The pdfescape package

Heiko Oberdiek*
<heiko.oberdiek at googlemail.com>

2016/05/16 v1.14

Abstract

This package implements pdfTeX's escape features (\pdfescapehex, \pdfunescapehex, \pdfescapename, \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 \EdefUnescapeName . . . 6
2.6 \EdefUnescapeString . . . 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 . . . 13
2.9 Conversions . . . 13
2.9.1 Conversion to hex string . . . 13
2.9.2 Character code to octal number . . . 14
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 . . . 19
3.3 Test with \pdfescape... commands . . . 19
3.4 Test without \pdfescape..., with ε-TeX . . . 19
3.5 Test without \pdfescape... and ε-TeX . . . 19
3.6 Test with LuaTeX . . . 19
3.7 Check/ensure test preconditions . . . 19
3.7.1 Check \pdfescape... . . . 19
3.7.2 Check ε-TeX . . . 19
3.7.3 Check LuaTeX . . . 20
3.8 Common part . . . 20
3.8.1 Test for iniTeX . . . 26

*Please report any issues at https://github.com/ho-tex/oberdiek/issues

4 Installation **28**
4.1 Download . . . 28
4.2 Bundle installation . . . 28
4.3 Package installation . . . 29
4.4 Refresh file name databases . . . 29
4.5 Some details for the interested . . . 29

5 Catalogue **30**

6 History **30**
[2007/02/21 v1.0] . . . 30
[2007/02/25 v1.1] . . . 30
[2007/03/20 v1.2] . . . 30
[2007/04/11 v1.3] . . . 31
[2007/04/21 v1.4] . . . 31
[2007/08/27 v1.5] . . . 31
[2007/09/09 v1.6] . . . 31
[2007/10/27 v1.7] . . . 31
[2007/11/11 v1.8] . . . 31
[2010/03/01 v1.9] . . . 31
[2010/11/12 v1.10] . . . 31
[2011/01/30 v1.11] . . . 31
[2011/04/04 v1.12] . . . 31
[2011/11/25 v1.13] . . . 31
[2016/05/16 v1.14] . . . 31

7 Index **31**

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\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 % #
```

```
  \catcode39=12 % '
  \catcode40=12 % (
  \catcode41=12 % )
  \catcode44=12 % ,
  \catcode45=12 % -
  \catcode46=12 % .
  \catcode47=12 % /
  \catcode58=12 % :
  \catcode64=11 % @
  \catcode91=12 % [
  \catcode93=12 % ]
  \catcode123=1 % {
  \catcode125=2 % }
  \expandafter\ifx\csname ProvidesPackage\endcsname\relax
    \def\x#1#2#3[#4]{\endgroup
      \immediate\write-1{Package: #3 #4}%
      \xdef#1{#4}%
    }%
  \else
    \def\x#1#2[#3]{\endgroup
      #2[{#3}]%
      \ifx#1\@undefined
        \xdef#1{#3}%
      \fi
      \ifx#1\relax
        \xdef#1{#3}%
      \fi
    }%
  \fi
\expandafter\x\csname ver@pdfescape.sty\endcsname
\ProvidesPackage{pdfescape}%
  [2016/05/16 v1.14 Implements pdfTeX's escape features (HO)]%
```

2.2 Catcodes

```
\begingroup\catcode61\catcode48\catcode32=10\relax%
  \catcode13=5 % ^^M
  \endlinechar=13 %
  \catcode123 1 % {
  \catcode125 2 % }
  \catcode64 11 %
  \def\x{\endgroup
    \expandafter\edef\csname PE@AtEnd\endcsname{%
      \endlinechar=\the\endlinechar\relax
      \catcode13=\the\catcode13\relax
      \catcode32=\the\catcode32\relax
      \catcode35=\the\catcode35\relax
      \catcode61=\the\catcode61\relax
      \catcode64=\the\catcode64\relax
      \catcode123=\the\catcode123\relax
      \catcode125=\the\catcode125\relax
    }%
  }%
\x\catcode61\catcode48\catcode32=10\relax%
\catcode13=5 % ^^M
\endlinechar=13 %
\catcode35=6 % #
\catcode64=11 % @
\catcode123=1 % {
\catcode125=2 % }
\def\TMP@EnsureCode#1#2#3{%
  \edef\PE@AtEnd{%
```

```
    \PE@AtEnd
    #1#2=\the#1#2\relax
  }%
  #1#2=#3\relax
}
\TMP@EnsureCode\catcode{0}{12}% ^^@
\TMP@EnsureCode\catcode{34}{12}% "
\TMP@EnsureCode\catcode{36}{3}% $
\TMP@EnsureCode\catcode{38}{4}% &
\TMP@EnsureCode\catcode{39}{12}% '
\TMP@EnsureCode\catcode{42}{12}% *
\TMP@EnsureCode\catcode{45}{12}% -
\TMP@EnsureCode\catcode{46}{12}% .
\TMP@EnsureCode\catcode{47}{12}% /
\TMP@EnsureCode\catcode{60}{12}% <
\TMP@EnsureCode\catcode{62}{12}% >
\TMP@EnsureCode\catcode{91}{12}% [
\TMP@EnsureCode\catcode{93}{12}% ]
\TMP@EnsureCode\catcode{94}{7}% ^
\TMP@EnsureCode\catcode{96}{12}% `
\TMP@EnsureCode\uccode{34}{0}% "
\TMP@EnsureCode\uccode{48}{0}% 0
\TMP@EnsureCode\uccode{61}{0}% =
\edef\PE@AtEnd{\PE@AtEnd\noexpand\endinput}
```

2.3 Load package

```
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname RequirePackage\endcsname\relax
  \def\TMP@RequirePackage#1[#2]{%
    \begingroup\expandafter\expandafter\expandafter\endgroup
    \expandafter\ifx\csname ver@#1.sty\endcsname\relax
      \input #1.sty\relax
    \fi
  }%
  \TMP@RequirePackage{ltxcmds}[2010/04/08]%
\else
  \RequirePackage{ltxcmds}[2010/04/08]%
\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.

```
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname detokenize\endcsname\relax
  \long\def\EdefSanitize#1#2{%
    \begingroup
      \csname @safe@activestrue\endcsname
      \edef#1{#2}%
      \PE@onelevel@sanitize#1%
    \expandafter\endgroup
    \expandafter\def\expandafter#1\expandafter{#1}%
  }%
  \begingroup\expandafter\expandafter\expandafter\endgroup
  \expandafter\ifx\csname @onelevel@sanitize\endcsname\relax
    \def\PE@onelevel@sanitize#1{%
      \edef#1{\expandafter\PE@strip@prefix\meaning#1}%
    }%
    \def\PE@strip@prefix#1>{}%
  \else
    \let\PE@onelevel@sanitize\@onelevel@sanitize
  \fi
\else
```

```
\long\def\EdefSanitize#1#2{%
  \begingroup
    \csname @safe@activestrue\endcsname
    \edef#1{#2}%
  \expandafter\endgroup
  \expandafter\def\expandafter#1\expandafter{%
    \detokenize\expandafter{#1}%
  }%
 }%
 \def\PE@onelevel@sanitize#1{%
  \edef#1{\detokenize\expandafter{#1}}%
 }%
\fi
```

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

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

2.4.1 Space characters

\PE@space@other

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

\PE@space@space

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

2.4.2 Space normalization

\PE@SanitizeSpaceOther

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

\PE@SpaceToOther

```
\def\PE@SpaceToOther#1 #2\relax{%
 #1%
 \ifx\\#2\\%
 \else
  \PE@space@other
  \ltx@ReturnAfterFi{%
   \PE@SpaceToOther#2\relax
  }%
 \fi
}
```

2.5 \EdefUnescapeName

\EdefUnescapeName

```
\def\EdefUnescapeName#1#2{%
 \EdefSanitize#1{#2}%
 \PE@SanitizeSpaceOther#1%
 \PE@UnescapeName#1%
 \PE@onelevel@sanitize#1%
}
```

\PE@UnescapeName

```
\begingroup
 \catcode`\$=6  % hash
 \catcode`\#=12 % other
```

```
\gdef\PE@UnescapeName$1{%
  \begingroup
    \PE@InitUccodeHexDigit
    \def\PE@result{}%
    \expandafter\PE@DeName$1#\relax\relax
  \expandafter\endgroup
  \expandafter\def\expandafter$1\expandafter{\PE@result}%
}%
\gdef\PE@DeName$1#$2$3{%
  \ifx\relax$2%
    \edef\PE@result{\PE@result$1}%
    \let\PE@next\relax
  \else
    \ifx\relax$3%
      % wrong escape sequence in input
      \edef\PE@result{\PE@result$1#}%
      \let\PE@next\relax
    \else
      \uppercase{%
        \def\PE@testA{$2}%
        \def\PE@testB{$3}%
      }%
      \ifcase\ifcase\expandafter\PE@TestUcHexDigit\PE@testA
             \ifcase\expandafter\PE@TestUcHexDigit\PE@testB
               \ltx@zero
             \else
               \ltx@one
             \fi
           \else
             \ltx@one
           \fi
        \uccode\ltx@zero=”\PE@testA\PE@testB\relax
        \uppercase{%
          \def\PE@temp{^^@}%
        }%
        \uccode\ltx@zero=\ltx@zero
        \edef\PE@result{\PE@result$1\PE@temp}%
        \let\PE@next\PE@DeName
      \else
        % wrong escape sequence in input
        \edef\PE@result{\PE@result$1#}%
        \def\PE@next{\PE@DeName$2$3}%
      \fi
    \fi
  \fi
  \PE@next
}%
\endgroup
```

\PE@InitUccodeHexDigit

```
\def\PE@InitUccodeHexDigit{%
  \uccode‘a=‘A\relax
  \uccode‘b=‘B\relax
  \uccode‘c=‘C\relax
  \uccode‘d=‘D\relax
  \uccode‘e=‘E\relax
  \uccode‘f=‘F\relax
  \uccode‘A=\ltx@zero
  \uccode‘B=\ltx@zero
  \uccode‘C=\ltx@zero
  \uccode‘D=\ltx@zero
  \uccode‘E=\ltx@zero
  \uccode‘F=\ltx@zero
```

```
  \uccode`0=\ltx@zero
  \uccode`1=\ltx@zero
  \uccode`2=\ltx@zero
  \uccode`3=\ltx@zero
  \uccode`4=\ltx@zero
  \uccode`5=\ltx@zero
  \uccode`6=\ltx@zero
  \uccode`7=\ltx@zero
  \uccode`8=\ltx@zero
  \uccode`9=\ltx@zero
}
```

\PE@TestUcHexDigit

```
\def\PE@TestUcHexDigit#1{%
  \ifnum`#1<48 % 0
    \ltx@one
  \else
    \ifnum`#1>70 % F
      \ltx@one
    \else
      \ifnum`#1>57 % 9
        \ifnum`#1<65 % A
          \ltx@one
        \else
          \ltx@zero
        \fi
      \else
        \ltx@zero
      \fi
    \fi
  \fi
}
```

2.6 \EdefUnescapeString

\EdefUnescapeString

```
\def\EdefUnescapeString#1#2{%
  \EdefSanitize#1{#2}%
  \PE@SanitizeSpaceOther#1%
  \PE@NormalizeLineEnd#1%
  \PE@UnescapeString#1%
  \PE@onelevel@sanitize#1%
}

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

\PE@NormalizeLineEnd

```
  \def\PE@NormalizeLineEnd##1{%
    \def\PE@result{}%
    \expandafter\PE@@NormalizeLineEnd##1#2\relax
    \let##1\PE@result
  }%
```

\PE@@NormalizeLineEnd

```
  \def\PE@@NormalizeLineEnd##1#2##2{%
    \ifx\relax##2%
      \edef\PE@result{\PE@result##1}%
      \let\PE@next\relax
    \else
```

```
    \edef\PE@result{\PE@result##1#1}%
    \ifx#1##2% lf
      \let\PE@next\PE@@NormalizeLineEnd
    \else
      \def\PE@next{\PE@@NormalizeLineEnd##2}%
    \fi
  \fi
  \PE@next
 }%
}%
\uppercase{%
 \x 89%
}

\begingroup
 \catcode`\|=0 %
 \catcode`\\=12 %
```

\PE@UnescapeString

```
 |gdef|PE@UnescapeString#1{%
  |begingroup
   |def|PE@result{}%
   |expandafter|PE@DeString#1\|relax
  |expandafter|endgroup
  |expandafter|def|expandafter#1|expandafter{|PE@result}%
 }%
```

\PE@DeString

```
 |gdef|PE@DeString#1\#2{%
  |ifx|relax#2%
   |edef|PE@result{|PE@result#1}%
   |let|PE@next|relax
  |else
   |if n#2%
    |uccode|ltx@zero=10 %
   |else|if r#2%
    |uccode|ltx@zero=13 %
   |else|if t#2%
    |uccode|ltx@zero=9 %
   |else|if b#2%
    |uccode|ltx@zero=8 %
   |else|if f#2%
    |uccode|ltx@zero=12 %
   |else
    |uccode|ltx@zero=|ltx@zero
   |fi|fi|fi|fi|fi
   |ifnum|uccode|ltx@zero>|ltx@zero
    |uppercase{%
     |edef|PE@temp{^^@}%
    }%
    |edef|PE@result{|PE@result#1|PE@temp}%
    |let|PE@next|PE@DeString
   |else
    |if\#2% backslash
     |edef|PE@result{|PE@result#1}%
     |let|PE@next|PE@CheckEndBackslash
    |else
     |ifnum`#2=10 % linefeed
      |edef|PE@result{|PE@result#1}%
      |let|PE@next|PE@DeString
     |else
      |ifcase|PE@TestOctDigit#2%
       |edef|PE@result{|PE@result#1}%
```

```
      |def|PE@next{|PE@OctI#2}%
    |else
      |edef|PE@result{|PE@result#1#2}%
      |let|PE@next|PE@DeString
    |fi
  |fi
 |fi
 |fi
 |PE@next
}%
```

\PE@CheckEndBackslash

```
|gdef|PE@CheckEndBackslash#1{%
 |ifx|relax#1%
 |else
  |edef|PE@result{|PE@result\}%
  |expandafter|PE@DeString|expandafter#1%
 |fi
}%

|endgroup
```

\PE@TestOctDigit

```
\def\PE@TestOctDigit#1{%
 \ifnum‘#1<48 % 0
  \ltx@one
 \else
  \ifnum‘#1>55 % 7
   \ltx@one
  \else
   \ltx@zero
  \fi
 \fi
}
```

\PE@OctI

```
\def\PE@OctI#1#2{%
 \ifcase\PE@TestOctDigit#2%
  \def\PE@next{\PE@OctII{#1#2}}%
 \else
  \def\PE@next{\PE@OctAll#1#2}%
 \fi
 \PE@next
}
```

\PE@OctII

```
\def\PE@OctII#1#2{%
 \ifcase\PE@TestOctDigit#2%
  \def\PE@next{\PE@OctIII#1#2}%
 \else
  \def\PE@next{\PE@OctAll{#1}#2}%
 \fi
 \PE@next
}

\ltx@IfUndefined{numexpr}{%
 \catcode‘\$=9 %
 \catcode‘\&=14 %
}{%
 \catcode‘\$=14 %
 \catcode‘\&=9 %
}
```

\PE@OctIII

```
\def\PE@OctIII#1#2#3{%
  \ifnum#1<4 %
    \def\PE@next{\PE@OctAll{#1#2#3}}%
  \else
$   \count\ltx@cclv#1 %
$   \advance\count\ltx@cclv -4 %
    \edef\PE@next{%
      \noexpand\PE@OctAll{%
$       \the\count\ltx@cclv
&       \the\numexpr#1-4\relax
        #2#3%
      }%
    }%
  \fi
  \PE@next
}
```

\PE@OctAll

```
\def\PE@OctAll#1{%
  \uccode\ltx@zero='#1\relax
  \uppercase{%
    \edef\PE@result{\PE@result^^@}%
  }%
  \PE@DeString
}
```

2.7 User macros (pdfTEX analogues)

```
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname RequirePackage\endcsname\relax
  \def\TMP@RequirePackage#1[#2]{%
    \begingroup\expandafter\expandafter\expandafter\endgroup
    \expandafter\ifx\csname ver@#1.sty\endcsname\relax
      \input #1.sty\relax
    \fi
  }%
  \TMP@RequirePackage{pdftexcmds}[2007/11/11]%
\else
  \RequirePackage{pdftexcmds}[2007/11/11]%
\fi
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname pdf@escapehex\endcsname\relax
```

\EdefEscapeHex

```
  \long\def\EdefEscapeHex#1#2{%
    \EdefSanitize#1{#2}%
    \PE@SanitizeSpaceOther#1%
    \PE@EscapeHex#1%
  }%
```

\EdefUnescapeHex

```
  \def\EdefUnescapeHex#1#2{%
    \EdefSanitize#1{#2}%
    \PE@UnescapeHex#1%
  }%
```

\EdefEscapeName

```
  \long\def\EdefEscapeName#1#2{%
    \EdefSanitize#1{#2}%
    \PE@SanitizeSpaceOther#1%
    \PE@EscapeName#1%
  }%
```

\EdefEscapeString

```
  \long\def\EdefEscapeString#1#2{%
    \EdefSanitize#1{#2}%
    \PE@SanitizeSpaceOther#1%
    \PE@EscapeString#1%
  }%

\else
```

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

```
  \long\def\PE@edefbabel#1#2#3{%
    \begingroup
      \csname @save@activestrue\endcsname
      \edef#1{#2{#3}}%
    \expandafter\endgroup
    \expandafter\def\expandafter#1\expandafter{#1}%
  }%
```

\EdefEscapeHex

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

\EdefUnescapeHex

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

\EdefEscapeName

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

\EdefEscapeString

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

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

2.8 Help macros

2.8.1 Characters

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

\PE@hash

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

\PE@backslash

```
\begingroup
  \escapechar=-1 %
\edef\x{\endgroup
  \def\noexpand\PE@backslash{\string\\}%
}
\x
```

2.8.2 Switch for ε-TEX

```
\ltx@newif\ifPE@etex
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname numexpr\endcsname\relax
\else
  \PE@etextrue
\fi
```

2.9 Conversions

2.9.1 Conversion to hex string

\PE@EscapeHex

```
\ifPE@etex
  \def\PE@EscapeHex#1{%
    \edef#1{\expandafter\PE@ToHex#1\relax}%
  }%
\else
  \def\PE@EscapeHex#1{%
    \def\PE@result{}%
    \expandafter\PE@ToHex#1\relax
    \let#1\PE@result
  }%
\fi
```

\PE@ToHex

```
\def\PE@ToHex#1{%
  \ifx\relax#1%
  \else
    \PE@HexChar{#1}%
    \expandafter\PE@ToHex
  \fi
}%
```

\PE@HexChar

```
\ifPE@etex
  \def\PE@HexChar#1{%
    \PE@HexDigit{\numexpr\dimexpr.0625\dimexpr`#1sp\relax\relax\relax}%
    \PE@HexDigit{%
      \numexpr`#1-16*\dimexpr.0625\dimexpr`#1sp\relax\relax\relax
    }%
  }%
\else
  \def\PE@HexChar#1{%
    \dimen0=`#1sp%
    \dimen2=.0625\dimen0 %
    \advance\dimen0-16\dimen2 %
    \edef\PE@result{%
      \PE@result
      \PE@HexDigit{\dimen2 }%
      \PE@HexDigit{\dimen0 }%
    }%
  }%
\fi
```

\PE@HexDigit

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

2.9.2 Character code to octal number

\PE@OctChar

```
\ifPE@etex
  \def\PE@OctChar#1{%
    \expandafter\PE@@OctChar
       \the\numexpr\dimexpr.015625\dimexpr‘#1sp\relax\relax
          \expandafter\relax
       \expandafter\relax
       \the\numexpr\dimexpr.125\dimexpr‘#1sp\relax\relax\relax
       \relax
       #1%
  }%
  \def\PE@@OctChar#1\relax#2\relax#3{%
    \PE@backslash
    #1%
    \the\numexpr#2-8*#1\relax
    \the\numexpr\dimexpr‘#3sp\relax-8*#2\relax
  }%
\else
  \def\PE@OctChar#1{%
    \dimen0=‘#1sp%
    \dimen2=.125\dimen0 %
    \dimen4=.125\dimen2 %
    \advance\dimen0-8\dimen2 %
    \advance\dimen2-8\dimen4 %
    \edef\PE@result{%
      \PE@result
      \PE@backslash
      \number\dimen4 %
      \number\dimen2 %
      \number\dimen0 %
    }%
  }%
\fi
```

2.9.3 Unpack hex string

\PE@UnescapeHex

```
\def\PE@UnescapeHex#1{%
  \begingroup
    \PE@InitUccodeHexDigit
    \def\PE@result{}%
    \expandafter\PE@DeHex#1\relax\relax
  \expandafter\endgroup
  \expandafter\def\expandafter#1\expandafter{\PE@result}%
}
```

\PE@DeHex

```
\def\PE@DeHex#1#2{%
  \ifx#2\relax
    \ifx#1\relax
      \let\PE@next\relax
    \else
      \uppercase{%
        \def\PE@testA{#1}%
      }%
      \ifcase\expandafter\PE@TestUcHexDigit\PE@testA
        \def\PE@next{%
          \PE@DeHex#10\relax\relax
        }%
      \else
```

```
      \let\PE@next\relax
    \fi
  \fi
 \else
  \uppercase{%
   \def\PE@testA{#1}%
   \def\PE@testB{#2}%
  }%
  \ifcase\expandafter\PE@TestUcHexDigit\PE@testA
   \ifcase\expandafter\PE@TestUcHexDigit\PE@testB
    \uccode\ltx@zero=”\PE@testA\PE@testB\relax
    \ifnum\uccode\ltx@zero=32 %
     \let\PE@temp\PE@space@space
    \else
     \uppercase{%
      \def\PE@temp{^^@}%
     }%
    \fi
    \edef\PE@result{\PE@result\PE@temp}%
    \let\PE@next\PE@DeHex
   \else
    % invalid input sequence
    \def\PE@next{%
     \PE@DeHex#1%
    }%
   \fi
  \else
   % invalid input sequence
   \def\PE@next{\PE@DeHex#2}%
  \fi
 \fi
 \PE@next
}
```

2.9.4 Conversion to PDF name

\PE@EscapeName

```
\ifPE@etex
 \def\PE@EscapeName#1{%
  \edef#1{\expandafter\PE@EscapeNameTokens#1\relax}%
 }%
\else
 \def\PE@EscapeName#1{%
  \def\PE@result{}%
  \expandafter\PE@EscapeNameTokens#1\relax
  \let#1\PE@result
 }%
\fi
```

\PE@EscapeNameTokens

```
\def\PE@EscapeNameTokens#1{%
 \ifx\relax#1%
 \else
  \ifnum‘#1<33 %
   \ifcase‘#1 %
    % drop illegal zero
   \else
    \PE@EscapeNameAdd\PE@hash
    \PE@HexChar#1%
   \fi
  \else
   \ifnum‘#1>126 %
```

```
      \PE@EscapeNameAdd\PE@hash
      \PE@HexChar#1%
    \else  \ifnum‘#1=35  \PE@EscapeNameHashChar 23% #
      \else\ifnum‘#1=37  \PE@EscapeNameHashChar 25% %
      \else\ifnum‘#1=40  \PE@EscapeNameHashChar 28% (
      \else\ifnum‘#1=41  \PE@EscapeNameHashChar 29% )
      \else\ifnum‘#1=47  \PE@EscapeNameHashChar 2F% /
      \else\ifnum‘#1=60  \PE@EscapeNameHashChar 3C% <
      \else\ifnum‘#1=62  \PE@EscapeNameHashChar 3E% >
      \else\ifnum‘#1=91  \PE@EscapeNameHashChar 5B% [
      \else\ifnum‘#1=93  \PE@EscapeNameHashChar 5D% ]
      \else\ifnum‘#1=123 \PE@EscapeNameHashChar 7B% {
      \else\ifnum‘#1=125 \PE@EscapeNameHashChar 7D% }
      \else
        \PE@EscapeNameAdd{#1}%
      \fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi
    \fi
  \fi
  \expandafter\PE@EscapeNameTokens
 \fi
}%
\def\PE@EscapeNameHashChar#1#2{%
 \PE@EscapeNameAdd{\PE@hash\string#1\string#2}%
}%
```

\PE@EscapeNameAdd

```
\ifPE@etex
 \def\PE@EscapeNameAdd#1{#1}%
\else
 \def\PE@EscapeNameAdd#1{%
  \edef\PE@result{%
   \PE@result
   #1%
  }%
 }%
\fi
```

2.9.5 Conversion to PDF string

\PE@EscapeString

```
\ifPE@etex
 \def\PE@EscapeString#1{%
  \edef#1{\expandafter\PE@EscapeStringTokens#1\relax}%
 }%
\else
 \def\PE@EscapeString#1{%
  \begingroup
   \def\PE@result{}%
   \expandafter\PE@EscapeStringTokens#1\relax
  \expandafter\endgroup
  \expandafter\def\expandafter#1\expandafter{\PE@result}%
 }%
\fi
```

\PE@EscapeStringTokens

```
\def\PE@EscapeStringTokens#1{%
 \ifx\relax#1%
 \else
  \ifnum‘#1<33 %
   \PE@OctChar#1%
  \else
   \ifnum‘#1>126 %
```

```
      \PE@OctChar#1%
    \else  \ifnum`#1=40 \PE@EscapeStringAdd{\string\(}% (
      \else\ifnum`#1=41 \PE@EscapeStringAdd{\string\)}% )
      \else\ifnum`#1=92 \PE@EscapeStringAdd{\string\\}% \
      \else
        \PE@EscapeStringAdd{#1}%
      \fi\fi\fi
    \fi
  \fi
  \expandafter\PE@EscapeStringTokens
 \fi
}%
```

\PE@EscapeStringAdd

```
\ifPE@etex
 \def\PE@EscapeStringAdd#1{#1}%
\else
 \def\PE@EscapeStringAdd#1{%
  \edef\PE@result{%
   \PE@result
   #1%
  }%
 }%
\fi

\PE@AtEnd%
```

⟨/package⟩

3 Test

3.1 Catcode checks for loading

⟨*test1⟩

```
\catcode`\{=1 %
\catcode`\}=2 %
\catcode`\#=6 %
\catcode`\@=11 %
\expandafter\ifx\csname count@\endcsname\relax
 \countdef\count@=255 %
\fi
\expandafter\ifx\csname @gobble\endcsname\relax
 \long\def\@gobble#1{}%
\fi
\expandafter\ifx\csname @firstofone\endcsname\relax
 \long\def\@firstofone#1{#1}%
\fi
\expandafter\ifx\csname loop\endcsname\relax
 \expandafter\@firstofone
\else
 \expandafter\@gobble
\fi
{%
 \def\loop#1\repeat{%
  \def\body{#1}%
  \iterate
 }%
 \def\iterate{%
  \body
   \let\next\iterate
  \else
   \let\next\relax
```

```
    \fi
    \next
  }%
  \let\repeat=\fi
}%
\def\RestoreCatcodes{}
\count@=0 %
\loop
  \edef\RestoreCatcodes{%
    \RestoreCatcodes
    \catcode\the\count@=\the\catcode\count@\relax
  }%
\ifnum\count@<255 %
  \advance\count@ 1 %
\repeat

\def\RangeCatcodeInvalid#1#2{%
  \count@=#1\relax
  \loop
    \catcode\count@=15 %
  \ifnum\count@<#2\relax
    \advance\count@ 1 %
  \repeat
}
\def\RangeCatcodeCheck#1#2#3{%
  \count@=#1\relax
  \loop
    \ifnum#3=\catcode\count@
    \else
      \errmessage{%
        Character \the\count@\space
        with wrong catcode \the\catcode\count@\space
        instead of \number#3%
      }%
    \fi
  \ifnum\count@<#2\relax
    \advance\count@ 1 %
  \repeat
}
\def\space{ }
\expandafter\ifx\csname LoadCommand\endcsname\relax
  \def\LoadCommand{\input pdfescape.sty\relax}%
\fi
\def\Test{%
  \RangeCatcodeInvalid{0}{47}%
  \RangeCatcodeInvalid{58}{64}%
  \RangeCatcodeInvalid{91}{96}%
  \RangeCatcodeInvalid{123}{255}%
  \catcode`\@=12 %
  \catcode`\\=0 %
  \catcode`\%=14 %
  \LoadCommand
  \RangeCatcodeCheck{0}{36}{15}%
  \RangeCatcodeCheck{37}{37}{14}%
  \RangeCatcodeCheck{38}{47}{15}%
  \RangeCatcodeCheck{48}{57}{12}%
  \RangeCatcodeCheck{58}{63}{15}%
  \RangeCatcodeCheck{64}{64}{12}%
  \RangeCatcodeCheck{65}{90}{11}%
  \RangeCatcodeCheck{91}{91}{15}%
  \RangeCatcodeCheck{92}{92}{0}%
  \RangeCatcodeCheck{93}{96}{15}%
```

```
  \RangeCatcodeCheck{97}{122}{11}%
  \RangeCatcodeCheck{123}{255}{15}%
  \RestoreCatcodes
}
\Test
\csname @@end\endcsname
\end
```

⟨/test1⟩

3.2 Macro tests

⟨*test2 | test3 | test4 | test5⟩

```
\NeedsTeXFormat{LaTeX2e}
\makeatletter
```

3.3 Test with \pdfescape... commands

⟨*test2⟩

```
\ProvidesFile{pdfescape-test2.tex}%
   [2016/05/16 v1.14 Test with \string\pdfescape... commands]%
```

⟨/test2⟩

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

⟨*test3⟩

```
\ProvidesFile{pdfescape-test3.tex}%
   [2016/05/16 v1.14 Test without \string\pdfescape..., with e-TeX]%
```

⟨/test3⟩

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

⟨*test4⟩

```
\ProvidesFile{pdfescape-test4.tex}%
   [2016/05/16 v1.14 Test without \string\pdfescape... and e-TeX]%
```

⟨/test4⟩

3.6 Test with LuaTEX

⟨*test5⟩

```
\ProvidesFile{pdfescape-test5.tex}%
   [2016/05/16 v1.14 Test with LuaTeX]%
```

⟨/test5⟩

3.7 Check/ensure test preconditions

3.7.1 Check \pdfescape...

⟨*test2⟩

```
\@ifundefined{pdfescapehex}{%
  \PackageError{pdfescape-test2}{%
    Missing \string\pdfescape... commands%
  }{Test aborted.}%
  \stop
}{}
```

⟨/test2⟩

⟨*test3 | test4⟩

```
\let\pdfescapehex\@undefined
\let\pdfunescapehex\@undefined
\let\pdfescapename\@undefined
\let\pdfescapestring\@undefined
```

⟨/test3 | test4⟩

3.7.2 Check ε-TEX

⟨*test3⟩

```
\@ifundefined{numexpr}{%
  \PackageError{pdfescape-test3}{%
```

```
    Missing \eTeX
  }{Test aborted.}%
  \stop
}{}
⟨/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

```
⟨*test5⟩
\@ifundefined{directlua}{%
  \PackageError{pdfescape-test5}{%
    Missing LuaTeX%
  }{Test aborted.}%
  \stop
}{}
⟨/test5⟩
```

3.8 Common part

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

```
\RequirePackage{qstest}
\IncludeTests{*}
\LogTests{log}{*}{*}

\newcommand*{\ExpectVar}[2]{%
  \ifx#1#2%
  \else
    \begingroup
      \@onelevel@sanitize#1%
      \@onelevel@sanitize#2%
      \typeout{[#1] <> [#2]}% hash-ok
    \endgroup
  \fi
  \Expect*{\ifx#1#2true\else false\fi}{true}%
}

\makeatletter
\begingroup
  \gdef\AllBytes{}%
  \count@=0 %
  \catcode0=12 %
  \@whilenum\count@<256 \do{%
    \lccode0=\count@
    \ifnum\count@=32 %
      \xdef\AllBytes{\AllBytes\space}%
    \else
      \lowercase{%
        \xdef\AllBytes{\AllBytes^^@}%
      }%
    \fi
    \advance\count@ by 1 %
  }%
\endgroup
\newcommand*{\AllBytesHex}{%
  000102030405060708090A0B0C0D0E0F%
  101112131415161718191A1B1C1D1E1F%
  202122232425262728292A2B2C2D2E2F%
  303132333435363738393A3B3C3D3E3F%
  404142434445464748494A4B4C4D4E4F%
  505152535455565758595A5B5C5D5E5F%
  606162636465666768696A6B6C6D6E6F%
```

```
  707172737475767778797A7B7C7D7E7F%
  808182838485868788898A8B8C8D8E8F%
  909192939495969798999A9B9C9D9E9F%
  A0A1A2A3A4A5A6A7A8A9AAABACADAEAF%
  B0B1B2B3B4B5B6B7B8B9BABBBCBDBEBF%
  C0C1C2C3C4C5C6C7C8C9CACBCCCDCECF%
  D0D1D2D3D4D5D6D7D8D9DADBDCDDDEDF%
  E0E1E2E3E4E5E6E7E8E9EAEBECEDEEEF%
  F0F1F2F3F4F5F6F7F8F9FAFBFCFDFEFF%
}
\@onelevel@sanitize\AllBytesHex
\expandafter\lowercase\expandafter{%
  \expandafter\newcommand\expandafter*\expandafter\AllBytesHexLC
     \expandafter{\AllBytesHex}%
}
\newcommand*{\AllBytesName}{}
\begingroup
  \catcode`\#=12 %
  \xdef\AllBytesName{%
    #01#02#03#04#05#06#07#08#09#0A#0B#0C#0D#0E#0F%
    #10#11#12#13#14#15#16#17#18#19#1A#1B#1C#1D#1E#1F%
    #20!"#23$#25&'#28#29*+,-.#2F%
    0123456789:;#3C=#3E?%
    @ABCDEFGHIJKLMNO%
    PQRSTUVWXYZ#5B\@backslashchar#5D^_%
    `abcdefghijklmno%
    pqrstuvwxyz#7B|#7D\string~#7F%
    #80#81#82#83#84#85#86#87#88#89#8A#8B#8C#8D#8E#8F%
    #90#91#92#93#94#95#96#97#98#99#9A#9B#9C#9D#9E#9F%
   #A0#A1#A2#A3#A4#A5#A6#A7#A8#A9#AA#AB#AC#AD#AE#AF%
    #B0#B1#B2#B3#B4#B5#B6#B7#B8#B9#BA#BB#BC#BD#BE#BF%
   #C0#C1#C2#C3#C4#C5#C6#C7#C8#C9#CA#CB#CC#CD#CE#CF%
   #D0#D1#D2#D3#D4#D5#D6#D7#D8#D9#DA#DB#DC#DD#DE#DF%
    #E0#E1#E2#E3#E4#E5#E6#E7#E8#E9#EA#EB#EC#ED#EE#EF%
    #F0#F1#F2#F3#F4#F5#F6#F7#F8#F9#FA#FB#FC#FD#FE#FF%
  }%
\endgroup
\@onelevel@sanitize\AllBytesName

\newcommand*{\AllBytesString}{}
\begingroup
  \def\|{|}%
  \edef\%{\@percentchar}%
  \catcode`\|=0 %
  \catcode`\#=12 %
  \catcode`\~=12 %
  \catcode`\\=12 %
  |xdef|AllBytesString{%
    \000\001\002\003\004\005\006\007\010\011\012\013\014\015\016\017%
    \020\021\022\023\024\025\026\027\030\031\032\033\034\035\036\037%
    \040!"#$|%&'\(\)*+,-./%
    0123456789:;<=>?%
    @ABCDEFGHIJKLMNO%
    PQRSTUVWXYZ[\\]^_%
    `abcdefghijklmno%
    pqrstuvwxyz{||}~\177%
    \200\201\202\203\204\205\206\207\210\211\212\213\214\215\216\217%
    \220\221\222\223\224\225\226\227\230\231\232\233\234\235\236\237%
    \240\241\242\243\244\245\246\247\250\251\252\253\254\255\256\257%
    \260\261\262\263\264\265\266\267\270\271\272\273\274\275\276\277%
    \300\301\302\303\304\305\306\307\310\311\312\313\314\315\316\317%
    \320\321\322\323\324\325\326\327\330\331\332\333\334\335\336\337%
```

```
    \340\341\342\343\344\345\346\347\350\351\352\353\354\355\356\357%
    \360\361\362\363\364\365\366\367\370\371\372\373\374\375\376\377%
  }%
|endgroup
\@onelevel@sanitize\AllBytesString

⟨*test4⟩
\let\org@detokenize\detokenize
\let\detokenize\@undefined
\let\org@numexpr\numexpr
\let\numexpr\@undefined
⟨/test4⟩
\RequirePackage{pdfescape}
⟨*test4⟩
\let\detokenize\org@detokenize
\let\numexpr\org@numexpr
⟨/test4⟩

\begin{qstest}{all-hex}{\AllBytes, escapehex}
  \EdefEscapeHex\x{\AllBytes}%
  \Expect*{\x}*{\AllBytesHex}%
  \ExpectVar\x\AllBytesHex
\end{qstest}

\begin{qstest}{all-unhex}{\AllBytesHex, unescapehex}
  \EdefUnescapeHex\x{\AllBytesHex}%
  \Expect*{\x}*{\AllBytes}%
  \ExpectVar\x\AllBytes
\end{qstest}

\begin{qstest}{all-unhex-lc}{\AllBytesHexLC, unescapehex, lowercase}
  \EdefUnescapeHex\x{\AllBytesHexLC}%
  \Expect*{\x}*{\AllBytes}%
  \ExpectVar\x\AllBytes
\end{qstest}

\begin{qstest}{unhex-incomplete}{unescapehex, incomplete}
  \EdefUnescapeHex\x{4}%
  \Expect*{\x}{@}%
\end{qstest}

\begin{qstest}{unhex-space}{unescapehex, space}
  \EdefUnescapeHex\x{20}%
  \Expect*{\x}{ }%
  \ExpectVar\x\space
\end{qstest}

\begin{qstest}{unhex-spaces}{unescapehex, spaces}
  \EdefUnescapeHex\x{204020204120}%
  \def\y#1{%
    \edef\z{#1\string @#1#1\string A#1}%
  }\y{ }%
  \Expect*{\x}*{\z}%
  \ExpectVar\x\z
\end{qstest}

\begin{qstest}{unhex-hash}{unescapehex, hash}
  \catcode`\#=12 %
  \EdefUnescapeHex\x{#20}%
  \ExpectVar\x\space
\end{qstest}
```

```
\begin{qstest}{unhex-invalid}{unescapehex, invalid}
  \def\test#1#2{%
    \EdefUnescapeHex\x{#1}%
    \edef\y{#2}%
    \@onelevel@sanitize\y
    \ExpectVar\x\y
  }%
⟨*test2⟩
  \edef\x{\pdfunescapehex{4X}}%
  \edef\y{\string @}%
  \ifx\x\y
  \else
    \def~{\space}%
    \typeout{**********************************}%
    \typeout{* Your pdfTeX contains bug 777.~~~~*}%
    \typeout{* This test is redefined as dummy, *}%
    \typeout{* because it triggers the bug.~~~~~*}%
    \typeout{**********************************}%
    \def\test#1#2{}%
  \fi
⟨/test2⟩
  \test{X}{}%
  \test{XY}{}%
  \test{XYZ}{}%
  \test{A}{^^a0}%
  \test{AX}{^^a0}%
  \test{XA}{^^a0}%
  \test{XXAXX}{^^a0}%
\end{qstest}

\begin{qstest}{all-name}{\AllBytes, escapename}
  \EdefEscapeName\x{\AllBytes}%
  \Expect*{\x}*{\AllBytesName}%
  \ExpectVar\x\AllBytesName
\end{qstest}

\begin{qstest}{all-string}{\AllBytes, escapestring}
  \EdefEscapeString\x{\AllBytes}%
  \Expect*{\x}*{\AllBytesString}%
  \ExpectVar\x\AllBytesString
\end{qstest}

\begin{qstest}{uchexdigit}{unescape, uppercase hex digit}
  \catcode`\@=11 %
  \catcode0=12 %
  \def\test#1#2{%
    \uccode0=#1\relax
    \uppercase{%
      \def\x{^^@}%
    }%
    \Expect*{%
      \ifcase\expandafter\PE@TestUcHexDigit\x
        true%
      \else
        false%
      \fi
    }{#2}%
  }%
  \def\range#1#2#3{%
    \count0=#1\relax
    \loop
    \ifnum\count0<#2\relax
```

```
      \test{\count0}{#3}%
      \advance\count0 by 1 %
    \repeat
  }%
  \range{0}{47}{false}%
  \range{48}{57}{true}%
  \range{58}{64}{false}%
  \range{65}{70}{true}%
  \range{71}{255}{false}%
\end{qstest}

\begin{qstest}{unescapename}{unescapename}
  \def\test#1#2{%
    \EdefUnescapeName\x{#1}%
    \edef\y{#2}%
    \@onelevel@sanitize\y
    \ExpectVar\x\y
  }%
  \catcode`\#=12 %
  \catcode0=12 %
  \test{}{}%
  \test{x}{x}%
  \test{xy}{xy}%
  \test{#}{#}%
  \test{##}{##}%
  \test{###}{###}%
  \test{####}{####}%
  \test{#x}{#x}%
  \test{#xy}{#xy}%
  \test{#1}{#1}%
  \test{#40}{@}%
  \test{#400}{@0}%
  \test{#4x0}{#4x0}%
  \test{#ab}{^^ab}%
  \test{#00}{^^@}%
  \test{x#40y#40z}{x@y@z}%
  \test{#40#40#40#40}{@@@@}%
  \test{a#x}{a#x}%
  \test{a#xy}{a#xy}%
  \test{a#1}{a#1}%
  \test{a#40}{a@}%
  \test{a#400}{a@0}%
  \test{#20}{ }%
  \test{a#20}{a }%
  \test{a#20b}{a b}%
  \test{a#20#20#20b}{a \space\space b}%
\end{qstest}

\begin{qstest}{unescapestring}{unescapestring}
  \def\test#1#2{%
    \EdefUnescapeString\x{#1}%
    \edef\y{#2}%
    \@onelevel@sanitize\y
    \ExpectVar\x\y
  }%
  \catcode0=12 %
  \def\DefChar#1#2{%
    \begingroup
      \uccode0=#2\relax
    \uppercase{\endgroup
      \def#1{^^@}%
    }%
```

```
}%
\DefChar\nul{0}%
\DefChar\one{1}%
\DefChar\bel{8}%
\DefChar\tab{9}%
\DefChar\lf{10}%
\DefChar\ff{12}%
\DefChar\cr{13}%
\DefChar\\{92}%
\test{}{}%
\test{a}{a}%
\test{\\}{}%
\test{\\\\}{\\}%
\test{\\\\y}{\\y}%
\test{\\000}{\nul}%
\test{\\b}{\bel}%
\test{\\t}{\tab}%
\test{\\n}{\lf}%
\test{\\f}{\ff}%
\test{\\r}{\cr}%
\test{\\(}{(}%
\test{\\)}{)}%
\test{\\040}{ }%
\test{\\100}{@}%
\test{\\40}{ }%
\test{\\1}{\one}%
\test{\\01}{\one}%
\test{\\001}{\one}%
\test{\\18}{\one8}%
\test{\\018}{\one8}%
\test{\\0018}{\one8}%
\test{x\\}{x}%
\test{x\\\\}{x\\}%
\test{x\\\\y}{x\\y}%
\test{x\\000}{x\nul}%
\test{x\\b}{x\bel}%
\test{x\\t}{x\tab}%
\test{x\\n}{x\lf}%
\test{x\\f}{x\ff}%
\test{x\\r}{x\cr}%
\test{x\\(}{x(}%
\test{x\\)}{x)}%
\test{x\\040}{x }%
\test{x\\100}{x@}%
\test{x\\40}{x }%
\test{x\\1}{x\one}%
\test{x\\01}{x\one}%
\test{x\\001}{x\one}%
\test{x\\18}{x\one8}%
\test{x\\018}{x\one8}%
\test{x\\0018}{x\one8}%
\test{\\b\\t\\n\\f\\r\\(\\)\\\\\\000\\040}{%
  \bel\tab\lf\ff\cr()\\\nul\space
}%
\test{\\\lf}{}%
\test{x\\\lf}{x}%
\test{\cr}{\lf}%
\test{\cr\lf}{\lf}%
\test{\lf}{\lf}%
\test{\lf\cr}{\lf\lf}%
\test{x\cr}{x\lf}%
\test{x\cr\lf}{x\lf}%
```

```
  \test{x\lf}{x\lf}%
  \test{x\lf\cr}{x\lf\lf}%
  \test{x\\\cr\lf y\cr}{xy\lf}%
%
  \test{\\409}{ 9}%
  \test{\\800}{800}%
  \test{\\900}{900}%
  \test{\\578}{/8}%
  \test{\\477}{?}%
  \test{\\377}{^^ff}%
  \test{\\777}{^^ff}%
  \test{\\7777}{^^ff7}%
\end{qstest}
\stop
```

⟨/test2 | test3 | test4 | test5⟩

3.8.1 Test for iniTEX

```
⟨*test6⟩
\input pdfescape.sty\relax
\catcode`\{=1 %
\catcode`\}=2 %
\catcode`\#=6 %
\catcode`\^=7 %
\catcode`\@=11 %
\begingroup
  \catcode`\@=11 %
  \countdef\count@=255 %
  \def\space{ }%
  \long\def\@whilenum#1\do #2{%
    \ifnum #1\relax
      #2\relax
      \@iwhilenum{#1\relax#2\relax}%
    \fi
  }%
  \long\def\@iwhilenum#1{%
    \ifnum #1%
      \expandafter\@iwhilenum
    \else
      \expandafter\ltx@gobble
    \fi
    {#1}%
  }%
  \gdef\AllBytes{}%
  \count@=0 %
  \catcode0=12 %
  \@whilenum\count@<256 \do{%
    \lccode0=\count@
    \ifnum\count@=32 %
      \xdef\AllBytes{\AllBytes\space}%
    \else
      \lowercase{%
        \xdef\AllBytes{\AllBytes^^@}%
      }%
    \fi
    \advance\count@ by 1 %
  }%
\endgroup
\def\AllBytesHex{%
  000102030405060708090A0B0C0D0E0F%
  101112131415161718191A1B1C1D1E1F%
  202122232425262728292A2B2C2D2E2F%
```

```
  303132333435363738393A3B3C3D3E3F%
  404142434445464748494A4B4C4D4E4F%
  505152535455565758595A5B5C5D5E5F%
  606162636465666768696A6B6C6D6E6F%
  707172737475767778797A7B7C7D7E7F%
  808182838485868788898A8B8C8D8E8F%
  909192939495969798999A9B9C9D9E9F%
  A0A1A2A3A4A5A6A7A8A9AAABACADAEAF%
  B0B1B2B3B4B5B6B7B8B9BABBBCBDBEBF%
  C0C1C2C3C4C5C6C7C8C9CACBCCCDCECF%
  D0D1D2D3D4D5D6D7D8D9DADBDCDDDEDF%
  E0E1E2E3E4E5E6E7E8E9EAEBECEDEEEF%
  F0F1F2F3F4F5F6F7F8F9FAFBFCFDFEFF%
}
\ltx@onelevel@sanitize\AllBytesHex
\expandafter\lowercase\expandafter{%
  \expandafter\def\expandafter\AllBytesHexLC
     \expandafter{\AllBytesHex}%
}
\begingroup
  \catcode`\#=12 %
  \xdef\AllBytesName{%
    #01#02#03#04#05#06#07#08#09#0A#0B#0C#0D#0E#0F%
    #10#11#12#13#14#15#16#17#18#19#1A#1B#1C#1D#1E#1F%
    #20!"#23$#25&'#28#29*+,-.#2F%
    0123456789:;#3C=#3E?%
    @ABCDEFGHIJKLMNO%
    PQRSTUVWXYZ#5B\ltx@backslashchar#5D^_%
    `abcdefghijklmno%
    pqrstuvwxyz#7B|#7D\string~#7F%
    #80#81#82#83#84#85#86#87#88#89#8A#8B#8C#8D#8E#8F%
    #90#91#92#93#94#95#96#97#98#99#9A#9B#9C#9D#9E#9F%
   #A0#A1#A2#A3#A4#A5#A6#A7#A8#A9#AA#AB#AC#AD#AE#AF%
    #B0#B1#B2#B3#B4#B5#B6#B7#B8#B9#BA#BB#BC#BD#BE#BF%
   #C0#C1#C2#C3#C4#C5#C6#C7#C8#C9#CA#CB#CC#CD#CE#CF%
   #D0#D1#D2#D3#D4#D5#D6#D7#D8#D9#DA#DB#DC#DD#DE#DF%
    #E0#E1#E2#E3#E4#E5#E6#E7#E8#E9#EA#EB#EC#ED#EE#EF%
    #F0#F1#F2#F3#F4#F5#F6#F7#F8#F9#FA#FB#FC#FD#FE#FF%
  }%
\endgroup
\ltx@onelevel@sanitize\AllBytesName
\edef\AllBytesFromName{\expandafter\ltx@gobble\AllBytes}

\begingroup
  \def\|{|}%
  \edef\%{\ltx@percentchar}%
  \catcode`\|=0 %
  \catcode`\#=12 %
  \catcode`\~=12 %
  \catcode`\\=12 %
  |xdef|AllBytesString{%
    \000\001\002\003\004\005\006\007\010\011\012\013\014\015\016\017%
    \020\021\022\023\024\025\026\027\030\031\032\033\034\035\036\037%
    \040!"#$|%&'\(\)*+,-./%
    0123456789:;<=>?%
    @ABCDEFGHIJKLMNO%
    PQRSTUVWXYZ[\\]^_%
    `abcdefghijklmno%
    pqrstuvwxyz{||}~\177%
    \200\201\202\203\204\205\206\207\210\211\212\213\214\215\216\217%
    \220\221\222\223\224\225\226\227\230\231\232\233\234\235\236\237%
    \240\241\242\243\244\245\246\247\250\251\252\253\254\255\256\257%
```

```
    \260\261\262\263\264\265\266\267\270\271\272\273\274\275\276\277%
    \300\301\302\303\304\305\306\307\310\311\312\313\314\315\316\317%
    \320\321\322\323\324\325\326\327\330\331\332\333\334\335\336\337%
    \340\341\342\343\344\345\346\347\350\351\352\353\354\355\356\357%
    \360\361\362\363\364\365\366\367\370\371\372\373\374\375\376\377%
  }%
|endgroup
\ltx@onelevel@sanitize\AllBytesString

\def\msg#{\immediate\write16}
\def\Test#1#2#3{%
  \begingroup
    #1\TestResult{#2}%
    \ifx\TestResult#3%
    \else
      \newlinechar=10 %
      \msg{Expect:^^J#3}%
      \msg{Result:^^J\TestResult}%
      \errmessage{\string#2 -\string#1-> \string#3}%
    \fi
  \endgroup
}
\Test\EdefEscapeHex\AllBytes\AllBytesHex
\Test\EdefUnescapeHex\AllBytesHex\AllBytes
\Test\EdefEscapeName\AllBytes\AllBytesName
\Test\EdefUnescapeName\AllBytesName\AllBytesFromName
\Test\EdefEscapeString\AllBytes\AllBytesString
\Test\EdefUnescapeString\AllBytesString\AllBytes

\csname @@end\endcsname\end
⟨/test6⟩
```

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] http://ctan.org/pkg/pdfescape

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
test/pdfescape-test6.tex → doc/latex/oberdiek/test/pdfescape-test6.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

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 Catalogue

The following XML file can be used as source for the TEX Catalogue. The elements caption and description are imported from the original XML file from the Catalogue. The name of the XML file in the Catalogue is pdfescape.xml.

```
⟨*catalogue⟩
<?xml version='1.0' encoding='us-ascii'?>
<!DOCTYPE entry SYSTEM 'catalogue.dtd'>
<entry datestamp='$Date$' modifier='$Author$' id='pdfescape'>
  <name>pdfescape</name>
  <caption>Implements pdfTeX's escape features using TeX or e-TeX.</caption>
  <authorref id='auth:oberdiek'/>
  <copyright owner='Heiko Oberdiek' year='2007,2010,2011'/>
  <license type='lppl1.3'/>
  <version number='1.14'/>
  <description>
    This package implements <xref refid='pdftex'>pdfTeX</xref>'s
    escape features (<tt>\pdfescapehex</tt>, <tt>\pdfunescapehex</tt>,
    <tt>\pdfescapename</tt>, <tt>\pdfescapestring</tt>) using TeX or
    e-TeX.
    <p/>
    The package is part of the <xref refid='oberdiek'>oberdiek</xref>
    bundle.
  </description>
  <documentation details='Package documentation'
      href='ctan:/macros/latex/contrib/oberdiek/pdfescape.pdf'/>
  <ctan file='true' path='/macros/latex/contrib/oberdiek/pdfescape.dtx'/>
  <miktex location='oberdiek'/>
  <texlive location='oberdiek'/>
  <install path='/macros/latex/contrib/oberdiek/oberdiek.tds.zip'/>
</entry>
⟨/catalogue⟩
```

6 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.

[2011/11/25 v1.13]

- Higher order bit of octal sequences in `\EdefUnescapeString` ignored according to the PDF specification (Bug found by Bruno Le Floch).

[2016/05/16 v1.14]

- Documentation updates.

7 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
\# 191, 327, 352, 740, 943, 970, 1045, 1130, 1255, 1315, 1342

\$ 190, 409, 412
\% 816, 968, 1340
\& 410, 413
\(. 714, 976, 1348
\) 715, 976, 1348
\@ 741, 814, 1093, 1257, 1259
\@backslashchar 950
\@firstofone 749, 752
\@gobble 746, 754
\@ifundefined 856, 870, 878
\@iwhilenum 1265, 1268, 1270
\@onelevel@sanitize 149, 893, 894, 936, 963, 992, 1054, 1127, 1164
\@percentchar 968
\@undefined 58, 864, 865, 866, 867, 996, 998
\@whilenum 906, 1262, 1279
\\ 175, 319, 497, 716, 815, 972, 979, 1182, 1185, 1186, 1187, 1188, 1189, 1190, 1191, 1192, 1193, 1194, 1195, 1196, 1197, 1198, 1199, 1200, 1201, 1202, 1203, 1204, 1205, 1206, 1207, 1208, 1209, 1210, 1211, 1212, 1213, 1214, 1215, 1216, 1217, 1218, 1219, 1220, 1221, 1222, 1223, 1224, 1225, 1226, 1228, 1229, 1238, 1240, 1241, 1242, 1243, 1244, 1245, 1246, 1247, 1344, 1351
\{ 738, 1253
\} 376, 739, 1254
\^ 1256
\| 318, 323, 967, 969, 1339, 1341
\~ 971, 1343

Numbers

\0 . . . . . 974, 975, 976, 1346, 1347, 1348
\1 . . . . . 981, 1353
\2 . . . . . 982, 983, 984, 985, 1354, 1355, 1356, 1357
\3 . . . . . 986, 987, 988, 989, 1358, 1359, 1360, 1361
\8 291
\9 292

\␣ 167, 716

A

\advance 420, 535, 571, 572, 779, 787, 802, 915, 1113, 1288
\aftergroup 29
\AllBytes 903, 909, 912, 1006, 1007, 1014, 1015, 1020, 1021, 1080, 1081, 1086, 1087, 1276, 1282, 1285, 1336, 1378, 1379, 1380, 1382, 1383
\AllBytesFromName 1336, 1381
\AllBytesHex 918, 936, 939, 1008, 1009, 1012, 1013, 1291, 1309, 1312, 1378, 1379
\AllBytesHexLC . 938, 1018, 1019, 1311
\AllBytesName 941, 944, 963, 1082, 1083, 1316, 1335, 1380, 1381
\AllBytesString 965, 992, 1088, 1089, 1364, 1382, 1383

B

\begin 1006, 1012, 1018, 1024, 1029, 1035, 1044, 1050, 1080, 1086, 1092, 1123, 1160
\bel 1177, 1189, 1209, 1226
\body 758, 762

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, 114, 115, 167, 190, 191, 318, 319, 409, 410, 412, 413, 738, 739, 740, 741, 776, 785, 793, 797, 814, 815, 816, 905, 943, 969, 970, 971, 972, 1045, 1093, 1094, 1130, 1131, 1167, 1253, 1254, 1255, 1256, 1257, 1259, 1278, 1315, 1341, 1342, 1343, 1344
\count 419, 420, 423, 1109, 1111, 1112, 1113
\count@ 743, 772, 776, 778, 779, 783, 785, 786, 787, 791, 793, 796, 797, 801, 802, 904, 906, 907, 908, 915, 1260, 1277, 1279, 1280, 1281, 1288
\countdef 743, 1260
\cr . . . 1181, 1193, 1213, 1226, 1230, 1231, 1233, 1234, 1235, 1237, 1238
\csname 14, 21, 50, 66, 76, 121, 124, 133, 136, 143, 154, 439, 442, 451, 474, 502, 742, 745, 748, 751, 806, 833, 1384

D

\DefChar 1168, 1175, 1176, 1177, 1178, 1179, 1180, 1181, 1182
\detokenize . . . 158, 162, 995, 996, 1002
\dimen 533, 534, 535, 538, 539, 568, 569, 570, 571, 572, 576, 577, 578
\dimexpr 526, 528, 553, 556, 564
\do 906, 1262, 1279

E

\EdefEscapeHex *2*, 452, 479, 1007, 1378
\EdefEscapeName . 461, 485, 1081, 1380
\EdefEscapeString . 466, 488, 1087, 1382
\EdefSanitize *3*, 132, 165, 184, 284, 453, 458, 462, 467
\EdefUnescapeHex 457, 482, 1013, 1019, 1025, 1030, 1036, 1046, 1052, 1379
\EdefUnescapeName *2*, 183, 1125, 1381

\EdefUnescapeString 3, 283, 1162, 1383
\empty 17, 18
\end 834, 1010, 1016, 1022, 1027, 1033, 1042, 1048, 1078, 1084, 1090, 1121, 1158, 1248, 1384
\endcsname 14, 21, 50, 66, 76, 121, 124, 133, 136, 143, 154, 439, 442, 451, 474, 502, 742, 745, 748, 751, 806, 833, 1384
\endinput 29, 119
\endlinechar 4, 35, 71, 77, 89
\errmessage 795, 1374
\escapechar 495
\eTeX 872
\Expect 898, 1008, 1014, 1020, 1026, 1031, 1040, 1082, 1088, 1100
\ExpectVar 889, 1009, 1015, 1021, 1032, 1041, 1047, 1055, 1083, 1089, 1128, 1165

F

\ff 1180, 1192, 1212, 1226

G

\gdef 192, 200, 903, 1276

I

\ifcase 214, 215, 393, 401, 545, 598, 611, 612, 651, 1101
\ifnum 265, 268, 271, 272, 382, 385, 416, 614, 650, 658, 661, 662, 663, 664, 665, 666, 667, 668, 669, 670, 671, 709, 712, 714, 715, 716, 778, 786, 793, 801, 908, 1111, 1263, 1269, 1281
\ifPE@etex 500, 506, 524, 550, 636, 683, 693, 725
\ifx . 15, 18, 21, 50, 58, 61, 121, 124, 133, 143, 175, 201, 205, 300, 305, 439, 442, 451, 502, 518, 591, 592, 648, 707, 742, 745, 748, 751, 806, 890, 898, 1060, 1369
\immediate 23, 52, 1365
\IncludeTests 886
\input 125, 443, 807, 1252
\iterate 759, 761, 763

L

\lccode 907, 1280
\lf 1179, 1191, 1211, 1226, 1228, 1229, 1230, 1231, 1232, 1233, 1234, 1235, 1236, 1237, 1238
\LoadCommand 807, 817
\LogTests 887
\loop 757, 773, 784, 792, 1110
\lowercase 911, 937, 1284, 1310
\ltx@backslashchar 1322
\ltx@cclv 419, 420, 423
\ltx@gobble 1272, 1336
\ltx@IfUndefined 408
\ltx@newif 500
\ltx@one 218, 221, 266, 269, 273, 383, 386
\ltx@onelevel@sanitize 1309, 1335, 1364
\ltx@percentchar 1340
\ltx@ReturnAfterFi 178
\ltx@zero 216, 223, 227, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 275, 278, 388, 432, 613, 614

M

\makeatletter 838, 901
\meaning 145
\msg 1365, 1372, 1373

N

\NeedsTeXFormat 837
\newcommand . . 889, 918, 938, 941, 965
\newlinechar 1371
\next 763, 765, 767
\nul 1175, 1188, 1208, 1226
\number 576, 577, 578, 798
\numexpr 424, 526, 528, 553, 556, 563, 564, 997, 998, 1003

O

\one 1176, 1199, 1200, 1201, 1202, 1203, 1204, 1219, 1220, 1221, 1222, 1223, 1224
\org@detokenize 995, 1002
\org@numexpr 997, 1003

P

\PackageError 857, 871, 879
\PackageInfo 26
\pdf@escapehex 480
\pdf@escapename 486
\pdf@escapestring 489
\pdf@unescapehex 483
\pdfescape 841, 845, 849, 858
\pdfescapehex 864, 1398
\pdfescapename 866, 1399
\pdfescapestring 867, 1399
\pdfunescapehex 865, 1058, 1398
\PE@@NormalizeLineEnd 296, 299
\PE@@OctChar 552, 560
\PE@AtEnd 95, 96, 119, 491, 735
\PE@backslash 494, 561, 575
\PE@CheckEndBackslash 373
\PE@DeHex 586, 590
\PE@DeName 196, 200, 229, 233
\PE@DeString 327, 436
\PE@edefbabel . 472, 480, 483, 486, 489
\PE@EscapeHex 455, 506
\PE@EscapeName 464, 636
\PE@EscapeNameAdd 654, 659, 673, 681, 683
\PE@EscapeNameHashChar 661, 662, 663, 664, 665, 666, 667, 668, 669, 670, 671, 680
\PE@EscapeNameTokens . 638, 643, 647
\PE@EscapeString 469, 693
\PE@EscapeStringAdd 714, 715, 716, 718, 725
\PE@EscapeStringTokens 695, 701, 706
\PE@etextrue 504

\PE@hash 493, 654, 659, 681
\PE@HexChar 520, 524, 655, 660
\PE@HexDigit . . 526, 527, 538, 539, 543
\PE@InitUccodeHexDigit . 194, 240, 584
\PE@next 203, 208, 229, 233, 237, 302, 306, 308, 311, 394, 396, 398, 402, 404, 406, 417, 421, 429, 593, 599, 603, 622, 625, 631, 634
\PE@NormalizeLineEnd 286, 294
\PE@OctAll . . . 396, 404, 417, 422, 431
\PE@OctChar 550, 710, 713
\PE@OctI 392
\PE@OctII 394, 400
\PE@OctIII 402, 415
\PE@onelevel@sanitize 138, 144, 149, 161, 187, 288
\PE@result 195, 198, 202, 207, 228, 232, 295, 297, 301, 304, 434, 512, 514, 536, 537, 573, 574, 585, 588, 621, 642, 644, 687, 688, 700, 703, 729, 730
\PE@sanitize 165
\PE@SanitizeSpaceOther 170, 185, 285, 454, 463, 468
\PE@space@other 166, 177
\PE@space@space 169, 615
\PE@SpaceToOther 171, 173
\PE@strip@prefix 145, 147
\PE@temp 225, 228, 615, 618, 621
\PE@testA 211, 214, 223, 596, 598, 608, 611, 613
\PE@testB . 212, 215, 223, 609, 612, 613
\PE@TestOctDigit 381, 393, 401
\PE@TestUcHexDigit 214, 215, 264, 598, 611, 612, 1101
\PE@ToHex 508, 513, 517
\PE@UnescapeHex 459, 582
\PE@UnescapeName 186, 189
\PE@UnescapeString 287, 320
\ProvidesFile 840, 844, 848, 852
\ProvidesPackage 19, 67

R

\range 1108, 1116, 1117, 1118, 1119, 1120
\RangeCatcodeCheck 790, 818, 819, 820, 821, 822, 823, 824, 825, 826, 827, 828, 829
\RangeCatcodeInvalid 782, 810, 811, 812, 813
\repeat . . 757, 769, 780, 788, 803, 1114
\RequirePackage . . . 130, 448, 885, 1000
\RestoreCatcodes . . 771, 774, 775, 830

S

\space 796, 797, 805, 909, 1032, 1047, 1062, 1157, 1226, 1261, 1282
\stop 860, 874, 882, 1249

T

\tab 1178, 1190, 1210, 1226
\Test 809, 832, 1366, 1378, 1379, 1380, 1381, 1382, 1383
\test . 1051, 1068, 1071, 1072, 1073, 1074, 1075, 1076, 1077, 1095, 1112, 1124, 1132, 1133, 1134, 1135, 1136, 1137, 1138, 1139, 1140, 1141, 1142, 1143, 1144, 1145, 1146, 1147, 1148, 1149, 1150, 1151, 1152, 1153, 1154, 1155, 1156, 1157, 1161, 1183, 1184, 1185, 1186, 1187, 1188, 1189, 1190, 1191, 1192, 1193, 1194, 1195, 1196, 1197, 1198, 1199, 1200, 1201, 1202, 1203, 1204, 1205, 1206, 1207, 1208, 1209, 1210, 1211, 1212, 1213, 1214, 1215, 1216, 1217, 1218, 1219, 1220, 1221, 1222, 1223, 1224, 1225, 1228, 1229, 1230, 1231, 1232, 1233, 1234, 1235, 1236, 1237, 1238, 1240, 1241, 1242, 1243, 1244, 1245, 1246, 1247
\TestResult 1368, 1369, 1373
\the 77, 78, 79, 80, 81, 82, 83, 84, 97, 423, 424, 553, 556, 563, 564, 776, 796, 797
\TMP@EnsureCode 94, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118
\TMP@RequirePackage 122, 128, 440, 446
\typeout 895, 1063, 1064, 1065, 1066, 1067

U

\uccode 116, 117, 118, 223, 227, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 291, 292, 432, 613, 614, 1096, 1170
\uppercase 210, 224, 314, 433, 595, 607, 617, 1097, 1171

W

\write 23, 52, 1365

X

\x 14, 15, 18, 22, 26, 28, 51, 56, 66, 75, 87, 168, 293, 315, 496, 499, 1007, 1008, 1009, 1013, 1014, 1015, 1019, 1020, 1021, 1025, 1026, 1030, 1031, 1032, 1036, 1040, 1041, 1046, 1047, 1052, 1055, 1058, 1060, 1081, 1082, 1083, 1087, 1088, 1089, 1098, 1101, 1125, 1128, 1162, 1165

Y

\y 1037, 1039, 1053, 1054, 1055, 1059, 1060, 1126, 1127, 1128, 1163, 1164, 1165

Z

\z 1038, 1040, 1041