

This package may be distributed and/or modified under the conditions of the L<sup>A</sup>T<sub>E</sub>X Project Public License (LPPL), either version 1.3 of this license or any later version. The LPPL maintenance status of this software is ‘author-maintained.’ This software is provided ‘as it is,’ without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. © MMXI

# The `keyval2e` Package<sup>☆</sup>

Robust and fast key parser

Ahmed Musa ✉

Preston, Lancashire, UK

15th August 2011

## CONTENTS

<b>1 Introduction</b>	<b>1</b>	<b>4 Examples</b>	<b>3</b>
<b>2 Package options</b>	<b>2</b>	<b>5 Version history</b>	<b>5</b>
<b>3 User commands</b>	<b>2</b>	<b>Index</b>	<b>6</b>
3.1 Utility macros . . . . .	3		

## 1 INTRODUCTION

The `keyval2e` package provides lightweight and robust facilities for creating and managing keys. Its machinery isn’t as extensive as that of, e.g., `ltxkeys` package but it is equally robust. Ease of use and speed of processing are the two main motives of this package. Some, indeed many, applications of the key-value syntax (while they call for robustness) don’t require the full armor of key-value processing as found in, e.g., the `ltxkeys` package. This package was prompted by a subscriber’s post on [comp.text.tex](http://comp.text.tex) in August 2011.

In the `keyval2e` package, ordinary, boolean, and choice keys can be created using only one command (`\kve@definekeys`). Keys can be initialized with their default values (with the command `\kve@setdefaults`) as soon as they are created, or at any time. And in any run the default values of keys can be used to set keys that have no current values. The latter task is accomplished by the command `\kve@setafterdefaults`, meaning ‘set keys with the current values after those without current values have been initialized/set with their default values.’ Normally, keys are set with the re-entrant command `\kve@setkeys`.

The `keyval2e` package has no service for processing package or class options. See the `ltxkeys` package for this service. I have seen users who require the services of keys only in documents, and not in package or class files. And some package authors still use L<sup>A</sup>T<sub>E</sub>X’s native option processing

---

<sup>☆</sup> The package is available at <http://www.ctan.org/tex-archive/macros/latex/contrib/keyval2e/>. This user manual corresponds to version 0.0.1a.

schemes. For those authors, the `keyval2e` package may be used to process keys (but not options) in package and class files.

The `keyval` package provides a simple and widely used interface, but it is not robust, in the sense that it strips off outer curly braces in key values. Also, it has no means to automatically call up default key values after the keys have been defined. Moreover, it automatically redefines existing keys.

## 2 PACKAGE OPTIONS

The `keyval2e` package currently has no options.

## 3 USER COMMANDS

New macros: `\kve@definekeys`, `\kve@setkeys`, etc

```

1  \kve@definekeys[\langle pref \rangle]{\langle fam \rangle}[\langle mp \rangle]{%
2  \langle key-1 \rangle/\langle defa-1 \rangle/\langle callback-1 \rangle,
3  \dots,
4  \langle key-n \rangle/\langle defa-n \rangle/\langle callback-n \rangle
5  }
6  \kve@definekeys*[\langle pref \rangle]{\langle fam \rangle}[\langle mp \rangle]{%
7  \langle key-1 \rangle/\langle defa-1 \rangle/\langle callback-1 \rangle,
8  \dots,
9  \langle key-n \rangle/\langle defa-n \rangle/\langle callback-n \rangle
10 }
11 \kve@setkeys[\langle pref \rangle]{\langle fam \rangle}[\langle na \rangle]{\langle keyval \rangle}
12 \kve@setdefaults[\langle pref \rangle]{\langle fam \rangle}[\langle na \rangle]
13 \kve@setafterdefaults[\langle pref \rangle]{\langle fam \rangle}[\langle na \rangle]{\langle keyval \rangle}

```

Here, `\langle pref \rangle` is the optional key prefix (default `KV`), `\langle fam \rangle` is the mandatory family, `\langle mp \rangle` is the key-value-holding macro prefix (default `kvmp@`), `\langle defa-i \rangle` is the default value of key ‘`i`’, `\langle callback-i \rangle` is the callback (i. e., the function that will be executed when the key is set) of key ‘`i`’, and `\langle keyal \rangle` is a list of `\langle key \rangle=\langle value \rangle` pairs.

`\langle na \rangle` is a comma-separated list of keys that should be ignored, ie, not set in the current run of setting keys\*. `\langle mp \rangle@key` will hold the current value of `\langle key \rangle`. The key macro (i. e., the one that holds the key’s callback) is always `\langle pref \rangle@key`.

The starred (\*) variant of the command `\kve@definekeys` will define only definable keys, in the sense of L<sup>A</sup>T<sub>E</sub>X’s `\newcommand`. In that case the commands `\langle mp \rangle@key` must also be unique, i. e., not previously defined. The plain form `\kve@definekeys` will always define the key, whether or not the key already exists; existing keys will thus be overwritten in this case.

\*When setting keys, undefined keys are reported by the `keyval2e` package as undefined and are not saved as ‘remaining keys’. Moreover, there are no ‘undefined key handlers’ and no ‘handled keys’ in this package. Please see the `ltxkeys` package for these facilities.

You can use `#1` in `<callback>` to access the value of the current key. Also the macros `\currpref`, `\currfam`, `\currkey`, `\currval` and `\currkeyval` are always available when setting keys and may be called in `<callback>`.

The command `\kve@setdefaults` will set all the keys in the given family `<fam>` and prefix `<pref>` with their default values. All boolean keys (i. e., those with a default in the set `{true | false}`) will be initialized with a default value of `false`. This is to avoid premature toggling of the state of such keys. The command `\kve@setwithdefaults` is an alias for `\kve@setdefaults`.

**Note 3.1** After the keys have been defined, they are automatically set with their default values using the command `\kve@setdefaults`. This provides default definitions for immediate use.

The command `\kve@setafterdefaults` will set the given `<key>=<value>` pairs after initializing to default values all those keys (in the given family and prefix) that are not listed in the accompanying `<key>=<value>` list. This provides a mechanism for (re)initializing to default values those keys that don't have values in `<key>=<value>`. This type of (re)initialization is often required in the deployment of keys, and it is useful to have a handy way of accomplishing this semi-automatically.

### 3.1 Utility macros

The following macros are utilities.

New macro: `\kve@checkchoice`

```
14 \kve@checkchoice{<teststring>}{<nominations>}{<nomatch>}
```

The `<nominations>` have the syntax

Nominations and callbacks

```
15 <nom-1>:<callback-1>, ..., <nom-n>:<callback-n>
```

Here, please note the colon `:`, which separates `<nom>` from `<callback>`. `<callback-i>` will be executed if `<nom-i>` matches `<teststring>`. The first match found takes priority over subsequent matches. The fallback `<nomatch>` will be executed if `<teststring>` doesn't match any of the `<nom>`'s.

New macro: `\kve@checkbool`

```
16 \kve@checkbool{<val>}{<true>}{<false>}
```

This checks if `<val>` is an admissible value of a boolean, namely, if it is in the set `{true | false}`. If `<val>` is valid, the text `<true>` will be executed; otherwise `<false>` will be executed.

New macro: `\kve@keyvalerr`

```
17 \kve@keyvalerr
```

This is a parameterless command that uses `\currkey` and `\currval` internally. It simply generates an error to indicate that the current value of a key is invalid. It will indicate the key name and the truncated version of the key value that is invalid.

## 4 EXAMPLES

Example: \kve@definekeys

```

18 \kve@definekeys [KV] {fam} [mp@] {%
19 % keya and keyb are boolean keys:
20 keya/true/\kve@checkboxbool{#1}{\def\result{OK}}{\kve@keyvalerr},
21 keyb/false/\kve@checkboxbool{#1}{%
22   \username{@tempswa#1}%
23   \if@tempswa\def\x{found}\else\def\x{not-found}\fi
24 }{%
25   \kve@keyvalerr
26 },
27 % keyc is a choice key:
28 keyc/left/\kve@checkboxchoice{#1}{left:\let\x\flushleft,
29   right:\let\x\flushright}{\kve@keyvalerr},
30 keyd/right/\def\x##1{#1xx##1},
31 % keyone has an empty default value:
32 keyone/\ifx\#1\def\x{no-val}\else\def\x{#1}\fi,
33 % keytwo has no callback:
34 keytwo/+,
35 % keythree has a braced default value:
36 keythree/{left}/\kve@setkeys [KV] {fam}{keyone=#1}
37 }

```

Remember that after the keys have been defined, they are automatically set with their default values using the command `\kve@setdefaults`.

The following command says ‘set the current keys with their current values, after the defaults have been set up’. Keys with current values will not be set with their default values:

Example: \kve@setafterdefaults

```

38 \kve@setafterdefaults [KV] {fam}{keyone=+,keytwo=abc,keythree=+,keyfour=xax}

```

Please see the example file for the fuller version of the following example:

Examples: Creating a key command

```

39 \documentclass{minimal}
40 \usepackage{keyval2e}
41 \makeatletter
42 \kve@definekeys [KV] {fam} [mp@] {%
43   keyone/+,
44   keytwo/+,
45   keythree/+,
46   keyfour/+
47 }
48 \def\fourplus{+,+,+,+}
49 \newcommand{\test}[2]{%
50   \kve@setafterdefaults [KV] {fam}{#2}%
51   \edef\tempa{\mp@keyone,\mp@keytwo,\mp@keythree,\mp@keyfour}%
52   Test #1: *\texttt{\tempa}*%
53   \ifxTF\tempa\fourplus{All values are defaults}{At least one value is set}%
54 }
55 \begin{document}
56 \ttfamily\noindent

```

```
57 \test{A}{}\\  
58 \test{B}{keythree=+}\\  
59 \test{C}{keythree=a}\\  
60 \end{document}
```

## 5 VERSION HISTORY

The star sign (\*) on the right-hand side of the following lists means the subject features in the package but is not reflected anywhere in this user guide.

### Version 0.0.1 [2011/08/13]

First public release. . . . . \*

### Version 0.0.1a [2011/08/14]

Completed the user guide. . . . . \*

## INDEX

Index numbers refer to page numbers.

<b>K</b>			
<code>\kve@checkbool</code> .....	<u>3</u>	<code>\kve@setwithdefaults</code> .....	<i>see</i> <code>\kve@setdefaults</code>
<code>\kve@checkchoice</code> .....	<u>3</u>		
<code>\kve@definekeys</code> .....	<u>2</u>		
<code>\kve@keyvalerr</code> .....	<u>3</u>		
<code>\kve@setafterdefaults</code> .....	<u>2</u>		
<code>\kve@setdefaults</code> .....	<u>2</u>		
<code>\kve@setkeys</code> .....	<u>2</u>		

  

<b>P</b>	
Packages .....	
<code>keyval</code> .....	2
<code>keyval2e</code> .....	1, 2
<code>ltxkeys</code> .....	1, 2