

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^AT_EX 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	4
2.4	Suffix generation	6
3	Test	7
3.1	Catcode checks for loading	7
4	Installation	8
4.1	Download	8
4.2	Bundle installation	9
4.3	Package installation	9
4.4	Refresh file name databases	9
4.5	Some details for the interested	9
5	History	10
	[2000/05/23 v1.0]	10
	[2003/04/28 v1.1]	10
	[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
6	Index	11

1 Usage

<code>\engord{<L^AT_EX counter name>}</code>
--

It prints the value of the L^AT_EX 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_EX.

`\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 L^AT_EX 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^AT_EX.

```
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 \catcode35 6 % #
19 \catcode123 1 % {
20 \catcode125 2 % }
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 \endgroup
30 \expandafter\endinput
31 \fi
32 \endgroup
```

Package identification:

```
33 \begingroup
34 \catcode35 6 % #
35 \catcode40 12 % (
36 \catcode41 12 % )
37 \catcode44 12 % ,
38 \catcode45 12 % -
39 \catcode46 12 % .
40 \catcode47 12 % /
41 \catcode58 12 % :
42 \catcode64 11 % @
43 \catcode123 1 % {
44 \catcode125 2 % }
45 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
46 \def\x#1#2#3[#4]{\endgroup
47 \immediate\write-1{Package: #3 #4}%
48 \xdef#1{#4}%
49 }%
50 \else
51 \def\x#1#2[#3]{\endgroup
52 #2[#3]}%
53 \ifx#1\relax
54 \xdef#1{#3}%
55 \fi
56 }%
```

```

57 \fi
58 \expandafter\x\csname ver@engord.sty\endcsname
59 \ProvidesPackage{engord}%
60 [2007/09/20 v1.6 Provides English ordinal numbers (H0)]

```

2.2 Help commands for plain compatibility

```

61 \begingroup
62 \catcode123 1 % {
63 \catcode125 2 % }
64 \def\x{\endgroup
65 \expandafter\edef\csname EO@AtEnd\endcsname{%
66 \catcode35 \the\catcode35\relax
67 \catcode64 \the\catcode64\relax
68 \catcode123 \the\catcode123\relax
69 \catcode125 \the\catcode125\relax
70 }%
71 }%
72 \x
73 \catcode35 6 % #
74 \catcode64 11 % @
75 \catcode123 1 % {
76 \catcode125 2 % }
77 \def\TMP@EnsureCode#1#2{%
78 \edef\EO@AtEnd{%
79 \EO@AtEnd
80 \catcode#1 \the\catcode#1\relax
81 }%
82 \catcode#1 #2\relax
83 }
84 \TMP@EnsureCode{33}{12}% !
85 \TMP@EnsureCode{36}{3}% $
86 \TMP@EnsureCode{39}{12}% '
87 \TMP@EnsureCode{42}{12}% *
88 \TMP@EnsureCode{46}{12}% .
89 \TMP@EnsureCode{47}{12}% /
90 \TMP@EnsureCode{60}{12}% <
91 \TMP@EnsureCode{94}{7}% ^ (superscript)
92 \TMP@EnsureCode{96}{12}% '

```

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

```

93 \begingroup\expandafter\expandafter\expandafter\endgroup
94 \expandafter\ifx\csname newcommand\endcsname\relax
95 \def\EO@def{\def}%
96 \else
97 \def\EO@def#1{%
98 \newcommand*{#1}{}%
99 \def#1%
100 }%
101 \fi

102 \begingroup\expandafter\expandafter\expandafter\endgroup
103 \expandafter\ifx\csname RequirePackage\endcsname\relax
104 \input infwarerr.sty\relax
105 \else
106 \RequirePackage{infwarerr}[2007/09/09]%
107 \fi

```

2.3 User macros

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

```

108 \newif\ifengordraise
109 \engordraisetrue

```

In L^AT_EX this also can be controlled by option normal or raise.

```

110 \begingroup\expandafter\expandafter\expandafter\endgroup
111 \expandafter\ifx\csname DeclareOption\endcsname\relax
112 \else
113   \DeclareOption{normal}{\engordraisefalse}%
114   \DeclareOption{raise}{\engordraisetrue}%
115   \ProcessOptions*\relax
116 \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`.

```

117 \expandafter\ifx\csname engordletters\endcsname\relax
118   \EO@def\engordletters{%
119     \ifengordraise
120       \expandafter\engordtextsuperscript
121     \fi
122   }%
123 \fi

```

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

```

124 \expandafter\ifx\csname engordtextsuperscript\endcsname\relax
125   \begingroup\expandafter\expandafter\expandafter\endgroup
126   \expandafter\ifx\csname textsuperscript\endcsname\relax
127     \def\engordtextsuperscript#1{%
128       \relax
129       \ifmmode
130         ^{\rm#1}%
131       \else
132         ${\rm#1}$%
133       \fi
134     }%
135   \else
136     \def\engordtextsuperscript{\textsuperscript}%
137   \fi
138 \fi

```

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

```

139 \expandafter\ifx\csname engorderror\endcsname\relax
140   \EO@def\engorderror#1{%
141     #1\engordletters{!ERROR!}%
142     \@PackageWarning{engord}{%
143       ‘#1’ is not an ordinal number%
144     }%
145   }%
146 \fi

```

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

```

147 \begingroup\expandafter\expandafter\expandafter\endgroup
148 \expandafter\ifx\csname newcounter\endcsname\relax
149 \else
150   \EO@def\engord#1{%
151     \engordnumber{\value{#1}}%
152   }%
153 \fi

```

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

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

```
154 \EO@def\engordnumber#1{%
155   \expandafter\EO@number\expandafter{\number#1}%
156 }
```

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:

```
157 \def\EO@number#1{%
158   \ifnum#1<1 % handle the error case
159     \engorderror{#1}%
160   \else
161     \ifnum#1<21 %
162       \EO@ord{#1}%
163     \else
164       \ifnum#1<100 %
165         \EO@twodigits#1%
166       \else
167         \@ReturnAfterFi{%
168           \EO@reverse#1\@nil}\EO@afterreverse
169         }%
170       \fi
171     \fi
172   \fi
173 }
```

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

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

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

#1: decimal digits, #1 < 21

```
175 \def\EO@ord#1{%
176   #1%
177   \expandafter\engordletters
178   \ifcase#1{th}\or
179     {st}\or
180     {nd}\or
181     {rd}\else
182     {th}%
183   \fi
184 }
```

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

```
185 \def\EO@twodigits#1#2{%
186   #1\EO@ord{#2}%
187 }
```

\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

```
188 \def\EO@reverse#1#2\@nil#3#4{%
189   \ifx\\#2\\%
190     #4{#1#3}%
191   \else
192     \@ReturnAfterFi{%
193       \EO@reverse#2\@nil{#1#3}{#4}%
194     }
```

```

194     }%
195   \fi
196 }

\EO@afterreverse \EO@afterreverse calls \EO@reverseback so that \EO@reverseback can inspect
the digits of the number.
197 \def\EO@afterreverse#1{%
198   \EO@reverseback#1\@nil
199 }

\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.
200 \def\EO@reverseback#1#2#3\@nil{%
201   \EO@reverse#3\@nil{}\@firstofone
202   \ifnum#2#1<21 %
203     \EO@ord{#2#1}%
204   \else
205     #2\EO@ord{#1}%
206   \fi
207 }

208 \EO@AtEnd
209 \end{package}

```

3 Test

3.1 Catcode checks for loading

```

210 \test1
211 \catcode'\{=1 %
212 \catcode'\}=2 %
213 \catcode'\#=6 %
214 \catcode'\@=11 %
215 \expandafter\ifx\csname count@\endcsname\relax
216   \countdef\count@=255 %
217 \fi
218 \expandafter\ifx\csname @gobble\endcsname\relax
219   \long\def\@gobble#1{}%
220 \fi
221 \expandafter\ifx\csname @firstofone\endcsname\relax
222   \long\def\@firstofone#1{#1}%
223 \fi
224 \expandafter\ifx\csname loop\endcsname\relax
225   \expandafter\@firstofone
226 \else
227   \expandafter\@gobble
228 \fi
229 {%
230   \def\loop#1\repeat{%
231     \def\body{#1}%
232     \iterate
233   }%
234   \def\iterate{%
235     \body
236     \let\next\iterate
237   \else
238     \let\next\relax
239   \fi

```

```

240     \next
241 }%
242 \let\repeat=\fi
243 }%
244 \def\RestoreCatcodes{}
245 \count@=0 %
246 \loop
247   \edef\RestoreCatcodes{%
248     \RestoreCatcodes
249     \catcode\the\count@=\the\catcode\count@\relax
250   }%
251 \ifnum\count@<255 %
252   \advance\count@ 1 %
253 \repeat
254
255 \def\RangeCatcodeInvalid#1#2{%
256   \count@=#1\relax
257   \loop
258     \catcode\count@=15 %
259   \ifnum\count@<#2\relax
260     \advance\count@ 1 %
261   \repeat
262 }
263 \expandafter\ifx\csname LoadCommand\endcsname\relax
264   \def\LoadCommand{\input engord.sty\relax}%
265 \fi
266 \def\Test{%
267   \RangeCatcodeInvalid{0}{47}%
268   \RangeCatcodeInvalid{58}{64}%
269   \RangeCatcodeInvalid{91}{96}%
270   \RangeCatcodeInvalid{123}{255}%
271   \catcode'\@=12 %
272   \catcode'\=0 %
273   \catcode'\{=1 %
274   \catcode'\}=2 %
275   \catcode'\#=6 %
276   \catcode'\[=12 %
277   \catcode'\]=12 %
278   \catcode'\%=14 %
279   \catcode'\ =10 %
280   \catcode13=5 %
281   \LoadCommand
282   \RestoreCatcodes
283 }
284 \Test
285 \csname @@end\endcsname
286 \end
287 </test1>

```

4 Installation

4.1 Download

Package. This package is available on CTAN¹:

[CTAN:macros/latex/contrib/oberdiek/engord.dtx](http://ctan.org/macros/latex/contrib/oberdiek/engord.dtx) The source file.

[CTAN:macros/latex/contrib/oberdiek/engord.pdf](http://ctan.org/macros/latex/contrib/oberdiek/engord.pdf) Documentation.

¹[ftp://ftp.ctan.org/tex-archive/](http://ftp.ctan.org/tex-archive/)

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 T_EX 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-T_EX:

```
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>	→ <code>tex/generic/oberdiek/engord.sty</code>
<code>engord.pdf</code>	→ <code>doc/latex/oberdiek/engord.pdf</code>
<code>test/engord-test1.tex</code>	→ <code>doc/latex/oberdiek/test/engord-test1.tex</code>
<code>engord.dtx</code>	→ <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 T_EX distribution (teT_EX, mikT_EX, ...) relies on file name databases, you must refresh these. For example, teT_EX users run `texhash` or `mktexlsr`.

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^AT_EX. The `.dtx` chooses its action depending on the format:

plain-T_EX: Run `docstrip` and extract the files.

L^AT_EX: Generate the documentation.

If you insist on using L^AT_EX for `docstrip` (really, `docstrip` does not need L^AT_EX), 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^AT_EX:

```
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_EX.
- 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>	213, 275
<code>\%</code>	278
<code>\@</code>	214, 271
<code>\@PackageWarning</code>	142
<code>\@ReturnAfterFi</code>	167, <u>174</u> , 192
<code>\@firstofone</code>	201, 222, 225
<code>\@gobble</code>	219, 227
<code>\@nil</code>	168, 188, 193, 198, 200, 201
<code>\[</code>	276
<code>\</code>	189, 272
<code>\{</code>	211, 273
<code>\}</code>	212, 274
<code>\]</code>	277
<code>_</code>	279
A	
<code>\advance</code>	252, 260
B	
<code>\body</code>	231, 235
C	
<code>\catcode</code> ...	3, 4, 5, 6, 7, 18, 19, 20, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 62, 63, 66, 67, 68, 69, 73, 74, 75, 76, 80, 82, 211, 212, 213, 214, 249, 258, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280
<code>\count@</code>	216, 245, 249, 251, 252, 256, 258, 259, 260
<code>\countdef</code>	216
<code>\csname</code>	8, 21, 45, 58, 65, 94, 103, 111, 117, 124, 126, 139, 148, 215, 218, 221, 224, 263, 285
D	
<code>\DeclareOption</code>	113, 114
E	
<code>\empty</code>	12
<code>\end</code>	286
<code>\endcsname</code>	8, 21, 45, 58, 65, 94, 103, 111, 117, 124, 126, 139, 148, 215, 218, 221, 224, 263, 285
<code>\endinput</code>	30
<code>\engord</code>	1, <u>147</u>
<code>\engorderror</code>	2, <u>139</u> , 159
<code>\engordletters</code>	2, <u>117</u> , 141, 177
<code>\engordnumber</code>	2, 151, <u>154</u>
<code>\engordraisefalse</code>	2, 113
<code>\engordraisetrue</code>	2, 109, 114
<code>\engordtextsuperscript</code>	120, <u>124</u>
<code>\EO@afterreverse</code>	168, <u>197</u>
<code>\EO@AtEnd</code>	78, 79, 208
<code>\EO@def</code>	93, 118, 140, 150, 154
<code>\EO@number</code>	155, <u>157</u>
<code>\EO@ord</code>	162, <u>175</u> , 186, 203, 205
<code>\EO@reverse</code>	168, <u>188</u> , 201
<code>\EO@reverseback</code>	198, <u>200</u>
<code>\EO@twodigits</code>	165, <u>185</u>
I	
<code>\ifcase</code>	9, 178
<code>\ifengordraise</code>	<u>108</u> , 119
<code>\ifmmode</code>	129
<code>\ifnum</code>	158, 161, 164, 202, 251, 259
<code>\ifx</code>	10, 12, 21, 45, 53, 94, 103, 111, 117, 124, 126, 139, 148, 189, 215, 218, 221, 224, 263
<code>\immediate</code>	23, 47
<code>\input</code>	104, 264
<code>\iterate</code>	232, 234, 236
L	
<code>\LoadCommand</code>	264, 281
<code>\loop</code>	230, 246, 257
N	
<code>\newcommand</code>	98
<code>\newif</code>	108
<code>\next</code>	236, 238, 240
<code>\number</code>	155
P	
<code>\PackageInfo</code>	26
<code>\ProcessOptions</code>	115
<code>\ProvidesPackage</code>	59
R	
<code>\RangeCatcodeInvalid</code>	255, 267, 268, 269, 270
<code>\repeat</code>	230, 242, 253, 261
<code>\RequirePackage</code>	106
<code>\RestoreCatcodes</code> ..	244, 247, 248, 282
<code>\rm</code>	130, 132
T	
<code>\Test</code>	266, 284
<code>\textsuperscript</code>	136
<code>\the</code>	66, 67, 68, 69, 80, 249
<code>\TMP@EnsureCode</code>	77, 84, 85, 86, 87, 88, 89, 90, 91, 92
V	
<code>\value</code>	151
W	
<code>\write</code>	23, 47
X	
<code>\x</code> ..	8, 10, 12, 22, 26, 28, 46, 51, 58, 64, 72