

The picture package

Heiko Oberdiek
<oberdiek@uni-freiburg.de>

2008/11/26 v1.2

Abstract

There are macro and environment arguments that expect numbers that will internally be multiplied with `\unitlength`. This package extends the syntax of these arguments that `dimens` with calculation support can be added for these arguments.

Contents

1	User interface	2
1.1	Introduction	2
1.2	Options	2
1.3	Example	2
1.4	Supported packages	2
2	Implementation	3
2.1	Identification	3
2.2	Options	3
2.3	Calculation method	3
2.3.1	Method <code>calc</code>	3
2.3.2	Method <code>etex</code>	4
2.3.3	Method <code>plain</code>	4
2.3.4	Help macros	4
2.4	Redefinitions	5
2.4.1	<code>L^AT_EX</code> base macros	6
2.4.2	Package <code>pspicture</code>	6
2.5	Check package loading order	6
3	Installation	6
3.1	Download	6
3.2	Bundle installation	7
3.3	Package installation	7
3.4	Refresh file name databases	7
3.5	Some details for the interested	7
4	History	8
	[2006/08/26 v1.0]	8
	[2007/04/11 v1.1]	8
	[2008/11/26 v1.2]	8
5	Index	8

1 User interface

1.1 Introduction

The environment `picture` and macros such as `\put`, `\line`, `\vector` and other macros have arguments that expect numbers that are used as factor for `\unitlength`. This package redefines such macros and adds code that detects whether such an argument is given as number or as length. In the latter case, the length is used directly without multiplying with `\unitlength`.

1.2 Options

Depending on the available features, also length expressions can be given. Option `calc` loads package `calc`. Then expressions of these package may be used. Otherwise `etex` wraps the length argument inside `\dimexpr...\relax`, if ϵ -TeX is available. Otherwise option `plain` uses plain assignments without calculation support.

The default is `calc` if package `calc` is loaded before package `picture`. If you specify option `calc` the loading of `calc` is ensured. Otherwise package `picture` looks whether `\dimexpr` is available and uses then option `etex` as default. If ϵ -TeX also could not be found, then `plain` is used.

1.3 Example

```
1 (*example)
2 \documentclass{article}
3
4 \usepackage[calc]{picture}
5
6 \begin{document}
7
8 \setlength{\unitlength}{1pt}
9
10 \begin{picture}(\widthof{Hello World}, 10mm)
11   \put(0, 0){\makebox(0,0)[lb]{Hello World}}%
12   \put(0, \heightof{Hello World} + \fboxsep){%
13     \line(1, 0){\widthof{Hello World}}}%
14   }%
15   \put(\widthof{Hello World}, 10mm){%
16     \line(0, -1){10mm}}%
17   }%
18 \end{picture}
19
20 \end{document}
21 \end{example}
```

1.4 Supported packages

Packages `pspicture` and `pict2e` are supported, but they must be loaded before package `picture`.

New macros can be supported by `\picture@redefine`. The first argument is the macro which contains the arguments in its parameter text that you want to support by package `picture`. The second argument contains the parameter text. Change `#` to `&` for the arguments in question. Examples (already used by package `picture`):

```
\picture@redefine\put{(&1,&2)}
\picture@redefine\line{(#1,#2)&3}
```

2 Implementation

2.1 Identification

```
22 (*package)
23 \NeedsTeXFormat{LaTeX2e}
24 \ProvidesPackage{picture}%
25 [2008/11/26 v1.2 Dimens for picture macros (HO)]%
```

2.2 Options

```
26 \def\Pc@calcname{calc}
27 \def\Pc@etexname{etex}
28 \def\Pc@plainname{plain}

\Pc@method Macro \Pc@method stores the method to use for calculations. Check which features
are available and set the default for \Pc@method.

29 \@ifpackageloaded{calc}{%
30   \let\Pc@method\Pc@calcname
31 }{%
32   \begingroup\expandafter\expandafter\expandafter\endgroup
33   \expandafter\ifx\csname dimexpr\endcsname\relax
34     \let\Pc@method\Pc@plainname
35   \else
36     \let\Pc@method\Pc@etexname
37   \fi
38 }

39 \DeclareOption{plain}{%
40   \let\Pc@method\Pc@plainname
41 }
42 \DeclareOption{etex}{%
43   \begingroup\expandafter\expandafter\expandafter\endgroup
44   \expandafter\ifx\csname dimexpr\endcsname\relax
45     \PackageError{picture}{%
46       e-TeX is not available%
47     }\@ehc
48   \else
49     \let\Pc@method\Pc@etexname
50   \fi
51 }
52 \DeclareOption{calc}{%
53   \let\Pc@method\Pc@calcname
54 }
55 \ProcessOptions*
56 \begingroup
57   \let\on@line\@empty
58   \PackageInfo{picture}{Calculation method: \Pc@method}%
59 \endgroup
```

2.3 Calculation method

```
60 \ifx\Pc@method\Pc@calcname
61   \RequirePackage{calc}%
62 \fi
```

2.3.1 Method calc

```
63 \ifx\Pc@method\Pc@calcname
64   \def\Pc@tokslength#1{%
65     \begingroup
66       \let\calc@error\Pc@calc@error
67       \setlength\dimen@{#1\unitlength}\Pc@next\Pc@nil{#1}%
68     }%
69   \let\Pc@org@calc@error\calc@error
```

```

70 \def\Pc@calc@error#1{%
71   \expandafter\ifx\expandafter\unitlength\noexpand#1\relax
72     \def\calc@next##1!{%
73       \endgroup
74       \aftergroup\afterassignment
75       \aftergroup\Pc@next
76     }%
77   \else
78     \@ReturnAfterFi{%
79       \PcOrg@calc@error{#1}%
80     }%
81   \fi
82 }%
83 \expandafter\@firstofone
84 \else
85   \expandafter\@gobble
86 \fi
87 {%
88   \long\def\@ReturnAfterFi#1\fi{\fi#1}%
89 }

```

2.3.2 Method etex

```

90 \ifx\Pc@method\Pc@etexname
91   \def\Pc@tokslength#1{%
92     \begingroup
93     \afterassignment\Pc@next
94     \dimen@=\dimexpr#1\unitlength\Pc@nil{#1}%
95   }%
96 \fi

```

2.3.3 Method plain

```

97 \ifx\Pc@method\Pc@plainname
98   \def\Pc@tokslength#1{%
99     \begingroup
100    \afterassignment\Pc@next
101    \dimen@=#1\unitlength\Pc@nil{#1}%
102  }%
103 \fi

```

2.3.4 Help macros

```

104 \def\Pc@next#1\Pc@nil#2{%
105   \ifx\#1\%
106     \endgroup
107     \Pc@addtoks{{#2}}%
108   \else
109     \expandafter\endgroup
110     \expandafter\Pc@addtoks\expandafter{%
111       \expandafter{\the\dimen@\@gobble}%
112     }%
113   \fi
114 }

```

\Pc@nil \Pc@nil must not have the meaning of \relax because of \dimexpr.

```

115 \let\Pc@nil\message

```

\Pc@addtoks

```

116 \def\Pc@addtoks#1{%
117   \toks@=\expandafter{\the\toks@#1}%
118 }

```

\Pc@init

```

119 \def\Pc@init#1{%

```

```

120 \begingroup
121   \toks@={#1}%
122 }

```

`\Pc@finish`

```

123 \def\Pc@finish#1{%
124   \expandafter\endgroup
125   \expandafter#1\the\toks@
126 }

```

2.4 Redefinitions

`\picture@redefine` #1: command name
 #2: parameter text, length parameter with & instead of #

```

127 \def\picture@redefine#1#2{%
128   \begingroup
129   \edef\reserved@a{%
130     \noexpand\noexpand
131     \expandafter\noexpand
132     \csname PcOrg@\expandafter\@gobble\string#1\endcsname
133   }%
134   \toks0={#1}%
135   \Pc@first#2&0%
136 }

```

`\Pc@first`

```

137 \def\Pc@first#1&{%
138   \toks1={#1}%
139   \toks2={\Pc@init{#1}}%
140   \Pc@scanlength
141 }

```

`\Pc@scanlength` #1: number of length parameter or zero

```

142 \def\Pc@scanlength#1{%
143   \ifcase#1 %
144     \expandafter\Pc@last
145   \else
146     \toks1=\expandafter{\the\toks1 ###1}%
147     \toks2=\expandafter{\the\toks2 \Pc@tokslength{###1}}%
148     \expandafter\Pc@scannext
149   \fi
150 }

```

`\Pc@scannext`

```

151 \def\Pc@scannext#1&{%
152   \ifx\#1\\%
153   \else
154     \toks1=\expandafter{\the\toks1 #1}%
155     \toks2=\expandafter{\the\toks2 \Pc@addtoks{#1}}%
156   \fi
157   \Pc@scanlength
158 }

```

`\Pc@last`

```

159 \def\Pc@last{%
160   \edef\x{%
161     \endgroup
162     \let\reserved@a\the\toks0 %
163     \def\the\toks0 \the\toks1 {%
164       \the\toks2 %
165       \noexpand\Pc@finish\reserved@a

```

```

166     }%
167 }%
168 \x
169 }

```

2.4.1 L^AT_EX base macros

```

170 \picture@redefine\@picture{(&1,&2)(&3,&4)}
171 \picture@redefine\put{(&1,&2)}
172 \picture@redefine\multiput{(&1,&2)}
173 \picture@redefine\@multiput{(&1,&2)}
174 \picture@redefine\line{(#1,#2)&3}
175 \picture@redefine\vector{(#1,#2)&3}
176 \picture@redefine\dashbox{&1(&2,&3)}
177 \picture@redefine\@circle{&1}
178 \picture@redefine\@dot{&1}
179 \picture@redefine\@bezier{#1(&2,&3)(&4,&5)(&6,&7)}
180 \picture@redefine\@makepicbox{(&1,&2)}

```

2.4.2 Package pspicture

Package `pspicture` changes the signature of `\@oval` by adding an optional argument.

```

181 \ifpackageloaded{pspicture}{%
182   \picture@redefine\@oval{[&1](&2,&3)}%
183   \picture@redefine\Line{(&1,&2)}%
184   \picture@redefine\Curve{(&1,&2)}%
185   \picture@redefine\Vector{(&1,&2)}%
186 }{%
187   \picture@redefine\@oval{(&1,&2)}%
188 }

```

2.5 Check package loading order

`\PC@checkpackage`

```

189 \def\PC@checkpackage#1{%
190   \@ifpackageloaded{#1}{%
191     }{%
192       \AtBeginDocument{%
193         \@ifpackageloaded{#1}{%
194           \PackageWarningNoLine{picture}{%
195             Package ‘#1’ is loaded after ‘picture’.\MessageBreak
196             Load package ‘picture’ afterwards to get full support%
197             \MessageBreak
198             of its additional syntax with length specifications%
199           }%
200         }{}%
201       }%
202     }%
203   }

204 \PC@checkpackage{pict2e}
205 \PC@checkpackage{pspicture}
206 </package>

```

3 Installation

3.1 Download

Package. This package is available on CTAN¹:

¹<http://ftp.ctan.org/tex-archive/>

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

`CTAN:macros/latex/contrib/oberdiek/picture.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 T_EX Files” (`CTAN:tds/tds.pdf`). Directories with `texmf` in their name are usually organized this way.

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
```

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/
```

3.3 Package installation

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

```
tex picture.dtx
```

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

<code>picture.sty</code>	→ <code>tex/latex/oberdiek/picture.sty</code>
<code>picture.pdf</code>	→ <code>doc/latex/oberdiek/picture.pdf</code>
<code>picture-example.tex</code>	→ <code>doc/latex/oberdiek/picture-example.tex</code>
<code>picture.dtx</code>	→ <code>source/latex/oberdiek/picture.dtx</code>

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 T_EX distribution (teT_EX, mikT_EX, ...) relies on file name databases, you must refresh these. For example, teT_EX users run `texhash` or `mktextlsr`.

3.5 Some details for the interested

Attached source. The PDF documentation on CTAN also includes the `.dtx` source file. It can be extracted by AcrobatReader 6 or higher. Another option is `pdftk`, e.g. unpack the file into the current directory:

```
pdftk picture.pdf unpack_files output .
```

Unpacking with L^AT_EX. The .dtx chooses its action depending on the format:

plain-T_EX: Run docstrip and extract the files.

L^AT_EX: Generate the documentation.

If you insist on using L^AT_EX for docstrip (really, docstrip does not need L^AT_EX), then inform the autodetect routine about your intention:

```
latex \let\install=y\input{picture.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 pdfL^AT_EX:

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

4 History

[2006/08/26 v1.0]

- First released version. (First start of the project was June/July 2002.)

[2007/04/11 v1.1]

- Line ends sanitized.

[2008/11/26 v1.2]

- Package pict2e added to documentation section “Supported packages”.
- Package order of supported packages is checked.

5 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.

Symbols		
\@ReturnAfterFi	78, 88	\@multiput 173
\@bezier	179	\@oval 182, 187
\@circle	177	\@picture 170
\@dot	178	\\ 105, 152
\@ehc	47	
\@empty	57	A
\@firstofone	83	\afterassignment 74, 93, 100
\@gobble	85, 111, 132	\aftergroup 74, 75
\@ifpackageloaded	29, 181, 190, 193	\AtBeginDocument 192
\@imakepicbox	180	
		B
		\begin 6, 10

C		<code>\PC@checkpackage</code> 189	
<code>\calc@error</code> 66, 69		<code>\Pc@checkpackage</code> 189, 204, 205	
<code>\calc@next</code> 72		<code>\Pc@etexname</code> 27, 36, 49, 90	
<code>\csname</code> 33, 44, 132		<code>\Pc@finish</code> 123, 165	
<code>\Curve</code> 184		<code>\Pc@first</code> 135, 137	
D		<code>\Pc@init</code> 119, 139	
<code>\dashbox</code> 176		<code>\Pc@last</code> 144, 159	
<code>\DeclareOption</code> 39, 42, 52		<code>\Pc@method</code>	
<code>\dimen@</code> 67, 94, 101, 111	 29, 40, 49, 53, 58, 60, 63, 90, 97	
<code>\dimexpr</code> 94		<code>\Pc@next</code> 67, 75, 93, 100, 104	
<code>\documentclass</code> 2		<code>\Pc@nil</code> 67, 94, 101, 104, 115	
E		<code>\Pc@plainname</code> 28, 34, 40, 97	
<code>\end</code> 18, 20		<code>\Pc@scanlength</code> 140, 142, 157	
<code>\endcsname</code> 33, 44, 132		<code>\Pc@scannext</code> 148, 151	
F		<code>\Pc@tokslength</code> 64, 91, 98, 147	
<code>\fboxsep</code> 12		<code>\PcOrg@calc@error</code> 69, 79	
H		<code>\picture@redefine</code> 127, 170, 171, 172,	
<code>\heightof</code> 12		173, 174, 175, 176, 177, 178,	
I		179, 180, 182, 183, 184, 185, 187	
<code>\ifcase</code> 143		<code>\ProcessOptions</code> 55	
<code>\ifx</code> . 33, 44, 60, 63, 71, 90, 97, 105, 152		<code>\ProvidesPackage</code> 24	
L		<code>\put</code> 11, 12, 15, 171	
<code>\Line</code> 183		R	
<code>\line</code> 13, 16, 174		<code>\RequirePackage</code> 61	
M		<code>\reserved@a</code> 129, 162, 165	
<code>\makebox</code> 11		S	
<code>\message</code> 115		<code>\setlength</code> 8, 67	
<code>\MessageBreak</code> 195, 197		T	
<code>\multiput</code> 172		<code>\the</code> 111, 117, 125,	
N		146, 147, 154, 155, 162, 163, 164	
<code>\NeedsTeXFormat</code> 23		<code>\toks</code> 134, 138, 139,	
O		146, 147, 154, 155, 162, 163, 164	
<code>\on@line</code> 57		<code>\toks@</code> 117, 121, 125	
P		U	
<code>\PackageError</code> 45		<code>\unitlength</code> 8, 67, 71, 94, 101	
<code>\PackageInfo</code> 58		<code>\usepackage</code> 4	
<code>\PackageWarningNoLine</code> 194		V	
<code>\Pc@addtoks</code> 107, 110, 116, 155		<code>\Vector</code> 185	
<code>\Pc@calc@error</code> 66, 70		<code>\vector</code> 175	
<code>\Pc@calcname</code> 26, 30, 53, 60, 63		W	
X		<code>\widthof</code> 10, 13, 15	
		<code>\x</code> 160, 168	