

# The combinedgraphics package\*

Christian Schneider  
<software(at)chschneider(dot)eu>

June 15, 2011

**Warning: This is alpha software and may contain serious bugs!  
Use with caution and on your own risk! Check output!**

## Contents

<b>1</b>	<b>The Combined EPS/L<sup>A</sup>T<sub>E</sub>X format</b>	<b>2</b>
<b>2</b>	<b>Usage</b>	<b>3</b>
2.1	The macro . . . . .	3
2.2	Basic macro options . . . . .	3
2.3	Extended macro options . . . . .	3
2.4	Package options . . . . .	4
<b>3</b>	<b>Bugs, problems, and suggestions</b>	<b>5</b>
<b>4</b>	<b>Implementation</b>	<b>5</b>
4.1	Package options . . . . .	5
4.2	Basic macro options . . . . .	5
4.3	Extended macro options . . . . .	6
4.4	The macro . . . . .	9
4.5	Helper macros . . . . .	11

---

\*This document corresponds to combinedgraphics v0.1.1-alpha, dated 2011/06/15. Copyright 2009,2011 Christian Schneider <software(at)chschneider(dot)eu>.

## Abstract

This package provides a macro (`\includecombinedgraphics`) for the inclusion of combined EPS/L<sup>A</sup>T<sub>E</sub>X and PDF/L<sup>A</sup>T<sub>E</sub>X graphics (an export format of Gnuplot, Xfig, and maybe other programs). Instead of including the graphics with a simple `\input`, the `\includecombinedgraphics` macro has some benefits:

- changing the font and color of the text of the L<sup>A</sup>T<sub>E</sub>X parts
- rescaling the graphics without affecting the font of the L<sup>A</sup>T<sub>E</sub>X parts
- automatic inclusion of the vector graphics parts, as far as L<sup>A</sup>T<sub>E</sub>X parts do not do it (e.g., for files exported from Gnuplot before version 4.2 or Xfig)
- changing the inclusion order of L<sup>A</sup>T<sub>E</sub>X and vector graphics parts (e.g., Gnuplot 4.4 exports files in a way that the vector graphics part overlays the L<sup>A</sup>T<sub>E</sub>X part, which means that text may be hidden under shaded areas)
- rescaling and rotating of complete graphics (similar to `\includegraphics` from `graphics` package)

## 1 The Combined EPS/L<sup>A</sup>T<sub>E</sub>X format

Graphics in combined EPS/L<sup>A</sup>T<sub>E</sub>X or PDF/L<sup>A</sup>T<sub>E</sub>X format consist of two files:

1. an EPS or PDF file containing the vector graphics part
2. an L<sup>A</sup>T<sub>E</sub>X file containing the text part

This format has the advantage that it is possible to generate a high-quality vector graphics with text containing all symbols, macros and other stuff provided by L<sup>A</sup>T<sub>E</sub>X. The text is formatted by L<sup>A</sup>T<sub>E</sub>X itself when including the graphics (*not* during creation of the graphics!) and thus it is possible to use everything provided by L<sup>A</sup>T<sub>E</sub>X up to the font of the document for text inside the graphics.

Typically, the L<sup>A</sup>T<sub>E</sub>X part of the graphics is included by `\input`. The vector graphics part is either included automatically by the L<sup>A</sup>T<sub>E</sub>X part with an `\includegraphics` command (or something similar) and overlaid by the text or the user will have to combine both parts manually like this (e.g., for gnuplot before version 4.2 or Xfig) for a pair of files `foobar.tex/foobar.eps`:

```
\begin{picture}(0,0)%  
  \includegraphics{foobar}%  
\end{picture}%  
\input{foobar}
```

Although the format has some nice advantages compared to other formats, the user has to know whether to include the vector graphics part for (some of) his graphics or not, font changes sometimes require some knowledge of the internal commands of the L<sup>A</sup>T<sub>E</sub>X part (depending on the tool that generated the graphics), and the user is not intended to change the scaling of just the vector graphics part without affecting the text format. Furthermore, rotating, scaling and resizing the whole graphics requires extra-macros, e.g. `\rotatebox`, `\scalebox`, or `\resizebox` from the `graphics/graphics` package.

The intention of this package is to provide a macro for easier handling of combined EPS/L<sup>A</sup>T<sub>E</sub>X or PDF/L<sup>A</sup>T<sub>E</sub>X graphics, just as easy as `\includegraphics`, and removing the limitations mentioned above.

## 2 Usage

### 2.1 The macro

`\includecombinedgraphics` The macro `\includecombinedgraphics[<option list>]{<graphics file>}` is used to include a combined EPS/L<sup>A</sup>T<sub>E</sub>X or PDF/L<sup>A</sup>T<sub>E</sub>X file, where *<graphics file>* is the name of the L<sup>A</sup>T<sub>E</sub>X part of the graphics (*without* `.tex` extension) and *<option list>* is an optional list of *<key>=<value>* pairs (see below).

### 2.2 Basic macro options

The following options are processed first. If an option is specified several times, the last appearance will overwrite the previous ones.

- `textfont=<value>` one or more commands for reformatting the text in L<sup>A</sup>T<sub>E</sub>X part (e. g., `textfont=\Large\bfseries` for large bold-face font or `textfont={}` for default font of graphics overwriting package option)
- `textcolor=<value>` one or more commands for recoloring the text in L<sup>A</sup>T<sub>E</sub>X part (e. g., `textcolor=\color{red}` for red text or `textcolor={}` for default text color of graphics overwriting package option)
- `vecscale=<value>` scaling factor for rescaling the graphics without affecting the font of the L<sup>A</sup>T<sub>E</sub>X part
- `vecinclude=<value>` generate code to include the vector graphics part; *<value>*s are: `auto` to automatically determine if code is required (default), `true/false` to always/never generate that code, or `overwrite` to disable any inclusion of a vector graphics part in the L<sup>A</sup>T<sub>E</sub>X part and generate new code
- `vecfirst=<value>` inclusion order of vector graphics part; valid *<value>*s are: `true` (default) or `false` to include the vector graphics part before or after the L<sup>A</sup>T<sub>E</sub>X part, respectively
- `vecfile=<value>` filename of vector graphics part, if different from filename of L<sup>A</sup>T<sub>E</sub>X part (except for extension); implies `vecinclude=overwrite`

Note: If the vector graphics part is already included by the L<sup>A</sup>T<sub>E</sub>X part, `vecinclude` and `vecscale` will require the vector graphics part to be included by an `\includegraphics` command for proper functioning. Furthermore, `vecscale` also requires that the L<sup>A</sup>T<sub>E</sub>X part puts its text inside a `picture` environment whose scaling is set with `\setlength{\unitlength}{<some length>}`. These two prerequisites are fulfilled by graphics exported from Gnuplot and Xfig.

### 2.3 Extended macro options

The following options are processed *after* the basic options *in the order* of their appearances and affect the *whole* graphics (L<sup>A</sup>T<sub>E</sub>X and vector graphics part). They may be specified several times.

## Rotating

Rotating is basically performed with the means of `\rotatebox` from the `graphics/graphics` package.

`angle=value` angle of rotation

The following options can be used to fine adjust the effect of a rotation and apply to only the next `angle` value specified in the options list.

`origin=value` up to two of `lrctbB` (`B` stands for the baseline)  
`x=value` the `x` coordinate of the center of rotation  
`y=value` the `y` coordinate of the center of rotation  
`units=value` units of rotation angle (e. g., `-360` denotes clockwise rotation)

## Rescaling

Rescaling is performed with the means of `\scalebox` from the `graphics` package.

`scale=value` scaling factor  
`hscale=value` horizontal scaling factor  
`vscale=value` vertical scaling factor

Note: A scaling via `hscale` and `vscale` will only be performed, if a pair of both values is specified.

## Resizing

Resizing is performed with the means of `\resizebox` from the `graphics` package.

`height=value` height of graphics  
`totalheight=value` height + depth of graphics  
`width=value` width of graphics  
`keepaspectratio` if only one of the values for `height/totalheight` or `width` is specified, the other one will be calculated to keep the aspect ratio

Note: Resizing will only be performed, if a pair of `height/totalheight` and `width` is specified or one of them is specified followed by `keepaspectratio`.

## 2.4 Package options

The following options can be passed to the package as defaults for the macro options. If an option is specified several times, the last appearance will overwrite the previous ones.

`textfontcmd=value` name of a command (*without* leading backslash) for re-formatting text in `LATEX` part  
`textcolorcmd=value` name of a command (*without* leading backslash) for recoloring text in `LATEX` part  
`vecscale=value` see section 2.2  
`vecinclude=value` see section 2.2  
`vecfirst=value` see section 2.2

### 3 Bugs, problems, and suggestions

Please report bugs and problems or send suggestions for this package to Christian Schneider. Check for updates before reporting bugs at the website mentioned above. Do *not* bother Markus Kohm or Frank Neukam with bugs, problems or suggestions concerning this package!

### 4 Implementation

Load required packages for graphics, color and  $\langle key \rangle = \langle value \rangle$  pairs:

```
1 \RequirePackage{keyval}
2 \RequirePackage{graphicx}
3 \RequirePackage{color}
```

#### 4.1 Package options

This `if` is required to remember the inclusion order.

```
4 \newif\ifcgr@vecfirst
```

Definitions of package options as  $\langle key \rangle = \langle value \rangle$  pairs. The  $\langle value \rangle$ s are saved in the specified macros that are undefined by default.

```
5 \newcommand*\cgr@defopts{\define@key{cgr}}
6 \newcommand*\cgr@setopts{\setkeys{cgr}}
7 \cgr@defopts{textfontcmd}{\gdef\cgr@textfont@default{\@nameuse{#1}}}
8 \cgr@defopts{textcolorcmd}{\gdef\cgr@textcolor@default{\@nameuse{#1}}}
9 \cgr@defopts{vecscale}{\gdef\cgr@vecscale@default{#1}}
10 \cgr@defopts{vecinclude}{\gdef\cgr@vecinclude@default{#1}}
11 \cgr@defopts{vecfirst}{\gdef\cgr@vecfirst@default{#1}}
```

Next the package options are processed.

```
12 \DeclareOption*{%
13   \expandafter\cgr@setopts\expandafter{\CurrentOption}%
14 }
15 \ProcessOptions\relax
```

#### 4.2 Basic macro options

Now the  $\langle key \rangle = \langle value \rangle$  pairs for the optional argument of `\includecombinedgraphics` are defined. The following two macro are just shortcuts for this purpose.

```
16 \newcommand*\cgr@defopts@combgrphcs{\define@key{cgr@combgrphcs}}
17 \newcommand*\cgr@setopts@combgrphcs{\setkeys{cgr@combgrphcs}}
```

The basic macro options set some macros that are executed while tinkering the graphics from its two parts.

`textfont` and `textcolor` set `\cgr@textfont` and `\cgr@textcolor` macro, respectively, that will be applied to the text in the  $\text{\LaTeX}$  part of graphics *and* disable further formatting of fonts and colors inside the  $\text{\LaTeX}$  part by overwriting some macros afterwards (if not empty).

```
18 \cgr@defopts@combgrphcs{textfont}{%
19   \def\@tempa{#1}%
20   \ifx\@tempa\@empty%
21     \def\cgr@textfont{}}%
```

```

22 \else%
23 \def\cgr@textfont{%
24 #1%
25 \def\SetFigFont{\cgr@gobblefive}% %% Xfig <= 3.2.4
26 \def\SetFigFontNFSS{\cgr@gobblefive}% %% Xfig >= 3.2.5-alpha
27 }%
28 \fi%
29 }
30 \cgr@defopts@combgrphcs{textcolor}{%
31 \def\@tempa{#1}%
32 \ifx\@tempa\@empty%
33 \def\cgr@textcolor{}%
34 \else%
35 \def\cgr@textcolor{%
36 #1%
37 \def\color{\cgr@gobble@optone}%
38 }%
39 \fi%
40 }

```

`vecscale` sets the macro `\cgr@vecscale` to the scaling factor. Furthermore, it overwrites `\setlength` and `\includegraphics` in order to apply the scaling factor to the `picture` environment inside the L<sup>A</sup>T<sub>E</sub>X part (via `\setlength{\unitlength}...`) and to the `\includegraphics` command for inclusion of the vector graphics part, respectively.

```

41 \cgr@defopts@combgrphcs{vecscale}{%
42 \def\cgr@vecscale{#1}%
43 \def\setlength{\cgr@setlength}%
44 \def\includegraphics{\cgr@includegraphics}%
45 }

```

`vecinclude` switches between the different *value*s by setting `\cgr@vecinclude` to one of the for macros `\cgr@requires@graphics{value}`.

```

46 \cgr@defopts@combgrphcs{vecinclude}{%
47 \def\cgr@vecinclude{\@nameuse{cgr@requires@graphics#1}}%
48 }

```

`vecfirst` switches between the *value*s `true` and `false` by setting `\cgr@vecfirsttrue` or `\cgr@vecfirstfalse`, respectively.

```

49 \cgr@defopts@combgrphcs{vecfirst}{%
50 \@nameuse{cgr@vecfirst#1}%
51 }

```

`vecfile` resets `\cgr@vecfile` that defaults to the mandatory argument of `\includecombinedgraphics` to something else *and* sets `vecinclude` to `overwrite`.

```

52 \cgr@defopts@combgrphcs{vecfile}{%
53 \def\cgr@vecfile{#1}%
54 \cgr@setopts@combgrphcs{vecinclude=overwrite}%
55 }

```

### 4.3 Extended macro options

`\includecombinedgraphics` first tinkers the graphics applying the basic macro options and saves the result in `\cgr@curr@pic`. The extended macro options will redefine `\cgr@curr@pic` in the order of their occurrence and put a `\rotatebox`,

`\scalebox` or `\resizebox` around it, if all information is already available to do so: `\cgr@curr@pic`  $\rightarrow$  `\dots box{\cgr@curr@pic}`. Otherwise the *value* will be saved in a macro for later processing (e.g., if `hscale` is set without `vscale` being set before).

```

56 \cgr@defopts@combgrphcs{angle}{%
57   \ifx\cgr@curr@scaleopts\@empty%
58     \cgr@raddto@macro[groupfirst]{\cgr@curr@pic}{\rotatebox{#1}}%
59   \else%
60     \cgr@raddto@macro[groupfirst]{\cgr@curr@pic}{}%
61     \cgr@raddto@macro[expand]{\cgr@curr@pic}{\cgr@curr@scaleopts}%
62     \cgr@raddto@macro{\cgr@curr@pic}{\rotatebox[]}%
63     \def\cgr@curr@scaleopts{}%
64   \fi%
65 }
66 \cgr@defopts@combgrphcs{origin}{\cgr@addto@macro{\cgr@curr@scaleopts}{%
67   origin=#1,}%
68 }
69 \cgr@defopts@combgrphcs{x}{\cgr@addto@macro{\cgr@curr@scaleopts}{x=#1,}%
70 \cgr@defopts@combgrphcs{y}{\cgr@addto@macro{\cgr@curr@scaleopts}{y=#1,}%
71 \cgr@defopts@combgrphcs{units}{\cgr@addto@macro{\cgr@curr@scaleopts}{units=#1,}%
72 \cgr@defopts@combgrphcs{scale}{%
73   \cgr@raddto@macro[groupfirst]{\cgr@curr@pic}{\scalebox{#1}}%
74 }
75 \cgr@defopts@combgrphcs{hscale}{%
76   \def\cgr@curr@hscale{#1}%
77   \ifx\cgr@curr@vscale\@empty%
78     \else%
79       \cgr@raddto@macro[groupfirst]{\cgr@curr@pic}{}%
80       \cgr@raddto@macro[expand]{\cgr@curr@pic}{\cgr@curr@vscale}%
81       \cgr@raddto@macro{\cgr@curr@pic}{[%
82         \cgr@raddto@macro[expand,groupsecond]{\cgr@curr@pic}{\cgr@curr@hscale}%
83         \cgr@raddto@macro{\cgr@curr@pic}{\scalebox}%
84       \def\cgr@curr@hscale{}%
85       \def\cgr@curr@vscale{}%
86     \fi%
87 }
88 \cgr@defopts@combgrphcs{vscale}{%
89   \def\cgr@curr@vscale{#1}%
90   \ifx\cgr@curr@hscale\@empty%
91     \else%
92       \cgr@raddto@macro[groupfirst]{\cgr@curr@pic}{}%
93       \cgr@raddto@macro[expand]{\cgr@curr@pic}{\cgr@curr@vscale}%
94       \cgr@raddto@macro{\cgr@curr@pic}{[%
95         \cgr@raddto@macro[expand,groupsecond]{\cgr@curr@pic}{\cgr@curr@hscale}%
96         \cgr@raddto@macro{\cgr@curr@pic}{\scalebox}%
97       \def\cgr@curr@hscale{}%
98       \def\cgr@curr@vscale{}%
99     \fi%
100 }
101 \cgr@defopts@combgrphcs{height}{%
102   \def\cgr@curr@height{#1}%
103   \ifx\cgr@curr@width\@empty%
104     \else%

```

```

105 \cgr@raddto@macro[groupfirst,groupsecond,expand]{\cgr@curr@pic}{%
106 \cgr@curr@height%
107 }%
108 \cgr@raddto@macro[groupsecond,expand]{\cgr@curr@pic}{\cgr@curr@width}%
109 \cgr@raddto@macro{\cgr@curr@pic}{\resizebox}%
110 \def\cgr@curr@height{}%
111 \def\cgr@curr@width{}%
112 \fi%
113 }
114 \cgr@defopts@combgrphcs{totalheight}{%
115 \def\cgr@curr@totalheight{#1}%
116 \ifx\cgr@curr@width\@empty%
117 \else%
118 \cgr@raddto@macro[groupfirst,groupsecond,expand]{\cgr@curr@pic}{%
119 \cgr@curr@totalheight%
120 }%
121 \cgr@raddto@macro[groupsecond,expand]{\cgr@curr@pic}{\cgr@curr@width}%
122 \cgr@raddto@macro{\cgr@curr@pic}{\resizebox*}%
123 \def\cgr@curr@totalheight{}%
124 \def\cgr@curr@width{}%
125 \fi%
126 }
127 \cgr@defopts@combgrphcs{width}{%
128 \def\cgr@curr@width{#1}%
129 \ifx\cgr@curr@height\@empty%
130 \ifx\cgr@curr@totalheight\@empty%
131 \else%
132 \cgr@raddto@macro[groupfirst,groupsecond,expand]{\cgr@curr@pic}{%
133 \cgr@curr@totalheight%
134 }%
135 \cgr@raddto@macro[groupsecond,expand]{\cgr@curr@pic}{\cgr@curr@width}%
136 \cgr@raddto@macro{\cgr@curr@pic}{\resizebox*}%
137 \def\cgr@curr@totalheight{}%
138 \def\cgr@curr@width{}%
139 \fi%
140 \else%
141 \cgr@raddto@macro[groupfirst,groupsecond,expand]{\cgr@curr@pic}{%
142 \cgr@curr@height%
143 }%
144 \cgr@raddto@macro[groupsecond,expand]{\cgr@curr@pic}{\cgr@curr@width}%
145 \cgr@raddto@macro{\cgr@curr@pic}{\resizebox}%
146 \def\cgr@curr@height{}%
147 \def\cgr@curr@totalheight{}%
148 \def\cgr@curr@width{}%
149 \fi%
150 }
151 \cgr@defopts@combgrphcs{keepaspectratio}[]{%
152 \ifx\cgr@curr@height\@empty%
153 \ifx\cgr@curr@totalheight\@empty%
154 \ifx\cgr@curr@width\@empty%
155 \else%
156 \cgr@setopts@combgrphcs{height=!}%
157 \fi%
158 \else%

```



```

159     \cgr@setopts@combgrphcs{width=!}%
160     \fi%
161   \else%
162     \cgr@setopts@combgrphcs{width=!}%
163     \def\cgr@curr@totalheight{}%
164     \fi%
165 }

```

#### 4.4 The macro

First of all, `\cgr@curr@pic` and the `\cgr@requires@graphics...` macros are initialized.

```

166 \long\def\cgr@curr@pic{}
167 \newif\ifcgr@requires@graphics\cgr@requires@graphicstrue
168 \def\cgr@requires@graphicsauto{}
169 \def\cgr@requires@graphicsoverwrite{%
170   \cgr@requires@graphicstrue%
171   \def\includegraphics{\cgr@gobble@optone}%
172 }

```

Now the macros and `if` for the basic macro options and for temporarily saving  $\langle value \rangle$ s of the extended macro options are initialized.

```

173 \def\cgr@textfont{}
174 \def\cgr@textcolor{}
175 \def\cgr@vecyscale{1}
176 \def\cgr@vecinclude{\cgr@requires@graphicsauto}
177 \cgr@vecfirsttrue
178 \def\cgr@vecfile{}
179 \def\cgr@curr@hscale{}
180 \def\cgr@curr@vscale{}
181 \def\cgr@curr@scalects{}
182 \def\cgr@curr@height{}
183 \def\cgr@curr@width{}
184 \def\cgr@curr@totalheight{}

```

At this point the macro itself is defined.

```

185 \newcommand{\includecombinedgraphics}[2] [] {%
186   \begingroup%

```

To get to know, whether the vector graphics are already included in the  $\LaTeX$  parts or not, a check for an appearance of `\includegraphics` in the  $\LaTeX$  part is performed: the  $\LaTeX$  parts are expanded inside a box with `\includegraphics` being redefined to set an appropriate conditional (and eat the arguments of `\includegraphics`).

```

187     \global\cgr@requires@graphicstrue%
188     \setbox\@tempboxa\hbox{%
189       \def\includegraphics{%
190         \global\cgr@requires@graphicsfalse\cgr@gobble@optone%
191       }%
192       \input{#2}%
193     }%

```

Now the the macros resulting from the basic macro options, the vector graphics part and the  $\LaTeX$  part are added to the (empty) `\cgr@curr@pic` macro. We have to distinguish between two cases: (1) If the vector graphics part is included

before the  $\LaTeX$  part (`vecfirst=true`), the vector graphics file will simply be loaded by `\includegraphics` inside a picture environment followed by the  $\LaTeX$  part included by `\input`. (2) If the vector graphics part is included after the  $\LaTeX$  part (`vecfirst=false`), the `\includegraphics` macro will be put inside the `picture` environment of the  $\LaTeX$  part at the offset position passed to the `picture` environment. Therefore, we will have to wrap the `\picture` macro to gain access to the offsets passed to the `picture` environment in the  $\LaTeX$  part and wrap the `\endpicture` macro to add `\includegraphics` into this `picture` environment.

```

194 \cgr@addto@macro{\cgr@curr@pic}{%
195   %% from basic macro options
196   \cgr@vecinclude\cgr@textfont\cgr@textcolor%
197   %% inclusion of vector graphics part
198   \ifcgr@requires@graphics%
199     \ifcgr@vecfirst%
200       \begin{picture}(0,0)%
201         \cgr@includegraphics@orig[scale=\cgr@vecsca]{\cgr@vecfile}%
202       \end{picture}%
203     \else%
204       \def\picture{\cgr@picture}%
205       \def\endpicture{%
206         \put(\cgr@picture@xoffs,\cgr@picture@yoffs)%
207           {\cgr@includegraphics@orig[scale=\cgr@vecsca]{\cgr@vecfile}}%
208         \cgr@endpicture@orig%
209       }%
210     \fi%
211   \fi%
212   %% inclusion of \LaTeX{} part
213   \input{#2}%
214 }%
```

If set, the package options will be processed. (Suggestions for easier handling of package options are welcome.)

```

215 \@ifundefined{cgr@vecsca@default}{}{%
216   \cgr@setopts@combgrphcs{vecsca=\cgr@vecsca@default}%
217 }%
218 \@ifundefined{cgr@textfont@default}{}{%
219   \cgr@setopts@combgrphcs{textfont=\cgr@textfont@default}%
220 }%
221 \@ifundefined{cgr@textcolor@default}{}{%
222   \cgr@setopts@combgrphcs{textcolor=\cgr@textcolor@default}%
223 }%
224 \@ifundefined{cgr@vecinclude@default}{}{%
225   \cgr@setopts@combgrphcs{vecinclude=\cgr@vecinclude@default}%
226 }%
227 \@ifundefined{cgr@vecfirst@default}{}{%
228   \cgr@setopts@combgrphcs{vecfirst=\cgr@vecfirst@default}%
229 }%
```

Afterwards, the macro containing the name of the vector graphics part is initialized and the options passed the the macro are processed.

```

230 \def\cgr@vecfile{#2}%
231 \cgr@setopts@combgrphcs{#1}%
```

Finally, the macro `\cgr@curr@pic` is complete and can be output.

```
232 \cgr@curr@pic%
233 \endgroup%
234 }
```

## 4.5 Helper macros

This macro eats five arguments (analogous to `\@gobble` or `\@gobbletwo` from the L<sup>A</sup>T<sub>E</sub>X kernel).

```
235 \long\def\cgr@gobblefive#1#2#3#4#5{}
```

This macro eats all stars (if any), all arguments in square brackets (if any) and one mandatory argument.

```
236 \long\def\cgr@gobble@optone{\@ifstar{\cgr@gobble@optone}{\cgr@gobble@optone@}}
237 \long\def\cgr@gobble@optone@{\@ifnextchar [{\cgr@gobble@optone@@}{\@gobble}}
238 \long\def\cgr@gobble@optone@@[#1]{\cgr@gobble@optone@}
```

In order to rescale the `picture` environment inside the L<sup>A</sup>T<sub>E</sub>X part, the `\setlength{\unitlength}{<some dimen>}` must be changed to scale *<some dimen>* by `\cgr@vecsca` at its first appearance in the L<sup>A</sup>T<sub>E</sub>X part. This is the `\setlength` substitute to do so.

```
239 \let\cgr@setlength@orig=\setlength
240 \def\cgr@setlength#1#2{%
241   \ifx#1\unitlength%
242     \@tempdima=#2%
243     \cgr@setlength@orig{#1}{\cgr@vecsca\@tempdima}%
244     \def\setlength{\cgr@setlength@orig}%
245   \else%
246     \cgr@setlength@orig{#1}{#2}%
247   \fi%
248 }
```

Additionally, a `scale=\cgr@vecsca` option must be passed to the `\includegraphics` macro inside the L<sup>A</sup>T<sub>E</sub>X part. This is the substitute of `\includegraphics` for this purpose.

```
249 \let\cgr@includegraphics@orig=\includegraphics
250 \def\cgr@includegraphics{%
251   \@ifstar{%
252     \cgr@includegraphics@s%
253   }{%
254     \cgr@includegraphics@%
255   }%
256 }
257 \def\cgr@includegraphics@{%
258   \@ifnextchar [{%
259     \cgr@includegraphics@@%
260   }{%
261     \cgr@includegraphics@orig[scale=\cgr@vecsca]%
262   }%
263 }
264 \def\cgr@includegraphics@@[#1]{%
265   \@ifnextchar [{%
266     \cgr@includegraphics@@@[#1]%
267   }{%
```

```

268   \cgr@includegraphics@orig[#1,scale=\cgr@vecscalescale]%
269 }%
270 }
271 \def\cgr@includegraphics@@@[#1][#2]{%
272   \cgr@includegraphics@orig[#1][#2,scale=\cgr@vecscalescale]%
273 }
274 \def\cgr@includegraphics@s@{%
275   \@ifnextchar [{%
276     \cgr@includegraphics@s@@%
277   }{%
278     \cgr@includegraphics@orig*[scale=\cgr@vecscalescale]%
279   }%
280 }
281 \def\cgr@includegraphics@s@@[#1]{%
282   \@ifnextchar [{%
283     \cgr@includegraphics@s@@@[#1]%
284   }{%
285     \cgr@includegraphics@orig*[#1,scale=\cgr@vecscalescale]%
286   }%
287 }
288 \def\cgr@includegraphics@s@@@[#1][#2]{%
289   \cgr@includegraphics@orig*[#1][#2,scale=\cgr@vecscalescale]%
290 }

```

To play the trick of including the vector graphics part after the L<sup>A</sup>T<sub>E</sub>X part (`vecfirst=false`), we need to know the offset passed to the `picture` environment in the L<sup>A</sup>T<sub>E</sub>X part. Therefore, the `\picture` macro is redefined.

```

291 \let\cgr@picture@orig=\picture
292 \let\cgr@endpicture@orig=\endpicture
293 \long\def\cgr@picture(#1,#2){%
294   \@ifnextchar({%
295     \cgr@picture@(#1,#2)%
296   }{%
297     \cgr@picture@(#1,#2)(0,0)%
298   }%
299 }
300 \def\cgr@picture@(#1,#2)(#3,#4){%
301   \def\cgr@picture@xoffs{#3}%
302   \def\cgr@picture@yoffs{#4}%
303   \cgr@picture@orig(#1,#2)(#3,#4)%
304 }

```

This macro is identical to `\l@addto@macro` from koma-script bundle. It adds the stuff passed to its second argument to the end of the macro from its first argument.

```

305 \newcommand{\cgr@addto@macro}[2]{%
306   \begingroup\toks@\expandafter{#1#2}%
307   \edef\@tempa{\endgroup\def\noexpand#1{\the\toks@}}%
308   \@tempa%
309 }

```

The next macro is similar, but adds the stuff passed to its second argument to the *beginning* of the macro from its first argument. An optional argument allows for fine tuning: A comma-separated list containing `expand` (expands the stuff from the second argument before adding it), `groupfirst`, and/or `groupsecond` (puts

the stuff from the first/second argument in braces before adding) may be passed.

```

310 \newcommand*\cgr@defopts@raddto{\define@key{cgr@raddto}}
311 \newcommand*\cgr@setopts@raddto{\setkeys{cgr@raddto}}
312 \newif\ifcgr@raddto@expand\cgr@raddto@expandfalse
313 \newif\ifcgr@raddto@groupfirst\cgr@raddto@groupfirstfalse
314 \newif\ifcgr@raddto@groupsecond\cgr@raddto@groupsecondfalse
315 \cgr@defopts@raddto{expand}[true]{\@nameuse{cgr@raddto@expand#1}}
316 \cgr@defopts@raddto{groupfirst}[true]{\@nameuse{cgr@raddto@groupfirst#1}}
317 \cgr@defopts@raddto{groupsecond}[true]{\@nameuse{cgr@raddto@groupsecond#1}}
318 \newtoks\cgr@token@a
319 \newtoks\cgr@token@b
320 \newcommand{\cgr@raddto@macro}[3][[]]{%
321   \begingroup%
322     \cgr@setopts@raddto{#1}%
323     \cgr@token@a\expandafter{#2}%
324     \ifcgr@raddto@expand%
325       \cgr@token@b\expandafter{#3}%
326     \else%
327       \cgr@token@b{#3}%
328     \fi%
329     \ifcgr@raddto@groupfirst%
330       \ifcgr@raddto@groupsecond%
331         \edef\@tempa{\endgroup%
332           \def\noexpand#2{\the\cgr@token@b}{\the\cgr@token@a}}%
333         }%
334       \else%
335         \edef\@tempa{\endgroup%
336           \def\noexpand#2{\the\cgr@token@b{\the\cgr@token@a}}%
337         }%
338       \fi%
339     \else%
340       \ifcgr@raddto@groupsecond%
341         \edef\@tempa{\endgroup%
342           \def\noexpand#2{\the\cgr@token@b}\the\cgr@token@a}%
343         }%
344       \else%
345         \edef\@tempa{\endgroup%
346           \def\noexpand#2{\the\cgr@token@b\the\cgr@token@a}}%
347         }%
348       \fi%
349     \fi%
350   \@tempa%
351 }

```

## Change History

v0.0.1-alpha		ing the inclusion order of L <sup>A</sup> T <sub>E</sub> X
General: initial .dtx version	1	and vector graphics parts
v0.0.2-alpha		v0.1.1-alpha
General: initial release	1	General: fix in Makefiles of package
v0.1.0-alpha		
General: added options for chang-		

## 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; numbers in roman refer to the code lines where the entry is used.

<b>Symbols</b>	<code>\cgr@curr@width</code> ...	<code>\cgr@picture@xoffs</code> .
<code>\@empty</code> .. 20, 32, 57,	. 103, 108, 111,	..... 206, 301
77, 90, 103, 116,	116, 121, 124,	<code>\cgr@picture@yoffs</code> .
129, 130, 152–154	128, 135, 138,	..... 206, 302
<code>\@gobble</code> ..... 237	144, 148, 154, 183	<code>\cgr@raddto@expandfalse</code>
<code>\@ifnextchar</code> 237, 258,	<code>\cgr@defopts</code> ... 5, 7–11	..... 312
265, 275, 282, 294	<code>\cgr@defopts@combgrphcs</code>	<code>\cgr@raddto@groupfirstfalse</code>
<code>\@ifstar</code> ..... 236, 251	..... 16,	..... 313
<code>\@ifundefined</code> . 215,	18, 30, 41, 46,	<code>\cgr@raddto@groupsecondfalse</code>
218, 221, 224, 227	49, 52, 56, 66,	..... 314
<code>\@nameuse</code> ..... 7,	69–72, 75, 88,	<code>\cgr@raddto@macro</code> .
8, 47, 50, 315–317	101, 114, 127, 151	. 58, 60–62, 73,
<code>\@tempa</code> 19, 20, 31, 32,	<code>\cgr@defopts@raddto</code>	79–83, 92–96,
307, 308, 331,	.... 310, 315–317	105, 108, 109,
335, 341, 345, 350	<code>\cgr@endpicture@orig</code>	118, 121, 122,
<code>\@tempboxa</code> ..... 188	..... 208, 292	132, 135, 136,
<code>\@tempdima</code> ... 242, 243	<code>\cgr@gobble@optone</code> .	141, 144, 145, 320
	. 37, 171, 190, 236	<code>\cgr@requires@graphicsauto</code>
<b>B</b>	<code>\cgr@gobble@optone@</code>	..... 168, 176
<code>\begin</code> ..... 200	..... 236–238	<code>\cgr@requires@graphicsfalse</code>
	<code>\cgr@gobble@optone@@</code>	..... 190
<b>C</b>	..... 237, 238	<code>\cgr@requires@graphicsoverwrite</code>
<code>\cgr@addto@macro</code> ..	<code>\cgr@gobblefive</code> ...	..... 169
66, 69–71, 194, 305	..... 25, 26, 235	<code>\cgr@requires@graphicstrue</code>
<code>\cgr@curr@height</code> 102,	<code>\cgr@includegraphics</code>	.... 167, 170, 187
106, 110, 129,	..... 44, 250	<code>\cgr@setlength</code> . 43, 240
142, 146, 152, 182	<code>\cgr@includegraphics@</code>	<code>\cgr@setlength@orig</code>
<code>\cgr@curr@hscale</code> ..	..... 254, 257	. 239, 243, 244, 246
..... 76, 82,	<code>\cgr@includegraphics@@</code>	<code>\cgr@setopts</code> ..... 6, 13
84, 90, 95, 97, 179	..... 259, 264	<code>\cgr@setopts@combgrphcs</code>
<code>\cgr@curr@pic</code> 58, 60–	<code>\cgr@includegraphics@@@</code>	17, 54, 156, 159,
62, 73, 79–83,	..... 266, 271	162, 216, 219,
92–96, 105, 108,	<code>\cgr@includegraphics@orig</code>	222, 225, 228, 231
109, 118, 121,	.... 201, 207,	<code>\cgr@setopts@raddto</code>
122, 132, 135,	249, 261, 268,	..... 311, 322
136, 141, 144,	272, 278, 285, 289	<code>\cgr@textcolor</code> ....
145, 166, 194, 232	<code>\cgr@includegraphics@s@</code>	.. 33, 35, 174, 196
<code>\cgr@curr@scaleopts</code>	..... 252, 274	<code>\cgr@textcolor@default</code>
..... 57, 61,	<code>\cgr@includegraphics@s@@</code>	..... 8, 222
63, 66, 69–71, 181	..... 276, 281	<code>\cgr@textfont</code> .....
<code>\cgr@curr@totalheight</code>	<code>\cgr@includegraphics@s@@@</code>	.. 21, 23, 173, 196
. 115, 119, 123,	..... 283, 288	<code>\cgr@textfont@default</code>
130, 133, 137,	<code>\cgr@picture</code> .. 204, 293	..... 7, 219
147, 153, 163, 184	<code>\cgr@picture@</code> .....	<code>\cgr@token@a</code> 318, 323,
<code>\cgr@curr@vscale</code> ..	.... 295, 297, 300	332, 336, 342, 346
..... 77, 80,	<code>\cgr@picture@orig</code> .	<code>\cgr@token@b</code> .....
85, 89, 93, 98, 180	..... 291, 303	. 319, 325, 327,
		332, 336, 342, 346

<code>\cgr@vecfile</code> ... 53, 178, 201, 207, 230	<b>G</b>	<code>\newif</code> .. 4, 167, 312–314
<code>\cgr@vecfirst@default</code> ..... 11, 228	<code>\gdef</code> ..... 7–11	<code>\newtoks</code> ..... 318, 319
<code>\cgr@vecfirsttrue</code> . 177	<code>\global</code> ..... 187, 190	<code>\noexpand</code> .... 307, 332, 336, 342, 346
<code>\cgr@vecinclude</code> ... ..... 47, 176, 196	<b>H</b>	<b>P</b>
<code>\cgr@vecinclude@default</code> ..... 10, 225	<code>\hbox</code> ..... 188	<code>\picture</code> ..... 204, 291
<code>\cgr@vecscale</code> .. 42, 175, 201, 207, 243, 261, 268, 272, 278, 285, 289	<b>I</b>	<code>\ProcessOptions</code> ... 15
<code>\cgr@vecscale@default</code> ..... 9, 216	<code>\ifcgr@raddto@expand</code> ..... 312, 324	<code>\put</code> ..... 206
<code>\color</code> ..... 37	<code>\ifcgr@raddto@groupfirst</code> ..... 313, 329	<b>R</b>
<code>\CurrentOption</code> .... 13	<code>\ifcgr@raddto@groupsecond</code> .... 314, 330, 340	<code>\relax</code> ..... 15
<b>D</b>	<code>\ifcgr@requires@graphics</code> ..... 167, 198	<code>\RequirePackage</code> ... 1–3
<code>\DeclareOption</code> .... 12	<code>\ifcgr@vecfirst</code> . 4, 199	<code>\resizebox</code> ..... . 109, 122, 136, 145
<code>\define@key</code> .. 5, 16, 310	<code>\ifx</code> 20, 32, 57, 77, 90, 103, 116, 129, 130, 152–154, 241	<code>\rotatebox</code> ..... 58, 62
<b>E</b>	<code>\includecombinedgraphics</code> ..... 3, 185	<b>S</b>
<code>\end</code> ..... 202	<code>\includegraphics</code> .. . 44, 171, 189, 249	<code>\scalebox</code> ... 73, 83, 96
<code>\endpicture</code> ... 205, 292	<code>\input</code> ..... 192, 213	<code>\setbox</code> ..... 188
<code>\expandafter</code> ..... . 13, 306, 323, 325	<b>L</b>	<code>\SetFigFont</code> ..... 25
<b>F</b>	<code>\LaTeX</code> ..... 212	<code>\SetFigFontNFSS</code> ... 26
<code>\fi</code> ..... 28, 39, 64, 86, 99, 112, 125, 139, 149, 157, 160, 164, 210, 211, 247, 328, 338, 348, 349	<code>\let</code> .. 239, 249, 291, 292	<code>\setkeys</code> .... 6, 17, 311
	<code>\long</code> . 166, 235–238, 293	<code>\setlength</code> 43, 239, 244
	<b>N</b>	<b>T</b>
	<code>\newcommand</code> ..... 5, 6, 16, 17, 185, 305, 310, 311, 320	<code>\the</code> ..... 307, 332, 336, 342, 346
		<code>\toks@</code> ..... 306, 307
	<b>U</b>	
		<code>\unitlength</code> ..... 241