

The pdfescape package

Heiko Oberdiek
<oberdiek@uni-freiburg.de>

2006/02/25 v1.1

Abstract

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

Contents

1	Documentation	2
2	Implementation	2
2.1	Reload check and package identification	2
2.2	Preparations	3
2.2.1	Catcodes	3
2.3	User macros	3
2.4	Help macros	4
2.4.1	Characters	4
2.4.2	Switch for ϵ -TeX	5
2.5	Sanitizing	5
2.6	Conversions	6
2.6.1	Conversion to hex string	6
2.6.2	Character code to octal number	7
2.6.3	Unpack hex string	7
2.6.4	Conversion to PDF name	8
2.6.5	Conversion to PDF string	9
3	Test	10
3.1	Test with <code>\pdfescape...</code> commands	10
3.2	Test without <code>\pdfescape...</code> , with ϵ -TeX	10
3.3	Test without <code>\pdfescape...</code> and ϵ -TeX	10
3.4	Check/ensure test preconditions	10
3.4.1	Check <code>\pdfescape...</code>	10
3.4.2	Check ϵ -TeX	10
3.5	Common part	10
4	Installation	13
4.1	Some details for the interested	14
5	History	14
	[2007/02/21 v1.0]	14
	[2007/02/25 v1.1]	15
6	Index	15

1 Documentation

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

These commands convert $\langle string \rangle$ and store 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 ≥ 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`.

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

Package identification:

```
25 \begingroup  
26   \expandafter\ifx\csname ProvidesPackage\endcsname\relax  
27     \def\x#1#2#3[#4]{\endgroup  
28       \immediate\write-1{Package: #3 #4}%  
29       \xdef#1{#4}%  
30     }%  
31   \else  
32     \def\x#1#2[#3]{\endgroup
```

```

33      #2[#{#3}]%
34      \ifx#1\relax
35      \xdef#1{#3}%
36      \fi
37    }%
38  \fi
39 \expandafter\x\csname ver@pdfescape.sty\endcsname
40 \ProvidesPackage{pdfescape}%
41 [2006/02/25 v1.1 Provides hex, PDF name and string conversions (H0)]

```

2.2 Preparations

2.2.1 Catcodes

```

42 \expandafter\edef\csname PE@AtEnd\endcsname{%
43   \catcode64 \the\catcode64\relax
44 }
45 \catcode64 11 % @
46 \def\PE@EnsureCode#1#2#3{%
47   \edef\PE@AtEnd{%
48     \PE@AtEnd
49     #1#2 \the#1#2\relax
50   }%
51   #1#2 #3\relax
52 }
53 \PE@EnsureCode\catcode{0}{12}% ^^@
54 \PE@EnsureCode\catcode{34}{12}% "
55 \PE@EnsureCode\catcode{42}{12}% *
56 \PE@EnsureCode\catcode{45}{12}% -
57 \PE@EnsureCode\catcode{46}{12}% .
58 \PE@EnsureCode\catcode{60}{12}% <
59 \PE@EnsureCode\catcode{61}{12}% =
60 \PE@EnsureCode\catcode{62}{12}% >
61 \PE@EnsureCode\catcode{94}{7}% ^
62 \PE@EnsureCode\catcode{96}{12}% ‘
63 \PE@EnsureCode\uccode{34}{0}% "
64 \PE@EnsureCode\uccode{48}{0}% 0
65 \PE@EnsureCode\uccode{61}{0}% =

```

2.3 User macros

```

66 \begingroup\expandafter\expandafter\expandafter\endgroup
67 \expandafter\ifx\csname pdfescapehex\endcsname\relax

```

\EdefEscapeHex

```

68   \long\def\EdefEscapeHex#1#2{%
69     \PE@sanitize#1{#2}%
70     \edef#1{\expandafter\PE@SpaceToOther#1 \relax}%
71     \PE@EscapeHex#1%
72   }%

```

\EdefUnescapeHex

```

73   \def\EdefUnescapeHex#1#2{%
74     \PE@sanitize#1{#2}%
75     \PE@UnescapeHex#1%
76   }%

```

\EdefEscapeName

```

77   \long\def\EdefEscapeName#1#2{%
78     \PE@sanitize#1{#2}%
79     \edef#1{\expandafter\PE@SpaceToOther#1 \relax}%
80     \PE@EscapeName#1%
81   }%

```

```

\edefEscapeString
82 \long\def\edefEscapeString#1#2{%
83   \PE@sanitize#1{#2}%
84   \edef#1{\expandafter\PE@SpaceToOther#1 \relax}%
85   \PE@EscapeString#1%
86 }%

87 \else

\PE@edefbabel Help macro that adds support for babel's shorthand characters.
88 \long\def\PE@edefbabel#1#2#3{%
89   \begingroup
90     \csname @save@activetrue\endcsname
91     \edef#1{#2{#3}}%
92   \expandafter\endgroup
93   \expandafter\def\expandafter#1\expandafter{#1}%
94 }

\edefEscapeHex
95 \long\def\edefEscapeHex#1#2{%
96   \PE@edefbabel#1\pdfescapehex{#2}%
97 }%

\edefUnescapeHex
98 \def\edefUnescapeHex#1#2{%
99   \PE@edefbabel#1\pdfunescapehex{#2}%
100 }%

\edefEscapeName
101 \long\def\edefEscapeName#1#2{%
102   \PE@edefbabel#1\pdfescapename{#2}%
103 }%

\edefEscapeString
104 \long\def\edefEscapeString#1#2{%
105   \PE@edefbabel#1\pdfescapestring{#2}%
106 }%

107 \PE@AtEnd
108 \expandafter\endinput
109 \fi

```

2.4 Help macros

2.4.1 Characters

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

```

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

\PE@space@other
111 \begingroup
112   \catcode'\ =12\relax%
113 \def\x{\endgroup\def\PE@space@other{ }}\x\relax

\PE@space@space
114 \def\PE@space@space{ }

```

\PE@backslash

```

115 \begingroup
116   \long\def\@gobble#1{}%
117   \escapechar=92 %
118 \edef\x{\endgroup
119   \def\noexpand\PE@backslash{\expandafter\@gobble\string\\}%
120 }
121 \x

```

2.4.2 Switch for ε -TeX

```

122 \newif\ifPE@etex
123 \begingroup\expandafter\expandafter\expandafter\endgroup
124 \expandafter\ifx\csname numexpr\endcsname\relax
125 \else
126   \PE@etextrue
127 \fi

```

2.5 Sanitizing

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

```

128 \begingroup\expandafter\expandafter\expandafter\endgroup
129 \expandafter\ifx\csname detokenize\endcsname\relax
130   \long\def\PE@sanitize#1#2{%
131     \begingroup
132       \csname @safe@activetrue\endcsname
133       \edef#1{#2}%
134       \PE@onelevel@sanitize#1%
135     \expandafter\endgroup
136     \expandafter\def\expandafter#1\expandafter{#1}%
137   }%
138 \begingroup\expandafter\expandafter\expandafter\endgroup
139 \expandafter\ifx\csname @onelevel@sanitize\endcsname\relax
140   \def\PE@onelevel@sanitize#1{%
141     \edef#1{\expandafter\PE@strip@prefix\meaning#1}%
142   }%
143   \def\PE@strip@prefix#1>{}%
144 \else
145   \let\PE@onelevel@sanitize\@onelevel@sanitize
146 \fi
147 \else
148   \long\def\PE@sanitize#1#2{%
149     \begingroup
150       \csname @safe@activetrue\endcsname
151       \edef#1{#2}%
152       \edef#1{\detokenize\expandafter{#1}}%
153     \expandafter\endgroup
154     \expandafter\def\expandafter#1\expandafter{#1}%
155   }%
156 \fi

```

\PE@SpaceToOther

```

157 \def\PE@SpaceToOther#1 #2\relax{%
158   #1%
159   \ifx\\#2\\%
160   \else
161     \PE@space@other
162     \@ReturnAfterFi{%
163       \PE@SpaceToOther#2\relax
164     }%
165   \fi
166 }

```

\@ReturnAfterFi

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

2.6 Conversions

2.6.1 Conversion to hex string

\PE@EscapeHex

```
168 \ifPE@etex
169   \def\PE@EscapeHex#1{%
170     \edef#1{\expandafter\PE@ToHex#1\relax}%
171   }%
172 \else
173   \def\PE@EscapeHex#1{%
174     \def\PE@result{%
175       \expandafter\PE@ToHex#1\relax
176       \let#1\PE@result
177     }%
178 \fi
```

\PE@ToHex

```
179 \def\PE@ToHex#1{%
180   \ifx\relax#1%
181   \else
182     \PE@HexChar{#1}%
183     \expandafter\PE@ToHex
184   \fi
185 }%
```

\PE@HexChar

```
186 \ifPE@etex
187   \def\PE@HexChar#1{%
188     \PE@HexDigit{\numexpr\dimexpr.0625\dimexpr'#1sp\relax\relax\relax}%
189     \PE@HexDigit{%
190       \numexpr'#1-16*\dimexpr.0625\dimexpr'#1sp\relax\relax\relax
191     }%
192   }%
193 \else
194   \def\PE@HexChar#1{%
195     \dimen0='#1sp%
196     \dimen2=.0625\dimen0 %
197     \advance\dimen0-16\dimen2 %
198     \edef\PE@result{%
199       \PE@result
200       \PE@HexDigit{\dimen2 }%
201       \PE@HexDigit{\dimen0 }%
202     }%
203   }%
204 \fi
```

\PE@HexDigit

```
205 \def\PE@HexDigit#1{%
206   \expandafter\string
207   \ifcase#1%
208     0\or 1\or 2\or 3\or 4\or 5\or 6\or 7\or 8\or 9\or
209     A\or B\or C\or D\or E\or F%
210   \fi
211 }
```

2.6.2 Character code to octal number

\PE@OctChar

```
212 \ifPE@etex
213   \def\PE@OctChar#1{%
214     \expandafter\PE@OctChar
215       \the\numexpr\dimexpr.015625\dimexpr'#1sp\relax\relax
216       \expandafter\relax
217       \expandafter\relax
218       \the\numexpr\dimexpr.125\dimexpr'#1sp\relax\relax\relax
219       \relax
220       #1%
221   }%
222   \def\PE@@OctChar#1\relax#2\relax#3{%
223     \PE@backslash
224     #1%
225     \the\numexpr#2-8*#1\relax
226     \the\numexpr\dimexpr'#3sp\relax-8*#2\relax
227   }%
228 \else
229   \def\PE@OctChar#1{%
230     \dimen0='#1sp%
231     \dimen2=.125\dimen0 %
232     \dimen4=.125\dimen2 %
233     \advance\dimen0-8\dimen2 %
234     \advance\dimen2-8\dimen4 %
235     \edef\PE@result{%
236       \PE@result
237       \PE@backslash
238       \number\dimen4 %
239       \number\dimen2 %
240       \number\dimen0 %
241     }%
242   }%
243 \fi
```

2.6.3 Unpack hex string

\PE@UnescapeHex

```
244 \def\PE@UnescapeHex#1{%
245   \begingroup
246   \def\PE@result{}%
247   \expandafter\PE@DeHex#1\relax\relax
248   \expandafter\endgroup
249   \expandafter\def\expandafter#1\expandafter{\PE@result}%
250 }
```

\PE@DeHex

```
251 \def\PE@DeHex#1#2{%
252   \ifx#2\relax
253     \ifx#1\relax
254       \else
255         \PE@DeHex#10\relax\relax
256       \fi
257     \else
258       \uppercase{\lccode0="#1#2}\relax
259       \ifnum\lccode0=32 %
260         \edef\PE@result{\PE@result\PE@space@space}%
261       \else
262         \lowercase{\def\PE@temp{^^@}}%
263         \edef\PE@result{\PE@result\PE@temp}%
264       \fi
```



```

319 \edef\PE@result{%
320 \PE@result
321 #1%
322 }%
323 }%
324 \fi

```

2.6.5 Conversion to PDF string

\PE@EscapeString

```

325 \ifPE@etex
326 \def\PE@EscapeString#1{%
327 \edef#1{\expandafter\PE@EscapeStringTokens#1\relax}%
328 }%
329 \else
330 \def\PE@EscapeString#1{%
331 \begingroup
332 \def\PE@result{%
333 \expandafter\PE@EscapeStringTokens#1\relax
334 \expandafter\endgroup
335 \expandafter\def\expandafter#1\expandafter{\PE@result}%
336 }%
337 \fi

```

\PE@EscapeStringTokens

```

338 \def\PE@EscapeStringTokens#1{%
339 \ifx\relax#1%
340 \else
341 \ifnum'#1<33 %
342 \PE@OctChar#1%
343 \else
344 \ifnum'#1>126 %
345 \PE@OctChar#1%
346 \else \ifnum'#1=40 \PE@EscapeStringAdd{\string\}% (
347 \else\ifnum'#1=41 \PE@EscapeStringAdd{\string\)}% )
348 \else\ifnum'#1=92 \PE@EscapeStringAdd{\string\\}% \
349 \else
350 \PE@EscapeStringAdd{#1}%
351 \fi\fi\fi
352 \fi
353 \fi
354 \expandafter\PE@EscapeStringTokens
355 \fi
356 }%

```

\PE@EscapeStringAdd

```

357 \ifPE@etex
358 \def\PE@EscapeStringAdd#1{#1}%
359 \else
360 \def\PE@EscapeStringAdd#1{%
361 \edef\PE@result{%
362 \PE@result
363 #1%
364 }%
365 }%
366 \fi

367 \PE@AtEnd
368 \end{package}

```

3 Test

```
369 <*test1 | test2 | test3>
370 \NeedsTeXFormat{LaTeX2e}
371 \makeatletter
```

3.1 Test with \pdfescape... commands

```
372 <*test1>
373 \ProvidesFile{pdfescape-test1.tex}%
374 [2006/02/25 v1.1 Test with \string\pdfescape... commands]
375 </test1>
```

3.2 Test without \pdfescape..., with ε -TeX

```
376 <*test2>
377 \ProvidesFile{pdfescape-test2.tex}%
378 [2006/02/25 v1.1 Test without \string\pdfescape..., with e-TeX]
379 </test2>
```

3.3 Test without \pdfescape... and ε -TeX

```
380 <*test3>
381 \ProvidesFile{pdfescape-test3.tex}%
382 [2006/02/25 v1.1 Test without \string\pdfescape... and e-TeX]
383 </test3>
```

3.4 Check/ensure test preconditions

3.4.1 Check \pdfescape...

```
384 <*test1>
385 \@ifundefined{pdfescapehex}{%
386   \PackageError{pdfescape-test1}{%
387     Missing \string\pdfescape... commands%
388   }{Test aborted.}%
389   \stop
390 }{}
391 </test1>

392 <*test2 | test3>
393 \let\pdfescapehex\@undefined
394 \let\pdfunescapehex\@undefined
395 \let\pdfescapename\@undefined
396 \let\pdfescapestring\@undefined
397 </test2 | test3>
```

3.4.2 Check ε -TeX

```
398 <*test2>
399 \@ifundefined{numexpr}{%
400   \PackageError{pdfescape-test2}{%
401     Missing \eTeX
402   }{Test aborted.}%
403   \stop
404 }{}
405 </test2>
```

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

3.5 Common part

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

```
406 \RequirePackage{qstest}
407 \IncludeTests{*}
408 \LogTests{lgout}{*}{*}
```

```

409
410 \newcommand*{\ExpectVar}[2]{%
411   \Expect*{\ifx#1#2true\else false\fi}{true}%
412 }
413
414 \makeatletter
415
416
417 \begin{group}
418   \makeatletter
419   \gdef\AllBytes{}
420   \count@=0
421   \catcode0=12 %
422   \@whilenum\count@<256 \do{%
423     \lccode0=\count@
424     \ifnum\count@=32 %
425       \xdef\AllBytes{\AllBytes\space}%
426     \else
427       \lowercase{%
428         \xdef\AllBytes{\AllBytes^^@}%
429       }%
430     \fi
431     \advance\count@ by 1 %
432   }%
433 \end{group}
434 \newcommand*{\AllBytesHex}{%
435   000102030405060708090A0B0C0D0E0F%
436   101112131415161718191A1B1C1D1E1F%
437   202122232425262728292A2B2C2D2E2F%
438   303132333435363738393A3B3C3D3E3F%
439   404142434445464748494A4B4C4D4E4F%
440   505152535455565758595A5B5C5D5E5F%
441   606162636465666768696A6B6C6D6E6F%
442   707172737475767778797A7B7C7D7E7F%
443   808182838485868788898A8B8C8D8E8F%
444   909192939495969798999A9B9C9D9E9F%
445   A0A1A2A3A4A5A6A7A8A9AABACADA EAF%
446   B0B1B2B3B4B5B6B7B8B9BABBB CBDEBF%
447   C0C1C2C3C4C5C6C7C8C9CACBCCDCECF%
448   D0D1D2D3D4D5D6D7D8D9DADBDCDDDEDF%
449   E0E1E2E3E4E5E6E7E8E9EAEBECEDEEEF%
450   F0F1F2F3F4F5F6F7F8F9FABFBCFDFEFF%
451 }
452 \@onelevel@sanitize\AllBytesHex
453 \expandafter\lowercase\expandafter{%
454   \expandafter\newcommand\expandafter*\expandafter\AllBytesHexLC
455     \expandafter{\AllBytesHex}%
456 }
457 \newcommand*{\AllBytesName}{%
458 \begin{group}
459   \catcode'\#=12 %
460   \xdef\AllBytesName{%
461     #01#02#03#04#05#06#07#08#09#0A#0B#0C#0D#0E#0F%
462     #10#11#12#13#14#15#16#17#18#19#1A#1B#1C#1D#1E#1F%
463     #20!"#23$#25&'#28#29*+,-.#2F%
464     0123456789:;#3C=#3E?%
465     @ABCDEFGHIJKLMNO%
466     PQRSTUVWXYZ#5B\@backslashchar#5D^_%
467     'abcdefghijklmnopqrstuvwxyz%
468     pqrstuvwxyz#7B|#7D\string~#7F%
469     #80#81#82#83#84#85#86#87#88#89#8A#8B#8C#8D#8E#8F%
470     #90#91#92#93#94#95#96#97#98#99#9A#9B#9C#9D#9E#9F%

```

```

471   #A0#A1#A2#A3#A4#A5#A6#A7#A8#A9#AA#AB#AC#AD#AE#AF%
472   #B0#B1#B2#B3#B4#B5#B6#B7#B8#B9#BA#BB#BC#BD#BE#BF%
473   #C0#C1#C2#C3#C4#C5#C6#C7#C8#C9#CA#CB#CC#CD#CE#CF%
474   #D0#D1#D2#D3#D4#D5#D6#D7#D8#D9#DA#DB#DC#DD#DE#DF%
475   #E0#E1#E2#E3#E4#E5#E6#E7#E8#E9#EA#EB#EC#ED#EE#EF%
476   #F0#F1#F2#F3#F4#F5#F6#F7#F8#F9#FA#FB#FC#FD#FE#FF%
477   }
478 \endgroup
479 \@onelevel@sanitize\AllBytesName
480
481 \newcommand*{\AllBytesString}{
482 \begingroup
483   \def\|{|}
484   \edef%\{@percentchar}
485   \catcode'\|0 %
486   \catcode'\#=12 %
487   \catcode'\~=12 %
488   \catcode'\|=12 %
489   \xdef\AllBytesString{%
490     \000\001\002\003\004\005\006\007\010\011\012\013\014\015\016\017%
491     \020\021\022\023\024\025\026\027\030\031\032\033\034\035\036\037%
492     \040!"#$%&'(\)*+,-./%
493     0123456789:;<=>?%
494     @ABCDEFGHIJKLMNO%
495     PQRSTUVWXYZ[\]^_%
496     'abcdefghijklmno%
497     pqrstuvwxyz{|}~\177%
498     \200\201\202\203\204\205\206\207\210\211\212\213\214\215\216\217%
499     \220\221\222\223\224\225\226\227\230\231\232\233\234\235\236\237%
500     \240\241\242\243\244\245\246\247\250\251\252\253\254\255\256\257%
501     \260\261\262\263\264\265\266\267\270\271\272\273\274\275\276\277%
502     \300\301\302\303\304\305\306\307\310\311\312\313\314\315\316\317%
503     \320\321\322\323\324\325\326\327\330\331\332\333\334\335\336\337%
504     \340\341\342\343\344\345\346\347\350\351\352\353\354\355\356\357%
505     \360\361\362\363\364\365\366\367\370\371\372\373\374\375\376\377%
506   }
507 \endgroup
508 \@onelevel@sanitize\AllBytesString
509
510 <test3>
511 \let\org@detokenize\detokenize
512 \let\detokenize\@undefined
513 \let\org@numexpr\numexpr
514 \let\numexpr\@undefined
515 </test3>
516 \RequirePackage{pdfescape}
517 <test3>
518 \let\detokenize\org@detokenize
519 \let\numexpr\org@numexpr
520 </test3>
521
522 \begin{qstest}{all-hex}{\AllBytes, escapehex}
523   \EdefEscapeHex\x{\AllBytes}
524   \Expect*{\x}*{\AllBytesHex}%
525   \ExpectVar\x\AllBytesHex
526 \end{qstest}
527
528 \begin{qstest}{all-unhex}{\AllBytesHex, unescapehex}
529   \EdefUnescapeHex\x{\AllBytesHex}
530   \Expect*{\x}*{\AllBytes}
531   \ExpectVar\x\AllBytes
532 \end{qstest}

```

```

533
534 \begin{qstest}{all-unhex-lc}{\AllBytesHexLC, unescapehex, lowercase}
535   \EdefUnescapeHex\x{\AllBytesHexLC}
536   \Expect*{\x}*{\AllBytes}
537   \ExpectVar\x\AllBytes
538 \end{qstest}
539
540 \begin{qstest}{unhex-incomplete}{unescapehex, incomplete}
541   \EdefUnescapeHex\x{4}
542   \Expect*{\x}{0}
543 \end{qstest}
544
545 \begin{qstest}{unhex-space}{unescapehex, space}
546   \EdefUnescapeHex\x{20}
547   \Expect*{\x}{ }
548   \ExpectVar\x\space
549 \end{qstest}
550
551 \begin{qstest}{unhex-spaces}{unescapehex, spaces}
552   \EdefUnescapeHex\x{204020204120}
553   \def\y#1{%
554     \edef\z{#1\string @#1#1\string A#1}%
555     }\y{ }
556   \Expect*{\x}*{\z}
557   \ExpectVar\x\z
558 \end{qstest}
559
560 \begin{qstest}{all-name}{\AllBytes, escapename}
561   \EdefEscapeName\x{\AllBytes}
562   \Expect*{\x}*{\AllBytesName}
563   \ExpectVar\x\AllBytesName
564 \end{qstest}
565
566 \begin{qstest}{all-string}{\AllBytes, escapestring}
567   \EdefEscapeString\x{\AllBytes}
568   \Expect*{\x}*{\AllBytesString}
569   \ExpectVar\x\AllBytesString
570 \end{qstest}
571
572 \stop
573 </test1 | test2 | test3>

```

4 Installation

CTAN. This package is available on CTAN¹:

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

[CTAN:macros/latex/contrib/oberdiek/pdfescape.pdf](#) Documentation.

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):

¹<http://ftp.ctan.org/tex-archive/>

pdfescape.sty	→	tex/generic/oberdiek/pdfescape.sty
pdfescape.pdf	→	doc/latex/oberdiek/pdfescape.pdf
pdfescape-test1.tex	→	doc/latex/oberdiek/pdfescape-test1.tex
pdfescape-test2.tex	→	doc/latex/oberdiek/pdfescape-test2.tex
pdfescape-test3.tex	→	doc/latex/oberdiek/pdfescape-test3.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`.

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

4.1 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 `pdf \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.

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	C
<code>\#</code> 459, 486	<code>\catcode</code> 43, 45, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 112, 421, 459, 485, 486, 487, 488
<code>\%</code> 484	<code>\count@</code> 420, 422, 423, 424, 431
<code>\(</code> 346, 492	<code>\csname</code> 3, 13, 26, 39, 42, 67, 90, 124, 129, 132, 139, 150
<code>\)</code> 347, 492	
<code>\@ReturnAfterFi</code> 162, <u>167</u>	
<code>\@backslashchar</code> 466	
<code>\@gobble</code> 116, 119	
<code>\@ifundefined</code> 385, 399	
<code>\@onelevel@sanitize</code> 145, 452, 479, 508	
<code>\@percentchar</code> 484	
<code>\@undefined</code> 393, 394, 395, 396, 512, 514	
<code>\@whilenum</code> 422	
<code>\\</code> 119, 159, 348, 488, 495	
<code>\ </code> 483, 485	
<code>\~</code> 487	
Numbers	D
<code>\0</code> 490, 491, 492	<code>\detokenize</code> 152, 511, 512, 518
<code>\1</code> 497	<code>\dimen</code> . 195, 196, 197, 200, 201, 230, 231, 232, 233, 234, 238, 239, 240
<code>\2</code> 498, 499, 500, 501	<code>\dimexpr</code> 188, 190, 215, 218, 226
<code>\3</code> 502, 503, 504, 505	<code>\do</code> 422
<code>_</code> 112, 348	
A	E
<code>\advance</code> 197, 233, 234, 431	<code>\EdefEscapeHex</code> 2, 68, 95, 523
<code>\AllBytes</code> 419, 425, 428, 522, 523, 530, 531, 536, 537, 560, 561, 566, 567	<code>\EdefEscapeName</code> 77, <u>101</u> , 561
<code>\AllBytesHex</code> 434, 452, 455, 524, 525, 528, 529	<code>\EdefEscapeString</code> 82, <u>104</u> , 567
<code>\AllBytesHexLC</code> 454, 534, 535	<code>\EdefUnescapeHex</code> 73, 98, 529, 535, 541, 546, 552
<code>\AllBytesName</code> . 457, 460, 479, 562, 563	<code>\empty</code> 7
<code>\AllBytesString</code> ... 481, 508, 568, 569	<code>\end</code> 526, 532, 538, 543, 549, 558, 564, 570
B	<code>\endcsname</code> 3, 13, 26, 39, 42, 67, 90, 124, 129, 132, 139, 150
<code>\begin</code> 522, 528, 534, 540, 545, 551, 560, 566	<code>\endinput</code> 22, 108
	<code>\escapechar</code> 117
	<code>\eTeX</code> 401
	<code>\Expect</code> 411, 524, 530, 536, 542, 547, 556, 562, 568
	<code>\ExpectVar</code> 410, 525, 531, 537, 548, 557, 563, 569
	G
	<code>\gdef</code> 419
	I
	<code>\ifcase</code> 4, 207, 283
	<code>\ifnum</code> 259, 282, 290, 293, 294, 295, 296,

297, 298, 299, 300, 301, 302, 303, 341, 344, 346, 347, 348, 424	\PE@EscapeString 85, <u>325</u>
\ifPE@etex 122, 168, 186, 212, 268, 315, 325, 357	\PE@EscapeStringAdd 346, 347, 348, 350, <u>357</u>
\ifx 5, 7, 13, 26, 34, 67, 124, 129, 139, 159, 180, 252, 253, 280, 339, 411	\PE@EscapeStringTokens 327, 333, <u>338</u>
\immediate 15, 28	\PE@etextrue 126
\IncludeTests 407	\PE@hash 110, 286, 291, 313
L	\PE@HexChar 182, <u>186</u> , 287, 292
\lccode 258, 259, 423	\PE@HexDigit .. 188, 189, 200, 201, <u>205</u>
\LogTests 408	\PE@OctChar <u>212</u> , 342, 345
\lowercase 262, 427, 453	\PE@onelevel@sanitize . 134, 140, 145
M	\PE@result 174, 176, 198, 199, 235, 236, 246, 249, 260, 263, 274, 276, 319, 320, 332, 335, 361, 362
\makeatletter 371, 414, 418	\PE@sanitize 69, 74, 78, 83, <u>128</u>
\meaning 141	\PE@space@other <u>111</u> , 161
N	\PE@space@space <u>114</u> , 260
\NeedsTeXFormat 370	\PE@SpaceToOther 70, 79, 84, <u>157</u>
\newcommand ... 410, 434, 454, 457, 481	\PE@strip@prefix 141, 143
\newif 122	\PE@temp 262, 263
\number 238, 239, 240	\PE@ToHex 170, 175, <u>179</u>
\numexpr 188, 190, 215, 218, 225, 226, 513, 514, 519	\PE@UnescapeHex 75, <u>244</u>
	\ProvidesFile 373, 377, 381
	\ProvidesPackage 40
	R
	\RequirePackage 406, 516
O	S
\org@detokenize 511, 518	\space 425, 548
\org@numexpr 513, 519	\stop 389, 403, 572
P	T
\PackageError 386, 400	\the 43, 49, 215, 218, 225, 226
\PackageInfo 18	U
\pdfescape 374, 378, 382, 387	\uccode 63, 64, 65
\pdfescapehex 96, 393	\uppercase 258
\pdfescapehex 102, 395	W
\pdfescapestring 105, 396	\write 15, 28
\pdfunescapehex 99, 394	X
\PE@@OctChar 214, 222	\x 3, 5, 7, 14, 18, 20, 27, 32, 39, 113, 118, 121, 523, 524, 525, 529, 530, 531, 535, 536, 537, 541, 542, 546, 547, 548, 552, 556, 557, 561, 562, 563, 567, 568, 569
\PE@AtEnd 47, 48, 107, 367	Y
\PE@backslash <u>115</u> , 223, 237	\y 553, 555
\PE@DeHex 247, <u>251</u>	Z
\PE@edefbabel <u>88</u> , 96, 99, 102, 105	\z 554, 556, 557
\PE@EnsureCode 46, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65	
\PE@EscapeHex 71, <u>168</u>	
\PE@EscapeName 80, <u>268</u>	
\PE@EscapeNameAdd 286, 291, 305, 313, <u>315</u>	
\PE@EscapeNameHashChar 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 312	
\PE@EscapeNameTokens .. 270, 275, <u>279</u>	