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

\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{\textbackslash engordnumber\{(any \TeX\ number)\} }

It prints the number as English ordinal number.

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

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

\texttt{\textbackslash engordererror\{#1\} }

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

\texttt{\textbackslash engordraisetrue \textbackslash 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{\textbackslash usepackage[\texttt{normal}]{engord} }

\texttt{\textbackslash engordnumber\{1\} \rightarrow 1st} \\
\texttt{\textbackslash engordnumber\{12\} \rightarrow 12th} \\
\texttt{\textbackslash engordnumber\{123\} \rightarrow 123rd} \\
\texttt{\textbackslash engord\{page\} \rightarrow 1st (if page has the value of one)} \\
\texttt{\engordraisetrue} \\
\texttt{\textbackslash engordraisefalse} \\
\texttt{\textbackslash engordnumber\{12\} \rightarrow 12th} \\

Default is \texttt{raise}.

1.2 Examples

• \texttt{\usepackage[\texttt{normal}]{engord} }

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

• 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}: 

2
\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

Reload check, especially if the package is not used with \LaTeX.
\begingroup\catcode61\catcode48\catcode32=10\relax%
\catcode13=5 % ^^M
\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\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
\fi
\endgroup%

Package identification:
\begingroup\catcode61\catcode48\catcode32=10\relax%
\catcode13=5 % ^^M
\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\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
\fi
\endgroup%
\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\if\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 \% 
\catcode13=\the\catcode13 \% 
\catcode32=\the\catcode32 \relax 
\catcode35=\the\catcode35 \% 
\catcode61=\the\catcode61 \relax 
\catcode123=\the\catcode123 \relax 
\catcode64=\the\catcode64 \relax 
\catcode125=\the\catcode125 \relax 
\} \% 
\} \% 
\x\catcode61\catcode48\catcode32=10\relax \% 
\catcode13=5 \% ^^M 
\endlinechar=13 \% 
\catcode35=6 \% # 
\catcode64=11 \% @ 
\catcode123=1 \% { 
\catcode125=2 \% }

4
94 \def\TMP@EnsureCode#1#2{
95 \edef\EO@AtEnd{
96 \catcode#1=`the\catcode#1\relax
97 }%  
98 }%
99 \catcode#1=#2\relax
100 }
101 \TMP@EnsureCode{33}{12}% !
102 \TMP@EnsureCode{36}{3}% $
103 \TMP@EnsureCode{39}{12}% ',
104 \TMP@EnsureCode{42}{12}% *
105 \TMP@EnsureCode{46}{12}% .
106 \TMP@EnsureCode{47}{12}% /
107 \TMP@EnsureCode{60}{12}% <
108 \TMP@EnsureCode{91}{12}% [.
109 \TMP@EnsureCode{93}{12}% ]
110 \TMP@EnsureCode{94}{7}% ^(superscript)
111 \TMP@EnsureCode{96}{12}% '
112 \edef\EO@AtEnd\EO@def
113 \EO@def
114 \begingroup\expandafter\expandafter\expandafter\endgroup
115 \expandafter\ifx\csname newcommand\endcsname\relax
116 \def\EO@def\{\def}%
117 \else
118 \def\EO@def#1{%
119 \newcommand*{#1}{}%
120 }%
121 \fi
122 \begingroup\expandafter\expandafter\expandafter\endgroup
123 \expandafter\ifx\csname RequirePackage\endcsname\relax
124 \input infwarerr.sty\relax
125 \input ltxcmds.sty\relax
126 \else
127 \RequirePackage{infwarerr}[2007/09/09]%
128 \RequirePackage{ltxcmds}[2016/05/16]%
129 \fi

\EO@def Definitions, \newcommand does not exist in plain \TeX.
130 \begingroup\expandafter\expandafter\expandafter\endgroup
131 \expandafter\ifx\csname DeclareOption\endcsname\relax
132 \else
133 \DeclareOption{normal}{\engordraisefalse}%
134 \DeclareOption{raise}{\engordraisetrue}%
135 \ProcessOptions*\relax
136 \fi

2.3 User macros

\ifengordraise The switch \ifengordraise, whether the ordinal letters are raised or not. Default
is raised because of compatibility.
130 \ltx@newif\ifengordraise
131 \engordraisetrue
132 \begingroup\expandafter\expandafter\expandafter\endgroup
133 \expandafter\ifx\csname DeclareOption\endcsname\relax
134 \else
135 \DeclareOption{normal}{\engordraisefalse}%
136 \DeclareOption{raise}{\engordraisetrue}%
137 \ProcessOptions*\relax
138 \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\ifc@nameengordletters\endc@name\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\ifc@nameengordtextsuperscript\endc@name\relax
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifc@nametextsuperscript\endc@name\relax
\def\engordtextsuperscript#1{%
  \relax
  \ifmmode
    ^{\text{#1}}\%
  \else
    $^{\text{#1}}$\%
  \fi
}\else
  \def\engordtextsuperscript{\textsuperscript}\i
\endgroup
\engord\engord is called, if the number is zero or negative.
\expandafter\ifc@nameengorder\endc@name\relax
\EO@def\engorder#1{%
  \ifmmode
    {\text{#1}}\%
  \else
    '{\text{#1}}$\%$
  \fi
}\fi
\engorder\engorder expects a \LaTeX counter name as argument and calls \engordnumber. It is defined only, if \LaTeX is used.
\begingroup\expandafter\expandafter\expandafter\endgroup
\EO@def\engord#1{%
  \engordnumber{\value{#1}}\%\fi
}\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:

\EO@number
\EO@number
\EO@number

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{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,
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.
\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.

3 Installation

3.1 Download

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


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 docu-
mation 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 pd\LaTeX:

```
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.

---

2Url: [https://groups.google.com/group/de.comp.text.tex/msg/738e2cb4c51759d6](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>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>\PackageWarning</td>
<td>164</td>
<td>\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</td>
</tr>
<tr>
<td>\ReturnAfterFi</td>
<td>189, 196, 214</td>
<td></td>
</tr>
<tr>
<td>\firstofone</td>
<td>223</td>
<td></td>
</tr>
<tr>
<td>\nil</td>
<td>190, 210, 215, 220, 222, 223</td>
<td>70, 72, 73, 74, 78, 79, 80, 81, 82, 83, 84, 87, 88, 90, 91, 92, 93, 97, 99</td>
</tr>
<tr>
<td>\undefined</td>
<td>58</td>
<td>\csname 14, 21, 50, 66, 76, 114, 123, 133, 139, 146, 148, 161, 170</td>
</tr>
<tr>
<td>\</td>
<td>211</td>
<td></td>
</tr>
<tr>
<td>A\aftergroup\DeclareOption</td>
<td>29</td>
<td>135, 136</td>
</tr>
</tbody>
</table>