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} \rightarrow 1st
  \engordnumber{12} \rightarrow 12th
  \engordnumber{123} \rightarrow 123rd
  \engord{page} \rightarrow 1st (if page has the value of one)
  \engordraisetrue
  \engordnumber{12} \rightarrow 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}
\nThis 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
  \begin{verbatim}
  \engordraisefalse
  \end{verbatim}

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

\begin{verbatim}
\renewcommand*{\engordletters}[1]{#1}
\end{verbatim}

\section{Implementation}

\subsection{Reload check and identification}

\begin{verbatim}
\begin{group}
\catcode61\catcode48\catcode32=10\relax
\endgroup
\catcode13=5 % ^^M
\endlinechar=13 %
\catcode35=6 % #
\catcode39=12 % '
\catcode44=12 % ,
\catcode45=12 % -
\catcode46=12 % .
\catcode58=12 % :
\catcode64=11 % @
\catcode123=1 % {
\catcode125=2 % }
\expandafter\let\expandafter\x\csname ver@engord.sty\endcsname
\ifx\x\relax % plain-TeX, first loading
\else
\def\empty{}%
\ifx\x\empty % LaTeX, first loading, % variable is initialized, but \ProvidesPackage not yet seen
\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
\fi
\end{verbatim}
\begin{verbatim}
\begin{group}
\catcode61\catcode48\catcode32=10\relax
\endgroup
\catcode13=5 % ^^M
\endlinechar=13 %
\catcode35=6 % #
\catcode39=12 % '
\catcode44=12 % ,
\catcode45=12 % -
\catcode46=12 % .
\catcode58=12 % :
\catcode64=11 % @
\catcode123=1 % {
\catcode125=2 % }
\expandafter\let\expandafter\x\csname ver@engord.sty\endcsname
\ifx\x\relax % plain-TeX, first loading
\else
\def\empty{}%
\ifx\x\empty % LaTeX, first loading, % variable is initialized, but \ProvidesPackage not yet seen
\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
\fi
\end{verbatim}
\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\catcode65=10\relax
\catcode13=5 % ^^M
\endlinechar=13 %
\catcode123=1 % {
\catcode125=2 % }
\def\x{\endgroup
\expandafter\edef\csname EO@AtEnd\endcsname{\endlinechar=\the\endlinechar\relax
\catcode13=\the\catcode13\relax
\catcode61=\the\catcode61\relax
\catcode32=\the\catcode32\relax
\catcode48=\the\catcode48\relax
\catcode13=\the\catcode13\relax
\catcode65=\the\catcode65\relax
\catcode123=\the\catcode123\relax
\catcode64=\the\catcode64\relax
\catcode125=\the\catcode125\relax
}}%}
\x\catcode61=10\catcode48=10\relax
\catcode13=5 % ^^M
\endlinechar=13 %
\catcode32=6 % #
\catcode48=11 % @
\catcode123=1 % {
\catcode125=2 % }
\def\TMP@EnsureCode#1#2{\% 
\edef\EO@AtEnd{\% 
\catcode#1=\the\catcode#1\relax 
\catcode#1=#2\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.
\begin{group}\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname newcommand\endcsname\relax
\else
\\newcommand\def\EO@def{\def}\% 
\\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

\textbf{2.3 User macros}

\engordraise \textbf{The switch} \engordraise, \textbf{whether the ordinal letters are raised or not}. Default is raised because of compatibility.
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname DeclareOption\endcsname\relax
\else
\DeclareOption{normal}{\engordraisefalse}% 
\DeclareOption{raise}{\engordraisetrue}%
\ProcessOptions\relax
\fi
\begingroup\expandafter\expandafter\expandafter\endgroup
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname DeclareOption\endcsname\relax
\else
\DeclareOption{normal}{\engordraisefalse}% 
\DeclareOption{raise}{\engordraisetrue}%
\ProcessOptions\relax
\fi
\begingroup\expandafter\expandafter\expandafter\endgroup
\begingroup\expandafter\expandafter\expandafter\endgroup
\begingroup\expandafter\expandafter\expandafter\endgroup
\begin{group}\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname DeclareOption\endcsname\relax
\else
\DeclareOption{normal}{\engordraisefalse}% 
\DeclareOption{raise}{\engordraisetrue}%
\ProcessOptions\relax
\fi
\begingroup\expandafter\expandafter\expandafter\endgroup
\begingroup\expandafter\expandafter\expandafter\endgroup
\begingroup\expandafter\expandafter\expandafter\endgroup
\EngordLetters \textbf{\EngordLetters} is called with one argument, the english ordinal letters, and contains the code to format them. It defaults to \textsuperscript depending on \engordraise.
For plain \TeX\ the definition is quite ugly, redefine \texttt{\engordtextsuperscript} if you have a better one.

\begin{verbatim}
\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
\fi
\end{verbatim}

\texttt{\engorderror} is called, if the number is zero or negative.

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

\texttt{\engord} expects a \LaTeX\ counter name as argument and calls \texttt{\engordnumber}. It is defined only, if \LaTeX\ is used.

\begin{verbatim}
\begingroup\expandafter\expandafter\expandafter\endgroup
\EO@def\engord#1{\engordnumber{\value{#1}}}\fi
\end{verbatim}

\texttt{\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.

\begin{verbatim}
\EO@def\engordnumber#1{}\EO@number\expandafter{\number#1}\
\end{verbatim}

\section{Suffix generation}

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

\begin{verbatim}
\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:
\end{verbatim}
\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\fi
}\@ReturnAfterFi

\@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\fi}

\EO@twodigits \EO@twodigits expects a number with two digits, \#2 < 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.
\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.

\EO@AtEnd%
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?"

[2003/04/28 v1.1]

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

---

\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.

<table>
<thead>
<tr>
<th>Symbols</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>@PackageWarning</td>
<td>164</td>
</tr>
<tr>
<td>@ReturnAfterFi</td>
<td>189, 196, 214</td>
</tr>
<tr>
<td>@firstofone</td>
<td>223</td>
</tr>
<tr>
<td>@nil</td>
<td>190, 210, 215, 220, 222, 223</td>
</tr>
<tr>
<td>@undefined</td>
<td>58</td>
</tr>
<tr>
<td>\</td>
<td>211</td>
</tr>
<tr>
<td>\aftergroup</td>
<td>29</td>
</tr>
<tr>
<td>\DeclareOption</td>
<td>135, 136</td>
</tr>
</tbody>
</table>