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

The starting point for this package was a feature request of Arno Trautmann in the mailing list `texhax`[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:

---

[1]Info page for mailing list `texhax`: https://tug.org/mailman/listinfo/texhax

```
\eolsection My Title
```

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 LaTeX are supported by the package, see the example for `\eolsection` below.

`\eolgrab {⟨code⟩} ⟨argument⟩ ⟨EOL⟩`

Macro `\eolgrab` takes two arguments. The first argument is ⟨code⟩, a classical undelimited TeX macro argument. The second argument is delimited by the line end ⟨EOL⟩. The macro calls ⟨code⟩ with ⟨argument⟩ 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 `\protected` or LaTeX's `\DeclareRobustCommand` is available.

`\eolgrabopt {⟨code⟩} ⟨argument⟩ ⟨EOL⟩`

Macro `\eolgrabopt` passes ⟨argument⟩ as optional argument to ⟨code⟩ if ⟨argument⟩ is not empty.

```
\eolgrabopt\item foo
```

becomes to

```
\item[{foo}]
```

The curly argument braces are added to support square brackets inside ⟨argument⟩. If the ⟨argument⟩ 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 `\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    ⟨*example-sec⟩
2    \documentclass{article}
3    \usepackage{eolgrab}
4    \makeatletter
5    \newcommand*{\eolsection}{%
6      \@ifstar{%
7        \eolgrab{\section*}%
8      }{%
9        \@ifnextchar[{%
10         \eoloptsection
11       }{%
12         \eolgrab\section
13       }%
14     }%
15   }
16   \newcommand*{\eoloptsection}[1][]{%
17     \eolgrab{\section[{#1}]}%
18   }
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   ⟨/example-sec⟩
```

1.1.1 Small LaTeX document as example

```
27 ⟨∗example-ltx⟩
28 \RequirePackage{eolgrab}
29 \eolgrab\documentclass article
30 \eolgrab\begin document
31 \eolgrab\section Hello World
32 \eolgrab\emph Some text
33 \eolgrab\end document
34 ⟨/example-ltx⟩
```

1.1.2 LaTeX document with environments

```
35 ⟨∗example-env⟩
36 \documentclass{article}
37 \usepackage{eolgrab}
38 \newcommand*{\Begin}{\eolgrab    \begin}
39 \newcommand*{\End  }{\eolgrab    \end  }
40 \newcommand*{\Item }{\eolgrabopt\item }
41 \Begin document
42   \Begin itemize
43     \Item
44       first item
45     \Item
46       second item
47   \End itemize
48   \Begin description
49     \Item foo
50       is the first syllable of foobar.
51     \Item bar
52       is the second syllable of foobar.
53   \End description
54 \End document
55 ⟨/example-env⟩
```

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 \endlinechar at the end of the line. The category code of the inserted character is given by the current value of its \catcode. If \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 ^^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 LaTeX.

```
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   \catcode58=12 % :
100  \catcode64=11 % @
101  \catcode91=12 % [
```

```
102  \catcode93=12 % ]
103  \catcode123=1 % {
104  \catcode125=2 % }
105  \expandafter\ifx\csname ProvidesPackage\endcsname\relax
106    \def\x#1#2#3[#4]{\endgroup
107      \immediate\write-1{Package: #3 #4}%
108      \xdef#1{#4}%
109    }%
110  \else
111    \def\x#1#2[#3]{\endgroup
112      #2[{#3}]%
113      \ifx#1\@undefined
114        \xdef#1{#3}%
115      \fi
116      \ifx#1\relax
117        \xdef#1{#3}%
118      \fi
119    }%
120  \fi
121 \expandafter\x\csname ver@eolgrab.sty\endcsname
122 \ProvidesPackage{eolgrab}%
123    [2016/05/16 v1.1 Catch arguments delimited by end of line (HO)]%
```

2.2   Catcodes

```
124 \begingroup\catcode61\catcode48\catcode32=10\relax%
125  \catcode13=5 % ^^M
126  \endlinechar=13 %
127  \catcode123=1 % {
128  \catcode125=2 % }
129  \catcode64=11 % @
130  \def\x{\endgroup
131    \expandafter\edef\csname eolgrab@AtEnd\endcsname{%
132      \endlinechar=\the\endlinechar\relax
133      \catcode13=\the\catcode13\relax
134      \catcode32=\the\catcode32\relax
135      \catcode35=\the\catcode35\relax
136      \catcode61=\the\catcode61\relax
137      \catcode64=\the\catcode64\relax
138      \catcode123=\the\catcode123\relax
139      \catcode125=\the\catcode125\relax
140    }%
141  }%
142 \x\catcode61\catcode48\catcode32=10\relax%
143 \catcode13=5 % ^^M
144 \endlinechar=13 %
145 \catcode35=6 % #
146 \catcode64=11 % @
147 \catcode123=1 % {
148 \catcode125=2 % }
149 \def\TMP@EnsureCode#1#2{%
150  \edef\eolgrab@AtEnd{%
151    \eolgrab@AtEnd
152    \catcode#1=\the\catcode#1\relax
153  }%
154  \catcode#1=#2\relax
155 }
156 \TMP@EnsureCode{40}{12}% (
```

```
157 \TMP@EnsureCode{41}{12}% )
158 \TMP@EnsureCode{42}{12}% *
159 \TMP@EnsureCode{46}{12}% .
160 \TMP@EnsureCode{47}{12}% /
161 \TMP@EnsureCode{91}{12}% [
162 \TMP@EnsureCode{93}{12}% ]
163 \TMP@EnsureCode{94}{7}% ^
164 \edef\eolgrab@AtEnd{\eolgrab@AtEnd\noexpand\endinput}
```

2.3   Resources

```
165 \begingroup\expandafter\expandafter\expandafter\endgroup
166 \expandafter\ifx\csname RequirePackage\endcsname\relax
167   \input ltxcmds.sty\relax
168   \input infwarerr.sty\relax
169 \else
170   \RequirePackage{ltxcmds}[2010/12/04]%
171   \RequirePackage{infwarerr}[2010/04/08]%
172 \fi
```

\eolgrab@ifdefinable

```
173 \ltx@IfUndefined{@ifdefinable}{%
174   \def\eolgrab@ifdefinable#1#2{%
175     \ltx@ifundefined{#1}{#2}{%
176       \@PackageError{eolgrab}{%
177         Command \ltx@backslashchar#1 already defined%
178       }\@ehc
179     }%
180   }%
181 }{%
182   \def\eolgrab@ifdefinable#1{%
183     \expandafter\@ifdefinable\csname#1\endcsname
184   }%
185 }
```

2.4   Macro \eolgrab

\eolgrab

```
186 \eolgrab@ifdefinable{eolgrab}{%
187   \ltx@IfUndefined{protected}{%
188     \ltx@IfUndefined{DeclareRobustCommand}{%
189       \def\eolgrab#1%
190     }{%
191       \newcommand\eolgrab{}%
192       \DeclareRobustCommand*\eolgrab
193     }%
194   }{%
195     \protected\def\eolgrab#1%
196   }{%
197     \begingroup
198     \endlinechar=13 %
199     \catcode13=\ltx@active
200     \eolgrab@{#1}%
201   }%
202 }
```

\eolgrabopt

```
203 \eolgrab@ifdefinable{eolgrabopt}{%
204   \ltx@IfUndefined{protected}{%
```

```
205    \ltx@IfUndefined{DeclareRobustCommand}{%
206      \def\eolgrabopt#1%
207    }{%
208      \newcommand\eolgrabopt{}%
209      \DeclareRobustCommand*\eolgrabopt
210    }%
211  }{%
212    \protected\def\eolgrabopt#1%
213  }{%
214    \begingroup
215    \endlinechar=13 %
216    \catcode13=\ltx@active
217    \eolgrab@opt{#1}%
218  }%
219 }

220 \begingroup
221   \catcode13=\ltx@active %
222 \ltx@firstofone{\endgroup %
```

\eolgrab@

```
223    \def\eolgrab@#1#2^^M{%
224      \endgroup %
225      #1{#2}%
226    }%
```

\eolgrab@opt

```
227    \def\eolgrab@opt#1#2^^M{%
228      \endgroup %
229      \ltx@ifempty{#2}{%
230        #1%
231      }{%
232        #1[{#2}]%
233      }%
234    }%

235 }%

236 \eolgrab@AtEnd%

237 ⟨/package⟩
```

3 Installation

3.1 Download

Package. This package is available on CTAN[2]:

CTAN:macros/latex/contrib/oberdiek/eolgrab.dtx The source file.

CTAN:macros/latex/contrib/oberdiek/eolgrab.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:pkg/tds). Directories with texmf in their name are usually organized this way.

---

[2][CTAN:pkg/eolgrab](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):

```
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:

```
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 \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:

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

4 References

[1] Arno Trautmann, *[texhax] read argument until EOL*; mailing list texthax@tug.org, 2011-01-06; https://tug.org/pipermail/texhax/2011-January/016517.html.

[2] Philip Taylor, *Re: [texhax] read argument until EOL*; mailing list texhax@tug.org, 2011-01-06; https://tug.org/pipermail/texhax/2011-January/016519.html.

[3] Heiko Oberdiek, *Re: [texhax] read argument until EOL*; mailing list texhax@tug.org, 2011-01-06; https://tug.org/pipermail/texhax/2011-January/016526.html.

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.