The thepdfnumber package

Heiko Oberdiek*

2016/05/16 v1.1

Abstract

The package converts real numbers to a minimal representation that is stripped from leading or trailing zeros, plus signs and decimal point if not necessary.

Contents

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

1 Documentation

1.1 Introduction

Dealing with the PDF format, there is sometimes the need to write some low level PDF stuff. In case of numbers, the numbers can arise from user input (e.g. color or transparency specifications) or can be calculated. For example, LaTeX's `\strip@pt` makes a good job to output a real number. It automatically suppresses the decimal part if the number is an integer. However it leaves a leading zero for numbers greater zero and smaller one. Thus the package provides macros that can be used with different formats, even with iniTeX and generates numbers that are valid numbers of the PDF format and whose length is minimal.

1.2 Usage

The package thepdfnumber can be used with LaTeX, plain TeX or even with iniTeX:

```
\RequirePackage{thepdfnumber} % LaTeX
\input thepdfnumber.sty % plain TeX/iniTeX
```

The package does not need and have package options.

1.3 User macros

All user macros are expandable in exact two expansion steps.

`\thepdfnumber` {⟨*number*⟩}

Macro `\thepdfnumber` takes a number as argument and expands to a minimal representation of that number. Some examples:

```
1.: +123      → 123
2.: --123     → 123
3.: -01       → -1
4.: 0045      → 45
5.: 1.0       → 1
6.: 1.20      → 1.2
7.: 0.0       → 0
8.: 0.78      → .78
9.: +012.340 → 12.34
```

It reduces the length of the number representation:

- The signs are collapsed and only one minus sign is output if the number is negative (see examples 1, 2, 3, 9).
- Leading zeros are removed (4, 8, 9) unless the number is zero (7).
- The decimal part is omitted, if the number is an integer (5, 7).
- Trailing zeros from the decimal part are stripped (5, 6, 7, 9).

The resulting number representation can be catched with one of the following Perl regular expressions:

- `^0$` (zero)
- `^-?[1-9][0-9]*$` (integer)
- `^-?[0-9]*\.[0-9]*[1-9]$` (real)

This is a valid numeric object of the PDF specification [1, "7.3.3 Numeric Objects"].

`\thepdfnumberNormZeroOne`

There are various places in the PDF specification where the number is in the domain 0.0 upto 1.0. Macro thepdfnumberNormZeroOne automatically adjusts the number to fit into that range. Negative numbers are mapped to `0` and numbers greater than one are replaced by `1`. Thus the result fits one of the following regular expressions:

- `^0$`
- `^\.[0-9]*[1-9]$`
- `^1$`

Examples:

```
-456   → 0
-0.001 → 0
0.0    → 0
0.010  → .01
0.456  → .456
1.0    → 1
01.001 → 1
4      → 1
```

1.4 Input number

The user macros expect a number as argument. The number can either be given explicitely or as macro that expands in one step to an explicite number, because the first token of the argument is expanded once.

The explicite number consists of

- optional signs '`+`' and '`-`',
- digits '`0`' upto '`9`' and
- an optional dot '`.`'.

All tokens must have catcode 12 (other), the default catcodes for these characters in LaTeX, plain TeX or iniTeX. As Perl regular expression the number is expected in one of the following forms:

- `^[+-]*[0-9]+$`
- `^[+-]*[0-9]*\.[0-9]*$`

At least one digit or the dot must be present.

1.5 Error handling

The package is not intended for validating numbers or to decide if an argument is a number. Therefore it is an usage error to use the user macros with arguments that are not explicite numbers as specified in the previous sections. Nevertheless some error conditions are sometimes recognized. Errors are given in form of an undefined command sequence. It is the only way to notify TEX in expandable context. Expanding to some error text would invalidate the output. Currently the following errors are thrown:

`\thepdfnumber@ErrorEndMarker`: Internally the argument parsing uses an end marker that is never called directly. If it is called with valid user input, then this is a bug. Otherwise it means the user input contains nasty stuff.

`\thepdfnumber@ErrorUnexpectedEnd`: The macros expect at least one digit or the dot, otherwise if the argument is empty or only contains signs, then this error is called.

`\thepdfnumber@ErrorInvalidToken`: It is called if the number contains other tokens than signs, digits or the dot or the token at the wrong place (e.g. a sign after a digit). In case of `\thepdfnumberNormZeroOne` this error condition might not always be detected, because the number parsing might stop at an early point, when the result is already clear (e.g. if the number is negative or will be greater than one).

Improper alphabetic constant: This error might be thrown by TEX, if the number contains command tokens instead of characters.

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@thepdfnumber.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.}%
```

```
      }%
    \else
      \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
    \fi
    \x{thepdfnumber}{The package is already loaded}%
    \aftergroup\endinput
  \fi
\fi
\endgroup%
```

Package identification:

```
\begingroup\catcode61\catcode48\catcode32=10\relax%
  \catcode13=5 % ^^M
  \endlinechar=13 %
  \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@thepdfnumber.sty\endcsname
\ProvidesPackage{thepdfnumber}%
  [2016/05/16 v1.1 Print PDF numbers with minimal digits (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 ThPdNu@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{%
  \edef\ThPdNu@AtEnd{%
    \ThPdNu@AtEnd
    \catcode#1=\the\catcode#1\relax
  }%
  \catcode#1=#2\relax
}
\TMP@EnsureCode{33}{12}% !
\TMP@EnsureCode{36}{3}% $
\TMP@EnsureCode{38}{4}% &
\TMP@EnsureCode{42}{12}% *
\TMP@EnsureCode{43}{12}% +
\TMP@EnsureCode{45}{12}% -
\TMP@EnsureCode{46}{12}% .
\TMP@EnsureCode{60}{12}% <
\TMP@EnsureCode{62}{12}% >
\TMP@EnsureCode{96}{12}% `
\edef\ThPdNu@AtEnd{\ThPdNu@AtEnd\noexpand\endinput}
```

2.3 Helper macros

\ThPdNu@FIN

```
\def\ThPdNu@FIN{\thepdfnumber@ErrorEndMarker}
```

\ThPdNu@space

```
\def\ThPdNu@space{ }
```

\ThPdNu@zero

```
\chardef\ThPdNu@zero=0 %
```

\ThPdNu@one

```
\chardef\ThPdNu@one=1 %
```

\ThPdNu@firstoftwo

```
\long\def\ThPdNu@firstoftwo#1#2{#1}
```

\ThPdNu@secondoftwo

```
\long\def\ThPdNu@secondoftwo#1#2{#2}
```

2.4 Detect ε-TEX

```
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname detokenize\endcsname\relax
  \catcode`\&=14 %
  \catcode`\$=9 %
\else
  \catcode`\&=9 %
  \catcode`\$=14 %
\fi
```

2.5 User macro \thepdfnumber

\thepdfnumber

```
\def\thepdfnumber#1{%
  \romannumeral
& \iftrue\expandafter\ThPdNu@State@Plus\expandafter\fi
& \detokenize\expandafter{#1}%
& \ThPdNu@FIN
$ \ifx\ThPdNu@FIN#1\ThPdNu@FIN
$   \expandafter\ThPdNu@firstoftwo
$ \else
$   \expandafter\ThPdNu@secondoftwo
$ \fi
$ {%
$   \ThPdNu@zero
$   0\thepdfnumber@ErrorUnexpectedEnd
$ }{%
$   \iftrue\expandafter\ThPdNu@State@Plus\expandafter\fi#1\ThPdNu@FIN
$ }%
}
```

2.5.1 State definitions for sign

\ThPdNu@State@Plus

```
\def\ThPdNu@State@Plus#1\fi#2{%
  \fi
  \ifcase\ifx\ThPdNu@FIN#2%
            0%
&         \else\ifx-#2%
$         \else\ifnum`#2=45 % -
            1%
          \else\ifx0#2%
            2%
          \else\ifnum`#2>48 %
            \ifnum`#2<58 %
              3%
            \else
              9%
            \fi
&         \else\ifx.#2%
$         \else\ifnum`#2=46 % .
            4%
&         \else\ifx+#2%
$         \else\ifnum`#2=43 % +
            5%
          \else
            9%
          \fi\fi\fi\fi\fi\fi\ThPdNu@space
```

```
    \expandafter\ThPdNu@zero
    \expandafter0%
    \expandafter\thepdfnumber@ErrorUnexpectedEnd
  \or
    \ThPdNu@State@Minus
  \or
    \ThPdNu@State@SkipZeros!%
  \or
    \ThPdNu@State@Int!#2!%
  \or
    \ThPdNu@State@Dot!\ThPdNu@zero*\ThPdNu@zero!!%
  \or
    \ThPdNu@State@Plus
  \else
    \ThPdNu@ReturnError{0}%
  \fi
}
```

\ThPdNu@State@Minus

```
\def\ThPdNu@State@Minus#1\fi#2{%
  \fi
  \ifcase\ifx\ThPdNu@FIN#2%
            0%
          \else\ifx0#2%
            1%
          \else\ifnum`#2>48 %
            \ifnum`#2<58 %
              2%
            \else
              9%
            \fi
&         \else\ifx.#2%
$         \else\ifnum`#2=46 % .
            3%
&         \else\ifx-#2%
$         \else\ifnum`#2=45 % -
            4%
&         \else\ifx+#2%
$         \else\ifnum`#2=43 % +
            5%
          \else
            9%
          \fi\fi\fi\fi\fi\fi\ThPdNu@space
    \expandafter\ThPdNu@zero
    \expandafter0%
    \expandafter\thepdfnumber@ErrorUnexpectedEnd
  \or
    \ThPdNu@State@SkipZeros-!%
  \or
    \ThPdNu@State@Int-!#2!%
  \or
    \ThPdNu@State@Dot-!\ThPdNu@zero*\ThPdNu@zero!!%
  \or
    \ThPdNu@State@Plus
  \or
    \ThPdNu@State@Minus
  \else
    \ThPdNu@ReturnError{0}%
```

```
  \fi
}
```

`\ThPdNu@ReturnError`

```
\def\ThPdNu@ReturnError#1#2\fi#3\ThPdNu@FIN{%
  \fi
  \ThPdNu@zero
  #1%
  \thepdfnumber@ErrorInvalidToken
}
```

2.5.2 State definitions for integer part

`\ThPdNu@State@SkipZeros`

```
\def\ThPdNu@State@SkipZeros#1!#2\fi#3{%
  \fi
  \ifcase\ifx\ThPdNu@FIN#3%
            0%
          \else\ifx0#3%
            1%
          \else\ifnum`#3>48 %
            \ifnum`#3<58 %
              2%
            \else
              9%
            \fi
&         \else\ifx.#3%
$         \else\ifnum`#3=46 % .
            3%
          \else
            9%
          \fi\fi\fi\fi\ThPdNu@space
    \expandafter\ThPdNu@zero
    \expandafter0%
  \or
   \ThPdNu@State@SkipZeros#1!%
  \or
    \ThPdNu@State@Int#1!#3!%
  \or
    \ThPdNu@State@Dot#1!\ThPdNu@zero*\ThPdNu@zero!!%
  \else
    \ThPdNu@ReturnError{0}%
  \fi
}
```

`\ThPdNu@State@Int`

```
\def\ThPdNu@State@Int#1!#2!#3\fi#4{%
  \fi
  \ifcase\ifx\ThPdNu@FIN#4%
            0%
          \else\ifnum`#4>47 %
            \ifnum`#4<58 %
              1%
            \else
              9%
            \fi
&         \else\ifx.#4%
$         \else\ifnum`#4=46 % .
```

```
            2%
          \else
            9%
          \fi\fi\fi\ThPdNu@space
      \ThPdNu@ReturnInt{#1#2}%
    \or
      \ThPdNu@State@Int#1!#2#4!%
    \or
      \ThPdNu@State@Dot#1!\ThPdNu@one#2*\ThPdNu@zero!!%
    \else
      \ThPdNu@ReturnError{#1#2}%
    \fi
}
```

`\ThPdNu@ReturnInt`

```
\def\ThPdNu@ReturnInt#1#2\fi{%
  \fi
  \ThPdNu@zero
  #1%
}
```

2.5.3 State definitions for decimal digits

`\ThPdNu@State@Dot`

```
\def\ThPdNu@State@Dot#1*#2#3!#4!#5\fi#6{%
  \fi
  \ifcase\ifx\ThPdNu@FIN#6%
            0%
          \else\ifnum`#6>48 %
            \ifnum`#6<58 %
              1%
            \else
              9%
            \fi
          \else\ifx0#6%
            2%
          \else
            9%
          \fi\fi\fi\ThPdNu@space
      \ThPdNu@ReturnNumber#1*#2#3!%
    \or
      \ThPdNu@State@Dot#1*\ThPdNu@one#3#4#6!!%
    \or
      \ThPdNu@State@DotZero#1*#2#3!#4#6!%
    \else
      \ThPdNu@ReturnNumberInvalid#1*#2#3!%
    \fi
}
```

`\ThPdNu@State@DotZero`

```
\def\ThPdNu@State@DotZero#1*#2#3!#4!#5\fi#6{%
  \fi
  \ifcase\ifx\ThPdNu@FIN#6%
            0%
          \else\ifnum`#6>48 %
            \ifnum`#6<58 %
              1%
            \else
```

```
              9%
            \fi
          \else\ifx0#6%
            2%
          \else
            9%
          \fi\fi\fi\ThPdNu@space
    \ThPdNu@ReturnNumber#1*#2#3!%
  \or
    \ThPdNu@State@Dot#1*\ThPdNu@one#3#4#6!!%
  \or
    \ThPdNu@State@DotZero#1*#2#3!#4#6!%
  \else
    \ThPdNu@ReturnNumber#1*#2#3!%
  \fi
}
```

`\ThPdNu@ReturnNumber`

```
\def\ThPdNu@ReturnNumber#1!#2#3*#4#5!#6\fi{%
  \fi
  \ifcase#2%
    \expandafter\ThPdNu@firstoftwo
  \else
    \expandafter\ThPdNu@secondoftwo
  \fi
  {%
    \ifcase#4%
      \expandafter\ThPdNu@firstoftwo
    \else
      \expandafter\ThPdNu@secondoftwo
    \fi
    {\ThPdNu@zero 0}%
    {\ThPdNu@zero #1.#5}%
  }{%
    \ifcase#4%
      \expandafter\ThPdNu@firstoftwo
    \else
      \expandafter\ThPdNu@secondoftwo
    \fi
    {\ThPdNu@zero #1#3}%
    {\ThPdNu@zero #1#3.#5}%
  }%
}
```

`\ThPdNu@ReturnNumberInvalid`

```
\def\ThPdNu@ReturnNumberInvalid#1*#2!#3\fi#4\ThPdNu@FIN{%
  \fi
  \iftrue\ThPdNu@ReturnNumber#1*#2!\fi
  \thepdfnumber@ErrorInvalidToken
}
```

2.6 Norm macro

`\thepdfnumberNormZeroOne`

```
\def\thepdfnumberNormZeroOne#1{%
  \romannumeral
& \iftrue\expandafter\ThPbNu@StateN@Plus\expandafter\fi
& \detokenize\expandafter{#1}%
```

```
& \ThPdNu@FIN
$ \ifx\ThPdNu@FIN#1\ThPdNu@FIN
$   \expandafter\ThPdNu@firstoftwo
$ \else
$   \expandafter\ThPdNu@secondoftwo
$ \fi
$ {%
$   \ThPdNu@zero
$   0\thepdfnumber@ErrorUnexpectedEnd
$ }{%
$   \iftrue\expandafter\ThPbNu@StateN@Plus\expandafter\fi#1\ThPdNu@FIN
$ }%
}
```

2.6.1 State definitions for sign

\ThPbNu@StateN@Plus

```
\def\ThPbNu@StateN@Plus#1\fi#2{%
  \fi
  \ifcase\ifx\ThPdNu@FIN#2%
            0%
&         \else\ifx-#2%
$         \else\ifnum`#2=45 % -
            1%
          \else\ifx0#2%
            2%
          \else\ifnum`#2>48 %
            \ifnum`#2<58 %
              3%
            \else
              9%
            \fi
&         \else\ifx.#2%
$         \else\ifnum`#2=46 % .
            4%
&         \else\ifx+#2%
$         \else\ifnum`#2=43 % +
            5%
          \else
            9%
          \fi\fi\fi\fi\fi\fi\ThPdNu@space
    \expandafter\ThPdNu@zero
    \expandafter0%
    \expandafter\thepdfnumber@ErrorUnexpectedEnd
  \or
    \ThPbNu@StateN@Minus
  \or
    \ThPbNu@StateN@SkipZeros
  \or
    \ThPdNu@ReturnAndSkip{1}%
  \or
    \ThPbNu@StateN@Dot\ThPdNu@zero!!%
  \or
    \ThPbNu@StateN@Plus
  \else
    \ThPdNu@ReturnError{0}%
  \fi
}
```

\ThPbNu@StateN@Minus

```
\def\ThPbNu@StateN@Minus#1\fi#2{%
  \fi
  \ifcase\ifx\ThPdNu@FIN#2%
            0%
          \else\ifnum`#2>47 %
            \ifnum`#2<58 %
              1%
            \else
              9%
            \fi
&         \else\ifx.#2%
$         \else\ifnum`#2=46 % .
            1%
&         \else\ifx-#2%
$         \else\ifnum`#2=45 % -
            2%
&         \else\ifx+#2%
$         \else\ifnum`#2=43 % +
            3%
          \else
            9%
          \fi\fi\fi\fi\fi\ThPdNu@space
    \expandafter\ThPdNu@zero
    \expandafter0%
    \expandafter\thepdfnumber@ErrorUnexpectedEnd
  \or
    \ThPdNu@ReturnAndSkip{0}%
  \or
    \ThPbNu@StateN@Plus
  \or
    \ThPbNu@StateN@Minus
  \else
    \ThPdNu@ReturnError{0}%
  \fi
}
```

\ThPdNu@ReturnAndSkip

```
\def\ThPdNu@ReturnAndSkip#1#2\fi#3\ThPdNu@FIN{%
  \fi
  \ThPdNu@zero
  #1%
}
```

2.6.2 State definitions for integer part

\ThPbNu@StateN@SkipZeros

```
\def\ThPbNu@StateN@SkipZeros#1\fi#2{%
  \fi
  \ifcase\ifx\ThPdNu@FIN#2%
            0%
          \else\ifx0#2%
            1%
          \else\ifnum`#2>48 %
            \ifnum`#2<58 %
              2%
            \else
              9%
```

```
478           \fi
479 &       \else\ifx.#2%
480 $       \else\ifnum`#2=46 % .
481           3%
482         \else
483           9%
484         \fi\fi\fi\fi\ThPdNu@space
485     \expandafter\ThPdNu@zero
486     \expandafter0%
487   \or
488    \ThPbNu@StateN@SkipZeros%
489   \or
490     \ThPdNu@ReturnAndSkip{1}%
491   \or
492     \ThPbNu@StateN@Dot\ThPdNu@zero!!%
493   \else
494     \ThPdNu@ReturnError{0}%
495   \fi
496 }
```

2.6.3 State definitions for decimal digits

\ThPbNu@StateN@Dot

```
497 \def\ThPbNu@StateN@Dot#1#2!#3!#4\fi#5{%
498   \fi
499   \ifcase\ifx\ThPdNu@FIN#5%
500           0%
501         \else\ifnum`#5>48 %
502           \ifnum`#5<58 %
503             1%
504           \else
505             9%
506           \fi
507         \else\ifx0#5%
508           2%
509         \else
510           9%
511         \fi\fi\fi\ThPdNu@space
512     \ThPdNu@ReturnFracNumber#1#2!%
513   \or
514     \ThPbNu@StateN@Dot\ThPdNu@one#2#3#5!!%
515   \or
516     \ThPbNu@StateN@DotZero#1#2!#3#5!%
517   \else
518     \ThPdNu@ReturnFracNumberInvalid#1#2!%
519   \fi
520 }
```

\ThPbNu@StateN@DotZero

```
521 \def\ThPbNu@StateN@DotZero#1#2!#3!#4\fi#5{%
522   \fi
523   \ifcase\ifx\ThPdNu@FIN#5%
524           0%
525         \else\ifnum`#5>48 %
526           \ifnum`#5<58 %
527             1%
528           \else
529             9%
```

```
            \fi
          \else\ifx0#5%
            2%
          \else
            9%
          \fi\fi\fi\ThPdNu@space
      \ThPdNu@ReturnFracNumber#1#2!%
    \or
      \ThPbNu@StateN@Dot\ThPdNu@one#2#3#5!!%
    \or
      \ThPbNu@StateN@DotZero#1#2!#3#5!%
    \else
      \ThPdNu@ReturnFracNumberInvalid#1#2!%
    \fi
}
```

\ThPdNu@ReturnFracNumber

```
\def\ThPdNu@ReturnFracNumber#1#2!#3\fi{%
  \fi
  \ifcase#1%
    \expandafter\ThPdNu@firstoftwo
  \else
    \expandafter\ThPdNu@secondoftwo
  \fi
  {\ThPdNu@zero 0}%
  {\ThPdNu@zero .#2}%
}
```

\ThPdNu@ReturnFracNumberInvalid

```
\def\ThPdNu@ReturnFracNumberInvalid#1!#2\fi#3\ThPdNu@FIN{%
  \fi
  \iftrue\ThPdNu@ReturnFracNumber#1!\fi
  \thepdfnumber@ErrorInvalidToken
}

\ThPdNu@AtEnd%
⟨/package⟩
```

3 Installation

3.1 Download

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

CTAN:macros/latex/contrib/oberdiek/thepdfnumber.dtx The source file.

CTAN:macros/latex/contrib/oberdiek/thepdfnumber.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:pkg/tds). Directories with `texmf` in their name are usually organized this way.

[1] CTAN:pkg/thepdfnumber

3.2 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
```

3.3 Package installation

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

```
tex thepdfnumber.dtx
```

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

```
thepdfnumber.sty → tex/generic/oberdiek/thepdfnumber.sty
thepdfnumber.pdf → doc/latex/oberdiek/thepdfnumber.pdf
thepdfnumber.dtx → source/latex/oberdiek/thepdfnumber.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.

3.4 Refresh file name databases

If your TEX distribution (TEX Live, MiKTEX, ...) relies on file name databases, you must refresh these. For example, TEX Live users run `texhash` or `mktexlsr`.

3.5 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{thepdfnumber.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 thepdfnumber.dtx
makeindex -s gind.ist thepdfnumber.idx
pdflatex thepdfnumber.dtx
makeindex -s gind.ist thepdfnumber.idx
pdflatex thepdfnumber.dtx
```

4 References

[1] Adobe Systems Incorporated. *Document management – Portable document format – Part 1: PDF 1.7*. 1st ed. 2008-07-01. URL: https://www.adobe.com/content/dam/acom/en/devnet/pdf/pdfs/PDF32000_2008.pdf (visited on 2011-11-25).

5 History

[2011/11/24 v1.0]

- First version.

[2016/05/16 v1.1]

- Documentation updates.

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.

W

X