The engord package

Heiko Oberdiek*

2016/05/16 v1.9

Abstract

The package generates the suffix of English ordinal numbers. It can be used with plain and LaTeX formats.

Contents

1 Usage **2**
1.1 Package options . . . 2
1.2 Examples . . . 2

2 Implementation **3**
2.1 Reload check and identification . . . 3
2.2 Help commands for plain compatibility . . . 4
2.3 User macros . . . 5
2.4 Suffix generation . . . 6

3 Installation **8**
3.1 Download . . . 8
3.2 Bundle installation . . . 8
3.3 Package installation . . . 8
3.4 Refresh file name databases . . . 9
3.5 Some details for the interested . . . 9

4 History **9**
[2000/05/23 v1.0] . . . 9
[2003/04/28 v1.1] . . . 9
[2006/02/20 v1.2] . . . 10
[2007/04/11 v1.3] . . . 10
[2007/04/26 v1.4] . . . 10
[2007/09/09 v1.5] . . . 10
[2007/09/20 v1.6] . . . 10
[2008/08/11 v1.7] . . . 10
[2010/03/01 v1.8] . . . 10
[2016/05/16 v1.9] . . . 10

5 Index **10**

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

1 Usage

`\engord{`⟨*LaTeX counter name*⟩`}`

It prints the value of the LaTeX counter as English ordinal number. It can be used in the same way as `\arabic`, `\roman`, or `\alph`. The command is not available in plain TeX.

`\engordnumber{`⟨*any TeX number*⟩`}`

It prints the number as English ordinal number.

`\engordletters{#1}`

This command formats the English ordinal letters after the number. It defaults to `\textsuperscript`.

`\engorderror{#1}`

It can be redefined, if an other error handling is wanted. The argument is a negative number or zero.

`\engordraisetrue`
`\engordraisefalse`

These commands set the switch `\ifengordraise` that is asked by the default `\engordletters` before raising the ordinal letters.

1.1 Package options

normal: `\engordraisefalse`

raise: `\engordraisetrue`

Default is raise.

1.2 Examples

- ```
  \usepackage[normal]{engord}
  \engordnumber{1}   → 1st
  \engordnumber{12}  → 12th
  \engordnumber{123} → 123rd
  \engord{page}      → 1st (if page has the value of one)
  \engordraisetrue
  \engordnumber{12}  → 12th
  ```

- The default output of a counter can be redefined:

  ```
  \newcounter{mycounter}
  \renewcommand{\theengcounter}{\engord{mycounter}}
  ```

- Because the implementation of `\engord` and `\engordnumber` is kept expandable, these commands can be used to make command names with an appropriate definition of `\engordletters`:
  ```

```
\renewcommand*{\engordletters}[1]{#1}
\@namedef{My\engordnumber{3}Command}{...}
```

This generates the command name '`\My4rdCommand`'. Since version 1.2 the redefinition can be dropped if the letters are not raised.

- If the letters should not be raised, use LaTeX package option normal or use

  ```
  \engordraisefalse
  ```

  Also `\engordletters` could be redefined for this purpose:

  ```
  \renewcommand*{\engordletters}[1]{#1}
  ```

2 Implementation

2.1 Reload check and identification

1 ⟨*package⟩

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@engord.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{engord}{The package is already loaded}%
29      \aftergroup\endinput
30    \fi
31  \fi
32 \endgroup%
```

Package identification:

```
33 \begingroup\catcode61\catcode48\catcode32=10\relax%
34   \catcode13=5 % ^^M
35   \endlinechar=13 %
36   \catcode35=6 % #
37   \catcode39=12 % '
```

```
  \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@engord.sty\endcsname
\ProvidesPackage{engord}%
  [2016/05/16 v1.9 Provides English ordinal numbers (HO)]%
```

2.2 Help commands for plain compatibility

```
\begingroup\catcode61\catcode48\catcode32=10\relax%
  \catcode13=5 % ^^M
  \endlinechar=13 %
  \catcode123=1 % {
  \catcode125=2 % }
  \catcode64=11 % @
  \def\x{\endgroup
    \expandafter\edef\csname EO@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\EO@AtEnd{%
    \EO@AtEnd
    \catcode#1=\the\catcode#1\relax
  }%
  \catcode#1=#2\relax
}
\TMP@EnsureCode{33}{12}% !
\TMP@EnsureCode{36}{3}% $
\TMP@EnsureCode{39}{12}% ’
\TMP@EnsureCode{42}{12}% *
\TMP@EnsureCode{46}{12}% .
\TMP@EnsureCode{47}{12}% /
\TMP@EnsureCode{60}{12}% <
\TMP@EnsureCode{91}{12}% [
\TMP@EnsureCode{93}{12}% ]
\TMP@EnsureCode{94}{7}% ^(superscript)
\TMP@EnsureCode{96}{12}% ‘
\edef\EO@AtEnd{\EO@AtEnd\noexpand\endinput}
```

`\EO@def` Definitions, `\newcommand` does not exist in plain TEX.

```
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname newcommand\endcsname\relax
  \def\EO@def{\def}%
\else
  \def\EO@def#1{%
    \newcommand*{#1}{}%
    \def#1%
  }%
\fi
```

```
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname RequirePackage\endcsname\relax
  \input infwarerr.sty\relax
  \input ltxcmds.sty\relax
\else
  \RequirePackage{infwarerr}[2007/09/09]%
  \RequirePackage{ltxcmds}[2016/05/16]%
\fi
```

2.3 User macros

`\ifengordraise` The switch `\ifengordraise`, whether the ordinal letters are raised or not. Default is raised because of compatibility.

```
\ltx@newif\ifengordraise
\engordraisetrue
```

In LATEX this also can be controlled by option normal or raise.

```
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname DeclareOption\endcsname\relax
\else
  \DeclareOption{normal}{\engordraisefalse}%
  \DeclareOption{raise}{\engordraisetrue}%
  \ProcessOptions*\relax
\fi
```

`\engordletters` `\engordletters` is called with one argument, the english ordinal letters, and contains the code to format them. It defaults to `\textsuperscript` depending on `\ifengordraise`.

```
\expandafter\ifx\csname engordletters\endcsname\relax
  \EO@def\engordletters{%
    \ifengordraise
      \expandafter\engordtextsuperscript
    \fi
  }%
\fi
```

`\engordtextsuperscript` For plain TEX the definition is quite ugly, redefine `\engordtextsuperscript` if you have a better one.

```
\expandafter\ifx\csname engordtextsuperscript\endcsname\relax
  \begingroup\expandafter\expandafter\expandafter\endgroup
  \expandafter\ifx\csname textsuperscript\endcsname\relax
    \def\engordtextsuperscript#1{%
      \relax
      \ifmmode
        ^{\rm#1}%
      \else
        $^{\rm#1}$%
      \fi
    }%
  \else
    \def\engordtextsuperscript{\textsuperscript}%
  \fi
\fi
```

`\engorderror` `\engorderror` is called, if the number is zero or negative.

```
\expandafter\ifx\csname engorderror\endcsname\relax
  \EO@def\engorderror#1{%
    #1\engordletters{!ERROR!}%
    \@PackageWarning{engord}{%
      `#1' is not an ordinal number%
    }%
  }%
\fi
```

`\engord` `\engord` expects a LATEX counter name as argument and calls `\engordnumber`. It is defined only, if LATEX is used.

```
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname newcounter\endcsname\relax
\else
  \EO@def\engord#1{%
    \engordnumber{\value{#1}}%
  }%
\fi
```

`\engordnumber` `\engordnumber` is the user command to print a number as english ordinal number. The argument can be any TEX number like explicit numbers, register values, ...

In a safe way it converts the TEX number argument into a form that only consists of decimal digits.

```
\EO@def\engordnumber#1{%
  \expandafter\EO@number\expandafter{\number#1}%
}
```

2.4 Suffix generation

`\EO@number` `\EO@number` expects a number with decimal digits as argument and looks at the size of the number and the count of the digits:

```
\def\EO@number#1{%
  \ifnum#1<1 % handle the error case
    \engorderror{#1}%
  \else
    \ifnum#1<21 %
      \EO@ord{#1}%
    \else
      \ifnum#1<100 %
        \EO@twodigits#1%
      \else
        \@ReturnAfterFi{%
          \EO@reverse#1\@nil{}\EO@afterreverse
        }%
      \fi
    \fi
  \fi
}
```

`\@ReturnAfterFi` An internal help macro to prevent a too deep `\if` nesting.

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

`\EO@ord` `\EO@ord` prints the number with ord letters.
`#1`: decimal digits, `#1` < 21

```
\def\EO@ord#1{%
  #1%
  \expandafter\engordletters
  \ifcase#1{th}\or
    {st}\or
    {nd}\or
    {rd}\else
    {th}%
  \fi
}
```

`\EO@twodigits` `\EO@twodigits` expects a number with two digits,
$20 <$ number < 100

```
\def\EO@twodigits#1#2{%
  #1\EO@ord{#2}%
}
```

`\EO@reverse` `\EO@reverse` reverses the digits of the number.
`#1`: next digit
`#2`: rest of the digits
`#3`: already reversed digits
`#4`: next command to call with the reversed number as argument

```
\def\EO@reverse#1#2\@nil#3#4{%
  \ifx\\#2\\%
    #4{#1#3}%
  \else
    \@ReturnAfterFi{%
      \EO@reverse#2\@nil{#1#3}{#4}%
    }%
  \fi
}
```

`\EO@afterreverse` `\EO@afterreverse` calls `\EO@reverseback` so that `\EO@reverseback` can inspect the digits of the number.

```
\def\EO@afterreverse#1{%
  \EO@reverseback#1\@nil
}
```

`\EO@reverseback` `\EO@reverseback` reverses the reversion.
#1: the last digit of the number
#2: the second last digit of the number
#3: first digits of the number in reversed order, it is not empty, because `\EO@reverseback` is only called with numbers > 100.

```
\def\EO@reverseback#1#2#3\@nil{%
  \EO@reverse#3\@nil{}\@firstofone
  \ifnum#2#1<21 %
    \EO@ord{#2#1}%
  \else
    #2\EO@ord{#1}%
  \fi
}

\EO@AtEnd%
```

⟨/package⟩

3 Installation

3.1 Download

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

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

CTAN:macros/latex/contrib/oberdiek/engord.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.

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 engord.dtx
```

[1] CTAN:pkg/engord

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

```
engord.sty → tex/generic/oberdiek/engord.sty
engord.pdf → doc/latex/oberdiek/engord.pdf
engord.dtx → source/latex/oberdiek/engord.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{engord.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 engord.dtx
makeindex -s gind.ist engord.idx
pdflatex engord.dtx
makeindex -s gind.ist engord.idx
pdflatex engord.dtx
```

4 History

[2000/05/23 v1.0]

- First public release, published in newsgroup de.comp.text.tex: "Re: Ordinalzahlen in LaTeX?"[2]

[2003/04/28 v1.1]

- Bug fix for 30, 40, 50, ..., 100, 130, ...
- `\ordletters` renamed to documented `\engordletters`.

[2] Url: https://groups.google.com/group/de.comp.text.tex/msg/738e2cb4c51759d6

[2006/02/20 v1.2]

- Support for plain TEX.
- Switch `\ifengordraise` added.
- Package options raise and normal added.
- DTX framework.

[2007/04/11 v1.3]

- Line ends sanitized.

[2007/04/26 v1.4]

- Use of package infwarerr.

[2007/09/09 v1.5]

- Catcode section added.

[2007/09/20 v1.6]

- Short description fixed (George White).

[2008/08/11 v1.7]

- Code is not changed.
- URLs updated.

[2010/03/01 v1.8]

- Compatibility with iniTEX.

[2016/05/16 v1.9]

- Documentation updates.

5 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

`\@PackageWarning` 164
`\@ReturnAfterFi` 189, 196, 214
`\@firstofone` 223
`\@nil` 190, 210, 215, 220, 222, 223
`\@undefined` 58
`\\` 211

A

`\aftergroup` 29

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, 97, 99
`\csname` 14, 21, 50, 66, 76, 114, 123, 133, 139, 146, 148, 161, 170

D

`\DeclareOption` 135, 136

E

\empty 17, 18
\endcsname . 14, 21, 50, 66, 76, 114, 123, 133, 139, 146, 148, 161, 170
\endinput 29, 112
\endlinechar 4, 35, 71, 77, 89
\engord *2*, 169
\engorderror *2*, 161, 181
\engordletters *2*, 139, 163, 199
\engordnumber *2*, 173, 176
\engordraisefalse *2*, 135
\engordraisetrue *2*, 131, 136
\engordtextsuperscript 142, 146
\EO@afterreverse 190, 219
\EO@AtEnd 95, 96, 112, 230
\EO@def 113, 140, 162, 172, 176
\EO@number 177, 179
\EO@ord 184, 197, 208, 225, 227
\EO@reverse 190, 210, 223
\EO@reverseback 220, 222
\EO@twodigits 187, 207

I

\ifcase 200
\ifengordraise 130, 141
\ifmmode 151
\ifnum 180, 183, 186, 224
\ifx . 15, 18, 21, 50, 58, 61, 114, 123, 133, 139, 146, 148, 161, 170, 211
\immediate 23, 52
\input 124, 125

L

\ltx@newif 130

N

\newcommand 118
\number 177

P

\PackageInfo 26
\ProcessOptions 137
\ProvidesPackage 19, 67

R

\RequirePackage 127, 128
\rm 152, 154

T

\textsuperscript 158
\the . . . 77, 78, 79, 80, 81, 82, 83, 84, 97
\TMP@EnsureCode 94, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111

V

\value 173

W

\write 23, 52

X

\x 14, 15, 18, 22, 26, 28, 51, 56, 66, 75, 87