861, 862, 866, 867, 868, 872,
		873, 877, 878, 879, 883, 887,
	<b>U</b>	888, 893, 894, 899, 902, 905,
<code>\uccode</code>	..... 77, 78,	907, 928, 929, 930, 934, 935,
	79, 172, 176, 190, 191, 192, 193,	936, 945, 948, 972, 975, 1009, 1012
	194, 195, 196, 197, 198, 199,	
	200, 201, 202, 203, 204, 205,	<b>Y</b>
	206, 207, 208, 209, 210, 211,	<code>\y</code> .... 884, 886, 900, 901, 902, 906,
	240, 241, 358, 528, 529, 943, 1017	907, 973, 974, 975, 1010, 1011, 1012
<code>\uppercase</code>	..... 159, 173,	
	263, 359, 510, 522, 532, 944, 1018	<b>Z</b>
		<code>\z</code> ..... 885, 887, 888
	<b>W</b>	<code>\z@</code> ... 165, 172, 176, 196, 197, 198,
<code>\write</code>	..... 20, 41	199, 200, 201, 202, 203, 204,
		205, 206, 207, 208, 209, 210,
	<b>X</b>	211, 224, 227, 337, 358, 528, 529
<code>\x</code>	..... 8, 10, 12, 19,	