

The pdfescape package

Heiko Oberdiek
<heiko.oberdiek at gmail.com>

2011/04/04 v1.12

Abstract

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

Contents

| | | |
|----------|--|-----------|
| 1 | Documentation | 2 |
| 1.1 | Additional unescape macros | 2 |
| 1.2 | Sanitizing macro | 3 |
| 2 | Implementation | 3 |
| 2.1 | Reload check and package identification | 3 |
| 2.2 | Catcodes | 4 |
| 2.3 | Load package | 5 |
| 2.4 | Sanitizing | 5 |
| 2.4.1 | Space characters | 6 |
| 2.4.2 | Space normalization | 6 |
| 2.5 | <code>\EdefUnescapeName</code> | 6 |
| 2.6 | <code>\EdefUnescapeString</code> | 8 |
| 2.7 | User macros (pdfTeX analogues) | 11 |
| 2.8 | Help macros | 12 |
| 2.8.1 | Characters | 12 |
| 2.8.2 | Switch for ϵ -TeX | 12 |
| 2.9 | Conversions | 12 |
| 2.9.1 | Conversion to hex string | 12 |
| 2.9.2 | Character code to octal number | 13 |
| 2.9.3 | Unpack hex string | 14 |
| 2.9.4 | Conversion to PDF name | 15 |
| 2.9.5 | Conversion to PDF string | 16 |
| 3 | Test | 17 |
| 3.1 | Catcode checks for loading | 17 |
| 3.2 | Macro tests | 18 |
| 3.3 | Test with <code>\pdfescape...</code> commands | 18 |
| 3.4 | Test without <code>\pdfescape...</code> , with ϵ -TeX | 18 |
| 3.5 | Test without <code>\pdfescape...</code> and ϵ -TeX | 19 |
| 3.6 | Test with LuaTeX | 19 |
| 3.7 | Check/ensure test preconditions | 19 |
| 3.7.1 | Check <code>\pdfescape...</code> | 19 |
| 3.7.2 | Check ϵ -TeX | 19 |
| 3.7.3 | Check LuaTeX | 19 |
| 3.8 | Common part | 19 |
| 3.8.1 | Test for iniTeX | 25 |

| | | |
|--------------------|---------------------------------|-----------|
| 4 | Installation | 28 |
| 4.1 | Download | 28 |
| 4.2 | Bundle installation | 28 |
| 4.3 | Package installation | 28 |
| 4.4 | Refresh file name databases | 28 |
| 4.5 | Some details for the interested | 29 |
| 5 | History | 29 |
| [2007/02/21 v1.0] | | 29 |
| [2007/02/25 v1.1] | | 29 |
| [2007/03/20 v1.2] | | 29 |
| [2007/04/11 v1.3] | | 30 |
| [2007/04/21 v1.4] | | 30 |
| [2007/08/27 v1.5] | | 30 |
| [2007/09/09 v1.6] | | 30 |
| [2007/10/27 v1.7] | | 30 |
| [2007/11/11 v1.8] | | 30 |
| [2010/03/01 v1.9] | | 30 |
| [2010/11/12 v1.10] | | 30 |
| [2011/01/30 v1.11] | | 30 |
| [2011/04/04 v1.12] | | 30 |
| 6 | Index | 30 |

1 Documentation

```
\EdefEscapeHex {\cmd} {\string}
\EdefUnescapeHex {\cmd} {\string}
\EdefEscapeName {\cmd} {\string}
\EdefEscapeString {\cmd} {\string}
```

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 ≥ 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 {\langle cmd\rangle} {\langle string\rangle}`

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

`\EdefSanitize {\langle cmd\rangle} {\langle string\rangle}`

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 $\langle *package \rangle$

2.1 Reload check and package identification

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

```

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

Package identification:

```

33 \begingroup\catcode61\catcode48\catcode32=10\relax%
34  \catcode13=5 % ^^M
35  \endlinechar=13 %
36  \catcode35=6 % #
37  \catcode39=12 % '
38  \catcode40=12 % (
39  \catcode41=12 % )
40  \catcode44=12 % ,
```

```

41 \catcode45=12 % -
42 \catcode46=12 % .
43 \catcode47=12 % /
44 \catcode58=12 % :
45 \catcode64=11 % @
46 \catcode91=12 % [
47 \catcode93=12 % ]
48 \catcode123=1 % {
49 \catcode125=2 % }
50 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
51   \def\x#1#2#3[#4]{\endgroup
52     \immediate\write-1{Package: #3 #4}%
53     \xdef#1{#4}%
54   }%
55 \else
56   \def\x#1#2[#3]{\endgroup
57     #2[{#3}]%
58     \ifx#1\@undefined
59       \xdef#1{#3}%
60     \fi
61     \ifx#1\relax
62       \xdef#1{#3}%
63     \fi
64   }%
65 \fi
66 \expandafter\x\csname ver@pdfescape.sty\endcsname
67 \ProvidesPackage{pdfescape}%
68 [2011/04/04 v1.12 Provides string conversions (HO)]%

```

2.2 Catcodes

```

69 \begingroup\catcode61\catcode48\catcode32=10\relax%
70 \catcode13=5 % ^^M
71 \endlinechar=13 %
72 \catcode123 1 % {
73 \catcode125 2 % }
74 \catcode64 11 %
75 \def\x{\endgroup
76   \expandafter\edef\csname PE@AtEnd\endcsname{%
77     \endlinechar=\the\endlinechar\relax
78     \catcode13=\the\catcode13\relax
79     \catcode32=\the\catcode32\relax
80     \catcode35=\the\catcode35\relax
81     \catcode61=\the\catcode61\relax
82     \catcode64=\the\catcode64\relax
83     \catcode123=\the\catcode123\relax
84     \catcode125=\the\catcode125\relax
85   }%
86 }%
87 \x\catcode61\catcode48\catcode32=10\relax%
88 \catcode13=5 % ^^M
89 \endlinechar=13 %
90 \catcode35=6 % #
91 \catcode64=11 % @
92 \catcode123=1 % {
93 \catcode125=2 % }
94 \def\TMP@EnsureCode#1#2#3{%
95   \edef\PE@AtEnd{%
96     \PE@AtEnd
97     #1#2=\the#1#2\relax
98   }%
99   #1#2=#3\relax

```

```

100 }
101 \TMP@EnsureCode\catcode{0}{12}% ^^@
102 \TMP@EnsureCode\catcode{34}{12}% "
103 \TMP@EnsureCode\catcode{39}{12}% '
104 \TMP@EnsureCode\catcode{42}{12}% *
105 \TMP@EnsureCode\catcode{45}{12}% -
106 \TMP@EnsureCode\catcode{46}{12}% .
107 \TMP@EnsureCode\catcode{47}{12}% /
108 \TMP@EnsureCode\catcode{60}{12}% <
109 \TMP@EnsureCode\catcode{62}{12}% >
110 \TMP@EnsureCode\catcode{91}{12}% [
111 \TMP@EnsureCode\catcode{93}{12}% ]
112 \TMP@EnsureCode\catcode{94}{7}% ^
113 \TMP@EnsureCode\catcode{96}{12}% '
114 \TMP@EnsureCode\uccode{34}{0}% "
115 \TMP@EnsureCode\uccode{48}{0}% 0
116 \TMP@EnsureCode\uccode{61}{0}% =
117 \edef\PE@AtEnd{\PE@AtEnd\noexpand\endinput}

```

2.3 Load package

```

118 \begingroup\expandafter\expandafter\expandafter\endgroup
119 \expandafter\ifx\csname RequirePackage\endcsname\relax
120   \def\TMP@RequirePackage#1[#2]{%
121     \begingroup\expandafter\expandafter\expandafter\endgroup
122     \expandafter\ifx\csname ver@#1.sty\endcsname\relax
123       \input #1.sty\relax
124     \fi
125   }%
126   \TMP@RequirePackage{ltxcmds}[2010/04/08]%
127 \else
128   \RequirePackage{ltxcmds}[2010/04/08]%
129 \fi

```

2.4 Sanitizing

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

```

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

```

```

156     \detokenize\expandafter{#1}%
157   }%
158 }%
159 \def\PE@onelevel@sanitize#1{%
160   \edef#1{\detokenize\expandafter{#1}}%
161 }%
162 \fi

```

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

```

163 \let\PE@sanitize\EdefSanitize

```

2.4.1 Space characters

`\PE@space@other`

```

164 \begingroup
165   \catcode'\ =12\relax%
166 \def\x{\endgroup\def\PE@space@other{ }}\x\relax

```

`\PE@space@space`

```

167 \def\PE@space@space{ }

```

2.4.2 Space normalization

`\PE@SanitizeSpaceOther`

```

168 \def\PE@SanitizeSpaceOther#1{%
169   \edef#1{\expandafter\PE@SpaceToOther#1 \relax}%
170 }

```

`\PE@SpaceToOther`

```

171 \def\PE@SpaceToOther#1 #2\relax{%
172   #1%
173   \ifx\#2\%
174     \else
175       \PE@space@other
176       \ltx@ReturnAfterFi{%
177         \PE@SpaceToOther#2\relax
178       }%
179     \fi
180 }

```

2.5 \EdefUnescapeName

`\EdefUnescapeName`

```

181 \def\EdefUnescapeName#1#2{%
182   \EdefSanitize#1{#2}%
183   \PE@SanitizeSpaceOther#1%
184   \PE@UnescapeName#1%
185   \PE@onelevel@sanitize#1%
186 }

```

`\PE@UnescapeName`

```

187 \begingroup
188   \catcode'\$=6 % hash
189   \catcode'\#=12 % other
190   \gdef\PE@UnescapeName$1{%
191     \begingroup
192       \PE@InitUccodeHexDigit
193       \def\PE@result{}%
194       \expandafter\PE@DeName$1#\relax\relax

```

```

195 \expandafter\endgroup
196 \expandafter\def\expandafter$1\expandafter{\PE@result}%
197 }%
198 \gdef\PE@DeName$1#$2$3{%
199 \ifx\relax$2%
200 \edef\PE@result{\PE@result$1}%
201 \let\PE@next\relax
202 \else
203 \ifx\relax$3%
204 % wrong escape sequence in input
205 \edef\PE@result{\PE@result$1#}%
206 \let\PE@next\relax
207 \else
208 \uppercase{%
209 \def\PE@testA{$2}%
210 \def\PE@testB{$3}%
211 }%
212 \ifcase\ifcase\expandafter\PE@TestUcHexDigit\PE@testA
213 \ifcase\expandafter\PE@TestUcHexDigit\PE@testB
214 \ltx@zero
215 \else
216 \ltx@one
217 \fi
218 \else
219 \ltx@one
220 \fi
221 \uccode\ltx@zero="\PE@testA\PE@testB\relax
222 \uppercase{%
223 \def\PE@temp{^^@}%
224 }%
225 \uccode\ltx@zero=\ltx@zero
226 \edef\PE@result{\PE@result$1\PE@temp}%
227 \let\PE@next\PE@DeName
228 \else
229 % wrong escape sequence in input
230 \edef\PE@result{\PE@result$1#}%
231 \def\PE@next{\PE@DeName$2$3}%
232 \fi
233 \fi
234 \fi
235 \PE@next
236 }%
237 \endgroup

```

\PE@InitUccodeHexDigit

```

238 \def\PE@InitUccodeHexDigit{%
239 \uccode'a='A\relax
240 \uccode'b='B\relax
241 \uccode'c='C\relax
242 \uccode'd='D\relax
243 \uccode'e='E\relax
244 \uccode'f='F\relax
245 \uccode'A=\ltx@zero
246 \uccode'B=\ltx@zero
247 \uccode'C=\ltx@zero
248 \uccode'D=\ltx@zero
249 \uccode'E=\ltx@zero
250 \uccode'F=\ltx@zero
251 \uccode'0=\ltx@zero
252 \uccode'1=\ltx@zero
253 \uccode'2=\ltx@zero
254 \uccode'3=\ltx@zero
255 \uccode'4=\ltx@zero

```

```

256 \uccode'5=\ltx@zero
257 \uccode'6=\ltx@zero
258 \uccode'7=\ltx@zero
259 \uccode'8=\ltx@zero
260 \uccode'9=\ltx@zero
261 }

```

\PE@TestUcHexDigit

```

262 \def\PE@TestUcHexDigit#1{%
263 \ifnum'#1<48 % 0
264 \ltx@one
265 \else
266 \ifnum'#1>70 % F
267 \ltx@one
268 \else
269 \ifnum'#1>57 % 9
270 \ifnum'#1<65 % A
271 \ltx@one
272 \else
273 \ltx@zero
274 \fi
275 \else
276 \ltx@zero
277 \fi
278 \fi
279 \fi
280 }

```

2.6 \EdefUnescapeString

\EdefUnescapeString

```

281 \def\EdefUnescapeString#1#2{%
282 \EdefSanitize#1{#2}%
283 \PE@SanitizeSpaceOther#1%
284 \PE@NormalizeLineEnd#1%
285 \PE@UnescapeString#1%
286 \PE@onelevel@sanitize#1%
287 }

288 \begingroup
289 \uccode'\8=10 % lf
290 \uccode'\9=13 % cr
291 \def\x#1#2{\endgroup

```

\PE@NormalizeLineEnd

```

292 \def\PE@NormalizeLineEnd##1{%
293 \def\PE@result{}%
294 \expandafter\PE@@NormalizeLineEnd##1#2\relax
295 \let##1\PE@result
296 }%

```

\PE@@NormalizeLineEnd

```

297 \def\PE@@NormalizeLineEnd##1#2##2{%
298 \ifx\relax##2%
299 \edef\PE@result{\PE@result##1}%
300 \let\PE@next\relax
301 \else
302 \edef\PE@result{\PE@result##1#1}%
303 \ifx#1##2% lf
304 \let\PE@next\PE@@NormalizeLineEnd
305 \else
306 \def\PE@next{\PE@@NormalizeLineEnd##2}%

```



```

307     \fi
308     \fi
309     \PE@next
310 }%
311 }%
312 \uppercase{%
313   \x 89%
314 }

315 \begingroup
316   \catcode'\|=0 %
317   \catcode'\|=12 %

```

\PE@UnescapeString

```

318 |gdef|PE@UnescapeString#1{%
319   |begingroup
320     |def|PE@result{%}%
321     |expandafter|PE@DeString#1\|relax
322     |expandafter|endgroup
323     |expandafter|def|expandafter#1|expandafter{|PE@result}%
324 }%

```

\PE@DeString

```

325 |gdef|PE@DeString#1\#2{%
326   |ifx|relax#2%
327     |edef|PE@result{|PE@result#1}%
328     |let|PE@next|relax
329   |else
330     |if n#2%
331       |uccode|ltx@zero=10 %
332     |else|if r#2%
333       |uccode|ltx@zero=13 %
334     |else|if t#2%
335       |uccode|ltx@zero=9 %
336     |else|if b#2%
337       |uccode|ltx@zero=8 %
338     |else|if f#2%
339       |uccode|ltx@zero=12 %
340     |else
341       |uccode|ltx@zero=|ltx@zero
342     |fi|fi|fi|fi|fi
343     |ifnum|uccode|ltx@zero>|ltx@zero
344       |uppercase{%
345         |edef|PE@temp{^^@}%
346       }%
347       |edef|PE@result{|PE@result#1|PE@temp}%
348       |let|PE@next|PE@DeString
349     |else
350       |if\#2% backslash
351         |edef|PE@result{|PE@result#1}%
352         |let|PE@next|PE@CheckEndBackslash
353       |else
354         |ifnum'#2=10 % linefeed
355         |edef|PE@result{|PE@result#1}%
356         |let|PE@next|PE@DeString
357       |else
358         |ifcase|PE@TestOctDigit#2%
359         |edef|PE@result{|PE@result#1}%
360         |def|PE@next{|PE@OctI#2}%
361       |else
362         |edef|PE@result{|PE@result#1#2}%
363         |let|PE@next|PE@DeString

```

```

364         |fi
365     |fi
366 |fi
367 |fi
368 |fi
369 |PE@next
370 }%

\PE@CheckEndBackslash

371 |gdef|PE@CheckEndBackslash#1{%
372 |ifx|relax#1%
373 |else
374 |edef|PE@result{|PE@result\}%
375 |expandafter|PE@DeString|expandafter#1%
376 |fi
377 }%

378 |endgroup

\PE@TestOctDigit

379 \def\PE@TestOctDigit#1{%
380 \ifnum'#1<48 % 0
381 \ltx@one
382 \else
383 \ifnum'#1>55 % 7
384 \ltx@one
385 \else
386 \ltx@zero
387 \fi
388 \fi
389 }

\PE@OctI

390 \def\PE@OctI#1#2{%
391 \ifcase\PE@TestOctDigit#2%
392 \def\PE@next{\PE@OctII{#1#2}}%
393 \else
394 \def\PE@next{\PE@OctAll#1#2}%
395 \fi
396 \PE@next
397 }

\PE@OctII

398 \def\PE@OctII#1#2{%
399 \ifcase\PE@TestOctDigit#2%
400 \def\PE@next{\PE@OctAll{#1#2}}%
401 \else
402 \def\PE@next{\PE@OctAll{#1}#2}%
403 \fi
404 \PE@next
405 }

\PE@OctAll

406 \def\PE@OctAll#1{%
407 \uccode\ltx@zero='#1\relax
408 \uppercase{%
409 \edef\PE@result{\PE@result^^@}%
410 }%
411 \PE@DeString
412 }

```

2.7 User macros (pdfTeX analogues)

```

413 \begingroup\expandafter\expandafter\expandafter\endgroup
414 \expandafter\ifx\csname RequirePackage\endcsname\relax
415   \def\TMP@RequirePackage#1[#2]{%
416     \begingroup\expandafter\expandafter\expandafter\endgroup
417     \expandafter\ifx\csname ver@#1.sty\endcsname\relax
418       \input #1.sty\relax
419     \fi
420   }%
421   \TMP@RequirePackage{pdftexcmds}[2007/11/11]%
422 \else
423   \RequirePackage{pdftexcmds}[2007/11/11]%
424 \fi

425 \begingroup\expandafter\expandafter\expandafter\endgroup
426 \expandafter\ifx\csname pdf@escapehex\endcsname\relax

\edefEscapeHex
427   \long\def\edefEscapeHex#1#2{%
428     \edefSanitize#1{#2}%
429     \PE@SanitizeSpaceOther#1%
430     \PE@EscapeHex#1%
431   }%

\edefUnescapeHex
432   \def\edefUnescapeHex#1#2{%
433     \edefSanitize#1{#2}%
434     \PE@UnescapeHex#1%
435   }%

\edefEscapeName
436   \long\def\edefEscapeName#1#2{%
437     \edefSanitize#1{#2}%
438     \PE@SanitizeSpaceOther#1%
439     \PE@EscapeName#1%
440   }%

\edefEscapeString
441   \long\def\edefEscapeString#1#2{%
442     \edefSanitize#1{#2}%
443     \PE@SanitizeSpaceOther#1%
444     \PE@EscapeString#1%
445   }%
446 \else

\PE@edefbabel Help macro that adds support for babel's shorthand characters.
447   \long\def\PE@edefbabel#1#2#3{%
448     \begingroup
449       \csname @save@activetrue\endcsname
450       \edef#1{#2{#3}}%
451     \expandafter\endgroup
452     \expandafter\def\expandafter#1\expandafter{#1}%
453   }%

\edefEscapeHex
454   \long\def\edefEscapeHex#1#2{%
455     \PE@edefbabel#1\pdf@escapehex{#2}%
456   }%

```

`\EdefUnescapeHex`

```
457 \def\EdefUnescapeHex#1#2{%
458   \PE@edefbabel#1\pdf@unescapehex{#2}%
459 }%
```

`\EdefEscapeName`

```
460 \long\def\EdefEscapeName#1#2{%
461   \PE@edefbabel#1\pdf@escapename{#2}%
462 }%
```

`\EdefEscapeString`

```
463 \long\def\EdefEscapeString#1#2{%
464   \PE@edefbabel#1\pdf@escapestring{#2}%
465 }%

466 \expandafter\PE@AtEnd
467 \fi%
```

2.8 Help macros

2.8.1 Characters

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

`\PE@hash`

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

`\PE@backslash`

```
469 \begingroup
470   \escapechar=-1 %
471 \edef\x{\endgroup
472   \def\noexpand\PE@backslash{\string\\}%
473 }
474 \x
```

2.8.2 Switch for ϵ -TeX

```
475 \ltx@newif\ifPE@etex
476 \begingroup\expandafter\expandafter\expandafter\endgroup
477 \expandafter\ifx\csname numexpr\endcsname\relax
478 \else
479   \PE@etextrue
480 \fi
```

2.9 Conversions

2.9.1 Conversion to hex string

`\PE@EscapeHex`

```
481 \ifPE@etex
482   \def\PE@EscapeHex#1{%
483     \edef#1{\expandafter\PE@ToHex#1\relax}%
484   }%
485 \else
486   \def\PE@EscapeHex#1{%
487     \def\PE@result{%
488       \expandafter\PE@ToHex#1\relax
489     }\let#1\PE@result
490   }%
491 \fi
```

\PE@ToHex

```
492 \def\PE@ToHex#1{%
493   \ifx\relax#1%
494   \else
495     \PE@HexChar{#1}%
496     \expandafter\PE@ToHex
497   \fi
498 }%
```

\PE@HexChar

```
499 \ifPE@etex
500   \def\PE@HexChar#1{%
501     \PE@HexDigit{\numexpr\dimexpr.0625\dimexpr'#1sp\relax\relax\relax}%
502     \PE@HexDigit{%
503       \numexpr'#1-16*\dimexpr.0625\dimexpr'#1sp\relax\relax\relax
504     }%
505   }%
506 \else
507   \def\PE@HexChar#1{%
508     \dimen0='#1sp%
509     \dimen2=.0625\dimen0 %
510     \advance\dimen0-16\dimen2 %
511     \edef\PE@result{%
512       \PE@result
513       \PE@HexDigit{\dimen2 }%
514       \PE@HexDigit{\dimen0 }%
515     }%
516   }%
517 \fi
```

\PE@HexDigit

```
518 \def\PE@HexDigit#1{%
519   \expandafter\string
520   \ifcase#1%
521     0\or 1\or 2\or 3\or 4\or 5\or 6\or 7\or 8\or 9\or
522     A\or B\or C\or D\or E\or F%
523   \fi
524 }
```

2.9.2 Character code to octal number

\PE@OctChar

```
525 \ifPE@etex
526   \def\PE@OctChar#1{%
527     \expandafter\PE@@OctChar
528     \the\numexpr\dimexpr.015625\dimexpr'#1sp\relax\relax
529     \expandafter\relax
530     \expandafter\relax
531     \the\numexpr\dimexpr.125\dimexpr'#1sp\relax\relax\relax
532     \relax
533     #1%
534   }%
535   \def\PE@@OctChar#1\relax#2\relax#3{%
536     \PE@backslash
537     #1%
538     \the\numexpr#2-8*#1\relax
539     \the\numexpr\dimexpr'#3sp\relax-8*#2\relax
540   }%
541 \else
542   \def\PE@OctChar#1{%
543     \dimen0='#1sp%
```

```

544 \dimen2=.125\dimen0 %
545 \dimen4=.125\dimen2 %
546 \advance\dimen0-8\dimen2 %
547 \advance\dimen2-8\dimen4 %
548 \edef\PE@result{%
549 \PE@result
550 \PE@backslash
551 \number\dimen4 %
552 \number\dimen2 %
553 \number\dimen0 %
554 }%
555 }%
556 \fi

```

2.9.3 Unpack hex string

\PE@UnescapeHex

```

557 \def\PE@UnescapeHex#1{%
558 \begingroup
559 \PE@InitUccodeHexDigit
560 \def\PE@result{%
561 \expandafter\PE@DeHex#1\relax\relax
562 \expandafter\endgroup
563 \expandafter\def\expandafter#1\expandafter{\PE@result}%
564 }

```

\PE@DeHex

```

565 \def\PE@DeHex#1#2{%
566 \ifx#2\relax
567 \ifx#1\relax
568 \let\PE@next\relax
569 \else
570 \uppercase{%
571 \def\PE@testA{#1}%
572 }%
573 \ifcase\expandafter\PE@TestUcHexDigit\PE@testA
574 \def\PE@next{%
575 \PE@DeHex#10\relax\relax
576 }%
577 \else
578 \let\PE@next\relax
579 \fi
580 \fi
581 \else
582 \uppercase{%
583 \def\PE@testA{#1}%
584 \def\PE@testB{#2}%
585 }%
586 \ifcase\expandafter\PE@TestUcHexDigit\PE@testA
587 \ifcase\expandafter\PE@TestUcHexDigit\PE@testB
588 \uccode\ltx@zero="\PE@testA\PE@testB\relax
589 \ifnum\uccode\ltx@zero=32 %
590 \let\PE@temp\PE@space@space
591 \else
592 \uppercase{%
593 \def\PE@temp{^^@}%
594 }%
595 \fi
596 \edef\PE@result{\PE@result\PE@temp}%
597 \let\PE@next\PE@DeHex
598 \else
599 % invalid input sequence

```



```

656 \PE@EscapeNameAdd{\PE@hash\string#1\string#2}%
657 }%

```

\PE@EscapeNameAdd

```

658 \ifPE@etex
659 \def\PE@EscapeNameAdd#1{#1}%
660 \else
661 \def\PE@EscapeNameAdd#1{%
662 \edef\PE@result{%
663 \PE@result
664 #1%
665 }%
666 }%
667 \fi

```

2.9.5 Conversion to PDF string

\PE@EscapeString

```

668 \ifPE@etex
669 \def\PE@EscapeString#1{%
670 \edef#1{\expandafter\PE@EscapeStringTokens#1\relax}%
671 }%
672 \else
673 \def\PE@EscapeString#1{%
674 \begingroup
675 \def\PE@result{}%
676 \expandafter\PE@EscapeStringTokens#1\relax
677 \expandafter\endgroup
678 \expandafter\def\expandafter#1\expandafter{\PE@result}%
679 }%
680 \fi

```

\PE@EscapeStringTokens

```

681 \def\PE@EscapeStringTokens#1{%
682 \ifx\relax#1%
683 \else
684 \ifnum'#1<33 %
685 \PE@OctChar#1%
686 \else
687 \ifnum'#1>126 %
688 \PE@OctChar#1%
689 \else \ifnum'#1=40 \PE@EscapeStringAdd{\string\}% (
690 \else\ifnum'#1=41 \PE@EscapeStringAdd{\string\)}% )
691 \else\ifnum'#1=92 \PE@EscapeStringAdd{\string\\}% \
692 \else
693 \PE@EscapeStringAdd{#1}%
694 \fi\fi\fi
695 \fi
696 \fi
697 \expandafter\PE@EscapeStringTokens
698 \fi
699 }%

```

\PE@EscapeStringAdd

```

700 \ifPE@etex
701 \def\PE@EscapeStringAdd#1{#1}%
702 \else
703 \def\PE@EscapeStringAdd#1{%
704 \edef\PE@result{%
705 \PE@result
706 #1%

```



```

707    }%
708  }%
709 \fi

710 \PE@AtEnd%
711 \</package>

```

3 Test

3.1 Catcode checks for loading

```

712 <*test1>

713 \catcode'\{=1 %
714 \catcode'\}=2 %
715 \catcode'\#=6 %
716 \catcode'\@=11 %
717 \expandafter\ifx\csname count@\endcsname\relax
718   \countdef\count@=255 %
719 \fi
720 \expandafter\ifx\csname @gobble\endcsname\relax
721   \long\def\@gobble#1{}%
722 \fi
723 \expandafter\ifx\csname @firstofone\endcsname\relax
724   \long\def\@firstofone#1{#1}%
725 \fi
726 \expandafter\ifx\csname loop\endcsname\relax
727   \expandafter\@firstofone
728 \else
729   \expandafter\@gobble
730 \fi
731 {%
732   \def\loop#1\repeat{%
733     \def\body{#1}%
734     \iterate
735   }%
736   \def\iterate{%
737     \body
738     \let\next\iterate
739   \else
740     \let\next\relax
741   \fi
742   \next
743 }%
744 \let\repeat=\fi
745 }%
746 \def\RestoreCatcodes{}
747 \count@=0 %
748 \loop
749   \edef\RestoreCatcodes{%
750     \RestoreCatcodes
751     \catcode\the\count@=\the\catcode\count@\relax
752   }%
753 \ifnum\count@<255 %
754   \advance\count@ 1 %
755 \repeat
756
757 \def\RangeCatcodeInvalid#1#2{%
758   \count@=#1\relax
759   \loop
760     \catcode\count@=15 %
761   \ifnum\count@<#2\relax

```

```

762 \advance\count@ 1 %
763 \repeat
764 }
765 \def\RangeCatcodeCheck#1#2#3{%
766 \count@=#1\relax
767 \loop
768 \ifnum#3=\catcode\count@
769 \else
770 \errmessage{%
771 Character \the\count@\space
772 with wrong catcode \the\catcode\count@\space
773 instead of \number#3%
774 }%
775 \fi
776 \ifnum\count@<#2\relax
777 \advance\count@ 1 %
778 \repeat
779 }
780 \def\space{ }
781 \expandafter\ifx\csname LoadCommand\endcsname\relax
782 \def\LoadCommand{\input pdfescape.sty\relax}%
783 \fi
784 \def\Test{%
785 \RangeCatcodeInvalid{0}{47}%
786 \RangeCatcodeInvalid{58}{64}%
787 \RangeCatcodeInvalid{91}{96}%
788 \RangeCatcodeInvalid{123}{255}%
789 \catcode'\@=12 %
790 \catcode'\=0 %
791 \catcode'\%=14 %
792 \LoadCommand
793 \RangeCatcodeCheck{0}{36}{15}%
794 \RangeCatcodeCheck{37}{37}{14}%
795 \RangeCatcodeCheck{38}{47}{15}%
796 \RangeCatcodeCheck{48}{57}{12}%
797 \RangeCatcodeCheck{58}{63}{15}%
798 \RangeCatcodeCheck{64}{64}{12}%
799 \RangeCatcodeCheck{65}{90}{11}%
800 \RangeCatcodeCheck{91}{91}{15}%
801 \RangeCatcodeCheck{92}{92}{0}%
802 \RangeCatcodeCheck{93}{96}{15}%
803 \RangeCatcodeCheck{97}{122}{11}%
804 \RangeCatcodeCheck{123}{255}{15}%
805 \RestoreCatcodes
806 }
807 \Test
808 \csname @@end\endcsname
809 \end
810 </test1>

```

3.2 Macro tests

```

811 <*test2 | test3 | test4 | test5>
812 \NeedsTeXFormat{LaTeX2e}
813 \makeatletter

```

3.3 Test with \pdfescape... commands

```

814 <*test2>
815 \ProvidesFile{pdfescape-test2.tex}%
816 [2011/04/04 v1.12 Test with \string\pdfescape... commands]%
817 </test2>

```

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

```

818 <*test3>
819 \ProvidesFile{pdfescape-test3.tex}%
820 [2011/04/04 v1.12 Test without \string\pdfescape..., with e-TeX]%
821 </test3>

```

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

```

822 <*test4>
823 \ProvidesFile{pdfescape-test4.tex}%
824 [2011/04/04 v1.12 Test without \string\pdfescape... and e-TeX]%
825 </test4>

```

3.6 Test with LuaTeX

```

826 <*test5>
827 \ProvidesFile{pdfescape-test5.tex}%
828 [2011/04/04 v1.12 Test with LuaTeX]%
829 </test5>

```

3.7 Check/ensure test preconditions

3.7.1 Check \pdfescape...

```

830 <*test2>
831 \@ifundefined{pdfescapehex}{%
832 \PackageError{pdfescape-test2}{%
833 Missing \string\pdfescape... commands%
834 }{Test aborted.}%
835 \stop
836 }{}
837 </test2>

838 <*test3 | test4>
839 \let\pdfescapehex\@undefined
840 \let\pdfunescapehex\@undefined
841 \let\pdfescapename\@undefined
842 \let\pdfescapestring\@undefined
843 </test3 | test4>

```

3.7.2 Check ε -TeX

```

844 <*test3>
845 \@ifundefined{numexpr}{%
846 \PackageError{pdfescape-test3}{%
847 Missing \eTeX
848 }{Test aborted.}%
849 \stop
850 }{}
851 </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

```

852 <*test5>
853 \@ifundefined{directlua}{%
854 \PackageError{pdfescape-test5}{%
855 Missing LuaTeX%
856 }{Test aborted.}%
857 \stop
858 }{}
859 </test5>

```

3.8 Common part

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

```

860 \RequirePackage{qstest}

```

```

861 \IncludeTests{*}
862 \LogTests{log}{*}{*}
863
864 \newcommand*{\ExpectVar}[2]{%
865   \ifx#1#2%
866   \else
867     \begingroup
868       \@onelevel@sanitize#1%
869       \@onelevel@sanitize#2%
870       \typeout{[#1] <> [#2]}% hash-ok
871     \endgroup
872   \fi
873   \Expect*{\ifx#1#2true\else false\fi}{true}%
874 }
875
876 \makeatletter
877 \begingroup
878   \gdef\AllBytes{}%
879   \count@=0 %
880   \catcode0=12 %
881   \@whilenum\count@<256 \do{%
882     \lccode0=\count@
883     \ifnum\count@=32 %
884       \xdef\AllBytes{\AllBytes\space}%
885     \else
886       \lowercase{%
887         \xdef\AllBytes{\AllBytes^^@}%
888       }%
889     \fi
890     \advance\count@ by 1 %
891   }%
892 \endgroup
893 \newcommand*{\AllBytesHex}{%
894   000102030405060708090A0B0C0D0E0F%
895   101112131415161718191A1B1C1D1E1F%
896   202122232425262728292A2B2C2D2E2F%
897   303132333435363738393A3B3C3D3E3F%
898   404142434445464748494A4B4C4D4E4F%
899   505152535455565758595A5B5C5D5E5F%
900   606162636465666768696A6B6C6D6E6F%
901   707172737475767778797A7B7C7D7E7F%
902   808182838485868788898A8B8C8D8E8F%
903   909192939495969798999A9B9C9D9E9F%
904   A0A1A2A3A4A5A6A7A8A9AABACADA EAF%
905   B0B1B2B3B4B5B6B7B8B9BABBBBCDBEBF%
906   C0C1C2C3C4C5C6C7C8C9CACBCCDCECF%
907   D0D1D2D3D4D5D6D7D8D9DADBDCDDDEDF%
908   E0E1E2E3E4E5E6E7E8E9EAEBECEDEEEF%
909   F0F1F2F3F4F5F6F7F8F9FABFCFDFEFF%
910 }
911 \@onelevel@sanitize\AllBytesHex
912 \expandafter\lowercase\expandafter{%
913   \expandafter\newcommand\expandafter*\expandafter\AllBytesHexLC
914     \expandafter{\AllBytesHex}%
915 }
916 \newcommand*{\AllBytesName}{%
917 \begingroup
918   \catcode'\# =12 %
919   \xdef\AllBytesName{%
920     #01#02#03#04#05#06#07#08#09#0A#0B#0C#0D#0E#0F%
921     #10#11#12#13#14#15#16#17#18#19#1A#1B#1C#1D#1E#1F%
922     #20!"#23$#25&'#28#29*+,-.#2F%

```

```

923 0123456789:;#3C=#3E?%
924 @ABCDEFGHIJKLMNO%
925 PQRSTUVWXYZ#5B\@backslashchar#5D^_%
926 'abcdefghijklmno%
927 pqrstuvwxyz#7B|#7D\string~#7F%
928 #80#81#82#83#84#85#86#87#88#89#8A#8B#8C#8D#8E#8F%
929 #90#91#92#93#94#95#96#97#98#99#9A#9B#9C#9D#9E#9F%
930 #A0#A1#A2#A3#A4#A5#A6#A7#A8#A9#AA#AB#AC#AD#AE#AF%
931 #B0#B1#B2#B3#B4#B5#B6#B7#B8#B9#BA#BB#BC#BD#BE#BF%
932 #C0#C1#C2#C3#C4#C5#C6#C7#C8#C9#CA#CB#CC#CD#CE#CF%
933 #D0#D1#D2#D3#D4#D5#D6#D7#D8#D9#DA#DB#DC#DD#DE#DF%
934 #E0#E1#E2#E3#E4#E5#E6#E7#E8#E9#EA#EB#EC#ED#EE#EF%
935 #F0#F1#F2#F3#F4#F5#F6#F7#F8#F9#FA#FB#FC#FD#FE#FF%
936 }%
937 \endgroup
938 \@onelevel@sanitize\AllBytesName
939
940 \newcommand*{\AllBytesString}{%
941 \begin{group}
942 \def\|{|}%
943 \edef\%{\@percentchar}%
944 \catcode'\|=0 %
945 \catcode'\#=12 %
946 \catcode'\~=12 %
947 \catcode'\|=12 %
948 |xdef|AllBytesString{%
949 \000\001\002\003\004\005\006\007\010\011\012\013\014\015\016\017%
950 \020\021\022\023\024\025\026\027\030\031\032\033\034\035\036\037%
951 \040!"#$%&'(\)*+,-./%
952 0123456789:;<=>?%
953 @ABCDEFGHIJKLMNO%
954 PQRSTUVWXYZ[\]^_%
955 'abcdefghijklmno%
956 pqrstuvwxyz{|}~\177%
957 \200\201\202\203\204\205\206\207\210\211\212\213\214\215\216\217%
958 \220\221\222\223\224\225\226\227\230\231\232\233\234\235\236\237%
959 \240\241\242\243\244\245\246\247\250\251\252\253\254\255\256\257%
960 \260\261\262\263\264\265\266\267\270\271\272\273\274\275\276\277%
961 \300\301\302\303\304\305\306\307\310\311\312\313\314\315\316\317%
962 \320\321\322\323\324\325\326\327\330\331\332\333\334\335\336\337%
963 \340\341\342\343\344\345\346\347\350\351\352\353\354\355\356\357%
964 \360\361\362\363\364\365\366\367\370\371\372\373\374\375\376\377%
965 }%
966 \endgroup
967 \@onelevel@sanitize\AllBytesString
968
969 <*test4>
970 \let\org@detokenize\detokenize
971 \let\detokenize\@undefined
972 \let\org@numexpr\numexpr
973 \let\numexpr\@undefined
974 </test4>
975 \RequirePackage{pdfescape}
976 <*test4>
977 \let\detokenize\org@detokenize
978 \let\numexpr\org@numexpr
979 </test4>
980
981 \begin{qstest}{all-hex}{\AllBytes, escapehex}
982 \EdefEscapeHex\x{\AllBytes}%
983 \Expect*{\x}*{\AllBytesHex}%
984 \ExpectVar\x\AllBytesHex

```

```

985 \end{qstest}
986
987 \begin{qstest}{all-unhex}{\AllBytesHex, unescapehex}
988   \EdefUnescapeHex\x{\AllBytesHex}%
989   \Expect*{\x}*{\AllBytes}%
990   \ExpectVar\x\AllBytes
991 \end{qstest}
992
993 \begin{qstest}{all-unhex-lc}{\AllBytesHexLC, unescapehex, lowercase}
994   \EdefUnescapeHex\x{\AllBytesHexLC}%
995   \Expect*{\x}*{\AllBytes}%
996   \ExpectVar\x\AllBytes
997 \end{qstest}
998
999 \begin{qstest}{unhex-incomplete}{unescapehex, incomplete}
1000   \EdefUnescapeHex\x{4}%
1001   \Expect*{\x}{@}%
1002 \end{qstest}
1003
1004 \begin{qstest}{unhex-space}{unescapehex, space}
1005   \EdefUnescapeHex\x{20}%
1006   \Expect*{\x}{ }%
1007   \ExpectVar\x\space
1008 \end{qstest}
1009
1010 \begin{qstest}{unhex-spaces}{unescapehex, spaces}
1011   \EdefUnescapeHex\x{204020204120}%
1012   \def\y#1{%
1013     \edef\z{#1\string @#1#1\string A#1}%
1014   }\y{ }%
1015   \Expect*{\x}*{\z}%
1016   \ExpectVar\x\z
1017 \end{qstest}
1018
1019 \begin{qstest}{unhex-hash}{unescapehex, hash}
1020   \catcode'\#=12 %
1021   \EdefUnescapeHex\x{#20}%
1022   \ExpectVar\x\space
1023 \end{qstest}
1024
1025 \begin{qstest}{unhex-invalid}{unescapehex, invalid}
1026   \def\test#1#2{%
1027     \EdefUnescapeHex\x{#1}%
1028     \edef\y{#2}%
1029     \@onelevel@sanitize\y
1030     \ExpectVar\x\y
1031   }%
1032 \test2
1033   \edef\x{\pdfunescapehex{4X}}%
1034   \edef\y{\string @}%
1035   \ifx\x\y
1036   \else
1037     \def~{\space}%
1038     \typeout{*****}%
1039     \typeout{* Your pdfTeX contains bug 777.~~~*}%
1040     \typeout{* This test is redefined as dummy, *}%
1041     \typeout{* because it triggers the bug.~~~~*}%
1042     \typeout{*****}%
1043     \def\test#1#2{}%
1044   \fi
1045 \test2
1046   \test{X}{}%

```

```

1047 \test{XY}{}%
1048 \test{XYZ}{}%
1049 \test{A}{^~a0}%
1050 \test{AX}{^^a0}%
1051 \test{XA}{^^a0}%
1052 \test{XXAXX}{^^a0}%
1053 \end{qstest}
1054
1055 \begin{qstest}{all-name}{\AllBytes, escapename}
1056 \EdefEscapeName\x{\AllBytes}%
1057 \Expect*{\x}*{\AllBytesName}%
1058 \ExpectVar\x\AllBytesName
1059 \end{qstest}
1060
1061 \begin{qstest}{all-string}{\AllBytes, escapestring}
1062 \EdefEscapeString\x{\AllBytes}%
1063 \Expect*{\x}*{\AllBytesString}%
1064 \ExpectVar\x\AllBytesString
1065 \end{qstest}
1066
1067 \begin{qstest}{uchexdigit}{unescape, uppercase hex digit}
1068 \catcode'\@=11 %
1069 \catcode0=12 %
1070 \def\test#1#2{%
1071 \uccode0=#1\relax
1072 \uppercase{%
1073 \def\x{^~@}%
1074 }%
1075 \Expect*{%
1076 \ifcase\expandafter\PE@TestUcHexDigit\x
1077 true%
1078 \else
1079 false%
1080 \fi
1081 }{#2}%
1082 }%
1083 \def\range#1#2#3{%
1084 \count0=#1\relax
1085 \loop
1086 \ifnum\count0<#2\relax
1087 \test{\count0}{#3}%
1088 \advance\count0 by 1 %
1089 \repeat
1090 }%
1091 \range{0}{47}{false}%
1092 \range{48}{57}{true}%
1093 \range{58}{64}{false}%
1094 \range{65}{70}{true}%
1095 \range{71}{255}{false}%
1096 \end{qstest}
1097
1098 \begin{qstest}{unescapename}{unescapename}
1099 \def\test#1#2{%
1100 \EdefUnescapeName\x{#1}%
1101 \edef\y{#2}%
1102 \@onelevel@sanitize\y
1103 \ExpectVar\x\y
1104 }%
1105 \catcode'\#=12 %
1106 \catcode0=12 %
1107 \test{}{}%
1108 \test{x}{x}%

```

```

1109 \test{xy}{xy}%
1110 \test{#}{#}%
1111 \test{##}{##}%
1112 \test{###}{###}%
1113 \test{####}{####}%
1114 \test{#x}{#x}%
1115 \test{#xy}{#xy}%
1116 \test{#1}{#1}%
1117 \test{#40}{@}%
1118 \test{#400}{@0}%
1119 \test{#4x0}{#4x0}%
1120 \test{#ab}{~^ab}%
1121 \test{#00}{~^@}%
1122 \test{x#40y#40z}{x@y@z}%
1123 \test{#40#40#40#40}{@@@}%
1124 \test{a#x}{a#x}%
1125 \test{a#xy}{a#xy}%
1126 \test{a#1}{a#1}%
1127 \test{a#40}{a@}%
1128 \test{a#400}{a@0}%
1129 \test{#20}{ }%
1130 \test{a#20}{a }%
1131 \test{a#20b}{a b}%
1132 \test{a#20#20#20b}{a \space\space b}%
1133 \end{qstest}
1134
1135 \begin{qstest}{unescapestring}{unescapestring}
1136 \def\test#1#2{%
1137   \EdefUnescapeString\x{#1}%
1138   \edef\y{#2}%
1139   \@onelevel@sanitize\y
1140   \ExpectVar\x\y
1141 }%
1142 \catcode0=12 %
1143 \def\DefChar#1#2{%
1144   \begingroup
1145     \uccode0=#2\relax
1146     \uppercase{\endgroup
1147       \def#1{~^@}%
1148     }%
1149 }%
1150 \DefChar\nul{0}%
1151 \DefChar\one{1}%
1152 \DefChar\bel{8}%
1153 \DefChar\tab{9}%
1154 \DefChar\lf{10}%
1155 \DefChar\ff{12}%
1156 \DefChar\cr{13}%
1157 \DefChar\{92}%
1158 \test{}{}%
1159 \test{a}{a}%
1160 \test{\}{}%
1161 \test{\\}{}%
1162 \test{\\y}{\y}%
1163 \test{\\000}{\nul}%
1164 \test{\\b}{\bel}%
1165 \test{\\t}{\tab}%
1166 \test{\\n}{\lf}%
1167 \test{\\f}{\ff}%
1168 \test{\\r}{\cr}%
1169 \test{\\}{}%
1170 \test{\\}{}%

```



```

1171 \test{\040}{ }%
1172 \test{\100}{@}%
1173 \test{\40}{ }%
1174 \test{\1}{\one}%
1175 \test{\01}{\one}%
1176 \test{\001}{\one}%
1177 \test{\18}{\one8}%
1178 \test{\018}{\one8}%
1179 \test{\0018}{\one8}%
1180 \test{x\}{x}%
1181 \test{x\\\}{x\}%
1182 \test{x\\y}{x\y}%
1183 \test{x\000}{x\nul}%
1184 \test{x\b}{x\bel}%
1185 \test{x\t}{x\tab}%
1186 \test{x\n}{x\lf}%
1187 \test{x\f}{x\ff}%
1188 \test{x\r}{x\cr}%
1189 \test{x\(){x}%
1190 \test{x\)}{x}%
1191 \test{x\040}{x }%
1192 \test{x\100}{x@}%
1193 \test{x\40}{x }%
1194 \test{x\1}{x\one}%
1195 \test{x\01}{x\one}%
1196 \test{x\001}{x\one}%
1197 \test{x\18}{x\one8}%
1198 \test{x\018}{x\one8}%
1199 \test{x\0018}{x\one8}%
1200 \test{\b\t\n\f\r\(\)\000\040}{%
1201 \bel\tab\lf\ff\cr()\nul\space
1202 }%
1203 \test{\\\lf}{}%
1204 \test{x\\\lf}{x}%
1205 \test{\cr}{\lf}%
1206 \test{\cr\lf}{\lf}%
1207 \test{\lf}{\lf}%
1208 \test{\lf\cr}{\lf\lf}%
1209 \test{x\cr}{x\lf}%
1210 \test{x\cr\lf}{x\lf}%
1211 \test{x\lf}{x\lf}%
1212 \test{x\lf\cr}{x\lf\lf}%
1213 \test{x\\\cr\lf y\cr}{xy\lf}%
1214 \end{qstest}
1215 \stop

1216 </test2 | test3 | test4 | test5>

```

3.8.1 Test for iniT_EX

```

1217 <*test6>
1218 \input pdfescape.sty\relax
1219 \catcode'\{=1 %
1220 \catcode'\}=2 %
1221 \catcode'\#=6 %
1222 \catcode'\^=7 %
1223 \catcode'\@=11 %

1224 \begingroup
1225 \catcode'\@=11 %
1226 \countdef\count@=255 %
1227 \def\space{ }%
1228 \long\def\@whilenum#1\do #2{%
1229 \ifnum #1\relax

```

```

1230      #2\relax
1231      \@iwhilenum{#1\relax#2\relax}%
1232    \fi
1233  }%
1234  \long\def\@iwhilenum#1{%
1235    \ifnum #1%
1236      \expandafter\@iwhilenum
1237    \else
1238      \expandafter\ltx@gobble
1239    \fi
1240    {#1}%
1241  }%
1242  \gdef\AllBytes{}%
1243  \count@=0 %
1244  \catcode0=12 %
1245  \@whilenum\count@<256 \do{%
1246    \lccode0=\count@
1247    \ifnum\count@=32 %
1248      \xdef\AllBytes{\AllBytes\space}%
1249    \else
1250      \lowercase{%
1251        \xdef\AllBytes{\AllBytes^^@}%
1252      }%
1253    \fi
1254    \advance\count@ by 1 %
1255  }%
1256 \endgroup
1257 \def\AllBytesHex{%
1258 000102030405060708090A0B0C0D0E0F%
1259 101112131415161718191A1B1C1D1E1F%
1260 202122232425262728292A2B2C2D2E2F%
1261 303132333435363738393A3B3C3D3E3F%
1262 404142434445464748494A4B4C4D4E4F%
1263 505152535455565758595A5B5C5D5E5F%
1264 606162636465666768696A6B6C6D6E6F%
1265 707172737475767778797A7B7C7D7E7F%
1266 808182838485868788898A8B8C8D8E8F%
1267 909192939495969798999A9B9C9D9E9F%
1268 A0A1A2A3A4A5A6A7A8A9AABACADAFAF%
1269 B0B1B2B3B4B5B6B7B8B9BABBBBCDBEBF%
1270 C0C1C2C3C4C5C6C7C8C9CACBCCDCECF%
1271 D0D1D2D3D4D5D6D7D8D9DADBDCDDDEDF%
1272 E0E1E2E3E4E5E6E7E8E9EAEBECEDEEEF%
1273 F0F1F2F3F4F5F6F7F8F9FABFBCFDFEFF%
1274 }
1275 \ltx@onelevel@sanitize\AllBytesHex
1276 \expandafter\lowercase\expandafter{%
1277 \expandafter\def\expandafter\AllBytesHexLC
1278   \expandafter{\AllBytesHex}%
1279 }
1280 \begingroup
1281 \catcode'\# =12 %
1282 \xdef\AllBytesName{%
1283 #01#02#03#04#05#06#07#08#09#0A#0B#0C#0D#0E#0F%
1284 #10#11#12#13#14#15#16#17#18#19#1A#1B#1C#1D#1E#1F%
1285 #20!"#23$#25&'#28#29*+,-.#2F%
1286 0123456789:;#3C=#3E?%
1287 @ABCDEFGHIJKLMNO%
1288 PQRSTUVWXYZ#5B\ltx@backslashchar#5D^_%
1289 'abcdefghijklmnopqrstuvwxyz%
1290 pqrstuvwxyz#7B|#7D\string~#7F%
1291 #80#81#82#83#84#85#86#87#88#89#8A#8B#8C#8D#8E#8F%

```

```

1292      #90#91#92#93#94#95#96#97#98#99#9A#9B#9C#9D#9E#9F%
1293      #A0#A1#A2#A3#A4#A5#A6#A7#A8#A9#AA#AB#AC#AD#AE#AF%
1294      #B0#B1#B2#B3#B4#B5#B6#B7#B8#B9#BA#BB#BC#BD#BE#BF%
1295      #C0#C1#C2#C3#C4#C5#C6#C7#C8#C9#CA#CB#CC#CD#CE#CF%
1296      #D0#D1#D2#D3#D4#D5#D6#D7#D8#D9#DA#DB#DC#DD#DE#DF%
1297      #E0#E1#E2#E3#E4#E5#E6#E7#E8#E9#EA#EB#EC#ED#EE#EF%
1298      #F0#F1#F2#F3#F4#F5#F6#F7#F8#F9#FA#FB#FC#FD#FE#FF%
1299  }%
1300 \endgroup
1301 \ltx@onelevel@sanitize\AllBytesName
1302 \edef\AllBytesFromName{\expandafter\ltx@gobble\AllBytes}
1303
1304 \beginingroup
1305   \def\{|}%
1306   \edef\%{\ltx@percentchar}%
1307   \catcode'\|=0 %
1308   \catcode'\#=12 %
1309   \catcode'\~=12 %
1310   \catcode'\|=12 %
1311   |xdef|AllBytesString{%
1312     \000\001\002\003\004\005\006\007\010\011\012\013\014\015\016\017%
1313     \020\021\022\023\024\025\026\027\030\031\032\033\034\035\036\037%
1314     \040!"#$%&'(\)*+,-./%
1315     0123456789;<=>?%
1316     @ABCDEFGHIJKLMNO%
1317     PQRSTUVWXYZ[\]^_ %
1318     'abcdefghijklmnopqrstuvwxyz%
1319     pqrstuvwxyz{|}~\177%
1320     \200\201\202\203\204\205\206\207\210\211\212\213\214\215\216\217%
1321     \220\221\222\223\224\225\226\227\230\231\232\233\234\235\236\237%
1322     \240\241\242\243\244\245\246\247\250\251\252\253\254\255\256\257%
1323     \260\261\262\263\264\265\266\267\270\271\272\273\274\275\276\277%
1324     \300\301\302\303\304\305\306\307\310\311\312\313\314\315\316\317%
1325     \320\321\322\323\324\325\326\327\330\331\332\333\334\335\336\337%
1326     \340\341\342\343\344\345\346\347\350\351\352\353\354\355\356\357%
1327     \360\361\362\363\364\365\366\367\370\371\372\373\374\375\376\377%
1328   }%
1329 \endgroup
1330 \ltx@onelevel@sanitize\AllBytesString
1331 \def\msg#{\immediate\write16}
1332 \def\Test#1#2#3{%
1333   \beginingroup
1334     #1\TestResult{#2}%
1335     \ifx\TestResult#3%
1336     \else
1337       \newlinechar=10 %
1338       \msg{Expect:^^J#3}%
1339       \msg{Result:^^J\TestResult}%
1340       \errmessage{\string#2 -\string#1-> \string#3}%
1341     \fi
1342   \endgroup
1343 }
1344 \Test\edef\EscapeHex\AllBytes\AllBytesHex
1345 \Test\edef\UnescapeHex\AllBytesHex\AllBytes
1346 \Test\edef\EscapeName\AllBytes\AllBytesName
1347 \Test\edef\UnescapeName\AllBytesName\AllBytesFromName
1348 \Test\edef\EscapeString\AllBytes\AllBytesString
1349 \Test\edef\UnescapeString\AllBytesString\AllBytes
1350 \csname @@end\endcsname\end
1351 </test6>

```

4 Installation

4.1 Download

Package. This package is available on CTAN¹:

[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 T_EX Files” ([CTAN:tds/tds.pdf](#)). Directories with `texmf` in their name are usually organized this way.

4.2 Bundle installation

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

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

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

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

4.3 Package installation

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

```
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):

| | |
|---------------------------------------|--|
| <code>pdfescape.sty</code> | → <code>tex/generic/oberdiek/pdfescape.sty</code> |
| <code>pdfescape.pdf</code> | → <code>doc/latex/oberdiek/pdfescape.pdf</code> |
| <code>test/pdfescape-test1.tex</code> | → <code>doc/latex/oberdiek/test/pdfescape-test1.tex</code> |
| <code>test/pdfescape-test2.tex</code> | → <code>doc/latex/oberdiek/test/pdfescape-test2.tex</code> |
| <code>test/pdfescape-test3.tex</code> | → <code>doc/latex/oberdiek/test/pdfescape-test3.tex</code> |
| <code>test/pdfescape-test4.tex</code> | → <code>doc/latex/oberdiek/test/pdfescape-test4.tex</code> |
| <code>test/pdfescape-test5.tex</code> | → <code>doc/latex/oberdiek/test/pdfescape-test5.tex</code> |
| <code>test/pdfescape-test6.tex</code> | → <code>doc/latex/oberdiek/test/pdfescape-test6.tex</code> |
| <code>pdfescape.dtx</code> | → <code>source/latex/oberdiek/pdfescape.dtx</code> |

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 T_EX distribution (teT_EX, miK_TE_X, ...) relies on file name databases, you must refresh these. For example, teT_EX users run `texhash` or `mktextlsr`.

¹<http://ftp.ctan.org/tex-archive/>

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 L^AT_EX. The `.dtx` chooses its action depending on the format:

plain T_EX: Run `docstrip` and extract the files.

L^AT_EX: Generate the documentation.

If you insist on using L^AT_EX for `docstrip` (really, `docstrip` does not need L^AT_EX), 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 pdfL^AT_EX:

```
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.

[2010/03/01 v1.9]

- Compatibility with `iniTeX`.

[2010/11/12 v1.10]

- Use of package `ltxcmds`.
- Fix for compatibility with `iniTeX`.

[2011/01/30 v1.11]

- Already loaded package files are not input in plain `TeX`.

[2011/04/04 v1.12]

- Further fixes for compatibility for `iniTeX`.
- Test file for `iniTeX` added.

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; plain numbers refer to the code lines where the entry is used.

| Symbols | | |
|------------------------------------|---|---|
| <code>\#</code> | 189, 325, 350, 715, 918, 945, 1020, 1105, 1221, 1281, 1308 | <code>\@gobble</code> 721, 729 |
| <code>\\$</code> | 188 | <code>\@ifundefined</code> 831, 845, 853 |
| <code>\%</code> | 791, 943, 1306 | <code>\@iwhilenum</code> 1231, 1234, 1236 |
| <code>\(</code> | 689, 951, 1314 | <code>\@onelevel@sanitize</code> 147, 868, 869, 911, 938, 967, 1029, 1102, 1139 |
| <code>\)</code> | 690, 951, 1314 | <code>\@percentchar</code> 943 |
| <code>\@</code> | 716, 789, 1068, 1223, 1225 | <code>\@undefined</code> 58, 839, 840, 841, 842, 971, 973 |
| <code>\@backslashchar</code> | 925 | <code>\@whilenum</code> 881, 1228, 1245 |
| <code>\@firstofone</code> | 724, 727 | <code>\@</code> 173, 317, 472, |

| | | | |
|-------------------|---|---------------------|--|
| | 691, 790, 947, 954, 1157, 1160, 1161, 1162, 1163, 1164, 1165, 1166, 1167, 1168, 1169, 1170, 1171, 1172, 1173, 1174, 1175, 1176, 1177, 1178, 1179, 1180, 1181, 1182, 1183, 1184, 1185, 1186, 1187, 1188, 1189, 1190, 1191, 1192, 1193, 1194, 1195, 1196, 1197, 1198, 1199, 1200, 1201, 1203, 1204, 1213, 1310, 1317 | | 165, 188, 189, 316, 317, 713, 714, 715, 716, 751, 760, 768, 772, 789, 790, 791, 880, 918, 944, 945, 946, 947, 1020, 1068, 1069, 1105, 1106, 1142, 1219, 1220, 1221, 1222, 1223, 1225, 1244, 1281, 1307, 1308, 1309, 1310 |
| \{ | 713, 1219 | \count | 1084, 1086, 1087, 1088 |
| \} | 374, 714, 1220 | \count@ | 718, 747, 751, 753, 754, 758, 760, 761, 762, 766, 768, 771, 772, 776, 777, 879, 881, 882, 883, 890, 1226, 1243, 1245, 1246, 1247, 1254 |
| \^ | 1222 | \countdef | 718, 1226 |
| \ | 316, 321, 942, 944, 1305, 1307 | \cr | 1156, 1168, 1188, 1201, 1205, 1206, 1208, 1209, 1210, 1212, 1213 |
| \~ | 946, 1309 | \csname | 14, 21, 50, 66, 76, 119, 122, 131, 134, 141, 152, 414, 417, 426, 449, 477, 717, 720, 723, 726, 781, 808, 1350 |
| Numbers | | | |
| \0 | 949, 950, 951, 1312, 1313, 1314 | D | |
| \1 | 956, 1319 | \DefChar | 1143, 1150, 1151, 1152, 1153, 1154, 1155, 1156, 1157 |
| \2 | 957, 958, 959, 960, 1320, 1321, 1322, 1323 | \detokenize | 156, 160, 970, 971, 977 |
| \3 | 961, 962, 963, 964, 1324, 1325, 1326, 1327 | \dimen | 508, 509, 510, 513, 514, 543, 544, 545, 546, 547, 551, 552, 553 |
| \8 | 289 | \dimexpr | 501, 503, 528, 531, 539 |
| \9 | 290 | \do | 881, 1228, 1245 |
| _ | 165, 691 | E | |
| A | | | |
| \advance | 510, 546, 547, 754, 762, 777, 890, 1088, 1254 | \EdefEscapeHex | 2, 427, 454, 982, 1344 |
| \aftergroup | 29 | \EdefEscapeName | 436, 460, 1056, 1346 |
| \AllBytes | 878, 884, 887, 981, 982, 989, 990, 995, 996, 1055, 1056, 1061, 1062, 1242, 1248, 1251, 1302, 1344, 1345, 1346, 1348, 1349 | \EdefEscapeString | 441, 463, 1062, 1348 |
| \AllBytesFromName | 1302, 1347 | \EdefSanitize | 3, 130, 163, 182, 282, 428, 433, 437, 442 |
| \AllBytesHex | 893, 911, 914, 983, 984, 987, 988, 1257, 1275, 1278, 1344, 1345 | \EdefUnescapeHex | 432, 457, 988, 994, 1000, 1005, 1011, 1021, 1027, 1345 |
| \AllBytesHexLC | 913, 993, 994, 1277 | \EdefUnescapeName | 2, 181, 1100, 1347 |
| \AllBytesName | 916, 919, 938, 1057, 1058, 1282, 1301, 1346, 1347 | \EdefUnescapeString | 3, 281, 1137, 1349 |
| \AllBytesString | 940, 967, 1063, 1064, 1330, 1348, 1349 | \empty | 17, 18 |
| B | | | |
| \begin | 981, 987, 993, 999, 1004, 1010, 1019, 1025, 1055, 1061, 1067, 1098, 1135 | \end | 809, 985, 991, 997, 1002, 1008, 1017, 1023, 1053, 1059, 1065, 1096, 1133, 1214, 1350 |
| \bel | 1152, 1164, 1184, 1201 | \endcsname | 14, 21, 50, 66, 76, 119, 122, 131, 134, 141, 152, 414, 417, 426, 449, 477, 717, 720, 723, 726, 781, 808, 1350 |
| \body | 733, 737 | \endinput | 29, 117 |
| C | | | |
| \catcode | 2, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 33, 34, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 69, 70, 72, 73, 74, 78, 79, 80, 81, 82, 83, 84, 87, 88, 90, 91, 92, 93, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, | \endlinechar | 4, 35, 71, 77, 89 |
| | | \errmessage | 770, 1340 |
| | | \escapechar | 470 |
| | | \eTeX | 847 |
| | | \Expect | 873, 983, 989, 995, 1001, 1006, 1015, 1057, 1063, 1075 |
| | | \ExpectVar | 864, 984, 990, 996, 1007, 1016, 1022, 1030, 1058, 1064, 1103, 1140 |
| F | | | |
| | | \ff | 1155, 1167, 1187, 1201 |
| G | | | |
| | | \gdef | 190, 198, 878, 1242 |

| | | |
|---|---|--|
| I | | <code>\org@numexpr</code> 972, 978 |
| <code>\ifcase</code> | 212, 213, 391, 399, 520, 573, 586, 587, 626, 1076 | P |
| <code>\ifnum</code> | 263, 266, 269, 270, 380, 383, 589, 625, 633, 636, 637, 638, 639, 640, 641, 642, 643, 644, 645, 646, 684, 687, 689, 690, 691, 753, 761, 768, 776, 883, 1086, 1229, 1235, 1247 | <code>\PackageError</code> 832, 846, 854 |
| <code>\ifPE@etex</code> | 475, 481, 499, 525, 611, 658, 668, 700 | <code>\PackageInfo</code> 26 |
| <code>\ifx</code> | 15, 18, 21, 50, 58, 61, 119, 122, 131, 141, 173, 199, 203, 298, 303, 414, 417, 426, 477, 493, 566, 567, 623, 682, 717, 720, 723, 726, 781, 865, 873, 1035, 1335 | <code>\pdf@escapehex</code> 455 |
| <code>\immediate</code> | 23, 52, 1331 | <code>\pdf@escapename</code> 461 |
| <code>\IncludeTests</code> | 861 | <code>\pdf@escapestring</code> 464 |
| <code>\input</code> | 123, 418, 782, 1218 | <code>\pdf@unescapehex</code> 458 |
| <code>\iterate</code> | 734, 736, 738 | <code>\pdfescape</code> 816, 820, 824, 833 |
| L | | <code>\pdfescapehex</code> 839 |
| <code>\lccode</code> | 882, 1246 | <code>\pdfescapename</code> 841 |
| <code>\lf</code> | 1154, 1166, 1186, 1201, 1203, 1204, 1205, 1206, 1207, 1208, 1209, 1210, 1211, 1212, 1213 | <code>\pdfescapestring</code> 842 |
| <code>\LoadCommand</code> | 782, 792 | <code>\pdfunescapehex</code> 840, 1033 |
| <code>\LogTests</code> | 862 | <code>\PE@@NormalizeLineEnd</code> 294, 297 |
| <code>\loop</code> | 732, 748, 759, 767, 1085 | <code>\PE@@OctChar</code> 527, 535 |
| <code>\lowercase</code> | 886, 912, 1250, 1276 | <code>\PE@AtEnd</code> 95, 96, 117, 466, 710 |
| <code>\ltx@backslashchar</code> | 1288 | <code>\PE@backslash</code> 469, 536, 550 |
| <code>\ltx@gobble</code> | 1238, 1302 | <code>\PE@CheckEndBackslash</code> 371 |
| <code>\ltx@newif</code> | 475 | <code>\PE@DeHex</code> 561, 565 |
| <code>\ltx@one</code> 216, 219, 264, 267, 271, 381, 384 | | <code>\PE@DeName</code> 194, 198, 227, 231 |
| <code>\ltx@onelevel@sanitize</code> | 1275, 1301, 1330 | <code>\PE@DeString</code> 325, 411 |
| <code>\ltx@percentchar</code> | 1306 | <code>\PE@edefbabel</code> 447, 455, 458, 461, 464 |
| <code>\ltx@ReturnAfterFi</code> | 176 | <code>\PE@EscapeHex</code> 430, 481 |
| <code>\ltx@zero</code> 214, 221, 225, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 273, 276, 386, 407, 588, 589 | | <code>\PE@EscapeName</code> 439, 611 |
| M | | <code>\PE@EscapeNameAdd</code> 629, 634, 648, 656, 658 |
| <code>\makeatletter</code> | 813, 876 | <code>\PE@EscapeNameHashChar</code> 636, 637, 638, 639, 640, 641, 642, 643, 644, 645, 646, 655 |
| <code>\meaning</code> | 143 | <code>\PE@EscapeNameTokens</code> 613, 618, 622 |
| <code>\msg</code> | 1331, 1338, 1339 | <code>\PE@EscapeString</code> 444, 668 |
| N | | <code>\PE@EscapeStringAdd</code> 689, 690, 691, 693, 700 |
| <code>\NeedsTeXFormat</code> | 812 | <code>\PE@EscapeStringTokens</code> 670, 676, 681 |
| <code>\newcommand</code> | 864, 893, 913, 916, 940 | <code>\PE@etextrue</code> 479 |
| <code>\newlinechar</code> | 1337 | <code>\PE@hash</code> 468, 629, 634, 656 |
| <code>\next</code> | 738, 740, 742 | <code>\PE@HexChar</code> 495, 499, 630, 635 |
| <code>\nul</code> | 1150, 1163, 1183, 1201 | <code>\PE@HexDigit</code> 501, 502, 513, 514, 518 |
| <code>\number</code> | 551, 552, 553, 773 | <code>\PE@InitUccodeHexDigit</code> 192, 238, 559 |
| <code>\numexpr</code> | 501, 503, 528, 531, 538, 539, 972, 973, 978 | <code>\PE@next</code> 201, 206, 227, 231, 235, 300, 304, 306, 309, 392, 394, 396, 400, 402, 404, 568, 574, 578, 597, 600, 606, 609 |
| O | | <code>\PE@NormalizeLineEnd</code> 284, 292 |
| <code>\one</code> | 1151, 1174, 1175, 1176, 1177, 1178, 1179, 1194, 1195, 1196, 1197, 1198, 1199 | <code>\PE@OctAll</code> 394, 400, 402, 406 |
| <code>\org@detokenize</code> | 970, 977 | <code>\PE@OctChar</code> 525, 685, 688 |
| | | <code>\PE@OctI</code> 390 |
| | | <code>\PE@OctII</code> 392, 398 |
| | | <code>\PE@onelevel@sanitize</code> 136, 142, 147, 159, 185, 286 |
| | | <code>\PE@result</code> 193, 196, 200, 205, 226, 230, 293, 295, 299, 302, 409, 487, 489, 511, 512, 548, 549, 560, 563, 596, 617, 619, 662, 663, 675, 678, 704, 705 |
| | | <code>\PE@sanitize</code> 163 |
| | | <code>\PE@SanitizeSpaceOther</code> 168, 183, 283, 429, 438, 443 |
| | | <code>\PE@space@other</code> 164, 175 |
| | | <code>\PE@space@space</code> 167, 590 |
| | | <code>\PE@SpaceToOther</code> 169, 171 |
| | | <code>\PE@strip@prefix</code> 143, 145 |

| | | |
|--|-------------------------------------|---|
| <code>\PE@temp</code> | 223, 226, 590, 593, 596 | 1181, 1182, 1183, 1184, 1185, |
| <code>\PE@testA</code> | 209, | 1186, 1187, 1188, 1189, 1190, |
| | 212, 221, 571, 573, 583, 586, 588 | 1191, 1192, 1193, 1194, 1195, |
| <code>\PE@testB</code> | 210, 213, 221, 584, 587, 588 | 1196, 1197, 1198, 1199, 1200, |
| <code>\PE@TestOctDigit</code> | 379, 391, 399 | 1203, 1204, 1205, 1206, 1207, |
| <code>\PE@TestUcHexDigit</code> | | 1208, 1209, 1210, 1211, 1212, 1213 |
| | 212, 213, 262, 573, 586, 587, 1076 | <code>\TestResult</code> |
| <code>\PE@ToHex</code> | 483, 488, 492 | 1334, 1335, 1339 |
| <code>\PE@UnescapeHex</code> | 434, 557 | <code>\the</code> |
| <code>\PE@UnescapeName</code> | 184, 187 | 77, 78, 79, 80, 81, 82, 83, 84, |
| <code>\PE@UnescapeString</code> | 285, 318 | 97, 528, 531, 538, 539, 751, 771, 772 |
| <code>\ProvidesFile</code> | 815, 819, 823, 827 | <code>\TMP@EnsureCode</code> |
| <code>\ProvidesPackage</code> | 19, 67 | 94, 101, 102, 103, |
| | | 104, 105, 106, 107, 108, 109, |
| | | 110, 111, 112, 113, 114, 115, 116 |
| | | <code>\TMP@RequirePackage</code> 120, 126, 415, 421 |
| | | <code>\typeout</code> 870, 1038, 1039, 1040, 1041, 1042 |
| R | | |
| <code>\range</code> 1083, 1091, 1092, 1093, 1094, 1095 | | |
| <code>\RangeCatcodeCheck</code> | | |
| | 765, 793, 794, 795, 796, 797, | |
| | 798, 799, 800, 801, 802, 803, 804 | |
| <code>\RangeCatcodeInvalid</code> | | |
| | 757, 785, 786, 787, 788 | |
| <code>\repeat</code> | 732, 744, 755, 763, 778, 1089 | |
| <code>\RequirePackage</code> | 128, 423, 860, 975 | |
| <code>\RestoreCatcodes</code> | 746, 749, 750, 805 | |
| S | | |
| <code>\space</code> | 771, 772, 780, 884, 1007, | |
| | 1022, 1037, 1132, 1201, 1227, 1248 | |
| <code>\stop</code> | 835, 849, 857, 1215 | |
| T | | |
| <code>\tab</code> | 1153, 1165, 1185, 1201 | |
| <code>\Test</code> | 784, 807, 1332, | |
| | 1344, 1345, 1346, 1347, 1348, 1349 | |
| <code>\test</code> | 1026, 1043, | |
| | 1046, 1047, 1048, 1049, 1050, | |
| | 1051, 1052, 1070, 1087, 1099, | |
| | 1107, 1108, 1109, 1110, 1111, | |
| | 1112, 1113, 1114, 1115, 1116, | |
| | 1117, 1118, 1119, 1120, 1121, | |
| | 1122, 1123, 1124, 1125, 1126, | |
| | 1127, 1128, 1129, 1130, 1131, | |
| | 1132, 1136, 1158, 1159, 1160, | |
| | 1161, 1162, 1163, 1164, 1165, | |
| | 1166, 1167, 1168, 1169, 1170, | |
| | 1171, 1172, 1173, 1174, 1175, | |
| | 1176, 1177, 1178, 1179, 1180, | |
| U | | |
| <code>\uccode</code> | 114, 115, 116, | |
| | 221, 225, 239, 240, 241, 242, | |
| | 243, 244, 245, 246, 247, 248, | |
| | 249, 250, 251, 252, 253, 254, | |
| | 255, 256, 257, 258, 259, 260, | |
| | 289, 290, 407, 588, 589, 1071, 1145 | |
| <code>\uppercase</code> | 208, 222, | |
| | 312, 408, 570, 582, 592, 1072, 1146 | |
| W | | |
| <code>\write</code> | 23, 52, 1331 | |
| X | | |
| <code>\x</code> | 14, 15, 18, 22, 26, | |
| | 28, 51, 56, 66, 75, 87, 166, 291, | |
| | 313, 471, 474, 982, 983, 984, | |
| | 988, 989, 990, 994, 995, 996, | |
| | 1000, 1001, 1005, 1006, 1007, | |
| | 1011, 1015, 1016, 1021, 1022, | |
| | 1027, 1030, 1033, 1035, 1056, | |
| | 1057, 1058, 1062, 1063, 1064, | |
| | 1073, 1076, 1100, 1103, 1137, 1140 | |
| Y | | |
| <code>\y</code> | 1012, 1014, | |
| | 1028, 1029, 1030, 1034, 1035, | |
| | 1101, 1102, 1103, 1138, 1139, 1140 | |
| Z | | |
| <code>\z</code> | 1013, 1015, 1016 | |