The engord package

Heiko Oberdiek[*]
<heiko.oberdiek at googlemail.com>

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

[*] 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}   → 1st
  \engordnumber{12}  → 12th
  \engordnumber{123} → 123rd
  \engord{page}      → 1st (if page has the value of one)
  \engordraisetrue
  \engordnumber{12}  → 12^th
  ```
- 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 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

```
⟨*package⟩
```

Reload check, especially if the package is not used with LaTeX.

```
\begingroup\catcode61\catcode48\catcode32=10\relax%
  \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
      \x{engord}{The package is already loaded}%
      \aftergroup\endinput
    \fi
  \fi
\endgroup%
```

Package identification:

```
\begingroup\catcode61\catcode48\catcode32=10\relax%
  \catcode13=5 % ^^M
  \endlinechar=13 %
  \catcode35=6 % #
  \catcode39=12 % '
  \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\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\relax
      \catcode13=\the\catcode13\relax
      \catcode32=\the\catcode32\relax
      \catcode35=\the\catcode35\relax
      \catcode61=\the\catcode61\relax
      \catcode64=\the\catcode64\relax
      \catcode123=\the\catcode123\relax
      \catcode125=\the\catcode125\relax
    }%
  }%
\x\catcode61\catcode48\catcode32=10\relax%
\catcode13=5 % ^^M
\endlinechar=13 %
\catcode35=6 % #
\catcode64=11 % @
\catcode123=1 % {
\catcode125=2 % }
\def\TMP@EnsureCode#1#2{%
  \edef\EO@AtEnd{%
    \EO@AtEnd
    \catcode#1=\the\catcode#1\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.

```
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname newcommand\endcsname\relax
  \def\EO@def{\def}%
\else
  \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
```

2.3 User macros

\ifengordraise

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

```
\ltx@newif\ifengordraise
\engordraisetrue
```

In LATEX this also can be controlled by option normal or raise.

```
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname DeclareOption\endcsname\relax
\else
  \DeclareOption{normal}{\engordraisefalse}%
  \DeclareOption{raise}{\engordraisetrue}%
  \ProcessOptions*\relax
\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\ifx\csname engordletters\endcsname\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\ifx\csname engordtextsuperscript\endcsname\relax
  \begingroup\expandafter\expandafter\expandafter\endgroup
  \expandafter\ifx\csname textsuperscript\endcsname\relax
    \def\engordtextsuperscript#1{%
      \relax
      \ifmmode
        ^{\rm#1}%
      \else
        $^{\rm#1}$%
      \fi
    }%
  \else
    \def\engordtextsuperscript{\textsuperscript}%
  \fi
\fi
```

\engorderror

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

```
\expandafter\ifx\csname engorderror\endcsname\relax
  \EO@def\engorderror#1{%
    #1\engordletters{!ERROR!}%
    \@PackageWarning{engord}{%
      `#1' is not an ordinal number%
    }%
  }%
\fi
```

\engord

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

```
\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname newcounter\endcsname\relax
\else
  \EO@def\engord#1{%
    \engordnumber{\value{#1}}%
  }%
\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.

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

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#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
}
```

\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.

```
\def\EO@afterreverse#1{%
  \EO@reverseback#1\@nil
}
```

\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.

```
\def\EO@reverseback#1#2#3\@nil{%
  \EO@reverse#3\@nil{}\@firstofone
  \ifnum#2#1<21 %
    \EO@ord{#2#1}%
  \else
    #2\EO@ord{#1}%
  \fi
}
```

```
\EO@AtEnd%
```

```
⟨/package⟩
```

3 Test

3.1 Catcode checks for loading

```
⟨*test1⟩
\catcode`\{=1 %
\catcode`\}=2 %
\catcode`\#=6 %
\catcode`\@=11 %
\expandafter\ifx\csname count@\endcsname\relax
  \countdef\count@=255 %
\fi
\expandafter\ifx\csname @gobble\endcsname\relax
  \long\def\@gobble#1{}%
\fi
\expandafter\ifx\csname @firstofone\endcsname\relax
  \long\def\@firstofone#1{#1}%
\fi
\expandafter\ifx\csname loop\endcsname\relax
  \expandafter\@firstofone
\else
  \expandafter\@gobble
\fi
{%
  \def\loop#1\repeat{%
    \def\body{#1}%
    \iterate
  }%
  \def\iterate{%
    \body
      \let\next\iterate
    \else
      \let\next\relax
    \fi
    \next
  }%
  \let\repeat=\fi
}%
\def\RestoreCatcodes{}
\count@=0 %
\loop
  \edef\RestoreCatcodes{%
    \RestoreCatcodes
    \catcode\the\count@=\the\catcode\count@\relax
  }%
\ifnum\count@<255 %
  \advance\count@ 1 %
\repeat

\def\RangeCatcodeInvalid#1#2{%
  \count@=#1\relax
  \loop
    \catcode\count@=15 %
  \ifnum\count@<#2\relax
    \advance\count@ 1 %
  \repeat
}
\def\RangeCatcodeCheck#1#2#3{%
  \count@=#1\relax
  \loop
    \ifnum#3=\catcode\count@
    \else
```

```
      \errmessage{%
        Character \the\count@\space
        with wrong catcode \the\catcode\count@\space
        instead of \number#3%
      }%
    \fi
  \ifnum\count@<#2\relax
    \advance\count@ 1 %
  \repeat
}
\def\space{ }
\expandafter\ifx\csname LoadCommand\endcsname\relax
  \def\LoadCommand{\input engord.sty\relax}%
\fi
\def\Test{%
  \RangeCatcodeInvalid{0}{47}%
  \RangeCatcodeInvalid{58}{64}%
  \RangeCatcodeInvalid{91}{96}%
  \RangeCatcodeInvalid{123}{255}%
  \catcode`\@=12 %
  \catcode`\\=0 %
  \catcode`\%=14 %
  \LoadCommand
  \RangeCatcodeCheck{0}{36}{15}%
  \RangeCatcodeCheck{37}{37}{14}%
  \RangeCatcodeCheck{38}{47}{15}%
  \RangeCatcodeCheck{48}{57}{12}%
  \RangeCatcodeCheck{58}{63}{15}%
  \RangeCatcodeCheck{64}{64}{12}%
  \RangeCatcodeCheck{65}{90}{11}%
  \RangeCatcodeCheck{91}{91}{15}%
  \RangeCatcodeCheck{92}{92}{0}%
  \RangeCatcodeCheck{93}{96}{15}%
  \RangeCatcodeCheck{97}{122}{11}%
  \RangeCatcodeCheck{123}{255}{15}%
  \RestoreCatcodes
}
\Test
\csname @@end\endcsname
\end
⟨/test1⟩
```

4 Installation

4.1 Download

Package. This package is available on CTAN[1]:

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.

[1] http://ctan.org/pkg/engord

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 (teTEX, mikTEX, …) relies on file name databases, you must refresh these. For example, teTEX users run texhash or mktexlsr.

4.5 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
```

5 Catalogue

The following XML file can be used as source for the TEX 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.

```
⟨*catalogue⟩
<?xml version='1.0' encoding='us-ascii'?>
<!DOCTYPE entry SYSTEM 'catalogue.dtd'>
<entry datestamp='$Date$' modifier='$Author$' id='engord'>
  <name>engord</name>
  <caption>Converts numbers to English ordinal numbers.</caption>
  <authorref id='auth:oberdiek'/>
  <copyright owner='Heiko Oberdiek' year='2000,2003,2006-2008,2010'/>
  <license type='lppl1.3'/>
  <version number='1.9'/>
  <description>
    Defines <tt>\engord</tt> (used like <tt>\arabic</tt>,
    <tt>\roman</tt>, etc.), and <tt>\engordnumber</tt> (which formats
    a &#x201C;TeX number&#x201D;).
    <p/>
    So <tt>\pagenumbering{engord}</tt> gives page numbers <tt>1st,
    2nd, 3rd, ...</tt>
    <p/>
    The package is part of the <xref refid='oberdiek'>oberdiek</xref>
    bundle.
  </description>
  <documentation details='Package documentation'
      href='ctan:/macros/latex/contrib/oberdiek/engord.pdf'/>
  <ctan file='true' path='/macros/latex/contrib/oberdiek/engord.dtx'/>
  <miktex location='oberdiek'/>
  <texlive location='oberdiek'/>
  <install path='/macros/latex/contrib/oberdiek/oberdiek.tds.zip'/>
</entry>
⟨/catalogue⟩
```

6 History

[2000/05/23 v1.0]

- First public release, published in newsgroup de.comp.text.tex: “Re: Ordinalzahlen in LaTeX?”[2]

[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.

[2] Url: http://groups.google.com/group/de.comp.text.tex/msg/738e2cb4c51759d6

- 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.

[2016/05/16 v1.9]

- Documentation updates.

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.