

# The kvdefinekeys package

Heiko Oberdiek  
<heiko.oberdiek at gmail.com>

2010/03/01 v1.0

## Abstract

Package kvdefinekeys provides `\kv@define@key` to define keys the same way as keyval's `\define@key`. However, it works also using ini-TeX.

## Contents

<b>1</b>	<b>Documentation</b>	<b>1</b>
1.1	Motivation . . . . .	1
<b>2</b>	<b>Implementation</b>	<b>2</b>
2.1	Identification . . . . .	2
2.2	Package loading . . . . .	3
2.3	Provide key defining macro . . . . .	3
<b>3</b>	<b>Test</b>	<b>4</b>
3.1	Catcode checks for loading . . . . .	4
<b>4</b>	<b>Installation</b>	<b>5</b>
4.1	Download . . . . .	5
4.2	Bundle installation . . . . .	5
4.3	Package installation . . . . .	5
4.4	Refresh file name databases . . . . .	6
4.5	Some details for the interested . . . . .	6
<b>5</b>	<b>References</b>	<b>6</b>
<b>6</b>	<b>History</b>	<b>6</b>
	[2010/03/01 v1.0] . . . . .	6
<b>7</b>	<b>Index</b>	<b>7</b>

## 1 Documentation

### 1.1 Motivation

`\kvsetkeys` serves as replacement for keyval's `\setkeys`. This package adds macros to define keys, closing the gap `\kvsetkeys` leaves.

`\kv@define@key {<family>} {<key>},[<default>]`

Macro `\kv@define@key` reimplements keyval's `\define@key`. As difference to the original the defined keys also allow `\par` inside values.

## 2 Implementation

### 2.1 Identification

```
1 (*package)
```

Reload check, especially if the package is not used with L<sup>A</sup>T<sub>E</sub>X.

```
2 \begingroup
3  \catcode44 12 % ,
4  \catcode45 12 % -
5  \catcode46 12 % .
6  \catcode58 12 % :
7  \catcode64 11 % @
8  \catcode123 1 % {
9  \catcode125 2 % }
10 \expandafter\let\expandafter\x\csname ver@kvdefinekeys.sty\endcsname
11 \ifx\x\relax % plain-TeX, first loading
12 \else
13   \def\empty{}%
14   \ifx\x\empty % LaTeX, first loading,
15     % variable is initialized, but \ProvidesPackage not yet seen
16   \else
17     \catcode35 6 % #
18     \expandafter\ifx\csname PackageInfo\endcsname\relax
19       \def\x#1#2{%
20         \immediate\write-1{Package #1 Info: #2.}%
21       }%
22     \else
23       \def\x#1#2{\PackageInfo{#1}{#2, stopped}}%
24     \fi
25     \x{kvdefinekeys}{The package is already loaded}%
26     \aftergroup\endinput
27   \fi
28 \fi
29 \endgroup
```

Package identification:

```
30 \begingroup
31  \catcode35 6 % #
32  \catcode40 12 % (
33  \catcode41 12 % )
34  \catcode44 12 % ,
35  \catcode45 12 % -
36  \catcode46 12 % .
37  \catcode47 12 % /
38  \catcode58 12 % :
39  \catcode64 11 % @
40  \catcode91 12 % [
41  \catcode93 12 % ]
42  \catcode123 1 % {
43  \catcode125 2 % }
44  \expandafter\ifx\csname ProvidesPackage\endcsname\relax
45    \def\x#1#2#3[#4]{\endgroup
46      \immediate\write-1{Package: #3 #4}%
47      \xdef#1{#4}%
48    }%
49  \else
50    \def\x#1#2[#3]{\endgroup
51      #2[#{#3}]%
52      \ifx#1\@undefined
53        \xdef#1{#3}%
54      \fi
55      \ifx#1\relax
56        \xdef#1{#3}%

```

```

57     \fi
58   }%
59 \fi
60 \expandafter\x\csname ver@kvdefinekeys.sty\endcsname
61 \ProvidesPackage{kvdefinekeys}%
62 [2010/03/01 v1.0 Defining keys (H0)]
63 \begingroup
64   \catcode123 1 % {
65   \catcode125 2 % }
66   \def\x{\endgroup
67     \expandafter\edef\csname KVD@AtEnd\endcsname{%
68       \catcode35 \the\catcode35\relax
69       \catcode64 \the\catcode64\relax
70       \catcode123 \the\catcode123\relax
71       \catcode125 \the\catcode125\relax
72     }%
73   }%
74 \x
75 \catcode35 6 % #
76 \catcode64 11 % @
77 \catcode123 1 % {
78 \catcode125 2 % }
79 \def\TMP@EnsureCode#1#2{%
80   \edef\KVD@AtEnd{%
81     \KVD@AtEnd
82     \catcode#1 \the\catcode#1\relax
83   }%
84   \catcode#1 #2\relax
85 }
86 \TMP@EnsureCode{46}{12}% .
87 \TMP@EnsureCode{47}{12}% /

```

## 2.2 Package loading

```

88 \begingroup\expandafter\expandafter\expandafter\endgroup
89 \expandafter\ifx\csname RequirePackage\endcsname\relax
90   \input ltxcmds.sty\relax
91 \else
92   \RequirePackage{ltxcmds}[2010/03/01]%
93 \fi

```

## 2.3 Provide key defining macro

`\kv@define@key`

```

94 \def\kv@define@key#1#2{%
95   \ltx@ifnextchar[{%
96     \KVD@DefineKeyWithDefault{#1}{#2}%
97   }{%
98     \long\expandafter\def\csname KV@#1@#2\endcsname##1%
99     }%
100 }

```

`\KVD@DefineKeyWithDefault`

```

101 \long\def\KVD@DefineKeyWithDefault#1#2[#3]{%
102   \expandafter\def\csname KV@#1@#2@default\endcsname
103   \expandafter{%
104     \csname KV@#1@#2\endcsname{#3}%
105   }%
106   \long\expandafter\def\csname KV@#1@#2\endcsname##1%
107 }

108 \KVD@AtEnd
109 \endpackage

```

## 3 Test

### 3.1 Catcode checks for loading

```
110 <*test1>
111 \catcode'\{=1 %
112 \catcode'\}=2 %
113 \catcode'\#=6 %
114 \catcode'\@=11 %
115 \expandafter\ifx\csname count@\endcsname\relax
116   \countdef\count@=255 %
117 \fi
118 \expandafter\ifx\csname @gobble\endcsname\relax
119   \long\def\@gobble#1{}%
120 \fi
121 \expandafter\ifx\csname @firstofone\endcsname\relax
122   \long\def\@firstofone#1{#1}%
123 \fi
124 \expandafter\ifx\csname loop\endcsname\relax
125   \expandafter\@firstofone
126 \else
127   \expandafter\@gobble
128 \fi
129 {%
130   \def\loop#1\repeat{%
131     \def\body{#1}%
132     \iterate
133   }%
134   \def\iterate{%
135     \body
136     \let\next\iterate
137   \else
138     \let\next\relax
139   \fi
140   \next
141 }%
142 \let\repeat=\fi
143 }%
144 \def\RestoreCatcodes{}
145 \count@=0 %
146 \loop
147   \edef\RestoreCatcodes{%
148     \RestoreCatcodes
149     \catcode\the\count@=\the\catcode\count@\relax
150   }%
151 \ifnum\count@<255 %
152   \advance\count@ 1 %
153 \repeat
154
155 \def\RangeCatcodeInvalid#1#2{%
156   \count@=#1\relax
157   \loop
158     \catcode\count@=15 %
159   \ifnum\count@<#2\relax
160     \advance\count@ 1 %
161   \repeat
162 }
163 \expandafter\ifx\csname LoadCommand\endcsname\relax
164   \def\LoadCommand{\input kvdefinekeys.sty\relax}%
165 \fi
166 \def\Test{%
167   \RangeCatcodeInvalid{0}{47}%
```

```

168 \RangeCatcodeInvalid{58}{64}%
169 \RangeCatcodeInvalid{91}{96}%
170 \RangeCatcodeInvalid{123}{255}%
171 \catcode'\@=12 %
172 \catcode'\=0 %
173 \catcode'\{=1 %
174 \catcode'\}=2 %
175 \catcode'\#=6 %
176 \catcode'\[=12 %
177 \catcode'\]=12 %
178 \catcode'\%=14 %
179 \catcode'\ =10 %
180 \catcode13=5 %
181 \LoadCommand
182 \RestoreCatcodes
183 }
184 \Test
185 \csname @@end\endcsname
186 \end
187 </test1>

```

## 4 Installation

### 4.1 Download

**Package.** This package is available on CTAN<sup>1</sup>:

[CTAN:macros/latex/contrib/oberdiek/kvdefinekeys.dtx](#) The source file.

[CTAN:macros/latex/contrib/oberdiek/kvdefinekeys.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: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
```

**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 kvdefinekeys.dtx
```

---

<sup>1</sup><http://ftp.ctan.org/tex-archive/>

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

```
kvdefinekeys.sty      → tex/generic/oberdiek/kvdefinekeys.sty
kvdefinekeys.pdf      → doc/latex/oberdiek/kvdefinekeys.pdf
test/kvdefinekeys-test1.tex → doc/latex/oberdiek/test/kvdefinekeys-test1.tex
kvdefinekeys.dtx      → source/latex/oberdiek/kvdefinekeys.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 (`teTeX`, `mikTeX`, ...) relies on file name databases, you must refresh these. For example, `teTeX` users run `texhash` or `mktextlsr`.

## 4.5 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 kvdefinekeys.pdf unpack_files output .
```

**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{kvdefinekeys.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 `pdf $\LaTeX$` :

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

## 5 References

- [1] David Carlisle: *The keyval package*; 1999/03/16 v1.13; [CTAN:macros/latex/required/graphics/keyval.dtx](#).

## 6 History

[2010/03/01 v1.0]

- First version.

## 7 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		
<code>\#</code>	113, 175	<code>\endinput</code> 26
<code>\%</code>	178	
<code>\@</code>	114, 171	<b>I</b>
<code>\@firstofone</code>	122, 125	<code>\ifnum</code> 151, 159
<code>\@gobble</code>	119, 127	<code>\ifx</code> 11, 14, 18, 44, 52, 55, 89, 115, 118, 121, 124, 163
<code>\@undefined</code>	52	<code>\immediate</code> 20, 46
<code>\[</code>	176	<code>\input</code> 90, 164
<code>\]</code>	172	<code>\iterate</code> 132, 134, 136
<code>\{</code>	111, 173	
<code>\}</code>	112, 174	<b>K</b>
<code>\]</code>	177	<code>\kv@define@key</code> 1, 94
		<code>\KVD@AtEnd</code> 80, 81, 108
		<code>\KVD@DefineKeyWithDefault</code> 96, 101
<code>\_</code>	179	
<b>A</b>		<b>L</b>
<code>\advance</code>	152, 160	<code>\LoadCommand</code> 164, 181
<code>\aftergroup</code>	26	<code>\loop</code> 130, 146, 157
		<code>\ltx@ifnextchar</code> 95
<b>B</b>		<b>N</b>
<code>\body</code>	131, 135	<code>\next</code> 136, 138, 140
<b>C</b>		<b>P</b>
<code>\catcode</code> 3, 4, 5, 6, 7, 8, 9, 17, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 64, 65, 68, 69, 70, 71, 75, 76, 77, 78, 82, 84, 111, 112, 113, 114, 149, 158, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180		<code>\PackageInfo</code> 23
<code>\count@</code> 116, 145, 149, 151, 152, 156, 158, 159, 160		<code>\ProvidesPackage</code> 15, 61
<code>\countdef</code> 116		
<code>\csname</code> 10, 18, 44, 60, 67, 89, 98, 102, 104, 106, 115, 118, 121, 124, 163, 185		<b>R</b>
		<code>\RangeCatcodeInvalid</code> 155, 167, 168, 169, 170
<b>E</b>		<code>\repeat</code> 130, 142, 153, 161
<code>\empty</code> 13, 14		<code>\RequirePackage</code> 92
<code>\end</code> 186		<code>\RestoreCatcodes</code> 144, 147, 148, 182
<code>\endcsname</code> 10, 18, 44, 60, 67, 89, 98, 102, 104, 106, 115, 118, 121, 124, 163, 185		<b>T</b>
		<code>\Test</code> 166, 184
		<code>\the</code> 68, 69, 70, 71, 82, 149
		<code>\TMP@EnsureCode</code> 79, 86, 87
		<b>W</b>
		<code>\write</code> 20, 46
		<b>X</b>
		<code>\x</code> 10, 11, 14, 19, 23, 25, 45, 50, 60, 66, 74