

# The engord package

Heiko Oberdiek  
<heiko.oberdiek at gmail.com>

2010/03/01 v1.8

## 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>2</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 . . . . .	4
2.3	User macros . . . . .	5
2.4	Suffix generation . . . . .	6
<b>3</b>	<b>Test</b>	<b>8</b>
3.1	Catcode checks for loading . . . . .	8
<b>4</b>	<b>Installation</b>	<b>9</b>
4.1	Download . . . . .	9
4.2	Bundle installation . . . . .	10
4.3	Package installation . . . . .	10
4.4	Refresh file name databases . . . . .	10
4.5	Some details for the interested . . . . .	10
<b>5</b>	<b>Catalogue</b>	<b>11</b>
<b>6</b>	<b>History</b>	<b>11</b>
	[2000/05/23 v1.0] . . . . .	11
	[2003/04/28 v1.1] . . . . .	12
	[2006/02/20 v1.2] . . . . .	12
	[2007/04/11 v1.3] . . . . .	12
	[2007/04/26 v1.4] . . . . .	12
	[2007/09/09 v1.5] . . . . .	12
	[2007/09/20 v1.6] . . . . .	12
	[2008/08/11 v1.7] . . . . .	12
	[2010/03/01 v1.8] . . . . .	12
<b>7</b>	<b>Index</b>	<b>12</b>

# 1 Usage

```
\engord{<LATEX counter name>}
```

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} → 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<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\catcode61\catcode48\catcode32=10\relax%
3 \catcode13=5 % ^~M
4 \endlinechar=13 %
5 \catcode35=6 % #
6 \catcode39=12 % '
7 \catcode44=12 % ,
8 \catcode45=12 % -
9 \catcode46=12 % .
10 \catcode58=12 % :
11 \catcode64=11 % @
12 \catcode123=1 % {
13 \catcode125=2 % }
14 \expandafter\let\expandafter\x\csname ver@engord.sty\endcsname
15 \ifx\x\relax % plain-TeX, first loading
16 \else
17 \def\empty{}%
18 \ifx\x\empty % LaTeX, first loading,
19 % variable is initialized, but \ProvidesPackage not yet seen
20 \else
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 \aftergroup\endinput
30 \fi
31 \fi
32 \endgroup%
```

Package identification:

```
33 \begingroup\catcode61\catcode48\catcode32=10\relax%
34 \catcode13=5 % ^~M
35 \endlinechar=13 %
36 \catcode35=6 % #
37 \catcode39=12 % '
38 \catcode40=12 % (
39 \catcode41=12 % )
40 \catcode44=12 % ,
41 \catcode45=12 % -
42 \catcode46=12 % .
43 \catcode47=12 % /
44 \catcode58=12 % :
45 \catcode64=11 % @
```

```

46 \catcode91=12 % [
47 \catcode93=12 % ]
48 \catcode123=1 % {
49 \catcode125=2 % }
50 \expandafter\ifx\csname ProvidesPackage\endcsname\relax
51   \def\x#1#2#3[#4]{\endgroup
52     \immediate\write-1{Package: #3 #4}%
53     \xdef#1{#4}%
54   }%
55 \else
56   \def\x#1#2[#3]{\endgroup
57     #2[#{3}]%
58     \ifx#1@\undefined
59       \xdef#1{#3}%
60     \fi
61     \ifx#1\relax
62       \xdef#1{#3}%
63     \fi
64   }%
65 \fi
66 \expandafter\x\csname ver@engord.sty\endcsname
67 \ProvidesPackage{engord}%
68 [2010/03/01 v1.8 Provides English ordinal numbers (H0)]%

```

## 2.2 Help commands for plain compatibility

```

69 \begingroup\catcode61\catcode48\catcode32=10\relax%
70 \catcode13=5 % ^^M
71 \endlinechar=13 %
72 \catcode123=1 % {
73 \catcode125=2 % }
74 \catcode64=11 % @
75 \def\x{\endgroup
76   \expandafter\edef\csname E0@AtEnd\endcsname{%
77     \endlinechar=\the\endlinechar\relax
78     \catcode13=\the\catcode13\relax
79     \catcode32=\the\catcode32\relax
80     \catcode35=\the\catcode35\relax
81     \catcode61=\the\catcode61\relax
82     \catcode64=\the\catcode64\relax
83     \catcode123=\the\catcode123\relax
84     \catcode125=\the\catcode125\relax
85   }%
86 }%
87 \x\catcode61\catcode48\catcode32=10\relax%
88 \catcode13=5 % ^^M
89 \endlinechar=13 %
90 \catcode35=6 % #
91 \catcode64=11 % @
92 \catcode123=1 % {
93 \catcode125=2 % }
94 \def\TMP@EnsureCode#1#2{%
95   \edef\E0@AtEnd{%
96     \E0@AtEnd
97     \catcode#1=\the\catcode#1\relax
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@AtEnd\noexpand\endinput}

\EO@def Definitions, \newcommand does not exist in plain TEX.
113 \begingroup\expandafter\expandafter\expandafter\endgroup
114 \expandafter\ifx\csname newcommand\endcsname\relax
115 \def\EO@def{\def}%
116 \else
117 \def\EO@def#1{%
118 \newcommand*{#1}{}%
119 \def#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}[2010/03/01]%
129 \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

```

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

```

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`.

```

139 \expandafter\ifx\csname engordletters\endcsname\relax
140 \EO@def\engordletters{%
141 \ifengordraise
142 \expandafter\engordtextsuperscript
143 \fi
144 }%
145 \fi

```

`\engordtextsuperscript` For plain T<sub>E</sub>X the definition is quite ugly, redefine `\engordtextsuperscript` if you have a better one.

```

146 \expandafter\ifx\csname engordtextsuperscript\endcsname\relax
147 \begingroup\expandafter\expandafter\expandafter\endgroup
148 \expandafter\ifx\csname textsuperscript\endcsname\relax
149 \def\engordtextsuperscript#1{%
150 \relax

```

```

151     \ifmmode
152       ^{\rm#1}%
153     \else
154       $\^{\rm#1}$%
155     \fi
156   }%
157 \else
158   \def\engordtextsuperscript{\textsuperscript}%
159 \fi
160 \fi

```

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

```

161 \expandafter\ifx\csname engorderror\endcsname\relax
162 \EO@def\engorderror#1{%
163   #1\engordletters{!ERROR!}%
164   \@PackageWarning{engord}{%
165     `#1' is not an ordinal number%
166   }%
167 }%
168 \fi

```

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

```

169 \begingroup\expandafter\expandafter\expandafter\endgroup
170 \expandafter\ifx\csname newcounter\endcsname\relax
171 \else
172   \EO@def\engord#1{%
173     \engordnumber{\value{#1}}%
174   }%
175 \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.

```

176 \EO@def\engordnumber#1{%
177   \expandafter\EO@number\expandafter{\number#1}%
178 }

```

## 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:

```

179 \def\EO@number#1{%
180   \ifnum#1<1 % handle the error case
181     \engorderror{#1}%
182   \else
183     \ifnum#1<21 %
184       \EO@ord{#1}%
185     \else
186       \ifnum#1<100 %
187         \EO@twodigits#1%
188       \else
189         \@ReturnAfterFi{%
190           \EO@reverse#1\@nil}\EO@afterreverse
191         }%
192       \fi
193     \fi
194 \fi
195 }

```

```

\@ReturnAfterFi An internal help macro to prevent a too deep \if nesting.
196 \long\def\@ReturnAfterFi#1\fi{\fi#1}

\EO@ord \EO@ord prints the number with ord letters.
#1: decimal digits, #1 < 21
197 \def\EO@ord#1{%
198 #1%
199 \expandafter\engordletters
200 \ifcase#1{th}\or
201 {st}\or
202 {nd}\or
203 {rd}\else
204 {th}%
205 \fi
206 }

\EO@twodigits \EO@twodigits expects a number with two digits,
20 < number < 100
207 \def\EO@twodigits#1#2{%
208 #1\EO@ord{#2}%
209 }

\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
210 \def\EO@reverse#1#2\@nil#3#4{%
211 \ifx\#2\%
212 #4{#1#3}%
213 \else
214 \@ReturnAfterFi{%
215 \EO@reverse#2\@nil{#1#3}{#4}%
216 }%
217 \fi
218 }

\EO@afterreverse \EO@afterreverse calls \EO@reverseback so that \EO@reverseback can inspect
the digits of the number.
219 \def\EO@afterreverse#1{%
220 \EO@reverseback#1\@nil
221 }

\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.
222 \def\EO@reverseback#1#2#3\@nil{%
223 \EO@reverse#3\@nil}\@firstofone
224 \ifnum#2#1<21 %
225 \EO@ord{#2#1}%
226 \else
227 #2\EO@ord{#1}%
228 \fi
229 }

230 \EO@AtEnd%
231 \end{package}

```

## 3 Test

### 3.1 Catcode checks for loading

```
232 {*test1}
233 \catcode`\{=1 %
234 \catcode`\}=2 %
235 \catcode`\#=6 %
236 \catcode`\@=11 %
237 \expandafter\ifx\csname count@\endcsname\relax
238   \countdef\count@=255 %
239 \fi
240 \expandafter\ifx\csname @gobble\endcsname\relax
241   \long\def\@gobble#1{}%
242 \fi
243 \expandafter\ifx\csname @firstofone\endcsname\relax
244   \long\def\@firstofone#1{#1}%
245 \fi
246 \expandafter\ifx\csname loop\endcsname\relax
247   \expandafter\@firstofone
248 \else
249   \expandafter\@gobble
250 \fi
251 {%
252   \def\loop#1\repeat{%
253     \def\body{#1}%
254     \iterate
255   }%
256   \def\iterate{%
257     \body
258     \let\next\iterate
259   \else
260     \let\next\relax
261   \fi
262   \next
263 }%
264 \let\repeat=\fi
265 }%
266 \def\RestoreCatcodes{}
267 \count@=0 %
268 \loop
269   \edef\RestoreCatcodes{%
270     \RestoreCatcodes
271     \catcode\the\count@=\the\catcode\count@\relax
272   }%
273 \ifnum\count@<255 %
274   \advance\count@ 1 %
275 \repeat
276
277 \def\RangeCatcodeInvalid#1#2{%
278   \count@=#1\relax
279   \loop
280     \catcode\count@=15 %
281     \ifnum\count@<#2\relax
282       \advance\count@ 1 %
283     \repeat
284 }
285 \def\RangeCatcodeCheck#1#2#3{%
286   \count@=#1\relax
287   \loop
288     \ifnum#3=\catcode\count@
289     \else
```



```

290     \errmessage{%
291         Character \the\count@\space
292         with wrong catcode \the\catcode\count@\space
293         instead of \number#3%
294     }%
295     \fi
296     \ifnum\count@<#2\relax
297         \advance\count@ 1 %
298     \repeat
299 }
300 \def\space{ }
301 \expandafter\ifx\csname LoadCommand\endcsname\relax
302 \def\LoadCommand{\input engord.sty\relax}%
303 \fi
304 \def\Test{%
305     \RangeCatcodeInvalid{0}{47}%
306     \RangeCatcodeInvalid{58}{64}%
307     \RangeCatcodeInvalid{91}{96}%
308     \RangeCatcodeInvalid{123}{255}%
309     \catcode`\@=12 %
310     \catcode`\=0 %
311     \catcode`\%=14 %
312     \LoadCommand
313     \RangeCatcodeCheck{0}{36}{15}%
314     \RangeCatcodeCheck{37}{37}{14}%
315     \RangeCatcodeCheck{38}{47}{15}%
316     \RangeCatcodeCheck{48}{57}{12}%
317     \RangeCatcodeCheck{58}{63}{15}%
318     \RangeCatcodeCheck{64}{64}{12}%
319     \RangeCatcodeCheck{65}{90}{11}%
320     \RangeCatcodeCheck{91}{91}{15}%
321     \RangeCatcodeCheck{92}{92}{0}%
322     \RangeCatcodeCheck{93}{96}{15}%
323     \RangeCatcodeCheck{97}{122}{11}%
324     \RangeCatcodeCheck{123}{255}{15}%
325     \RestoreCatcodes
326 }
327 \Test
328 \csname @@end\endcsname
329 \end
330 </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:install/macros/latex/contrib/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.

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

## 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):

```
engord.sty          → tex/generic/oberdiek/engord.sty
engord.pdf          → doc/latex/oberdiek/engord.pdf
test/engord-test1.tex → doc/latex/oberdiek/test/engord-test1.tex
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`.

## 4.4 Refresh file name databases

If your  $\TeX$  distribution (te $\TeX$ , mi $\TeX$ , ...) relies on file name databases, you must refresh these. For example, te $\TeX$  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  $\LaTeX$ .** 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 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 Catalogue

The following XML file can be used as source for the [T<sub>E</sub>X Catalogue](#). The elements `caption` and `description` are imported from the original XML file from the Catalogue. The name of the XML file in the Catalogue is `engord.xml`.

```
331 (*catalogue)
332 <?xml version='1.0' encoding='us-ascii'?>
333 <!DOCTYPE entry SYSTEM 'catalogue.dtd'>
334 <entry datestamp='$Date$' modifier='$Author$' id='engord'>
335   <name>engord</name>
336   <caption>Converts numbers to English ordinal numbers.</caption>
337   <authorref id='auth:oberdiek' />
338   <copyright owner='Heiko Oberdiek' year='2000,2003,2006-2008,2010' />
339   <license type='lppl1.3' />
340   <version number='1.8' />
341   <description>
342     Defines <tt>\engord</tt> (used like <tt>\arabic</tt>,
343     <tt>\roman</tt>, etc.), and <tt>\engordnumber</tt> (which formats
344     a &#x201C;TeX number&#x201D;).
345     <p/>
346     So <tt>\pagenumbering[engord]</tt> gives page numbers <tt>1st,
347     2nd, 3rd, ...</tt>
348     <p/>
349     The package is part of the <xref refid='oberdiek'>oberdiek</xref>
350     bundle.
351   </description>
352   <documentation details='Package documentation'
353     href='ctan:/macros/latex/contrib/oberdiek/engord.pdf' />
354   <ctan file='true' path='/macros/latex/contrib/oberdiek/engord.dtx' />
355   <miktex location='oberdiek' />
356   <texlive location='oberdiek' />
357   <install path='/macros/latex/contrib/oberdiek/oberdiek.tds.zip' />
358 </entry>
359 </catalogue>
```

## 6 History

[2000/05/23 v1.0]

- First public release, published in newsgroup `de.comp.text.tex`:  
“[Re: Ordinalzahlen in LaTeX?](#)”<sup>2</sup>

<sup>2</sup>Url: <http://groups.google.com/group/de.comp.text.tex/msg/738e2cb4c51759d6>

### [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 `ini- $\TeX$` .

## 7 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.

| Symbols                                                  | A                                                    |
|----------------------------------------------------------|------------------------------------------------------|
| <code>\#</code> ..... 235                                | <code>\advance</code> ..... 274, 282, 297            |
| <code>\%</code> ..... 311                                | <code>\aftergroup</code> ..... 29                    |
| <code>\@</code> ..... 236, 309                           | <code>\arabic</code> ..... 342                       |
| <code>\@PackageWarning</code> ..... 164                  | B                                                    |
| <code>\@ReturnAfterFi</code> ..... 189, <u>196</u> , 214 | <code>\body</code> ..... 253, 257                    |
| <code>\@firstofone</code> ..... 223, 244, 247            | C                                                    |
| <code>\@gobble</code> ..... 241, 249                     | <code>\catcode</code> ..... 2, 3, 5, 6, 7, 8, 9, 10, |
| <code>\@nil</code> ..... 190, 210, 215, 220, 222, 223    | 11, 12, 13, 33, 34, 36, 37, 38, 39,                  |
| <code>\@undefined</code> ..... 58                        | 40, 41, 42, 43, 44, 45, 46, 47, 48,                  |
| <code>\@</code> ..... 211, 310                           | 49, 69, 70, 72, 73, 74, 78, 79, 80,                  |
| <code>\{</code> ..... 233                                | 81, 82, 83, 84, 87, 88, 90, 91, 92,                  |
| <code>\}</code> ..... 234                                |                                                      |

93, 97, 99, 233, 234, 235, 236,  
271, 280, 288, 292, 309, 310, 311

`\count@` ..... 238, 267,  
271, 273, 274, 278, 280, 281,  
282, 286, 288, 291, 292, 296, 297

`\countdef` ..... 238

`\csname` ... 14, 21, 50, 66, 76, 114,  
123, 133, 139, 146, 148, 161,  
170, 237, 240, 243, 246, 301, 328

**D**

`\DeclareOption` ..... 135, 136

**E**

`\empty` ..... 17, 18

`\end` ..... 329

`\endcsname` . 14, 21, 50, 66, 76, 114,  
123, 133, 139, 146, 148, 161,  
170, 237, 240, 243, 246, 301, 328

`\endinput` ..... 29, 112

`\endlinechar` ..... 4, 35, 71, 77, 89

`\engord` ..... 2, 169, 342

`\engorderror` ..... 2, 161, 181

`\engordletters` ..... 2, 139, 163, 199

`\engordnumber` ..... 2, 173, 176, 343

`\engordraisefalse` ..... 2, 135

`\engordraisetrue` ..... 2, 131, 136

`\engordtextsuperscript` ..... 142, 146

`\EO@afterreverse` ..... 190, 219

`\EO@AtEnd` ..... 95, 96, 112, 230

`\EO@def` ..... 113, 140, 162, 172, 176

`\EO@number` ..... 177, 179

`\EO@ord` ..... 184, 197, 208, 225, 227

`\EO@reverse` ..... 190, 210, 223

`\EO@reverseback` ..... 220, 222

`\EO@twodigits` ..... 187, 207

`\errmessage` ..... 290

**I**

`\ifcase` ..... 200

`\ifengordraise` ..... 130, 141

`\ifmmode` ..... 151

`\ifnum` ..... 180,  
183, 186, 224, 273, 281, 288, 296

`\ifx` ... 15, 18, 21, 50, 58, 61, 114,  
123, 133, 139, 146, 148, 161,  
170, 211, 237, 240, 243, 246, 301

`\immediate` ..... 23, 52

`\input` ..... 124, 125, 302

`\iterate` ..... 254, 256, 258

**L**

`\LoadCommand` ..... 302, 312

`\loop` ..... 252, 268, 279, 287

`\ltx@newif` ..... 130

**N**

`\newcommand` ..... 118

`\next` ..... 258, 260, 262

`\number` ..... 177, 293

**P**

`\PackageInfo` ..... 26

`\pagenumbering` ..... 346

`\ProcessOptions` ..... 137

`\ProvidesPackage` ..... 19, 67

**R**

`\RangeCatcodeCheck` .....  
. 285, 313, 314, 315, 316, 317,  
318, 319, 320, 321, 322, 323, 324

`\RangeCatcodeInvalid` .....  
..... 277, 305, 306, 307, 308

`\repeat` ..... 252, 264, 275, 283, 298

`\RequirePackage` ..... 127, 128

`\RestoreCatcodes` .. 266, 269, 270, 325

`\rm` ..... 152, 154

`\roman` ..... 343

**S**

`\space` ..... 291, 292, 300

**T**

`\Test` ..... 304, 327

`\textsuperscript` ..... 158

`\the` ..... 77, 78, 79,  
80, 81, 82, 83, 84, 97, 271, 291, 292

`\TMP@EnsureCode` .....  
..... 94, 101, 102, 103, 104,  
105, 106, 107, 108, 109, 110, 111

**V**

`\value` ..... 173

**W**

`\write` ..... 23, 52

**X**

`\x` 14, 15, 18, 22, 26, 28, 51, 56, 66, 75, 87