# The pdfescape package

Heiko Oberdiek
<oberdiek@uni-freiburg.de>

2006/02/25 v1.1

**Abstract**

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

## Contents

# 1 Documentation

| |
|---|
| `\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.

# 2 Implementation

```
⟨*package⟩
```

## 2.1 Reload check and package identification

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

```
\begingroup
  \expandafter\let\expandafter\x\csname ver@pdfescape.sty\endcsname
  \ifcase 0%
    \ifx\x\relax % plain
    \else
      \ifx\x\empty % LaTeX
      \else
        1%
      \fi
    \fi
  \else
    \expandafter\ifx\csname PackageInfo\endcsname\relax
      \def\x#1#2{%
        \immediate\write-1{Package #1 Info: #2.}%
      }%
    \else
      \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
    \fi
    \x{pdfescape}{The package is already loaded}%
    \endgroup
    \expandafter\endinput
  \fi
\endgroup
```

Package identification:

```
\begingroup
  \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\relax
        \xdef#1{#3}%
      \fi
    }%
  \fi
\expandafter\x\csname ver@pdfescape.sty\endcsname
\ProvidesPackage{pdfescape}%
  [2006/02/25 v1.1 Provides hex, PDF name and string conversions (HO)]
```

## 2.2 Preparations

### 2.2.1 Catcodes

```
\expandafter\edef\csname PE@AtEnd\endcsname{%
  \catcode64 \the\catcode64\relax
}
\catcode64 11 % @
\def\PE@EnsureCode#1#2#3{%
  \edef\PE@AtEnd{%
    \PE@AtEnd
    #1#2 \the#1#2\relax
  }%
  #1#2 #3\relax
}
\PE@EnsureCode\catcode{0}{12}% ^^@
\PE@EnsureCode\catcode{34}{12}% "
\PE@EnsureCode\catcode{42}{12}% *
\PE@EnsureCode\catcode{45}{12}% -
\PE@EnsureCode\catcode{46}{12}% .
\PE@EnsureCode\catcode{60}{12}% <
\PE@EnsureCode\catcode{61}{12}% =
\PE@EnsureCode\catcode{62}{12}% >
\PE@EnsureCode\catcode{94}{7}% ^
\PE@EnsureCode\catcode{96}{12}% `
\PE@EnsureCode\uccode{34}{0}% "
\PE@EnsureCode\uccode{48}{0}% 0
\PE@EnsureCode\uccode{61}{0}% =
```

## 2.3 User macros

```
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname pdfescapehex\endcsname\relax
```

\EdefEscapeHex

```
  \long\def\EdefEscapeHex#1#2{%
    \PE@sanitize#1{#2}%
    \edef#1{\expandafter\PE@SpaceToOther#1 \relax}%
    \PE@EscapeHex#1%
  }%
```

\EdefUnescapeHex

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

\EdefEscapeName

```
  \long\def\EdefEscapeName#1#2{%
    \PE@sanitize#1{#2}%
    \edef#1{\expandafter\PE@SpaceToOther#1 \relax}%
    \PE@EscapeName#1%
  }%
```

\EdefEscapeString

```
  \long\def\EdefEscapeString#1#2{%
    \PE@sanitize#1{#2}%
    \edef#1{\expandafter\PE@SpaceToOther#1 \relax}%
    \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\pdfescapehex{#2}%
  }%
```

\EdefUnescapeHex

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

\EdefEscapeName

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

\EdefEscapeString

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

  \PE@AtEnd
  \expandafter\endinput
\fi
```

## 2.4 Help macros

### 2.4.1 Characters

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

\PE@hash

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

\PE@space@other

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

\PE@space@space

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

\PE@backslash

```
\begingroup
  \long\def\@gobble#1{}%
  \escapechar=92 %
\edef\x{\endgroup
  \def\noexpand\PE@backslash{\expandafter\@gobble\string\\}%
}
\x
```

### 2.4.2 Switch for $\varepsilon$-TEX

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

## 2.5 Sanitizing

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

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

\PE@SpaceToOther

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

\@ReturnAfterFi

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

## 2.6 Conversions

### 2.6.1 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
}
```

### 2.6.2 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
```

### 2.6.3 Unpack hex string

\PE@UnescapeHex

```
\def\PE@UnescapeHex#1{%
  \begingroup
    \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
    \else
      \PE@DeHex#10\relax\relax
    \fi
  \else
    \uppercase{\lccode0="#1#2}\relax
    \ifnum\lccode0=32 %
      \edef\PE@result{\PE@result\PE@space@space}%
    \else
      \lowercase{\def\PE@temp{^^@}}%
      \edef\PE@result{\PE@result\PE@temp}%
    \fi
```

```
    \expandafter\PE@DeHex
  \fi
}
```

### 2.6.4 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
```

### 2.6.5 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

```
⟨*test1 | test2 | test3⟩
\NeedsTeXFormat{LaTeX2e}
\makeatletter
```

## 3.1 Test with \pdfescape... commands

```
⟨*test1⟩
\ProvidesFile{pdfescape-test1.tex}%
    [2006/02/25 v1.1 Test with \string\pdfescape... commands]
⟨/test1⟩
```

## 3.2 Test without \pdfescape..., with ε-TEX

```
⟨*test2⟩
\ProvidesFile{pdfescape-test2.tex}%
    [2006/02/25 v1.1 Test without \string\pdfescape..., with e-TeX]
⟨/test2⟩
```

## 3.3 Test without \pdfescape... and ε-TEX

```
⟨*test3⟩
\ProvidesFile{pdfescape-test3.tex}%
    [2006/02/25 v1.1 Test without \string\pdfescape... and e-TeX]
⟨/test3⟩
```

## 3.4 Check/ensure test preconditions

### 3.4.1 Check \pdfescape...

```
⟨*test1⟩
\@ifundefined{pdfescapehex}{%
  \PackageError{pdfescape-test1}{%
    Missing \string\pdfescape... commands%
  }{Test aborted.}%
  \stop
}{}
⟨/test1⟩
```

```
⟨*test2 | test3⟩
\let\pdfescapehex\@undefined
\let\pdfunescapehex\@undefined
\let\pdfescapename\@undefined
\let\pdfescapestring\@undefined
⟨/test2 | test3⟩
```

### 3.4.2 Check ε-TEX

```
⟨*test2⟩
\@ifundefined{numexpr}{%
  \PackageError{pdfescape-test2}{%
    Missing \eTeX
  }{Test aborted.}%
  \stop
}{}
⟨/test2⟩
```

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

## 3.5 Common part

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

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

```
\newcommand*{\ExpectVar}[2]{%
  \Expect*{\ifx#1#2true\else false\fi}{true}%
}

\makeatletter


\begingroup
  \makeatletter
  \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

⟨*test3⟩
\let\org@detokenize\detokenize
\let\detokenize\@undefined
\let\org@numexpr\numexpr
\let\numexpr\@undefined
⟨/test3⟩
\RequirePackage{pdfescape}
⟨*test3⟩
\let\detokenize\org@detokenize
\let\numexpr\org@numexpr
⟨/test3⟩

\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}{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}

\stop
```

⟨/test1 | test2 | test3⟩

# 4 Installation

**CTAN.** 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.

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

```
tex pdfescape.dtx
```

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

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

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

If you have a `docstrip.cfg` that configures and enables docstrip's TDS installing feature, then some files can already be in the right place, see the documentation of docstrip.

**Refresh file databases.** If your TEX distribution (teTEX, mikTEX, ...) rely on file databases, you must refresh these. For example, teTEX users run `texhash` or `mktexlsr`.

### 4.1 Some details for the interested

**Attached source.** The PDF documentation on CTAN also includes the `.dtx` source file. It can be extracted by AcrobatReader 6 or higher. Another option is pdftk, e.g. unpack the file into the current directory:

```
pdftk pdfescape.pdf unpack_files output .
```

**Unpacking with LATEX.** The `.dtx` chooses its action depending on the format:

**plain-TEX:** Run docstrip and extract the files.

**LATEX:** Generate the documentation.

If you insist on using LATEX for docstrip (really, docstrip does not need LATEX), then inform the autodetect routine about your intention:

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

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

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

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

An example follows how to generate the documentation with pdfLATEX:

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

## 5 History

### [2007/02/21 v1.0]

- First version.

### [2007/02/25 v1.1]

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

# 6 Index

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