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 ............................................. 1
   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 ................................. 7
   3.1 Download ................................... 7
   3.2 Bundle installation ......................... 8
   3.3 Package installation ......................... 8
   3.4 Refresh file name databases .................. 8
   3.5 Some details for the interested .............. 8

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

5 Index ............................................ 9

1 Usage

\texttt{\textbackslash 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 \texttt{\arabic{}}, \texttt{\roman{}}, or \texttt{\alph{}}. The command is not available in plain \TeX{}.

\texttt{\engordnumber\{⟨any \TeX\ number⟩\}}

It prints the number as English ordinal number.

\texttt{\engordletters\{#1\}}

This command formats the English ordinal letters after the number. It defaults to \texttt{textsuperscript}.

\texttt{\engorderror\{#1\}}

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

\texttt{\engordraisetrue}
\texttt{\engordraisefalse}

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

1.1 Package options

\texttt{normal}: \texttt{\engordraisefalse}

\texttt{raise}: \texttt{\engordraisetrue}

Default is raise.

1.2 Examples

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

- The default output of a counter can be redefined:
  \texttt{\newcounter\{mycounter\}}
  \texttt{\renewcommand\{\theengcounter\}{\engord\{mycounter\}}}  

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

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

  This generates the command name ‘\texttt{\My4rdCommand}’. Since version 1.2 the redefinition can be dropped if the letters are not raised.

- If the letters should not be raised, use \texttt{\LaTeX} package option \texttt{normal} or use \texttt{\engordraisefalse}.

  Also \texttt{\engordletters} could be redefined for this purpose:

  \texttt{\renewcommand\{\{\engordletters\}[1]\{#1\}}
2 Implementation

2.1 Reload check and identification

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

\begin{verbatim}
\catcode13=5 \catcode48=10 \relax\% reload check \end{verbatim}

\catcode4=13 \% \catcode5=6 \% \catcode9=12 \% 
\catcode10=12 \% 
\catcode12=12 \% 
\catcode15=12 \% 
\expandafter\let\expandafter\x\csname ver@engord.sty\endcsname
\ifx\x\relax \else \def\empty{}\fi
\ifx\x\empty \def\PackageInfo\relax\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\fi\x{engord}{The package is already loaded}\aftergroup\endinput

\endgroup

\begin{verbatim}
\catcode61\catcode48\catcode32=10 \relax\%
\catcode13=5 \catcode48\catcode32=10 \relax\%
\catcode4=13 \% \catcode5=6 \% \catcode9=12 \%
\catcode10=12 \%
\catcode12=12 \%
\catcode15=12 \%
\expandafter\let\expandafter\x\csname ver@engord.sty\endcsname
\ifx\x\relax \else \def\empty{}\fi
\ifx\x\empty \def\PackageInfo\relax\else \expandafter\ifx\csname ProvidesPackage\endcsname\relax \def\x#1#2#3[#4]{\endgroup\immediate\write-1{Package: #3 #4}}\else \def\x#1#2[#3]{\endgroup\immediate\write-1{\PackageInfo{#1}{#2, stopped}}}\fi\edef#1{#4}\fi\x{engord}{The package is already loaded}\aftergroup\endinput
\endgroup

Package identification:

\begin{verbatim}
\catcode13=5 \catcode48\catcode32=10 \relax\%
\catcode4=13 \% \catcode5=6 \% \catcode9=12 \%
\catcode10=12 \%
\catcode12=12 \%
\catcode15=12 \%
\expandafter\let\expandafter\x\csname ver@engord.sty\endcsname
\ifx\x\relax \else \def\empty{}\fi
\ifx\x\empty \def\PackageInfo\relax\else \expandafter\ifx\csname ProvidesPackage\endcsname\relax \def\x#1#2#3[#4]{\endgroup\immediate\write-1{Package: #3 #4}}\else \def\x#1#2[#3]{\endgroup\immediate\write-1{\PackageInfo{#1}{#2, stopped}}}\fi\edef#1{#4}\fi\x{engord}{The package is already loaded}\aftergroup\endinput
\endgroup

\end{verbatim}
\begin{itemize}
\item \texttt{\#2[\#3]}% \\texttt{\expandafter\expandafter\expandafter\edef\csname EO\@AtEnd\endcsname{\endlinechar=\the\endlinechar\relax}}% \\texttt{\catcode\#1=\the\catcode\#1\relax}}% \\texttt{\catcode\#1=#2\relax}}% \\texttt{\edef\EO@AtEnd{\EO@AtEnd\noexpand\endinput}}% \\texttt{\EO@def\Definition, \newcommand does not exist in plain \TeX.}}
\item \texttt{\expandafter\expandafter\expandafter\edef\csname EO\@AtEnd\endcsname{\endlinechar=\the\endlinechar\relax}}% \\texttt{\catcode\#1=\the\catcode\#1\relax}}% \\texttt{\catcode\#1=#2\relax}}% \\texttt{\edef\EO@AtEnd{\EO@AtEnd\noexpand\endinput}}% \\texttt{\EO@def\Definition, \newcommand does not exist in plain \TeX.}}
\end{itemize}
2.3 User macros

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

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

\endgroup

\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.

\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname engordletters\endcsname\relax
\EO@def\engordletters{\
\ifengordraise
\expandafter\engordtextsuperscript
\fi
}\
\fi

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

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

\textsuperscript
\engorderror \engorderror is called, if the number is zero or negative.
\expandafter\ifx\csname engorderror\endcsname\relax
\EO@def\engorderror#1{\texttt{!ERROR!}}%
\PackageWarning{engord}{
'\texttt{#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{\EO@number{\number#1}}%
\EO@ord \EO@ord prints the number with ord letters.
\ifnum#1<21 \engorderror{#1}\
\else
\EO@ord{#1}\
\fi
\EO@afterord An internal help macro to prevent a too deep \if nesting.
\EO@ord\EO@ord
\EO@ord prints the number with ord letters.
#1: decimal digits, #1 < 21
\EO@ord\EO@ord
\EO@ord
\EO@ord \EO@ord 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{\EO@number{\number#1}}%
\EO@ord \EO@ord 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@twodigits  \EO@twodigits expects a number with two digits, \(20 < \text{number} < 100\).

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

\EO@reverse  \EO@reverse reverses the digits of the number.
\begin{itemize}
  \item \#1: next digit
  \item \#2: rest of the digits
  \item \#3: already reversed digits
  \item \#4: next command to call with the reversed number as argument
\end{itemize}
\def\EO@reverse#1#2\@nil#3#4{\ifx\#2\% \#4{#1#3}\% \else \@ReturnAfterFi{\% \EO@reverse#2\@nil{#1#3}{#4} \} \% \fi \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.
\begin{itemize}
  \item \#1: the last digit of the number
  \item \#2: the second last digit of the number
  \item \#3: first digits of the number in reversed order, it is not empty, because \EO@reverseback is only called with numbers > 100.
\end{itemize}
\def\EO@reverseback#1#2#3\@nil{\EO@reverse#3\@nil\@firstofone \ifnum#2#1<21 \% \EO@ord{#2#1}\% \else \@ReturnAfterFi{\% \EO@reverse#2\@nil{#1#3}{#4} \} \% \fi \fi}

3 Installation

3.1 Download

Package. This package is available on CTAN:\(^1\)

\begin{itemize}
  \item CTAN:macros/latex/contrib/oberdiek/engord.dtx The source file.
  \item CTAN:macros/latex/contrib/oberdiek/engord.pdf Documentation.
\end{itemize}

\(^1\)CTAN:pkg/engord
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

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 \TeX. The .dtx chooses its action depending on the format:

plain TeX: Run docstrip and extract the files.

\L\A\TeX: Generate the documentation.

If you insist on using \L\A\TeX for docstrip (really, docstrip does not need \L\A\TeX), 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?”

[2003/04/28 v1.1]
• Bug fix for 30, 40, 50, …, 100, 130, …
• \ordletters renamed to documented \engordletters.

[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.