The **pdfescape** package

Heiko Oberdiek

<oberdiek@uni-freiburg.de>

2007/11/11 v1.8

Abstract

This package implements pdfTeX's escape features (`\pdfescapehex`, `\pdfunescapehex`, `\pdfescapename`, `\pdfescapestring`) using TeX or ε-TeX.

Contents

1 Documentation

\EdefEscapeHex {⟨*cmd*⟩} {⟨*string*⟩}
\EdefUnescapeHex {⟨*cmd*⟩} {⟨*string*⟩}
\EdefEscapeName {⟨*cmd*⟩} {⟨*string*⟩}
\EdefEscapeString {⟨*cmd*⟩} {⟨*string*⟩}

These commands converts ⟨*string*⟩ and stores the result in macro ⟨*cmd*⟩. The conversion result is the same as the conversion of the corresponding pdfTEX's primitives. Note that the argument ⟨*string*⟩ is expanded before the conversion.

For example, if pdfTEX >= 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 ε-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

\EdefUnescapeName {⟨*cmd*⟩} {⟨*string*⟩}

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).

\EdefUnescapeString {⟨*cmd*⟩} {⟨*string*⟩}

Macro ⟨*cmd*⟩ stores the unescaped string in ⟨*string*⟩. 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

\EdefSanitize {⟨*cmd*⟩} {⟨*string*⟩}

Argument ⟨*string*⟩ is expanded, converted to a string of tokens with catcode 12 (other) and space tokens, and stored in macro ⟨*cmd*⟩.

2 Implementation

1 ⟨∗package⟩

2.1 Reload check and package identification

Reload check, especially if the package is not used with LaTeX.

```
 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     \catcode35 6 % #
19     \catcode123 1 % {
20     \catcode125 2 % }
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{pdfescape}{The package is already loaded}%
29     \endgroup
30     \expandafter\endinput
31   \fi
32 \endgroup
```

Package identification:

```
33 \begingroup
34   \catcode35 6 % #
35   \catcode40 12 % (
36   \catcode41 12 % )
37   \catcode44 12 % ,
38   \catcode45 12 % -
39   \catcode46 12 % .
40   \catcode47 12 % /
41   \catcode58 12 % :
42   \catcode64 11 % @
43   \catcode123 1 % {
44   \catcode125 2 % }
45   \expandafter\ifx\csname ProvidesPackage\endcsname\relax
46     \def\x#1#2#3[#4]{\endgroup
47       \immediate\write-1{Package: #3 #4}%
```

```
48      \xdef#1{#4}%
49    }%
50  \else
51    \def\x#1#2[#3]{\endgroup
52      #2[{#3}]%
53      \ifx#1\relax
54        \xdef#1{#3}%
55      \fi
56    }%
57  \fi
58 \expandafter\x\csname ver@pdfescape.sty\endcsname
59 \ProvidesPackage{pdfescape}%
60   [2007/11/11 v1.8 Provides hex, PDF name and string conversions (HO)]
```

2.2 Catcodes

```
61 \expandafter\edef\csname PE@AtEnd\endcsname{%
62   \catcode64 \the\catcode64\relax
63 }
64 \catcode64 11 % @
65 \def\TMP@EnsureCode#1#2#3{%
66   \edef\PE@AtEnd{%
67     \PE@AtEnd
68     #1#2 \the#1#2\relax
69   }%
70   #1#2 #3\relax
71 }
72 \TMP@EnsureCode\catcode{0}{12}% ^^@
73 \TMP@EnsureCode\catcode{34}{12}% "
74 \TMP@EnsureCode\catcode{39}{12}% '
75 \TMP@EnsureCode\catcode{42}{12}% *
76 \TMP@EnsureCode\catcode{45}{12}% -
77 \TMP@EnsureCode\catcode{46}{12}% .
78 \TMP@EnsureCode\catcode{47}{12}% /
79 \TMP@EnsureCode\catcode{60}{12}% <
80 \TMP@EnsureCode\catcode{61}{12}% =
81 \TMP@EnsureCode\catcode{62}{12}% >
82 \TMP@EnsureCode\catcode{94}{7}% ^
83 \TMP@EnsureCode\catcode{96}{12}% `
84 \TMP@EnsureCode\uccode{34}{0}% "
85 \TMP@EnsureCode\uccode{48}{0}% 0
86 \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.

```
 87 \begingroup\expandafter\expandafter\expandafter\endgroup
 88 \expandafter\ifx\csname detokenize\endcsname\relax
 89   \long\def\EdefSanitize#1#2{%
 90     \begingroup
 91       \csname @safe@activestrue\endcsname
 92       \edef#1{#2}%
 93       \PE@onelevel@sanitize#1%
 94     \expandafter\endgroup
 95     \expandafter\def\expandafter#1\expandafter{#1}%
 96   }%
 97   \begingroup\expandafter\expandafter\expandafter\endgroup
 98   \expandafter\ifx\csname @onelevel@sanitize\endcsname\relax
 99     \def\PE@onelevel@sanitize#1{%
100       \edef#1{\expandafter\PE@strip@prefix\meaning#1}%
101     }%
102     \def\PE@strip@prefix#1>{}%
```

4

```
103   \else
104     \let\PE@onelevel@sanitize\@onelevel@sanitize
105   \fi
106 \else
107   \long\def\EdefSanitize#1#2{%
108     \begingroup
109       \csname @safe@activestrue\endcsname
110       \edef#1{#2}%
111     \expandafter\endgroup
112     \expandafter\def\expandafter#1\expandafter{%
113       \detokenize\expandafter{#1}%
114     }%
115   }%
116   \def\PE@onelevel@sanitize#1{%
117     \edef#1{\detokenize\expandafter{#1}}%
118   }%
119 \fi
```

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

```
120 \let\PE@sanitize\EdefSanitize
```

2.3.1   Space characters

\PE@space@other

```
121 \begingroup
122   \catcode`\ =12\relax%
123 \def\x{\endgroup\def\PE@space@other{ }}\x\relax
```

\PE@space@space

```
124 \def\PE@space@space{ }
```

2.3.2   Space normalization

\PE@SanitizeSpaceOther

```
125 \def\PE@SanitizeSpaceOther#1{%
126   \edef#1{\expandafter\PE@SpaceToOther#1 \relax}%
127 }
```

\PE@SpaceToOther

```
128 \def\PE@SpaceToOther#1 #2\relax{%
129   #1%
130   \ifx\\#2\\%
131   \else
132     \PE@space@other
133     \@ReturnAfterFi{%
134       \PE@SpaceToOther#2\relax
135     }%
136   \fi
137 }
```

\@ReturnAfterFi

```
138 \long\def\@ReturnAfterFi#1\fi{\fi#1}
```

2.4   \EdefUnescapeName

\EdefUnescapeName

```
139 \def\EdefUnescapeName#1#2{%
140   \EdefSanitize#1{#2}%
141   \PE@SanitizeSpaceOther#1%
```

```
142    \PE@UnescapeName#1%
143    \PE@onelevel@sanitize#1%
144 }
```

\PE@UnescapeName

```
145 \begingroup
146   \catcode`\$=6  % hash
147   \catcode`\#=12 % other
148   \gdef\PE@UnescapeName$1{%
149     \begingroup
150       \PE@InitUccodeHexDigit
151       \def\PE@result{}%
152       \expandafter\PE@DeName$1#\relax\relax
153     \expandafter\endgroup
154     \expandafter\def\expandafter$1\expandafter{\PE@result}%
155   }%
156   \gdef\PE@DeName$1#$2$3{%
157     \ifx\relax$2%
158       \edef\PE@result{\PE@result$1}%
159       \let\PE@next\relax
160     \else
161       \ifx\relax$3%
162         % wrong escape sequence in input
163         \edef\PE@result{\PE@result$1#}%
164         \let\PE@next\relax
165       \else
166         \uppercase{%
167           \def\PE@testA{$2}%
168           \def\PE@testB{$3}%
169         }%
170         \ifcase\ifcase\expandafter\PE@TestUcHexDigit\PE@testA
171                  \ifcase\expandafter\PE@TestUcHexDigit\PE@testB
172                    \z@
173                  \else
174                    \@ne
175                  \fi
176                \else
177                  \@ne
178                \fi
179         \uccode\z@="\PE@testA\PE@testB\relax
180         \uppercase{%
181           \def\PE@temp{^^@}%
182         }%
183         \uccode\z@=\z@
184         \edef\PE@result{\PE@result$1\PE@temp}%
185         \let\PE@next\PE@DeName
186       \else
187         % wrong escape sequence in input
188         \edef\PE@result{\PE@result$1#}%
189         \def\PE@next{\PE@DeName$2$3}%
190       \fi
191     \fi
192   \fi
193   \PE@next
194 }%
195 \endgroup
```

\PE@InitUccodeHexDigit

```
196 \def\PE@InitUccodeHexDigit{%
197   \uccode`a=`A\relax
198   \uccode`b=`B\relax
199   \uccode`c=`C\relax
```

```
200   \uccode`d=`D\relax
201   \uccode`e=`E\relax
202   \uccode`f=`F\relax
203   \uccode`A=\z@
204   \uccode`B=\z@
205   \uccode`C=\z@
206   \uccode`D=\z@
207   \uccode`E=\z@
208   \uccode`F=\z@
209   \uccode`0=\z@
210   \uccode`1=\z@
211   \uccode`2=\z@
212   \uccode`3=\z@
213   \uccode`4=\z@
214   \uccode`5=\z@
215   \uccode`6=\z@
216   \uccode`7=\z@
217   \uccode`8=\z@
218   \uccode`9=\z@
219 }
```

\PE@TestUcHexDigit

```
220 \def\PE@TestUcHexDigit#1{%
221   \ifnum`#1<48 % 0
222     \@ne
223   \else
224     \ifnum`#1>70 % F
225       \@ne
226     \else
227       \ifnum`#1>57 % 9
228         \ifnum`#1<65 % A
229           \@ne
230         \else
231           \z@
232         \fi
233       \else
234         \z@
235       \fi
236     \fi
237   \fi
238 }
```

2.5 \EdefUnescapeString

\EdefUnescapeString

```
239 \def\EdefUnescapeString#1#2{%
240   \EdefSanitize#1{#2}%
241   \PE@SanitizeSpaceOther#1%
242   \PE@NormalizeLineEnd#1%
243   \PE@UnescapeString#1%
244   \PE@onelevel@sanitize#1%
245 }
```

```
246 \begingroup
247   \uccode`\8=10 % lf
248   \uccode`\9=13 % cr
249 \def\x#1#2{\endgroup
```

\PE@NormalizeLineEnd

```
250   \def\PE@NormalizeLineEnd##1{%
251     \def\PE@result{}%
252     \expandafter\PE@@NormalizeLineEnd##1#2\relax
```

```
255   \def\PE@@NormalizeLineEnd##1#2##2{%
256     \ifx\relax##2%
257       \edef\PE@result{\PE@result##1}%
258       \let\PE@next\relax
259     \else
260       \edef\PE@result{\PE@result##1#1}%
261       \ifx#1##2% lf
262         \let\PE@next\PE@@NormalizeLineEnd
263       \else
264         \def\PE@next{\PE@@NormalizeLineEnd##2}%
265       \fi
266     \fi
267     \PE@next
268   }%
269 }%
270 \uppercase{%
271   \x 89%
272 }

273 \begingroup
274   \catcode`\|=0 %
275   \catcode`\\=12 %
```

```
276   |gdef|PE@UnescapeString#1{%
277     |begingroup
278       |def|PE@result{}%
279       |expandafter|PE@DeString#1\|relax
280     |expandafter|endgroup
281     |expandafter|def|expandafter#1|expandafter{|PE@result}%
282   }%
```

```
283   |gdef|PE@DeString#1\#2{%
284     |ifx|relax#2%
285       |edef|PE@result{|PE@result#1}%
286       |let|PE@next|relax
287     |else
288       |if n#2%
289         |uccode|z@=10 %
290       |else|if r#2%
291         |uccode|z@=13 %
292       |else|if t#2%
293         |uccode|z@=9 %
294       |else|if b#2%
295         |uccode|z@=8 %
296       |else|if f#2%
297         |uccode|z@=12 %
298       |else
299         |uccode|z@=|z@
300       |fi|fi|fi|fi|fi
301       |ifnum|uccode|z@>|z@
302         |uppercase{%
303           |edef|PE@temp{^^@}%
304         }%
305         |edef|PE@result{|PE@result#1|PE@temp}%
306         |let|PE@next|PE@DeString
307       |else
```

```
308        |if\#2% backslash
309          |edef|PE@result{|PE@result#1}%
310          |let|PE@next|PE@CheckEndBackslash
311        |else
312          |ifnum`#2=10 % linefeed
313            |edef|PE@result{|PE@result#1}%
314            |let|PE@next|PE@DeString
315          |else
316            |ifcase|PE@TestOctDigit#2%
317              |edef|PE@result{|PE@result#1}%
318              |def|PE@next{|PE@OctI#2}%
319            |else
320              |edef|PE@result{|PE@result#1#2}%
321              |let|PE@next|PE@DeString
322            |fi
323          |fi
324        |fi
325      |fi
326    |fi
327    |PE@next
328  }%
```

\PE@CheckEndBackslash

```
329  |gdef|PE@CheckEndBackslash#1{%
330    |ifx|relax#1%
331    |else
332      |edef|PE@result{|PE@result\}%
333      |expandafter|PE@DeString|expandafter#1%
334    |fi
335  }%

336 |endgroup
```

\PE@TestOctDigit

```
337 \def\PE@TestOctDigit#1{%
338   \ifnum`#1<48 % 0
339     \@ne
340   \else
341     \ifnum`#1>55 % 7
342       \@ne
343     \else
344       \z@
345     \fi
346   \fi
347 }
```

\PE@OctI

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

\PE@OctII

```
356 \def\PE@OctII#1#2{%
357   \ifcase\PE@TestOctDigit#2%
358     \def\PE@next{\PE@OctAll{#1#2}}%
359   \else
360     \def\PE@next{\PE@OctAll{#1}#2}%
```

```
361    \fi
362    \PE@next
363  }
```

\PE@OctAll

```
364  \def\PE@OctAll#1{%
365    \uccode\z@='#1\relax
366    \uppercase{%
367      \edef\PE@result{\PE@result^^@}%
368    }%
369    \PE@DeString
370  }
```

2.6   User macros (pdfTeX analogues)

```
371  \begingroup\expandafter\expandafter\expandafter\endgroup
372  \expandafter\ifx\csname RequirePackage\endcsname\relax
373    \input pdftexcmds.sty\relax
374  \else
375    \RequirePackage{pdftexcmds}[2007/11/11]%
376  \fi
```

```
377  \begingroup\expandafter\expandafter\expandafter\endgroup
378  \expandafter\ifx\csname pdf@escapehex\endcsname\relax
```

\EdefEscapeHex

```
379    \long\def\EdefEscapeHex#1#2{%
380      \EdefSanitize#1{#2}%
381      \PE@SanitizeSpaceOther#1%
382      \PE@EscapeHex#1%
383    }%
```

\EdefUnescapeHex

```
384    \def\EdefUnescapeHex#1#2{%
385      \EdefSanitize#1{#2}%
386      \PE@UnescapeHex#1%
387    }%
```

\EdefEscapeName

```
388    \long\def\EdefEscapeName#1#2{%
389      \EdefSanitize#1{#2}%
390      \PE@SanitizeSpaceOther#1%
391      \PE@EscapeName#1%
392    }%
```

\EdefEscapeString

```
393    \long\def\EdefEscapeString#1#2{%
394      \EdefSanitize#1{#2}%
395      \PE@SanitizeSpaceOther#1%
396      \PE@EscapeString#1%
397    }%
```

```
398  \else
```

\PE@edefbabel   Help macro that adds support for babel's shorthand characters.

```
399    \long\def\PE@edefbabel#1#2#3{%
400      \begingroup
401        \csname @save@activestrue\endcsname
402        \edef#1{#2{#3}}%
403      \expandafter\endgroup
404      \expandafter\def\expandafter#1\expandafter{#1}%
405    }%
```

```
406   \long\def\EdefEscapeHex#1#2{%
407     \PE@edefbabel#1\pdf@escapehex{#2}%
408   }%
```

```
409   \def\EdefUnescapeHex#1#2{%
410     \PE@edefbabel#1\pdf@unescapehex{#2}%
411   }%
```

```
412   \long\def\EdefEscapeName#1#2{%
413     \PE@edefbabel#1\pdf@escapename{#2}%
414   }%
```

```
415   \long\def\EdefEscapeString#1#2{%
416     \PE@edefbabel#1\pdf@escapestring{#2}%
417   }%
```

```
418   \PE@AtEnd
419   \expandafter\endinput
420 \fi
```

2.7   Help macros

2.7.1   Characters

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

```
421 \edef\PE@hash{\string#}
```

```
422 \begingroup
423   \escapechar=-1 %
424 \edef\x{\endgroup
425   \def\noexpand\PE@backslash{\string\\}%
426 }
427 \x
```

2.7.2   Switch for ε-TeX

```
428 \newif\ifPE@etex
429 \begingroup\expandafter\expandafter\expandafter\endgroup
430 \expandafter\ifx\csname numexpr\endcsname\relax
431 \else
432   \PE@etextrue
433 \fi
```

2.8   Conversions

2.8.1   Conversion to hex string

```
434 \ifPE@etex
435   \def\PE@EscapeHex#1{%
436     \edef#1{\expandafter\PE@ToHex#1\relax}%
437   }%
438 \else
439   \def\PE@EscapeHex#1{%
440     \def\PE@result{}%
```

```
441     \expandafter\PE@ToHex#1\relax
442     \let#1\PE@result
443   }%
444 \fi
```

```
445 \def\PE@ToHex#1{%
446   \ifx\relax#1%
447   \else
448     \PE@HexChar{#1}%
449     \expandafter\PE@ToHex
450   \fi
451 }%
```

```
452 \ifPE@etex
453   \def\PE@HexChar#1{%
454     \PE@HexDigit{\numexpr\dimexpr.0625\dimexpr`#1sp\relax\relax\relax}%
455     \PE@HexDigit{%
456       \numexpr`#1-16*\dimexpr.0625\dimexpr`#1sp\relax\relax\relax
457     }%
458   }%
459 \else
460   \def\PE@HexChar#1{%
461     \dimen0=`#1sp%
462     \dimen2=.0625\dimen0 %
463     \advance\dimen0-16\dimen2 %
464     \edef\PE@result{%
465       \PE@result
466       \PE@HexDigit{\dimen2 }%
467       \PE@HexDigit{\dimen0 }%
468     }%
469   }%
470 \fi
```

```
471 \def\PE@HexDigit#1{%
472   \expandafter\string
473   \ifcase#1%
474     0\or 1\or 2\or 3\or 4\or 5\or 6\or 7\or 8\or 9\or
475     A\or B\or C\or D\or E\or F%
476   \fi
477 }
```

2.8.2 Character code to octal number

```
478 \ifPE@etex
479   \def\PE@OctChar#1{%
480     \expandafter\PE@@OctChar
481         \the\numexpr\dimexpr.015625\dimexpr`#1sp\relax\relax
482             \expandafter\relax
483         \expandafter\relax
484         \the\numexpr\dimexpr.125\dimexpr`#1sp\relax\relax\relax
485         \relax
486         #1%
487   }%
488   \def\PE@@OctChar#1\relax#2\relax#3{%
489     \PE@backslash
490     #1%
491     \the\numexpr#2-8*#1\relax
```

```
492        \the\numexpr\dimexpr`#3sp\relax-8*#2\relax
493     }%
494  \else
495     \def\PE@OctChar#1{%
496        \dimen0=`#1sp%
497        \dimen2=.125\dimen0 %
498        \dimen4=.125\dimen2 %
499        \advance\dimen0-8\dimen2 %
500        \advance\dimen2-8\dimen4 %
501        \edef\PE@result{%
502           \PE@result
503           \PE@backslash
504           \number\dimen4 %
505           \number\dimen2 %
506           \number\dimen0 %
507        }%
508     }%
509  \fi
```

2.8.3 Unpack hex string

\PE@UnescapeHex

```
510  \def\PE@UnescapeHex#1{%
511     \begingroup
512        \PE@InitUccodeHexDigit
513        \def\PE@result{}%
514        \expandafter\PE@DeHex#1\relax\relax
515     \expandafter\endgroup
516     \expandafter\def\expandafter#1\expandafter{\PE@result}%
517  }
```

\PE@DeHex

```
518  \def\PE@DeHex#1#2{%
519     \ifx#2\relax
520        \ifx#1\relax
521           \let\PE@next\relax
522        \else
523           \uppercase{%
524              \def\PE@testA{#1}%
525           }%
526           \ifcase\expandafter\PE@TestUcHexDigit\PE@testA
527              \def\PE@next{%
528                 \PE@DeHex#10\relax\relax
529              }%
530           \else
531              \let\PE@next\relax
532           \fi
533        \fi
534     \else
535        \uppercase{%
536           \def\PE@testA{#1}%
537           \def\PE@testB{#2}%
538        }%
539        \ifcase\expandafter\PE@TestUcHexDigit\PE@testA
540           \ifcase\expandafter\PE@TestUcHexDigit\PE@testB
541              \uccode\z@="\PE@testA\PE@testB\relax
542              \ifnum\uccode\z@=32 %
543                 \let\PE@temp\PE@space@space
544              \else
545                 \uppercase{%
546                    \def\PE@temp{^^@}%
547                 }%
```

13

```
548        \fi
549        \edef\PE@result{\PE@result\PE@temp}%
550        \let\PE@next\PE@DeHex
551      \else
552        % invalid input sequence
553        \def\PE@next{%
554          \PE@DeHex#1%
555        }%
556      \fi
557    \else
558      % invalid input sequence
559      \def\PE@next{\PE@DeHex#2}%
560    \fi
561  \fi
562  \PE@next
563 }
```

2.8.4 Conversion to PDF name

\PE@EscapeName

```
564 \ifPE@etex
565   \def\PE@EscapeName#1{%
566     \edef#1{\expandafter\PE@EscapeNameTokens#1\relax}%
567   }%
568 \else
569   \def\PE@EscapeName#1{%
570     \def\PE@result{}%
571     \expandafter\PE@EscapeNameTokens#1\relax
572     \let#1\PE@result
573   }%
574 \fi
```

\PE@EscapeNameTokens

```
575 \def\PE@EscapeNameTokens#1{%
576   \ifx\relax#1%
577   \else
578     \ifnum`#1<33 %
579       \ifcase`#1 %
580         % drop illegal zero
581       \else
582         \PE@EscapeNameAdd\PE@hash
583         \PE@HexChar#1%
584       \fi
585     \else
586       \ifnum`#1>126 %
587         \PE@EscapeNameAdd\PE@hash
588         \PE@HexChar#1%
589       \else \ifnum`#1=35  \PE@EscapeNameHashChar 23% #
590         \else\ifnum`#1=37  \PE@EscapeNameHashChar 25% %
591         \else\ifnum`#1=40  \PE@EscapeNameHashChar 28% (
592         \else\ifnum`#1=41  \PE@EscapeNameHashChar 29% )
593         \else\ifnum`#1=47  \PE@EscapeNameHashChar 2F% /
594         \else\ifnum`#1=60  \PE@EscapeNameHashChar 3C% <
595         \else\ifnum`#1=62  \PE@EscapeNameHashChar 3E% >
596         \else\ifnum`#1=91  \PE@EscapeNameHashChar 5B% [
597         \else\ifnum`#1=93  \PE@EscapeNameHashChar 5D% ]
598         \else\ifnum`#1=123 \PE@EscapeNameHashChar 7B% {
599         \else\ifnum`#1=125 \PE@EscapeNameHashChar 7D% }
600       \else
601         \PE@EscapeNameAdd{#1}%
602       \fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi
603     \fi
```

```
604     \fi
605     \expandafter\PE@EscapeNameTokens
606   \fi
607 }%
608 \def\PE@EscapeNameHashChar#1#2{%
609   \PE@EscapeNameAdd{\PE@hash\string#1\string#2}%
610 }%
```

\PE@EscapeNameAdd

```
611 \ifPE@etex
612   \def\PE@EscapeNameAdd#1{#1}%
613 \else
614   \def\PE@EscapeNameAdd#1{%
615     \edef\PE@result{%
616       \PE@result
617       #1%
618     }%
619   }%
620 \fi
```

2.8.5 Conversion to PDF string

\PE@EscapeString

```
621 \ifPE@etex
622   \def\PE@EscapeString#1{%
623     \edef#1{\expandafter\PE@EscapeStringTokens#1\relax}%
624   }%
625 \else
626   \def\PE@EscapeString#1{%
627     \begingroup
628       \def\PE@result{}%
629       \expandafter\PE@EscapeStringTokens#1\relax
630     \expandafter\endgroup
631     \expandafter\def\expandafter#1\expandafter{\PE@result}%
632   }%
633 \fi
```

\PE@EscapeStringTokens

```
634 \def\PE@EscapeStringTokens#1{%
635   \ifx\relax#1%
636   \else
637     \ifnum`#1<33 %
638       \PE@OctChar#1%
639     \else
640       \ifnum`#1>126 %
641         \PE@OctChar#1%
642       \else  \ifnum`#1=40 \PE@EscapeStringAdd{\string\(}% (
643         \else\ifnum`#1=41 \PE@EscapeStringAdd{\string\)}% )
644         \else\ifnum`#1=92 \PE@EscapeStringAdd{\string\\}% \
645         \else
646           \PE@EscapeStringAdd{#1}%
647         \fi\fi\fi
648       \fi
649     \fi
650     \expandafter\PE@EscapeStringTokens
651   \fi
652 }%
```

\PE@EscapeStringAdd

```
653 \ifPE@etex
654   \def\PE@EscapeStringAdd#1{#1}%
```

```
655 \else
656   \def\PE@EscapeStringAdd#1{%
657     \edef\PE@result{%
658       \PE@result
659       #1%
660     }%
661   }%
662 \fi

663 \PE@AtEnd

664 ⟨/package⟩
```

3 Test

3.1 Catcode checks for loading

```
665 ⟨∗test1⟩

666 \catcode`\{=1 %
667 \catcode`\}=2 %
668 \catcode`\#=6 %
669 \catcode`\@=11 %
670 \expandafter\ifx\csname count@\endcsname\relax
671   \countdef\count@=255 %
672 \fi
673 \expandafter\ifx\csname @gobble\endcsname\relax
674   \long\def\@gobble#1{}%
675 \fi
676 \expandafter\ifx\csname @firstofone\endcsname\relax
677   \long\def\@firstofone#1{#1}%
678 \fi
679 \expandafter\ifx\csname loop\endcsname\relax
680   \expandafter\@firstofone
681 \else
682   \expandafter\@gobble
683 \fi
684 {%
685   \def\loop#1\repeat{%
686     \def\body{#1}%
687     \iterate
688   }%
689   \def\iterate{%
690     \body
691       \let\next\iterate
692     \else
693       \let\next\relax
694     \fi
695     \next
696   }%
697   \let\repeat=\fi
698 }%
699 \def\RestoreCatcodes{}
700 \count@=0 %
701 \loop
702   \edef\RestoreCatcodes{%
703     \RestoreCatcodes
704     \catcode\the\count@=\the\catcode\count@\relax
705   }%
706 \ifnum\count@<255 %
707   \advance\count@ 1 %
708 \repeat
709
```

16

```
710 \def\RangeCatcodeInvalid#1#2{%
711   \count@=#1\relax
712   \loop
713     \catcode\count@=15 %
714   \ifnum\count@<#2\relax
715     \advance\count@ 1 %
716   \repeat
717 }
718 \expandafter\ifx\csname LoadCommand\endcsname\relax
719   \def\LoadCommand{\input pdfescape.sty\relax}%
720 \fi
721 \def\Test{%
722   \RangeCatcodeInvalid{0}{47}%
723   \RangeCatcodeInvalid{58}{64}%
724   \RangeCatcodeInvalid{91}{96}%
725   \RangeCatcodeInvalid{123}{255}%
726   \catcode'\@=12 %
727   \catcode'\\=0 %
728   \catcode'\{=1 %
729   \catcode'\}=2 %
730   \catcode'\#=6 %
731   \catcode'\[=12 %
732   \catcode'\]=12 %
733   \catcode'\%=14 %
734   \catcode'\ =10 %
735   \catcode13=5 %
736   \LoadCommand
737   \RestoreCatcodes
738 }
739 \Test
740 \csname @@end\endcsname
741 \end

742 ⟨/test1⟩
```

3.2   Macro tests

```
743 ⟨∗test2 | test3 | test4 | test5⟩
744 \NeedsTeXFormat{LaTeX2e}
745 \makeatletter
```

3.3   Test with \pdfescape... commands

```
746 ⟨∗test2⟩
747 \ProvidesFile{pdfescape-test2.tex}%
748     [2007/11/11 v1.8 Test with \string\pdfescape... commands]%
749 ⟨/test2⟩
```

3.4   Test without \pdfescape..., with ε-TEX

```
750 ⟨∗test3⟩
751 \ProvidesFile{pdfescape-test3.tex}%
752     [2007/11/11 v1.8 Test without \string\pdfescape..., with e-TeX]%
753 ⟨/test3⟩
```

3.5   Test without \pdfescape... and ε-TEX

```
754 ⟨∗test4⟩
755 \ProvidesFile{pdfescape-test4.tex}%
756     [2007/11/11 v1.8 Test without \string\pdfescape... and e-TeX]%
757 ⟨/test4⟩
```

3.6   Test with LuaTEX

```
758 ⟨∗test5⟩
759 \ProvidesFile{pdfescape-test5.tex}%
760     [2007/11/11 v1.8 Test with LuaTeX]%
```

761 ⟨/test5⟩

3.7 Check/ensure test preconditions

3.7.1 Check \pdfescape...

762 ⟨*test2⟩
763 \@ifundefined{pdfescapehex}{%
764 \PackageError{pdfescape-test2}{%
765 Missing \string\pdfescape... commands%
766 }{Test aborted.}%
767 \stop
768 }{}
769 ⟨/test2⟩

770 ⟨*test3 | test4⟩
771 \let\pdfescapehex\@undefined
772 \let\pdfunescapehex\@undefined
773 \let\pdfescapename\@undefined
774 \let\pdfescapestring\@undefined
775 ⟨/test3 | test4⟩

3.7.2 Check ε-TEX

776 ⟨*test3⟩
777 \@ifundefined{numexpr}{%
778 \PackageError{pdfescape-test3}{%
779 Missing \eTeX
780 }{Test aborted.}%
781 \stop
782 }{}
783 ⟨/test3⟩

Package qstest uses ε-TEX, thus ε-TEX's features can only be disabled later during
loading of package pdfescape.

3.7.3 Check LuaTEX

784 ⟨*test5⟩
785 \@ifundefined{directlua}{%
786 \PackageError{pdfescape-test5}{%
787 Missing LuaTeX%
788 }{Test aborted.}%
789 \stop
790 }{}
791 ⟨/test5⟩

3.8 Common part

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

792 \RequirePackage{qstest}
793 \IncludeTests{*}
794 \LogTests{log}{*}{*}
795
796 \newcommand*{\ExpectVar}[2]{%
797 \ifx#1#2%
798 \else
799 \begingroup
800 \@onelevel@sanitize#1%
801 \@onelevel@sanitize#2%
802 \typeout{[#1] <> [#2]}% hash-ok
803 \endgroup
804 \fi
805 \Expect*{\ifx#1#2true\else false\fi}{true}%
806 }
807

```
808 \makeatletter
809 \begingroup
810   \gdef\AllBytes{}%
811   \count@=0 %
812   \catcode0=12 %
813   \@whilenum\count@<256 \do{%
814     \lccode0=\count@
815     \ifnum\count@=32 %
816       \xdef\AllBytes{\AllBytes\space}%
817     \else
818       \lowercase{%
819         \xdef\AllBytes{\AllBytes^^@}%
820       }%
821     \fi
822     \advance\count@ by 1 %
823   }%
824 \endgroup
825 \newcommand*{\AllBytesHex}{%
826   000102030405060708090A0B0C0D0E0F%
827   101112131415161718191A1B1C1D1E1F%
828   202122232425262728292A2B2C2D2E2F%
829   303132333435363738393A3B3C3D3E3F%
830   404142434445464748494A4B4C4D4E4F%
831   505152535455565758595A5B5C5D5E5F%
832   606162636465666768696A6B6C6D6E6F%
833   707172737475767778797A7B7C7D7E7F%
834   808182838485868788898A8B8C8D8E8F%
835   909192939495969798999A9B9C9D9E9F%
836   A0A1A2A3A4A5A6A7A8A9AAABACADAEAF%
837   B0B1B2B3B4B5B6B7B8B9BABBBCBDBEBF%
838   C0C1C2C3C4C5C6C7C8C9CACBCCCDCECF%
839   D0D1D2D3D4D5D6D7D8D9DADBDCDDDEDF%
840   E0E1E2E3E4E5E6E7E8E9EAEBECEDEEEF%
841   F0F1F2F3F4F5F6F7F8F9FAFBFCFDFEFF%
842 }
843 \@onelevel@sanitize\AllBytesHex
844 \expandafter\lowercase\expandafter{%
845   \expandafter\newcommand\expandafter*\expandafter\AllBytesHexLC
846     \expandafter{\AllBytesHex}%
847 }
848 \newcommand*{\AllBytesName}{}
849 \begingroup
850   \catcode`\#=12 %
851   \xdef\AllBytesName{%
852     #01#02#03#04#05#06#07#08#09#0A#0B#0C#0D#0E#0F%
853     #10#11#12#13#14#15#16#17#18#19#1A#1B#1C#1D#1E#1F%
854     #20!"#23$#25&'#28#29*+,-.#2F%
855     0123456789:;#3C=#3E?%
856     @ABCDEFGHIJKLMNO%
857     PQRSTUVWXYZ#5B\@backslashchar#5D^_%
858     `abcdefghijklmno%
859     pqrstuvwxyz#7B|#7D\string~#7F%
860     #80#81#82#83#84#85#86#87#88#89#8A#8B#8C#8D#8E#8F%
861     #90#91#92#93#94#95#96#97#98#99#9A#9B#9C#9D#9E#9F%
862     #A0#A1#A2#A3#A4#A5#A6#A7#A8#A9#AA#AB#AC#AD#AE#AF%
863     #B0#B1#B2#B3#B4#B5#B6#B7#B8#B9#BA#BB#BC#BD#BE#BF%
864     #C0#C1#C2#C3#C4#C5#C6#C7#C8#C9#CA#CB#CC#CD#CE#CF%
865     #D0#D1#D2#D3#D4#D5#D6#D7#D8#D9#DA#DB#DC#DD#DE#DF%
866     #E0#E1#E2#E3#E4#E5#E6#E7#E8#E9#EA#EB#EC#ED#EE#EF%
867     #F0#F1#F2#F3#F4#F5#F6#F7#F8#F9#FA#FB#FC#FD#FE#FF%
868   }%
869 \endgroup
```

```
870 \@onelevel@sanitize\AllBytesName
871
872 \newcommand*{\AllBytesString}{}
873 \begingroup
874   \def\|{|}%
875   \edef\%{\@percentchar}%
876   \catcode'\|=0 %
877   \catcode'\#=12 %
878   \catcode'\~=12 %
879   \catcode'\\=12 %
880   |xdef|AllBytesString{%
881     \000\001\002\003\004\005\006\007\010\011\012\013\014\015\016\017%
882     \020\021\022\023\024\025\026\027\030\031\032\033\034\035\036\037%
883     \040!"#$|%&'\(\)*+,-./%
884     0123456789:;<=>?%
885     @ABCDEFGHIJKLMNO%
886     PQRSTUVWXYZ[\\]^_%
887     'abcdefghijklmno%
888     pqrstuvwxyz{||}~\177%
889     \200\201\202\203\204\205\206\207\210\211\212\213\214\215\216\217%
890     \220\221\222\223\224\225\226\227\230\231\232\233\234\235\236\237%
891     \240\241\242\243\244\245\246\247\250\251\252\253\254\255\256\257%
892     \260\261\262\263\264\265\266\267\270\271\272\273\274\275\276\277%
893     \300\301\302\303\304\305\306\307\310\311\312\313\314\315\316\317%
894     \320\321\322\323\324\325\326\327\330\331\332\333\334\335\336\337%
895     \340\341\342\343\344\345\346\347\350\351\352\353\354\355\356\357%
896     \360\361\362\363\364\365\366\367\370\371\372\373\374\375\376\377%
897   }%
898 |endgroup
899 \@onelevel@sanitize\AllBytesString
900
901 ⟨*test4⟩
902 \let\org@detokenize\detokenize
903 \let\detokenize\@undefined
904 \let\org@numexpr\numexpr
905 \let\numexpr\@undefined
906 ⟨/test4⟩
907 \RequirePackage{pdfescape}
908 ⟨*test4⟩
909 \let\detokenize\org@detokenize
910 \let\numexpr\org@numexpr
911 ⟨/test4⟩
912
913 \begin{qstest}{all-hex}{\AllBytes, escapehex}
914   \EdefEscapeHex\x{\AllBytes}%
915   \Expect*{\x}*{\AllBytesHex}%
916   \ExpectVar\x\AllBytesHex
917 \end{qstest}
918
919 \begin{qstest}{all-unhex}{\AllBytesHex, unescapehex}
920   \EdefUnescapeHex\x{\AllBytesHex}%
921   \Expect*{\x}*{\AllBytes}%
922   \ExpectVar\x\AllBytes
923 \end{qstest}
924
925 \begin{qstest}{all-unhex-lc}{\AllBytesHexLC, unescapehex, lowercase}
926   \EdefUnescapeHex\x{\AllBytesHexLC}%
927   \Expect*{\x}*{\AllBytes}%
928   \ExpectVar\x\AllBytes
929 \end{qstest}
930
931 \begin{qstest}{unhex-incomplete}{unescapehex, incomplete}
```

```
932   \EdefUnescapeHex\x{4}%
933   \Expect*{\x}{@}%
934 \end{qstest}
935
936 \begin{qstest}{unhex-space}{unescapehex, space}
937   \EdefUnescapeHex\x{20}%
938   \Expect*{\x}{ }%
939   \ExpectVar\x\space
940 \end{qstest}
941
942 \begin{qstest}{unhex-spaces}{unescapehex, spaces}
943   \EdefUnescapeHex\x{204020204120}%
944   \def\y#1{%
945     \edef\z{#1\string @#1#1\string A#1}%
946   }\y{ }%
947   \Expect*{\x}*{\z}%
948   \ExpectVar\x\z
949 \end{qstest}
950
951 \begin{qstest}{unhex-hash}{unescapehex, hash}
952   \catcode`\#=12 %
953   \EdefUnescapeHex\x{#20}%
954   \ExpectVar\x\space
955 \end{qstest}
956
957 \begin{qstest}{unhex-invalid}{unescapehex, invalid}
958   \def\test#1#2{%
959     \EdefUnescapeHex\x{#1}%
960     \edef\y{#2}%
961     \@onelevel@sanitize\y
962     \ExpectVar\x\y
963   }%
964 ⟨*test2⟩
965   \edef\x{\pdfunescapehex{4X}}%
966   \edef\y{\string @}%
967   \ifx\x\y
968   \else
969     \def~{\space}%
970     \typeout{**********************************}%
971     \typeout{* Your pdfTeX contains bug 777.~~~~*}%
972     \typeout{* This test is redefined as dummy, *}%
973     \typeout{* because it triggers the bug.~~~~*}%
974     \typeout{**********************************}%
975     \def\test#1#2{}%
976   \fi
977 ⟨/test2⟩
978   \test{X}{}%
979   \test{XY}{}%
980   \test{XYZ}{}%
981   \test{A}{^^a0}%
982   \test{AX}{^^a0}%
983   \test{XA}{^^a0}%
984   \test{XXAXX}{^^a0}%
985 \end{qstest}
986
987 \begin{qstest}{all-name}{\AllBytes, escapename}
988   \EdefEscapeName\x{\AllBytes}%
989   \Expect*{\x}*{\AllBytesName}%
990   \ExpectVar\x\AllBytesName
991 \end{qstest}
992
993 \begin{qstest}{all-string}{\AllBytes, escapestring}
```

```
994    \EdefEscapeString\x{\AllBytes}%
995    \Expect*{\x}*{\AllBytesString}%
996    \ExpectVar\x\AllBytesString
997  \end{qstest}
998
999  \begin{qstest}{uchexdigit}{unescape, uppercase hex digit}
1000    \catcode`\@=11 %
1001    \catcode0=12 %
1002    \def\test#1#2{%
1003      \uccode0=#1\relax
1004      \uppercase{%
1005        \def\x{^^@}%
1006      }%
1007      \Expect*{%
1008        \ifcase\expandafter\PE@TestUcHexDigit\x
1009          true%
1010        \else
1011          false%
1012        \fi
1013      }{#2}%
1014    }%
1015    \def\range#1#2#3{%
1016      \count0=#1\relax
1017      \loop
1018      \ifnum\count0<#2\relax
1019        \test{\count0}{#3}%
1020        \advance\count0 by 1 %
1021      \repeat
1022    }%
1023    \range{0}{47}{false}%
1024    \range{48}{57}{true}%
1025    \range{58}{64}{false}%
1026    \range{65}{70}{true}%
1027    \range{71}{255}{false}%
1028  \end{qstest}
1029
1030  \begin{qstest}{unescapename}{unescapename}
1031    \def\test#1#2{%
1032      \EdefUnescapeName\x{#1}%
1033      \edef\y{#2}%
1034      \@onelevel@sanitize\y
1035      \ExpectVar\x\y
1036    }%
1037    \catcode`\#=12 %
1038    \catcode0=12 %
1039    \test{}{}%
1040    \test{x}{x}%
1041    \test{xy}{xy}%
1042    \test{#}{#}%
1043    \test{##}{##}%
1044    \test{###}{###}%
1045    \test{####}{####}%
1046    \test{#x}{#x}%
1047    \test{#xy}{#xy}%
1048    \test{#1}{#1}%
1049    \test{#40}{@}%
1050    \test{#400}{@0}%
1051    \test{#4x0}{#4x0}%
1052    \test{#ab}{^^ab}%
1053    \test{#00}{^^@}%
1054    \test{x#40y#40z}{x@y@z}%
1055    \test{#40#40#40#40}{@@@@}%
```

```
1056    \test{a#x}{a#x}%
1057    \test{a#xy}{a#xy}%
1058    \test{a#1}{a#1}%
1059    \test{a#40}{a@}%
1060    \test{a#400}{a@0}%
1061    \test{#20}{ }%
1062    \test{a#20}{a }%
1063    \test{a#20b}{a b}%
1064    \test{a#20#20#20b}{a \space\space b}%
1065 \end{qstest}
1066
1067 \begin{qstest}{unescapestring}{unescapestring}
1068    \def\test#1#2{%
1069       \EdefUnescapeString\x{#1}%
1070       \edef\y{#2}%
1071       \@onelevel@sanitize\y
1072       \ExpectVar\x\y
1073    }%
1074    \catcode0=12 %
1075    \def\DefChar#1#2{%
1076       \begingroup
1077          \uccode0=#2\relax
1078       \uppercase{\endgroup
1079          \def#1{^^@}%
1080       }%
1081    }%
1082    \DefChar\nul{0}%
1083    \DefChar\one{1}%
1084    \DefChar\bel{8}%
1085    \DefChar\tab{9}%
1086    \DefChar\lf{10}%
1087    \DefChar\ff{12}%
1088    \DefChar\cr{13}%
1089    \DefChar\\{92}%
1090    \test{}{}%
1091    \test{a}{a}%
1092    \test{\\}{}%
1093    \test{\\\\}{\\}%
1094    \test{\\\\y}{\\y}%
1095    \test{\\000}{\nul}%
1096    \test{\\b}{\bel}%
1097    \test{\\t}{\tab}%
1098    \test{\\n}{\lf}%
1099    \test{\\f}{\ff}%
1100    \test{\\r}{\cr}%
1101    \test{\\(}{(}%
1102    \test{\\)}{)}%
1103    \test{\\040}{ }%
1104    \test{\\100}{@}%
1105    \test{\\40}{ }%
1106    \test{\\1}{\one}%
1107    \test{\\01}{\one}%
1108    \test{\\001}{\one}%
1109    \test{\\18}{\one8}%
1110    \test{\\018}{\one8}%
1111    \test{\\0018}{\one8}%
1112    \test{x\\}{x}%
1113    \test{x\\\\}{x\\}%
1114    \test{x\\\\y}{x\\y}%
1115    \test{x\\000}{x\nul}%
1116    \test{x\\b}{x\bel}%
1117    \test{x\\t}{x\tab}%
```

```
1118    \test{x\\n}{x\lf}%
1119    \test{x\\f}{x\ff}%
1120    \test{x\\r}{x\cr}%
1121    \test{x\\(}{x(}%
1122    \test{x\\)}{x)}%
1123    \test{x\\040}{x }%
1124    \test{x\\100}{x@}%
1125    \test{x\\40}{x }%
1126    \test{x\\1}{x\one}%
1127    \test{x\\01}{x\one}%
1128    \test{x\\001}{x\one}%
1129    \test{x\\18}{x\one8}%
1130    \test{x\\018}{x\one8}%
1131    \test{x\\0018}{x\one8}%
1132    \test{\\b\\t\\n\\f\\r\\(\\)\\\\\\\\000\\040}{%
1133      \bel\tab\lf\ff\cr()\\\nul\space
1134    }%
1135    \test{\\\lf}{}%
1136    \test{x\\\lf}{x}%
1137    \test{\cr}{\lf}%
1138    \test{\cr\lf}{\lf}%
1139    \test{\lf}{\lf}%
1140    \test{\lf\cr}{\lf\lf}%
1141    \test{x\cr}{x\lf}%
1142    \test{x\cr\lf}{x\lf}%
1143    \test{x\lf}{x\lf}%
1144    \test{x\lf\cr}{x\lf\lf}%
1145    \test{x\\\cr\lf y\cr}{xy\lf}%
1146 \end{qstest}
1147 \stop

1148 ⟨/test2 | test3 | test4 | test5⟩
```

4 Installation

4.1 Download

Package. This package is available on CTAN[1]:

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:install/macros/latex/contrib/oberdiek.tds.zip

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

---
[1]ftp://ftp.ctan.org/tex-archive/

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 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
test/pdfescape-test5.tex → doc/latex/oberdiek/test/pdfescape-test5.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 TEX distribution (teTEX, mikTEX, ...) relies on file name databases, you must refresh these. For example, teTEX users run `texhash` or `mktexlsr`.

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 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{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 pdfLaTeX:

```
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`.

[2007/11/11 v1.8]

- Use of package pdftexcmds for LUATEX support.

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.

28