

The `skeycommand` Package[☆]

Version 0.1

Ahmed Musa ✉
Preston, Lancashire, UK

13th May 2010

SUMMARY

The `skeycommand` package provides tools for defining L^AT_EX-style commands and environments using parameters and keys together. The advantages of keys over parameters include the facts that the former aren't limited to nine but can rise as desired by the user, and keys are much easier to match to their values than parameters to arguments, especially if the parameters are many. Moreover, keys can have natural functions. The design approach and user interfaces in the `skeycommand` package differ from those found in the `keycommand` package. This package also provides the `\newtwooptcmd` and `\newtwooptenv` macros for defining new commands and environments with two options/optional arguments. At both key command definition and invocation times there is no reference by the user to the semantics of key parsing and management. All the complex semantics and calculations involved in defining and setting keys are transparent to the user. The user of the `skeycommand` package has access to some (but not all) of the machinery of `xkeyval` and `skeyval` packages (including the pointer mechanism) at the much lesser cost of worrying only about the key names and their values. Native boolean keys are automatically recognized and handled appropriately. However, because of the need to keep the user interface simple, toggle boolean keys aren't available in this package; neither are choice keys.

LICENSE

This work (i.e., all the files in the `skeycommand` bundle) may be distributed and/or modified under the conditions of the L^AT_EX Project Public License (LPPL), either version 1.3 of this license or any later version.

The L^AT_EX Project Public License 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.

© MMX

[☆]The package is available at <http://www.ctan.org/tex-archive/macros/latex/contrib/skeycommand/>.

CONTENTS

1 Package options	2	2.3 Commands and environments with two optional arguments	4
2 User interfaces	2	2.3.1 Calling commands and environments	5
2.1 Defining new key commands and environments	2		
2.2 Invoking new key commands and environments	4	3 Examples	5

1 PACKAGE OPTIONS

The package has only one option (namely, `verbose`) and can be invoked at the time of loading the package or via the `\skeycommand` macro. The option `verbose` is a boolean, initially set to false (i.e., its complement, `silent`, is true by default). Setting `silent` to `false` is tantamount to setting `verbose` to `true`.

Example

```

1 % In style files:
2 \RequirePackage[verbose=true or false]{skeycommand}
3 % In document files:
4 \usepackage[verbose=true or false]{skeycommand}
5 % In all cases:
6 \skeycommand{verbose=true or false or on or off}

```

If you enter the boolean `verbose` (or `silent`) without value, the value is assumed to be `true`. The `verbose` option is simply passed on to the `skeyval` package to log informational messages in the transcript file. The major task of key parsing for the `skeycommand` package is undertaken by the `skeyval` package.

2 USER INTERFACES

2.1 Defining new key commands and environments

The user interfaces for defining new key commands and environments are as follows:

Macro

```

7 \newkeycmd⟨cmd⟩⟨⟨mprefix⟩⟩(⟨keyval⟩) [⟨narg⟩]
8   [⟨default1⟩]{⟨defn⟩}
9 \newkeycmd*⟨cmd⟩⟨⟨mprefix⟩⟩(⟨keyval⟩) [⟨narg⟩]
10   [⟨default1⟩]{⟨defn⟩}

11 \renewkeycmd⟨cmd⟩⟨⟨mprefix⟩⟩(⟨keyval⟩) [⟨narg⟩]
12   [⟨default1⟩]{⟨defn⟩}
13 \renewkeycmd*⟨cmd⟩⟨⟨mprefix⟩⟩(⟨keyval⟩) [⟨narg⟩]
14   [⟨default1⟩]{⟨defn⟩}

```

```

15 \newkeyenviron⟨env⟩⟨mprefix⟩⟨keyval⟩[⟨narg⟩]
16   [⟨default1⟩]{⟨defn⟩}
17 \newkeyenviron*⟨env⟩⟨mprefix⟩⟨keyval⟩[⟨narg⟩]
18   [⟨default1⟩]{⟨defn⟩}

19 \renewkeyenviron⟨env⟩⟨mprefix⟩⟨keyval⟩[⟨narg⟩]
20   [⟨default1⟩]{⟨defn⟩}
21 \renewkeyenviron*⟨env⟩⟨mprefix⟩⟨keyval⟩[⟨narg⟩]
22   [⟨default1⟩]{⟨defn⟩}

```

Here, $\langle narg \rangle$ is the number of user arguments, as usual in `\newcommand` and `\newenvironment` (excluding the $\langle keyval \rangle$ list); $\langle cmd \rangle$ is the new command sequence; $\langle env \rangle$ is the new environment name; $\langle mprefix \rangle$ is the prefix for macros deriving from the defined keys whose values will be used in the new command or environment (this is called the *macro prefix* in the parlance of keys); $\langle keyval \rangle$ is the key-value list [e.g., (`keya=valuea`, `keyb=valueb`)]; $\langle narg \rangle$ is the number of arguments for the new command or environment (excluding the keys), as you would enter it in `\newcommand` and `\newenvironment`; $\langle default1 \rangle$ is the default value for your optional argument (normally the first argument in `\newcommand` and `\newenvironment`); $\langle defn \rangle$ is the replacement text (as in `\newcommand` and `\newenvironment`). If you don't supply the optional $\langle mprefix \rangle$, the package will use the first three letters of the key command or environment name, excluding the escape character but including an added “at sign” (`@`). The aim of default “at sign” is to aid the visual separation of key names from macro prefixes.

Please note the angles surrounding $\langle mprefix \rangle$, and the parenthesis surrounding $\langle keyval \rangle$ in the above syntaxes. The $\langle mprefix \rangle$ can't be empty (i.e., don't enter `<>`) because it will be used by the package to build unique names for the macros that will hold the key values. You can choose not to enter anything for $\langle mprefix \rangle$, and the package will happily use the default prefix `<xxx@>`, where “xxx” represents the first three letters of the new command or environment name. Also, $\langle keyval \rangle$ can't be empty: if it was empty, then we should ask: why use key commands instead of L^AT_EX's `\newcommand` and `\newenvironment`?

In $\langle defn \rangle$, you refer to your arguments in the normal way. You refer to the values of the keys using macros whose first three characters (after the escape character) are the $\langle mprefix \rangle$ or, if $\langle mprefix \rangle$ is not supplied, the first three letters of the declared key command (excluding the escape character). The family name of the keys defined via a key command is the key command name itself (without the escape character). The package uses this internally in developing the keys. The key prefix is always “KV”. If any of your key values contains parenthesis, simply enclose it in braces, to avoid confusing it with $\langle keyval \rangle$ list.

The starred variants (`*`) give “short” macros, while the unstarred variants yield “long” macros.

Note: The optional $\langle mprefix \rangle$ will be useful if you fear clashes with previously defined key commands. Although, to be defined, key commands must be defin-

able, two key commands may have their first three or four characters identical, thereby leading to clashes of their key-value prefixes.

2.2 Invoking new key commands and environments

The syntaxes for calling new key commands and environments are as follows:

	Macro
23	<code>\cmd[⟨arg1⟩]{⟨arg2⟩}...{⟨argn⟩}(⟨keyval⟩)</code>
24	<code>\begin{env}[⟨arg1⟩]{⟨arg2⟩}...{⟨argn⟩}(⟨keyval⟩)</code>
25	<code>environment body</code>
26	<code>\end{env}</code>

where `\cmd` and `env` have been previously defined using key command and key environment. You refer to your arguments using parameter number one `#1` onwards, up to a maximum of `#8` (yes, `#8`, not `#9`). Here, `⟨keyval⟩` (including the parenthesis) are optional arguments: you can omit them if you want to use the values of the keys set at key command definition time. Using keys is preferable to using parameters: you don't have to match parameters to arguments and, in principle, there is no limit to the number of keys that are permissible.

2.3 Commands and environments with two optional arguments

The `skeycommand` package uses the following macros internally. They can be used to define new commands and environments with two optional arguments. Their philosophy, intent, and use syntaxes differ from those of the `twoopt` package. They may be useful to some users, although I recommend the use of the above key commands.

	Macro
27	<code>\newtwoptcmd⟨cmd⟩[⟨narg⟩][⟨default1⟩]{⟨defn⟩}</code>
28	<code>\newtwoptcmd*⟨cmd⟩[⟨narg⟩][⟨default1⟩]{⟨defn⟩}</code>
29	<code>\renewtwoptcmd⟨cmd⟩[⟨narg⟩][⟨default1⟩]{⟨defn⟩}</code>
30	<code>\renewtwoptcmd*⟨cmd⟩[⟨narg⟩][⟨default1⟩]{⟨defn⟩}</code>
31	<code>\newtwoptenvron⟨cmd⟩[⟨narg⟩][⟨default1⟩]{⟨defn⟩}</code>
32	<code>\newtwoptenvron*⟨cmd⟩[⟨narg⟩][⟨default1⟩]{⟨defn⟩}</code>
33	<code>\renewtwoptenvron⟨cmd⟩[⟨narg⟩][⟨default1⟩]{⟨defn⟩}</code>
34	<code>\renewtwoptenvron*⟨cmd⟩[⟨narg⟩][⟨default1⟩]{⟨defn⟩}</code>

`⟨narg⟩` is the total number of arguments, including the first and second optional arguments. Where are the second optional arguments here, you might be wondering? The second optional argument is usually empty and doesn't appear at command definition time. The second optional argument isn't the second argument of your command (as in `twoopt` package), but the last. At command invocation, if you don't supply a value for the second optional argument, the

command will assume it to be empty. But how do you supply a value for the second optional argument? The next section shows how.

2.3.1 Invoking commands and environments with two optional arguments

The syntaxes for calling commands and environments with two optional arguments are as follows:

	Macro
35	<code>\cmd[(1st optarg)]{<arg2>}...{<argn>}(<2nd optarg>)</code>
36	<code>\begin{env}[(1st optarg)]{<arg2>}...{<argn>}(<2nd optarg>)</code>
37	<code>environment body</code>
38	<code>\end{env}</code>

If `<2nd optarg>` is empty at command or environment invocation, the command or environment will assume it to be empty. Now you can see the conceptual link between `\newtwooptcmd` (and friends) and `\newkeycmd` (and friends).

3 EXAMPLES

	Example
39	<code>% The following is a macro of 3 parameters and 4 keys:</code>
40	<code>\newkeycmd*\demomacro(name=Steve,height=1.60m,weight=75kg,</code>
41	<code>tested=true)[3][Registered]{%</code>
42	<code>\def\x{#1}\def\y{#2}\def\z{#3}%</code>
43	<code>\edef\xxx{\dem@name}%</code>
44	<code>}</code>
45	<code>% \dem@name will hold the value supplied for 'name'</code>
46	<code>% by the user of \demomacro. 'dem' is from 'demomacro'.</code>
47	<code>% Observe the LaTeX-like syntax of this command.</code>
48	<code>% The user doesn't have to bother about the nitty-gritty</code>
49	<code>% of key infrastructure.</code>
50	<code>% You can use the following statement to instruct the user</code>
51	<code>% to always supply value for 'name' in \demomacro macro:</code>
52	<code>\uservaluekeys[KV]{demomacro}{name}</code>

Instead of using the macro `\uservaluekeys`, you can use the full pointer system of the `skeyval` package. For example, the following requires the user to supply a value for 'name' and the package will save that value for some later use:

	Example
53	<code>\newkeycmd*\demomacro(\uservalue\savevalue{name}=Steve,</code>
54	<code>height=1.60m,wieght=75kg,tested=true)[3][Registered]{%</code>
55	<code>\def\x{#1}\def\y{#2}\def\z{#3}%</code>
56	<code>\edef\xxx{\dem@name}%</code>

```

57 % User now calls the \demomacro macro:
58 \demomacro[data1]{data2}{data3}(name,height=1.55m,wieght=55kg,
59   tested=true)
60 → Error: no value supplied for 'name'

```

Now you can use `\demomacro`. If you don't supply a key-value pair here, the macro will use the above defaults:

Example

```

61 \demomacro[data1]{data2}{data3}(name=John,height=1.55m,
62   weight=55kg,tested=true)

```

Example

```

63 \newkeycmd*\firstmacro<sk@>(name=Steve,
64   height=1.6m)[8][xxx]{%
65   \noindent Name: \sk@name\newline
66   Height: \sk@height\newline
67   Details: #1#2#3#4#5#6#7#8%
68   \newline\newline
69 }

70 \begin{document}
71 \firstmacro[1]{-2}{-3}{-4}{-5}{-6}{-7}{-8}
72   (name=John {(Winne)}r,height=1.54m)
73 \end{document}

```

Example

```

74 \newkeyenviron*{vdescription}<skv@>(labelwidth=5pt,
75   labelsep=5pt)[2][\qqquad
76   {\begin{list}{\redef*\makelabel##1{\sffamily ##1:\hfil}%
77     \settowidth\labelwidth{\makelabel{#1}}%
78     \SKV@dimdef\leftmargin{\labelwidth+\skv@labelwidth
79     +\labelsep+\skv@labelsep}}%
80     \item[Description Preamble] #2%
81   }{\end{list}\ignorespacesafterend}

82 \begin{document}
83 \begin{vdescription}[Description Postamble]%
84   {${\star\star\star$(labelwidth=10pt,labelsep=10pt)
85     \item[Item 1] xxx
86     \item[Item 2] yyy
87     \item[Description Postamble] $\langle$End of my
88       environment$\rangle$
89   \end{vdescription}
90 \end{document}

```

Example

```

91 \def\@barbara{An author writing an article for publication
92   in TUGboat is encouraged to create it on a computer file

```

```

93     and submit it on magnetic tape.}
94 \def\barbara{BARBARA BEETON,\ \emph{How to Prepare a File For
95   Publication in TUGboat} (1981)}
96 \def\@hieronymus{The printer should refuse to employ wandering
97   men, foreigners who, after having committed some grievous
98   error, can easily disappear and return to their own country.}
99 \def\hieronymus{HIERONYMUS HORNSCHUCH (1608)}

100 % The macros \@barbara, \barbara, etc. are just shorthands:
101 % you can enter their contents directly in key commands.

102 \newkeyenviron{Quote}<mp@>(leftmargin=\leftmargin,
103   rightmargin=\rightmargin,mode=false,whoby=\null,
104   source=\null){\begin{list}{}{%
105     \setlength\leftmargin{\mp@leftmargin}%
106     \setlength\rightmargin{\mp@rightmargin}%
107   }%
108   \item[]\makebox[0pt][r]{‘}%
109 }{%
110   \unskip\makebox[0pt][l]{’}%
111   \item[] \flushright\mp@whoby
112   \item[] \flushleft\small Source: \mp@source
113   \end{list}
114   \endgraf\vskip\baselineskip
115 }
116 \usepackage{lipsum}
117 \lipsum[1]
118 \begin{Quote}(leftmargin=30pt,rightmargin=30pt,mode=false,
119   whoby=\barbara,source={The \TeX Book})
120   {\ifmp@mode\color{red}\else\color{blue}\fi \@barbara}
121 \end{Quote}

122 \lipsum[1]
123 \begin{Quote}(leftmargin=20pt,rightmargin=20pt,mode=true,
124   whoby=\hieronymus,source={The \TeX Book})
125   {\ifmp@mode\color{red}\else\color{blue}\fi \@hieronymus}
126 \end{Quote}

```

Example

```

127 \usepackage{lipsum}
128 \newcounter{notecnt}
129 \def\noteparameters{\labelsep=\notelabelsep
130   \itemindent=\noteitemindent \leftmargin=\noteleftmargin
131   \rightmargin=\noterightmargin \labelwidth=\notelabelwidth}

132 \newkeyenviron*{note}<note>(labelsep=8pt,itemindent=8pt,
133   leftmargin=\parindent,rightmargin=\parindent,labelwidth=0pt,
134   preskip=0ex)[1][\baselineskip]%
135   {\begin{list}{\textsc{Note}}~\arabic{notecnt}: }%

```

```

136     {\noteparameters\usecounter{notecnt}}}%
137     \vskip#1}%
138 {\end{list}\ignorespacesafterend}

139 \begin{document}
140 \noindent\lipsum[1]
141 \begin{note}[\notepreskip] (labelsep=8pt,itemindent=8pt,
142     leftmargin=30pt,rightmargin=30pt,labelwidth=0pt,preskip=2ex)
143 \item \lipsum[1]
144 \item \lipsum[1]
145 \end{note}
146 \end{document}

```

Example

```

147 \def\sitation{}
148 \def\sitparameters{\leftmargin=\sit@leftmargin
149     \rightmargin=\sit@rightmargin}
150 \newbox\sitname
151 \renewkeyenvron*{sitation}(leftmargin=\parindent,
152     rightmargin=\parindent,nolinebreak=1)[2][1]%
153 {\def\cnolinebreak{#1}\sitparameters\sbox\sitname
154     {\textit{#2}}}%
155     \begin{quote}%
156     }%
157     {\hspace*{\fill}\nolinebreak[\cnolinebreak]%
158         \quad\hspace*{\fill}\finalhyphendemerits\z@
159         \box\sitname
160     \end{quote}\ignorespacesafterend}

161 \begin{document}
162 \begin{sitation}[\sit@nolinebreak]%
163     {Hieronymus Hornschuch~(1608)}%
164     (leftmargin=30pt,rightmargin=30pt,nolinebreak=2)
165 The printer should refuse to employ wandering men,
166 foreigners who, after having committed some grievous error,
167 can easily disappear and return to their own country.
168 \end{sitation}

169 \begin{sitation}{William Stanley Jevons~(1878)}%
170 Some composers still object to work in offices where
171 type-composing machines are introduced. They are all afraid
172 that if the work is done too easily and rapidly, they will
173 not be wanted to do it. Justifying and page makeup can't be
174 done profitably by machines, so the employees needn't fear
175 losing their jobs.
176 \end{sitation}
177 \end{document}

```


Example

```

178 \newkeyenvirown*{vdescription}(labelwidth=5pt,
179   labelsep=5pt)[2][\quad]
180   {\begin{list}{}{\redef*\makelabel##1{\sffamily ##1:\hfil}%
181     \settowidth\labelwidth{\makelabel{#1}}%
182     \SKV@dimdef\leftmargin{\labelwidth+\vde@labelwidth
183       +\labelsep+\vde@labelsep}}%
184     \item[Description Preamble] #2%
185   }\end{list}\ignorespacesafterend}

186 \begin{document}
187 \begin{vdescription}[Description Postamble]{+++xxx+++}%
188   (labelwidth=10pt,labelsep=5pt)
189   \item[Item 1] xxx
190   \item[Item 2] yyy
191   \item[Description Postamble] $\langle$End of my
192     environment$\rangle$
193 \end{vdescription}
194 \end{document}

```

Example

```

195 \renewtwoptenvirown*{vdescription}[3][\quad]
196   {\begin{list}{}{\redef*\makelabel##1{\sffamily ##1:\hfil}%
197     \settowidth\labelwidth{\makelabel{#1}}%
198     \SKV@dimdef\leftmargin{\labelwidth+\labelsep+#2}}%
199     \item[Description Preamble] #3%
200   }\end{list}\ignorespacesafterend}

201 \begin{document}
202 \begin{vdescription}[Description Postamble]{4cm}({(Begin
203   environment no.\ 1)})
204   \item[Item 1] xxx
205   \item[Item 2] yyy
206   \item[Description Postamble] (End of environment no.\ 1)
207 \end{vdescription}
208 \end{document}

```

Example

```

209 \newtwoptenvirown*{udescription}[3][\hspace{1cm}]
210   {\begin{list}{}{\redef*\makelabel##1{\sffamily ##1:\hfil}%
211     \settowidth\labelwidth{\makelabel{#1}}%
212     \SKV@dimdef\leftmargin{\labelwidth+\labelsep+#3}}%
213     \item[Description Preamble] #2%
214   }\end{list}\ignorespacesafterend}

215 \begin{document}
216 \begin{udescription}[Description Postamble]{uuu}(4cm)
217   \item[Item 1] xxx
218   \item[Item 2] yyy

```

```

219 \item[Description Postamble] The End
220 \end{udescription}
221 \end{document}

```

Example

```

222 \def\firstmacro{
223 \renewkeycmd*\firstmacro<skv@>(name=Steve,
224   module=Martian logic,pass=true)[2][\@ptsize]{%
225   \edef\x{\skv@name}%
226   \wlog{\if0#1 10pt\else\if1#1 11pt\else
227     \if2#1 12pt\fi\fi\fi\space font used}%
228   \def\y{#2}%
229 }
230 \firstmacro[0]{aaa}(name=John,module=Philosophy,pass=false)

```

Example

```

231 \newtwooptcmd*\macro[3][xxx]{\def\x{#1}\def\y{#2}\def\z{#3}}
232 \macro[uuu]{vvv}(www)
233 \macro{vvv}(www)
234 \macro{vvv}

```

Example

```

235 \SKV@undef\macro
236 \newtwooptcmd*\macro[2][xxx]{\def\x{#1}\def\y{#2}}
237 \macro[uuu](vvv)
238 \macro(vvv)

```

Example

```

239 \renewtwooptcmd\macro[2][xxx]{\def\x{#1}\long\def\y{#2}}
240 \macro[uuu](\par)
241 \macro(\par)

```

Example

```

242 \let\tcl\textrcolor
243 \newkeycmd*\firstrule(raise=.5ex,width=1em,thick=2pt,
244   proclaim=false)[1]{%
245   \tcl{blue}{\rule[\fir@raise]{\fir@width}{\fir@thick}}}%
246   #1%
247   \tcl{cyan}{\rule[\fir@raise]{\fir@width}{\fir@thick}}}%
248   \iffir@proclaim \color{red}\fi\textdaggerdbl
249 }
250 \usepackage[left=2cm,right=2cm]{geometry}
251 \begin{document}
252 \parindent\z@
253 \begin{tabular*}{\textwidth}{lr}
254 \verb+\firstrule{Hello World}(width=2em,thick=4pt,

```

```

255     proclaim)+:&
256     \firstrule{Hello World}(width=2em,thick=4pt)\cr
257 \verb+\firstrule{Hello}(width=2em,thick=.5pt,
258     proclaim=true)+:&
259     \firstrule{Hello}(width=2em,thick=.5pt,proclaim=true)\cr
260 \verb+\firstrule{Hello World}(thick=2pt,
261     proclaim=true)+:&
262     \firstrule{Hello World}(thick=2pt,proclaim=true)\cr
263 \verb+\firstrule{Hello World}(raise=1ex,width=2em,
264     thick=1pt)+:&
265     \firstrule{Hello}(raise=1ex,width=2em,thick=1pt)
266 \end{tabular*}
267 \end{document}

```

Example

```

268 \let\tcl\textcolor
269 \newkeycmd\secondrule<mp@>(raise=.5ex,width=1em,thick=2pt,
270     proclaim=false) [2] [\tcl{magenta}{\$star$}]{%
271     \tcl{cyan}{\rule[\mp@raise]{\mp@width}{\mp@thick}}%
272     #1#2#1%
273     \tcl{blue}{\rule[\mp@raise]{\mp@width}{\mp@thick}}%
274     \ifmp@proclaim \color{red}\fi\textdaggerdbl
275 }
276 \usepackage[left=2cm,right=2cm]{geometry}
277 \begin{document}
278 \parindent\z@
279 \begin{tabular*}\textwidth{lr}
280 \verb+\secondrule[\textbullet]{Hello World}(width=2em,
281     thick=4pt,proclaim)+:&
282     \secondrule[\textbullet]{Hello World}(width=2em,
283     thick=4pt)\cr
284 \verb+\secondrule{Hello}(width=2em,thick=.5pt,
285     proclaim=true)+:&
286     \secondrule{Hello}(width=2em,thick=.5pt,proclaim=true)\cr
287 \verb+\secondrule{Hello World}(thick=2pt,
288     proclaim=true)+:&
289     \secondrule{Hello World}(thick=2pt,proclaim=true)\cr
290 \verb+\secondrule{Hello World}(raise=1ex,width=2em,
291     thick=1pt)+:&
292     \secondrule{Hello}(raise=1ex,width=2em,thick=1pt)
293 \end{tabular*}
294 \end{document}

```