The eolgrab package

Heiko Oberdiek*
<heiko.oberdiek at googlemail.com>
2016/05/16 v1.1

Abstract
This package implements a generic argument grabber to catch an argument that is delimited by the line end.

Contents

1 Documentation 2
  1.1 Examples ............................................. 2
    1.1.1 Small \TeX document as example ................. 4
    1.1.2 \TeX document with environments ............... 4
  1.2 Limitations ........................................ 4

2 Implementation 5
  2.1 Reload check and package identification ........... 5
  2.2 Catcodes ............................................. 6
  2.3 Resources ............................................. 6
  2.4 Macro \eolgrab ....................................... 7

3 Test 8
  3.1 Catcode checks for loading .......................... 8
  3.2 Tests for plain \TeX .................................. 10

4 Installation 10
  4.1 Download ............................................. 10
  4.2 Bundle installation .................................. 10
  4.3 Package installation ................................ 11
  4.4 Refresh file name databases ........................ 11
  4.5 Some details for the interested ................... 11

5 Catalogue 12

6 References 12

7 History 12
  [2011/01/12 v1.0] ....................................... 12
  [2016/05/16 v1.1] ....................................... 12

8 Index 13

*Please report any issues at https://github.com/ho-tex/oberdiek/issues
1 Documentation

The starting point for this package was a feature request of Arno Trautmann in the mailing list texhax\footnote{Info page for mailing list texhax: \url{http://tug.org/mailman/listinfo/texhax}}. A macro \texttt{\eolsection} should behave like \texttt{\section}, but the argument should be delimited by the line end instead of given in curly braces:

\eolsection My Title

Phil Taylor answered this with an implementation for \texttt{\eolsection}. Because this feature could be useful for other macros as well, I answered with an implementation of \texttt{\eolgrab} as general solution \cite{3}.

Both formats plain \TeX{} and \LaTeX{} are supported by the package, see the example for \texttt{\eolsection} below.

\begin{verbatim}
\eolgrab \{\langle code\rangle\} \langle argument\rangle \langle EOL\rangle
\end{verbatim}

Macro \texttt{\eolgrab} takes two arguments. The first argument is \langle code\rangle, a classical undelimited \TeX{} macro argument. The second argument is delimited by the line end \langle EOL\rangle. The macro calls \langle code\rangle with \langle argument\rangle as argument in curly braces. Because the catcode of the line end is changed, \texttt{\eolgrab} will not work in the argument of other macros. Macro \texttt{\eolgrab} is made robust if either \LATEX{}'s \texttt{\texttt{\protected}} or \LaTEX{}'s \texttt{\DeclareRobustCommand} is available.

\begin{verbatim}
\eolgrabopt \{\langle code\rangle\} \langle argument\rangle \langle EOL\rangle
\end{verbatim}

Macro \texttt{\eolgrabopt} passes \langle argument\rangle as optional argument to \langle code\rangle if \langle argument\rangle is not empty.

\eolgrabopt\item foo

becomes to

\item\{\{foo\}\}

The curly argument braces are added to support square brackets inside \langle argument\rangle. If the \langle argument\rangle is empty:

\begin{verbatim}
\eolgrabopt\item
\end{verbatim}

then

\item

is called without optional argument.

1.1 Examples

- The line

\begin{verbatim}
\eolgrab\section My Title
\end{verbatim}

is equivalent to

\begin{verbatim}
\section{My Title}
\end{verbatim}

- The next example uses the star form of \texttt{\section}. Then the command to be called consists of two tokens. Therefore the first argument of \texttt{\eolgrab} needs curly braces:

\begin{verbatim}
\eolgrab{\section*}My Title
\end{verbatim}
becomes

\section*{My Title}

• Now \LaTeX’s \PackageError is used. This macro has three arguments, the package or class name, the message text and the help text. A standard help text of \LaTeX{} is used as given in macro \@ehc. The second argument, the message text is used as argument, delimited by line end:

\eolgrab{\PackageError{foobar}}%
Some error message text\MessageBreak%
with several lines
\@ehc

In the first two lines of the example, the line end is suppressed by the comment character (percent), thus the argument is delimited by the line end of the third line. The result is:

\PackageError{foobar}{Some error message text\MessageBreak
with several lines}\@ehc

• The original request for macro \eolsection, see above, can be implemented easily with the help of \eolgrab. Example for \LaTeX{}:

\usepackage{eolgrab}
\newcommand*{\eolsection}{\eolgrab\section}

Example for plain \TeX{}:

\input eolgrab.sty\relax
\def\eolsection{\eolgrab\section}

And a sophisticated variant for \LaTeX{} that also supports the star syntax and the optional argument:

1 \langle*example-
2 \documentclass{article}
3 \usepackage{eolgrab}
4 \makeatletter
5 \newcommand*{\eolsection}{%\protect\section*{example-
6 \@ifstar{%\protect\section*{example-
7 \eolgrab{%section*}{example-
8 }{%\protect\section*{example-
9 \ifnextchar[{{\protect\eoloptsection
10 \protect\section*{example-
11 }{%\protect\section*{example-
12 \eolgrab{%section}{example-
13 }{%\protect\section*{example-
14 }{%\protect\section*{example-
15 }
16 \newcommand*{\eoloptsection}[1]{%\protect\section*{example-
17 \eolgrab{%section*{example-
18 }{%\protect\section*{example-
19 \makeatother
20 \begin{document}
21 \tableofcontents
22 \eolsection{Section without star and optional argument}
23 \eolsection*{Section with star}
24 \eolsection[Short section title]{Long section title}
25 \end{document}
26 \langle/example-

3
1.1.1 \LaTeX\ document as example

\begin{verbatim}
\documentclass{article}
\usepackage{eolgrab}
\newcommand*{\Begin}{\eolgrab \begin}
\newcommand*{\End}{\eolgrab \end}
\newcommand*{\Item}{\eolgrabopt \item}
\Begin document
\Begin itemize
\Item first item
\Item second item
\End itemize
\Begin description
\Item foo is the first syllable of foobar.
\Item bar is the second syllable of foobar.
\End description
\End document
\end{verbatim}

1.1.2 \LaTeX\ document with environments

\begin{verbatim}
\documentclass{article}
\usepackage{eolgrab}
\newcommand*{\Begin}{\eolgrab \begin}
\newcommand*{\End}{\eolgrab \end}
\newcommand*{\Item}{\eolgrabopt \item}
\Begin document
\Begin itemize
\Item first item
\Item second item
\End itemize
\Begin description
\Item foo is the first syllable of foobar.
\Item bar is the second syllable of foobar.
\End description
\End document
\end{verbatim}

1.2 Limitations

Macro \eolgrab needs to catch the line end. If \TeX\ reads a line, then it throws away the line end characters (carriage return, line feed) and removes spaces at the end of the line. Then it adds the character with the character code that is given by \texttt{\textbackslash endlinechar} at the end of the line. The category code of the inserted character is given by the current value of its \texttt{\textbackslash catcode}. If \texttt{\textbackslash endlinechar} is not a valid character code (especially if it is negative), then no character is added.

In plain \TeX\ and \LaTeX\ the standard settings of the inserted endline character is the character with code 13 (or \texttt{\textbackslash ^\textbackslash M} in \TeX\ notation) with catcode 5 (end of line). That means the inserted end of line character behaves like a space token. For example, it is removed after macro names. Therefore \eolgrab changes the catcode.

Therefore \eolgrab has some limitations:

- Like other verbatim stuff, the macro \eolgrab cannot be used in the argument of other macros. \eolgrab want to change the catcode of the end of line character. If this character is read before, because it is processed as argument of another macro, the catcode is already set and is not reassigned later if \eolgrab changes the category code for this character code.

- The argument must not contain the end of line character. Otherwise the first end of line character is already taken as delimiter, leaving the rest of the line outside the argument.

- Because \eolgrab is probably mostly used in the line with the delimited argument. Therefore changes of \texttt{\textbackslash endlinechar} will not affect the current line.
2 Implementation

2.1 Reload check and package identification

Reload check, especially if the package is not used with \LaTeX.
\providepackage{eolgrab} %
[2016/05/16 v1.1 Catch arguments delimited by end of line (HO)]%

2.2 Catcodes

\begingroup\catcode61\catcode48\catcode32=10\relax%
\catcode13=5 % ^^M
\endlinechar=13 %
\catcode123=1 % {
\catcode125=2 % }
\catcode64=11 % @
\def\x{\endgroup\expandafter\edef\csname eolgrab@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}
}\endgroup
\catcode61\catcode48\catcode32=10\relax%
\catcode13=5 % ^^M
\endlinechar=13 %
\catcode35=6 % #
\catcode64=11 % @
\def\TMP@EnsureCode#1#2{%\catcode#1=\the\catcode#1\relax\catcode#1=#2\relax}
\TMP@EnsureCode{40}{12}% (\TMP@EnsureCode{41}{12}% )
\TMP@EnsureCode{42}{12}% *
\TMP@EnsureCode{46}{12}% \\
\TMP@EnsureCode{47}{12}% /\ TMP@EnsureCode{91}{12}% [\ TMP@EnsureCode{93}{12}% ]
\TMP@EnsureCode{94}{7}% ~
\edef\eolgrab@AtEnd{\eolgrab@AtEnd\noexpand\endinput}

2.3 Resources

\begingroup\expandafter\expandafter\expandafter\endgroup
\expandafter\ifx\csname RequirePackage\endcsname\relax
\input ltxcmds.sty\relax
\input infwarerr.sty\relax
\input infwarerr.sty\relax
\input infwarerr.sty\relax
2.4 Macro \eolgrab

\eolgrab

\eolgrab@ifdefinable{eolgrab}{
  \ltx@ifundefined{protected}{}
  \ltx@ifundefined{DeclareRobustCommand}{}
  \def\eolgrab#1{}
  \newcommand\eolgrab{}
  \DeclareRobustCommand\eolgrab{}
}{
  \begingroup
  \endlinechar=13
  \catcode13=\ltx@active
  \eolgrab@{#1}
  \endgroup
}

\eolgrabopt

\eolgrab@ifdefinable{eolgrabopt}{
  \ltx@ifundefined{protected}{}
  \ltx@ifundefined{DeclareRobustCommand}{}
  \def\eolgrabopt#1{}
  \newcommand\eolgrabopt{}
  \DeclareRobustCommand\eolgrabopt{}
}{
  \begingroup
    \endlinechar=13
    \catcode13=\ltx@active
    \eolgrab@opt{#1}
    \endgroup
}

\begingroup
  \catcode13=\ltx@active
  \ltx@firstofone{\endgroup %

\eolgrab@ifdefinable

\ltx@ifundefined{@ifdefinable}{
  \def\eolgrab@ifdefinable#1#2{
    \ltx@ifundefined{#1}{#2}{}
  }
}{
  \def\eolgrab@ifdefinable#1{
    \expandafter\@ifdefinable\csname#1\endcsname
  }
}

\eolgrabopt

\def\eolgrab@ifdefinable#1{
  \protect\def\eolgrab@ifdefinable#1{
    \begingroup
      \endlinechar=13
      \catcode13=\ltx@active
      \eolgrab@opt{#1}
      \endgroup
  }
}{
  \def\eolgrab@ifdefinable#1{
    \expandafter\@ifdefinable\csname#1\endcsname
  }
}
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
\else
\expandafter\@gobble
\fi
\expandafter\@firstofone
{%
\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@ with wrong catcode \the\catcode\count@ instead of \number#3}%
    \fi
    \ifnum\count@<#2\relax
      \advance\count@ 1 %
    \repeat
}%
\space
\expandafter\ifx\csname LoadCommand\endcsname\relax
  \def\LoadCommand{\input eolgrab.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
3.2 Tests for plain \TeX

\input eolgrab.sty
\catcode`{=1 %
\catcode`}=2 %
eolgrab{\immediate\write16}Hello World
\def\TestExpected{foo bar}
eolgrab{\def\TestResult}foo bar
\ifx\TestExpected\TestResult
\immediate\write16{* Ok (foo bar)}%
\else
\errmessage{Test failed (foo bar)}%
\fi
\begingroup
\def\TestExpected{foobar}%
\endlinechar=-1 %
eolgrab{\def\TestResult}foo bar
\ifx\TestExpected\TestResult
\immediate\write16{* Ok (foobar)}%
\else
\errmessage{Test failed (foobar)}%
\fi
\endgroup
\csname @@end\endcsname

4 Installation

4.1 Download

Package. This package is available on CTAN\textsuperscript{2}:


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.

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
```

\textsuperscript{2}http://ctan.org/pkg/eolgrab
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 eolgrab.dtx
```

TDS. Now the different files must be moved into the different directories in your installation TDS tree (also known as texmf tree):

```
eolgrab.sty → tex/generic/oberdiek/eolgrab.sty
eolgrab.pdf → doc/latex/oberdiek/eolgrab.pdf
eexample/eolgrab-example-ltx.tex → doc/latex/oberdiek/example/eolgrab-example-ltx.tex
eexample/eolgrab-example-env.tex → doc/latex/oberdiek/example/eolgrab-example-env.tex
eexample/eolgrab-example-sec.tex → doc/latex/oberdiek/example/eolgrab-example-sec.tex
test/eolgrab-test1.tex → doc/latex/oberdiek/test/eolgrab-test1.tex
test/eolgrab-test2.tex → doc/latex/oberdiek/test/eolgrab-test2.tex
evolegrab dtx → source/latex/oberdiek/eolgrab.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 (\TeX, \miktex, \ldots) relies on file name databases, you must refresh these. For example, \TeX 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{eolgrab.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 \texttt{latexdoc.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 pdf\LaTeX:

```
pdflatex eolgrab.dtx
makeindex -s gind.ist eolgrab.idx
pdflatex eolgrab.dtx
makeindex -s gind.ist eolgrab.idx
pdflatex eolgrab.dtx
```
5 Catalogue

The following XML file can be used as source for the \TeX\ Catalogue. The elements \texttt{caption} and \texttt{description} are imported from the original XML file from the Catalogue. The name of the XML file in the Catalogue is \texttt{eolgrab.xml}.

\begin{verbatim}
<catalogue>
  <?xml version='1.0' encoding='us-ascii'?>
  <!DOCTYPE entry SYSTEM 'catalogue.dtd'>
  <entry datestamp='$Date$' modifier='$Author$' id='eolgrab'>
    <name>eolgrab</name>
    <caption>Catch arguments delimited by end of line.</caption>
    <authorref id='auth:oberdiek'/>
    <copyright owner='Heiko Oberdiek' year='2011'/>
    <license type='lppl1.3'/>
    <version number='1.1'/>
    <description>
      The package defines the command \texttt{\eolgrab} that reads an
      string terminated by end of line, and then calls a macro with that
      string as argument. Thus
      \begin{verbatim}
      \pre\eolgrab\section My title</pre>
      \end{verbatim}
      is equivalent to
      \begin{verbatim}
      \section{My title}
      \end{verbatim}
    </description>
    <documentation details='Package documentation'
      href='ctan:/macros/latex/contrib/oberdiek/eolgrab.pdf'/>
    <ctan file='true' path='/macros/latex/contrib/oberdiek/eolgrab.dtx'/>
    <miktex location='oberdiek'/>
    <texlive location='oberdiek'/>
    <install path='/macros/latex/contrib/oberdiek/oberdiek.tds.zip'/>
  </entry>
</catalogue>
\end{verbatim}

6 References

[1] Arno Trautmann, \texttt{[texhax] read argument until EOL}; mailing list \texttt{texthax@tug.org}, 2011-01-06;

[2] Philip Taylor, \texttt{Re: [texhax] read argument until EOL}; mailing list \texttt{texhax@tug.org}, 2011-01-06;
\url{http://tug.org/pipermail/texhax/2011-January/016519.html}.

[3] Heiko Oberdiek, \texttt{Re: [texhax] read argument until EOL}; mailing list \texttt{texhax@tug.org}, 2011-01-06;

7 History

[2011/01/12 v1.0]
- First public version.

[2016/05/16 v1.1]
- Documentation updates.
8 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

\# ........................................ 241
\% ........................................ 317
\@ ........................................ 242, 315
\@PackageError .......................... 176
\@ehc .................................. 178
\@firstofone ............................ 250, 253
\@gobble ................................. 247, 255
\@ifdefinable ......................... 183
\@ifnextchar ............................ 9
\@ifstar ................................ 6
\@undefined ............................. 113
\{ ...................................... 316
\} ...................................... 239, 339
\} ....................................... 240, 340

A
\advance ................................. 280, 288, 303
\aftergroup ................................ 84

B
\Begin ................................... 38, 41, 42, 48
\begin ................................... 20, 30, 38
\body ..................................... 259, 263

C
\count@ .................................. 244, 273, 277, 279, 280, 284, 286, 287, 289, 292, 294, 297, 298, 302, 303
\coundef .................................. 244
\csname .................................. 69, 76, 105, 121, 131, 166, 183, 243, 246, 249, 252, 307, 334, 364

D
\DeclareRobustCommand .................. 192, 209
\documentclass ........................ 2, 29, 36

E
\emph ..................................... 32
\empty .................................. 72, 73
\End ..................................... 39, 47, 53, 54
\end .................................... 25, 33, 39, 335, 364
\endsname ................................ 69, 76, 105, 121, 131, 166, 183, 243, 246, 249, 252, 307, 334, 364
\endinput ................................ 84, 164
\endlinechar ............................ 59, 90, 126, 132, 144, 198, 215, 353
\esolgrab ............................... 2, 7, 12, 17, 29, 30, 31, 32, 33
\esolgrab@ ............................... 38, 39, 186, 341, 343, 354, 377, 378
\esolgrab@AtEnd .......................... 173, 186, 203
\esolgrab@opt ........................... 217, 227
\esolgrabopty ........................... 2, 40, 203, 383
\esoloptsection ......................... 10, 16
\esolection .............................. 5, 22, 23, 24
\errmessage ............................ 296, 348, 360

I
\ifnum .................................. 279, 287, 294, 302
\ifx ................................... 70, 73, 76, 105, 113, 116, 166, 243, 246, 249, 252, 307, 345, 357
\immediate ................................ 78, 107, 341, 346, 358
\input .................................. 167, 168, 308, 338
\item .................................... 40, 43, 45, 49, 51
\item .................................... 40, 384
\iterate .................................. 260, 262, 264

L
\LoadCommand ............................ 308, 318
\loop ..................................... 258, 274, 285, 293
\ltx@active .............................. 199, 216, 221
\ltx@backslashchar ...................... 177
\ltx@firstofone .......................... 222
\ltx@fewptyp ............................. 229
\ltx@ifempty ............................. 173, 187, 188, 204, 205
\ltx@ifundefined ......................... 175

M
\makeatletter ................................ 4
\makeatother ................................ 19

N
\newcommand ................................ 5, 16, 38, 39, 40, 191, 208
\next .................................... 264, 266, 268
\number .................................. 299

P
\PackageInfo ................................ 81
\protected ................................ 195, 212
\ProvidesPackage ......................... 74, 122

R
\RangeCatcodeCheck ...................... 291, 319, 320, 321, 322, 323
\RangeCatcodeInvalid ................. 283, 311, 312, 313, 314
\repeat .................................. 258, 270, 281, 289, 304
\RequirePackage .......................... 28, 170, 171
\RestoreCatcodes ......................... 272, 275, 276, 331