

# The `engord` package

Heiko Oberdiek  
<oberdiek@uni-freiburg.de>

2007/09/20 v1.6

## Abstract

The package generates the suffix of English ordinal numbers. It can be used with plain and L<sup>A</sup>T<sub>E</sub>X formats.

## Contents

<b>1</b>	<b>Usage</b>	<b>1</b>
1.1	Package options . . . . .	2
1.2	Examples . . . . .	2
<b>2</b>	<b>Implementation</b>	<b>3</b>
2.1	Reload check and identification . . . . .	3
2.2	Help commands for plain compatibility . . . . .	3
2.3	User macros . . . . .	4
2.4	Suffix generation . . . . .	5
<b>3</b>	<b>Test</b>	<b>7</b>
3.1	Catcode checks for loading . . . . .	7
<b>4</b>	<b>Installation</b>	<b>8</b>
4.1	Download . . . . .	8
4.2	Bundle installation . . . . .	8
4.3	Package installation . . . . .	8
4.4	Refresh file name databases . . . . .	8
4.5	Some details for the interested . . . . .	9
<b>5</b>	<b>History</b>	<b>9</b>
	[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] . . . . .	10
	[2007/09/09 v1.5] . . . . .	10
	[2007/09/20 v1.6] . . . . .	10
<b>6</b>	<b>Index</b>	<b>10</b>

## 1 Usage

<code>\engord{&lt;L<sup>A</sup>T<sub>E</sub>X counter name&gt;}</code>
--

It prints the value of the L<sup>A</sup>T<sub>E</sub>X 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-T<sub>E</sub>X.

`\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}` → 12<sup>th</sup>

- 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 L<sup>A</sup>T<sub>E</sub>X 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 L<sup>A</sup>T<sub>E</sub>X.

```
2 \begingroup
3   \catcode44 12 % ,
4   \catcode45 12 % -
5   \catcode46 12 % .
6   \catcode58 12 % :
7   \catcode64 11 % @
8   \expandafter\let\expandafter\x\csname ver@engord.sty\endcsname
9   \ifcase 0%
10    \ifx\x\relax % plain
11    \else
12      \ifx\x\empty % LaTeX
13      \else
14        1%
15      \fi
16    \fi
17  \else
18    \expandafter\ifx\csname PackageInfo\endcsname\relax
19      \def\x#1#2{%
20        \immediate\write-1{Package #1 Info: #2.}%
21      }%
22    \else
23      \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
24    \fi
25    \x{engord}{The package is already loaded}%
26  \endgroup
27  \expandafter\endinput
28  \fi
29 \endgroup
```

Package identification:

```
30 \begingroup
31   \catcode40 12 % (
32   \catcode41 12 % )
33   \catcode44 12 % ,
34   \catcode45 12 % -
35   \catcode46 12 % .
36   \catcode47 12 % /
37   \catcode58 12 % :
38   \catcode64 11 % @
39   \expandafter\ifx\csname ProvidesPackage\endcsname\relax
40     \def\x#1#2#3[#4]{\endgroup
41       \immediate\write-1{Package: #3 #4}%
42       \xdef#1{#4}%
43     }%
44   \else
45     \def\x#1#2[#3]{\endgroup
46       #2[{#3}]%
47       \ifx#1\relax
48         \xdef#1{#3}%
49       \fi
50     }%
51   \fi
52   \expandafter\x\csname ver@engord.sty\endcsname
53   \ProvidesPackage{engord}%
54   [2007/09/20 v1.6 Provides English ordinal numbers (H0)]
```

### 2.2 Help commands for plain compatibility

```

55 \expandafter\edef\csname EO@AtEnd\endcsname{%
56   \catcode64 \the\catcode64\relax
57 }
58 \catcode64 11 % @
59 \def\TMP@EnsureCode#1#2{%
60   \edef\EO@AtEnd{%
61     \EO@AtEnd
62     \catcode#1 \the\catcode#1\relax
63   }%
64   \catcode#1 #2\relax
65 }
66 \TMP@EnsureCode{33}{12}% !
67 \TMP@EnsureCode{36}{3}% $
68 \TMP@EnsureCode{39}{12}% '
69 \TMP@EnsureCode{42}{12}% *
70 \TMP@EnsureCode{46}{12}% .
71 \TMP@EnsureCode{47}{12}% /
72 \TMP@EnsureCode{60}{12}% <
73 \TMP@EnsureCode{94}{7}% ^ (superscript)
74 \TMP@EnsureCode{96}{12}% '

```

\EO@def Definitions, \newcommand does not exist in plain-TeX.

```

75 \begingroup\expandafter\expandafter\expandafter\endgroup
76 \expandafter\ifx\csname newcommand\endcsname\relax
77   \def\EO@def{\def}%
78 \else
79   \def\EO@def#1{%
80     \newcommand*{#1}{}%
81     \def#1%
82   }%
83 \fi

84 \begingroup\expandafter\expandafter\expandafter\endgroup
85 \expandafter\ifx\csname RequirePackage\endcsname\relax
86   \input infwarerr.sty\relax
87 \else
88   \RequirePackage{infwarerr}[2007/09/09]%
89 \fi

```

## 2.3 User macros

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

```

90 \newif\ifengordraise
91 \engordraisetrue

```

In L<sup>A</sup>T<sub>E</sub>X this also can be controlled by option normal or raise.

```

92 \begingroup\expandafter\expandafter\expandafter\endgroup
93 \expandafter\ifx\csname DeclareOption\endcsname\relax
94 \else
95   \DeclareOption{normal}{\engordraisefalse}%
96   \DeclareOption{raise}{\engordraisetrue}%
97   \ProcessOptions*\relax
98 \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.

```

99 \expandafter\ifx\csname engordletters\endcsname\relax
100 \EO@def\engordletters{%
101   \ifengordraise
102     \expandafter\engordtextsuperscript

```

```

103   \fi
104 }%
105 \fi

```

`\engordtextsuperscript` For plain- $\text{\TeX}$  the definition is quite ugly, redefine `\engordtextsuperscript` if you have a better one.

```

106 \expandafter\ifx\csname engordtextsuperscript\endcsname\relax
107   \begingroup\expandafter\expandafter\expandafter\endgroup
108   \expandafter\ifx\csname textsuperscript\endcsname\relax
109     \def\engordtextsuperscript#1{%
110       \relax
111       \ifmode
112         ^{\rm#1}%
113       \else
114         $\sim{\rm#1}$%
115       \fi
116     }%
117   \else
118     \def\engordtextsuperscript{\textsuperscript}%
119   \fi
120 \fi

```

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

```

121 \expandafter\ifx\csname engorderror\endcsname\relax
122   \EO@def\engorderror#1{%
123     #1\engordletters{!ERROR!}%
124     \@PackageWarning{engord}{%
125       ‘#1’ is not an ordinal number%
126     }%
127   }%
128 \fi

```

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

```

129 \begingroup\expandafter\expandafter\expandafter\endgroup
130 \expandafter\ifx\csname newcounter\endcsname\relax
131 \else
132   \EO@def\engord#1{%
133     \engordnumber{\value{#1}}%
134   }%
135 \fi

```

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

In a safe way it converts the  $\text{\TeX}$  number argument into a form that only consists of decimal digits.

```

136 \EO@def\engordnumber#1{%
137   \expandafter\EO@number\expandafter{\number#1}%
138 }

```

## 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:

```

139 \def\EO@number#1{%
140   \ifnum#1<1 % handle the error case
141     \engorderror{#1}%
142   \else
143     \ifnum#1<21 %
144       \EO@ord{#1}%
145     \else

```

```

146     \ifnum#1<100 %
147         \EO@twodigits#1%
148     \else
149         \@ReturnAfterFi{%
150             \EO@reverse#1\@nil{}\EO@afterreverse
151         }%
152     \fi
153 \fi
154 \fi
155 }

```

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

```

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

```

**\EO@ord** \EO@ord prints the number with ord letters.

#1: decimal digits, #1 < 21

```

157 \def\EO@ord#1{%
158     #1%
159     \expandafter\engordletters
160     \ifcase#1{th}\or
161         {st}\or
162         {nd}\or
163         {rd}\else
164         {th}%
165     \fi
166 }

```

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

```

167 \def\EO@twodigits#1#2{%
168     #1\EO@ord{#2}%
169 }

```

**\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

```

170 \def\EO@reverse#1#2\@nil#3#4{%
171     \ifx\#2\%
172         #4{#1#3}%
173     \else
174         \@ReturnAfterFi{%
175             \EO@reverse#2\@nil{#1#3}{#4}%
176         }%
177     \fi
178 }

```

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

```

179 \def\EO@afterreverse#1{%
180     \EO@reverseback#1\@nil
181 }

```

**\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.

```

182 \def\EO@reverseback#1#2#3\@nil{%
183     \EO@reverse#3\@nil{}\@firstofone

```

```

184 \ifnum#2#1<21 %
185   \E0@ord{#2#1}%
186 \else
187   #2\E0@ord{#1}%
188 \fi
189 }

190 \E0@AtEnd
191 \end{package}

```

## 3 Test

### 3.1 Catcode checks for loading

```

192 \test1
193 \catcode'\@=11 %
194 \def\RestoreCatcodes{}
195 \count@=0 %
196 \loop
197   \edef\RestoreCatcodes{%
198     \RestoreCatcodes
199     \catcode\the\count@=\the\catcode\count@\relax
200   }%
201 \ifnum\count@<255 %
202   \advance\count@\@ne
203 \repeat
204
205 \def\RangeCatcodeInvalid#1#2{%
206   \count@=#1\relax
207   \loop
208     \catcode\count@=15 %
209     \ifnum\count@<#2\relax
210       \advance\count@\@ne
211     \repeat
212 }
213 \def\Test{%
214   \RangeCatcodeInvalid{0}{47}%
215   \RangeCatcodeInvalid{58}{64}%
216   \RangeCatcodeInvalid{91}{96}%
217   \RangeCatcodeInvalid{123}{255}%
218   \catcode'\@=12 %
219   \catcode'\=0 %
220   \catcode'\{=1 %
221   \catcode'\}=2 %
222   \catcode'\#=6 %
223   \catcode'\[=12 %
224   \catcode'\]=12 %
225   \catcode'\%=14 %
226   \catcode'\ =10 %
227   \catcode13=5 %
228   \input engord.sty\relax
229   \RestoreCatcodes
230 }
231 \Test
232 \csname @@end\endcsname
233 \end
234 \test1

```

## 4 Installation

### 4.1 Download

**Package.** This package is available on CTAN<sup>1</sup>:

[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:macros/latex/contrib/oberdiek/oberdiek-tds.zip](#)

*TDS* refers to the standard “A Directory Structure for  $\TeX$  Files” ([CTAN:tds/tds.pdf](#)). Directories with `texmf` in their name are usually organized this way.

### 4.2 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
```

**Script installation.** Check the directory `TDS:scripts/oberdiek/` for scripts that need further installation steps. Package `attachfile2` comes with the Perl script `pdfatfi.pl` that should be installed in such a way that it can be called as `pdfatfi`. Example (linux):

```
chmod +x scripts/oberdiek/pdfatfi.pl
cp scripts/oberdiek/pdfatfi.pl /usr/local/bin/
```

### 4.3 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):

<code>engord.sty</code>	$\rightarrow$ <code>tex/generic/oberdiek/engord.sty</code>
<code>engord.pdf</code>	$\rightarrow$ <code>doc/latex/oberdiek/engord.pdf</code>
<code>test/engord-test1.tex</code>	$\rightarrow$ <code>doc/latex/oberdiek/test/engord-test1.tex</code>
<code>engord.dtx</code>	$\rightarrow$ <code>source/latex/oberdiek/engord.dtx</code>

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`.

### 4.4 Refresh file name databases

If your  $\TeX$  distribution (te $\TeX$ , mik $\TeX$ , ...) relies on file name databases, you must refresh these. For example, te $\TeX$  users run `texhash` or `mktextlsr`.

---

<sup>1</sup><http://ftp.ctan.org/tex-archive/>



## 4.5 Some details for the interested

**Attached source.** The PDF documentation on CTAN also includes the `.dtx` source file. It can be extracted by AcrobatReader 6 or higher. Another option is `pdftk`, e.g. unpack the file into the current directory:

```
pdftk engord.pdf unpack_files output .
```

**Unpacking with L<sup>A</sup>T<sub>E</sub>X.** The `.dtx` chooses its action depending on the format:

**plain-T<sub>E</sub>X:** Run `docstrip` and extract the files.

**L<sup>A</sup>T<sub>E</sub>X:** Generate the documentation.

If you insist on using L<sup>A</sup>T<sub>E</sub>X for `docstrip` (really, `docstrip` does not need L<sup>A</sup>T<sub>E</sub>X), 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 pdfL<sup>A</sup>T<sub>E</sub>X:

```
pdflatex engord.dtx
makeindex -s gind.ist engord.idx
pdflatex engord.dtx
makeindex -s gind.ist engord.idx
pdflatex engord.dtx
```

## 5 History

[2000/05/23 v1.0]

- First public release.

[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-T<sub>E</sub>X.
- 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).

## 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; numbers in roman refer to the code lines where the entry is used.

Symbols	
<code>\#</code> .....	222
<code>\%</code> .....	225
<code>\@</code> .....	193, 218
<code>\@PackageWarning</code> .....	124
<code>\@ReturnAfterFi</code> .....	149, 156, 174
<code>\@firstofone</code> .....	183
<code>\@one</code> .....	202, 210
<code>\@nil</code> .....	150, 170, 175, 180, 182, 183
<code>\[</code> .....	223
<code>\</code> .....	171, 219
<code>\{</code> .....	220
<code>\}</code> .....	221
<code>\]</code> .....	224
<code>\_</code> .....	226
A	
<code>\advance</code> .....	202, 210
C	
<code>\catcode</code> .....	3, 4, 5, 6, 7, 31, 32, 33, 34, 35, 36, 37, 38, 56, 58, 62, 64, 193, 199, 208, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227
<code>\count@</code> .....	195, 199, 201, 202, 206, 208, 209, 210
<code>\csname</code> .....	8, 18, 39, 52, 55, 76, 85, 93, 99, 106, 108, 121, 130, 232
D	
<code>\DeclareOption</code> .....	95, 96
E	
<code>\empty</code> .....	12
<code>\end</code> .....	233
<code>\endcsname</code> ...	8, 18, 39, 52, 55, 76, 85, 93, 99, 106, 108, 121, 130, 232
<code>\endinput</code> .....	27
<code>\engord</code> .....	1, 129
<code>\engorderror</code> .....	2, 121, 141
<code>\engordletters</code> .....	2, 99, 123, 159
<code>\engordnumber</code> .....	2, 133, 136
<code>\engordraisefalse</code> .....	2, 95
<code>\engordraisetrue</code> .....	2, 91, 96
<code>\engordtextsuperscript</code> ....	102, 106
<code>\EO@afterreverse</code> .....	150, 179
<code>\EO@AtEnd</code> .....	60, 61, 190
<code>\EO@def</code> .....	75, 100, 122, 132, 136
<code>\EO@number</code> .....	137, 139
<code>\EO@ord</code> .....	144, 157, 168, 185, 187
<code>\EO@reverse</code> .....	150, 170, 183
<code>\EO@reverseback</code> .....	180, 182
<code>\EO@twodigits</code> .....	147, 167
I	
<code>\ifcase</code> .....	9, 160
<code>\ifengordraise</code> .....	90, 101
<code>\ifmmode</code> .....	111
<code>\ifnum</code> ....	140, 143, 146, 184, 201, 209
<code>\ifx</code> .....	10, 12, 18, 39, 47, 76, 85, 93, 99, 106, 108, 121, 130, 171
<code>\immediate</code> .....	20, 41
<code>\input</code> .....	86, 228
L	
<code>\loop</code> .....	196, 207
N	
<code>\newcommand</code> .....	80
<code>\newif</code> .....	90
<code>\number</code> .....	137
P	
<code>\PackageInfo</code> .....	23
<code>\ProcessOptions</code> .....	97
<code>\ProvidesPackage</code> .....	53
R	
<code>\RangeCatcodeInvalid</code> .....	205, 214, 215, 216, 217
<code>\repeat</code> .....	203, 211
<code>\RequirePackage</code> .....	88
<code>\RestoreCatcodes</code> ..	194, 197, 198, 229
<code>\rm</code> .....	112, 114

<b>T</b>		<b>V</b>	
<code>\Test</code> .....	213, 231	<code>\value</code> .....	133
<code>\textsuperscript</code> .....	118	<b>W</b>	
<code>\the</code> .....	56, 62, 199	<code>\write</code> .....	20, 41
<code>\TMP@EnsureCode</code> .....	59,	<b>X</b>	
66, 67, 68, 69, 70, 71, 72, 73, 74		<code>\x</code> .....	8, 10, 12, 19, 23, 25, 40, 45, 52