The eolgrab package

Heiko Oberdiek

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 1
  1.1 Examples .......................................................... 2
    1.1.1 Small \LaTeX{} document as example .................... 4
    1.1.2 \LaTeX{} 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 ........................................................ 7
  2.4 Macro \eolgrab .................................................... 7

3 Installation 8
  3.1 Download ........................................................ 8
  3.2 Bundle installation ............................................. 9
  3.3 Package installation ........................................... 9
  3.4 Refresh file name databases .................................. 9
  3.5 Some details for the interested .............................. 9

4 References 10

5 History 10
  [2011/01/12 v1.0] .................................................. 10
  [2016/05/16 v1.1] .................................................. 10

6 Index 10

1 Documentation

The starting point for this package was a feature request of Arno Trautmann in the mailing list texhax\textsuperscript{1} [1]. A macro \eolsection should behave like \section, but the argument should be delimited by the line end instead of given in curly braces:

\textsuperscript{*}Please report any issues at https://github.com/ho-tex/oberdiek/issues
\textsuperscript{1}Info page for mailing list texhax: https://tug.org/mailman/listinfo/texhax
Phil Taylor answered this with an implementation for \eolsection. Because this feature could be useful for other macros as well, I answered with an implementation of \eolgrab as general solution [3]. Both formats plain \TeX and L\TeX are supported by the package, see the example for \eolsection below.

\[\texttt{\eolgrab \{\langle code\rangle\} \langle argument\rangle \langle EOL\rangle}\]

Macro \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, \eolgrab will not work in the argument of other macros. Macro \eolgrab is made robust if either ε-\TeX’s \protect or L\TeX’s \DeclareRobustCommand is available.

\[\texttt{\eolgrabopt \{\langle code\rangle\} \langle argument\rangle \langle EOL\rangle}\]

Macro \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:

\eolgrabopt\item

then

\item

is called without optional argument.

1.1 Examples

- The line

\eolgrab\section My Title

is equivalent to

\section{My Title}

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

\eolgrab{\section*}My Title

becomes

\section*{My Title}
• Now \LaTeX's \texttt{\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 \texttt{@ehc}. The second argument, the message text is used as argument, delimited by line end:

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

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:

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

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

\begin{verbatim}
\usepackage{eolgrab}
\newcommand*{\eolsection}{\eolgrab\section}
\end{verbatim}

Example for plain \TeX:

\begin{verbatim}
\input eolgrab.sty\relax
\def\eolsection{\eolgrab\section}
\end{verbatim}

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

\begin{verbatim}
\documentclass{article}
\usepackage{eolgrab}
\makeatletter
\newcommand*{\eolsection}{%\@ifstar{\eolgrab{\section*}}{%\@ifnextchar[{{\eoloptsection}}{\eolgrab\section}}%
\newcommand*{\eoloptsection}[1]{\eolgrab{\section[#1]}}%
\makeatother
\begin{document}
\tableofcontents
\eolsection Section without star and optional argument
\eolsection* Section with star
\eolsection[Short section title]Long section title
\end{document}
\end{verbatim}
1.1.1 Small \LaTeX document as example

\begin{verbatim}
\RequirePackage{eolgrab}
eolgrab\documentclass article
\eolgrab\begin document
\section Hello World
\emph Some text
\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{\endlinechar} at the end of the line. The category code of the inserted character is given by the current value of its \texttt{\catcode}. If \texttt{\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 \textasciitilde 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 `\endlinechar` will not affect the current line.

2 Implementation

56 (*package)

2.1 Reload check and package identification

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

57 `\begingroup\catcode61\catcode48\catcode32=10\relax%
58 \catcode13=5 \^-M
59 \endlinechar=13 %
60 \catcode35=6 % #
61 \catcode39=12 % '
62 \catcode44=12 % ,
63 \catcode45=12 % -
64 \catcode46=12 % .
65 \catcode58=12 % :
66 \catcode64=11 % @
67 \catcode123=1 % {
68 \catcode125=2 % }
69 \expandafter\let\expandafter\x\csname ver@eolgrab.sty\endcsname
70 \ifx\x\relax % plain-\TeX, first loading
71 \else
72 \def\empty{}
73 \ifx\x\empty % \LaTeX, first loading,
74 \\% variable is initialized, but \ProvidesPackage not yet seen
75 \else
76 \expandafter\ifx\csname PackageInfo\endcsname\relax
77 \\def\x#1#2{%
78 \immediate\write-1{Package #1 Info: #2.}%
79 }
80 \else
81 \def\x#1#2{%\PackageInfo{#1}{#2, stopped}}%
82 \fi
83 \x{eolgrab}{The package is already loaded}%
84 \aftergroup\endinput
85 \fi
86 \fi
87 \endgroup%

Package identification:

88 `\begingroup\catcode61\catcode48\catcode32=10\relax%
89 \catcode13=5 \^-M
90 \endlinechar=13 %
91 \catcode35=6 % #
92 \catcode39=12 % '
93 \catcode40=12 % ( 
94 \catcode41=12 % )
95 \catcode44=12 % ,
96 \catcode45=12 % -
97 \catcode46=12 % .
98 \catcode47=12 % /
99 \catcode68=12 % : 
100 \catcode91=12 % [
\begin{verbatim}
\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@eolgrab.sty\endcsname
\ProvidesPackage{eolgrab}\
[2016/05/16 v1.1 Catch arguments delimited by end of line (HO)]%
\end{verbatim}

2.2 Catcodes

\begin{verbatim}
\begingroup\catcode61\catcode48\catcode32=10\relax%
\catcode13=5 \^^M
\endlinechar=13 %
\catcode123=1 \{
\catcode125=2 \}
\def\x\catcode61\catcode48\catcode32=10\relax%
\catcode13=5 \^^M
\endlinechar=13 %
\catcode123=1 \{
\catcode125=2 \}
\def\x#1\catcode61\catcode48\catcode32=10\relax%
\catcode13=5 \^^M
\endlinechar=13 %
\catcode123=1 \{
\catcode125=2 \}
\def\x#1\catcode61\catcode48\catcode32=10\relax%
\catcode13=5 \^^M
\endlinechar=13 %
\catcode123=1 \{
\catcode125=2 \}
\def\x#1\catcode61\catcode48\catcode32=10\relax%
\catcode13=5 \^^M
\endlinechar=13 %
\catcode123=1 \{
\catcode125=2 \}
\def\x#1\catcode61\catcode48\catcode32=10\relax%
\catcode13=5 \^^M
\endlinechar=13 %
\catcode123=1 \{
\catcode125=2 \}
\def\x#1\catcode61\catcode48\catcode32=10\relax%
\end{verbatim}

6
\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
\else
\RequirePackage{ltxcmds}[2010/12/04]%
\RequirePackage{infwarerr}[2010/04/08]%
\fi
\eolgrab@ifdefinable
\ltx@ifundefined{@ifdefinable}{\def\eolgrab@ifdefinable#1#2{\ltx@ifundefined{#1}{#2}{\@PackageError{eolgrab}{Command \ltx@backslashchar#1 already defined}\@ehc}}}{\def\eolgrab@ifdefinable#1{\expandafter\@ifdefinable\csname#1\endcsname}}

2.4 Macro \eolgrab

\eolgrab\eolgrab@ifdefinable{eolgrab}{%
\ltx@ifundefined{protected}{\def\eolgrab@ifdefinable#1{%\ltx@ifundefined{DeclareRobustCommand}{\def\eolgrab#1{\newcommand\eolgrab{}\ DeclareRobustCommand*\eolgrab}}{\protected\def\eolgrab#1{%}}}}{}}%
\begingroup
\endlinechar=13 %
\catcode13=\ltx@active
\eolgrab@{#1}%
\eolgrab@ifdefinable{eolgrabopt}{%
\ltx@ifundefined{protected}{\def\eolgrab@ifdefinable{eolgrabopt}{% \protected\def\eolgrabopt#1{%}}}{}}%
\begingroup
\endlinechar=13 %
\catcode13=\ltx@active
\eolgrabopt@{#1}%
3 Installation

3.1 Download

Package. This package is available on CTAN:


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:pkg/tds). Directories with \texttt{texmf} in their name are usually organized this way.

\footnote{CTAN:pkg/eolgrab}
3.2 Bundle installation

**Unpacking.** Unpack the oberdiek.tds.zip in the TDS tree (also known as texmf tree) of your choice. Example (linux):

```bash
unzip oberdiek.tds.zip -d “/texmf"
```

3.3 Package installation

**Unpacking.** The .dtx file is a self-extracting docstrip archive. The files are extracted by running the .dtx through plain \TeX:

```latex
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`
- `example/eolgrab-example-ltx.tex` → `doc/latex/oberdiek/example/eolgrab-example-ltx.tex`
- `example/eolgrab-example-env.tex` → `doc/latex/oberdiek/example/eolgrab-example-env.tex`
- `example/eolgrab-example-sec.tex` → `doc/latex/oberdiek/example/eolgrab-example-sec.tex`
- `eolgrab.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.

3.4 Refresh file name databases

If your \TeX\ distribution (\TeX\ Live, MiKTeX, …) relies on file name databases, you must refresh these. For example, \TeX\ Live users run `texhash` or `mktexlsr`.

3.5 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
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 `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:\n
```latex
pdflatex eolgr
makeindex -s gind.ist eolgr.idx
pdflatex eolgr
makeindex -s gind.ist eolgr.idx
pdflatex eolgr
```
4 References

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

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

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

5 History

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

[2016/05/16 v1.1]
• Documentation updates.

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; plain numbers refer to the code lines where the entry is used.

Symbols
\CodeError .......................... 176
\ehc .................................. 178
@ifdefinable ........................ 183
@ifnextchar .................................. 9
@ifstar ................................ 6
@undefined ......................... 113
aftergroup .......................... 84
\Begin ..................... 38, 41, 42, 48
\begin .................................. 20, 30, 38
\catcode 57, 58, 60, 61, 62, 63, 64, 65,
66, 67, 68, 88, 89, 91, 92, 93, 94,
95, 96, 97, 98, 99, 100, 101, 102,
103, 104, 124, 125, 127, 128,
129, 133, 134, 135, 136, 137,
138, 139, 142, 143, 145, 146,
147, 148, 152, 154, 199, 216, 221
\csname 69, 76, 105, 121, 131, 166, 183
\DeclareRobustCommand ....... 192, 209
\documentclass ............. 2, 29, 36
\emph ................................. 32
\empty ............................... 72, 73
\End ............................... 39, 47, 53, 54
\end .................................. 25, 33, 39
\endcsname 69, 76, 105, 121, 131, 166, 183
\endinput ......................... 84, 164
\endlinechar ...................... 59, 90, 126, 132, 144, 198, 215
\eolgrab ......................... 2, 7, 12,
17, 29, 30, 31, 32, 33, 38, 39, 186
\eolgrab@ ........................ 200, 223
\eolgrab@AtEnd ............. 150, 151, 164, 236
\eolgrab@ifdefinable .... 173, 186, 203
\eolgrab@opt .................. 217, 227
\eolgrabopt ................... 2, 40, 203
\eoloptsection .................. 10, 15
\eolsection .......................... 5, 22, 23, 24
\ifx .................................. 70, 73, 76, 105, 113, 116, 166
\immediate ...................... 78, 107
\input ............................ 167, 168
<table>
<thead>
<tr>
<th>Command</th>
<th>Lines</th>
</tr>
</thead>
<tbody>
<tr>
<td>\item</td>
<td>40</td>
</tr>
<tr>
<td>\ltx@active</td>
<td>199, 216, 221</td>
</tr>
<tr>
<td>\ltx@backslashchar</td>
<td>177</td>
</tr>
<tr>
<td>\ltx@firstofone</td>
<td>222</td>
</tr>
<tr>
<td>\ltx@ifempty</td>
<td>229</td>
</tr>
<tr>
<td>\ltx@ifundefined</td>
<td>173, 187, 188, 204, 205</td>
</tr>
<tr>
<td>\ltx@ifundefined</td>
<td>175</td>
</tr>
<tr>
<td>\makeatletter</td>
<td>4</td>
</tr>
<tr>
<td>\makeatother</td>
<td>19</td>
</tr>
<tr>
<td>\newcommand</td>
<td>5, 16, 38, 39, 40, 191, 208</td>
</tr>
<tr>
<td>\PackageInfo</td>
<td>81</td>
</tr>
<tr>
<td>\protected</td>
<td>195, 212</td>
</tr>
<tr>
<td>\ProvidesPackage</td>
<td>74, 122</td>
</tr>
<tr>
<td>\RequirePackage</td>
<td>28, 170, 171</td>
</tr>
<tr>
<td>\section</td>
<td>7, 12, 17, 31</td>
</tr>
<tr>
<td>\tableofcontents</td>
<td>21</td>
</tr>
<tr>
<td>\the</td>
<td>132, 133, 134, 135, 136, 137, 138, 139, 152</td>
</tr>
<tr>
<td>\TMP@EnsureCode</td>
<td>149, 156, 157, 158, 159, 160, 161, 162, 163</td>
</tr>
<tr>
<td>\usepackage</td>
<td>3, 37</td>
</tr>
<tr>
<td>\write</td>
<td>78, 107</td>
</tr>
<tr>
<td>\x</td>
<td>69, 70, 73, 77, 81, 83, 106, 111, 121, 130, 142</td>
</tr>
</tbody>
</table>