

conv-xkv: Convert xkeyval format style

D. P. Story
Email: `dpstory@uakron.edu`

processed January 4, 2017

Contents

1	Introduction	1
2	Preliminaries	2
3	Core commands for this package	4
4	Index	8
5	Change History	9

¹ `*package`

1 Introduction

This is a intellectual exercise for creating *alternate key-value* notation. The standard L^AT_EX notation is $\langle key \rangle = \langle value \rangle$. To change to the JavaScript object style of key-values ($\langle key \rangle : \langle value \rangle$), use `\cxkvsetkeys` as you would `\setkeys`:

`\cxkvsetkeys{\family}{\KV-pairs}`

to convert $\langle key \rangle : \langle value \rangle$ to $\langle key \rangle = \langle value \rangle$ and `xkeyval` processes the keys as it normally does. The comma (,) separates sets of key-value pairs and must not, therefore, be used as the delimiter that separates the $\langle key \rangle$ from the $\langle value \rangle$.

The package is more general than what is described above. You can define several key-value delimiters, for whatever reason, in your document or package. Declare a *named* delimiter:

`\DeclareDelimiter{\name}{\delimiter}`

Use the newly declared delimited as follows:

`\cxkvsetkeys(\name){\family}{\KV-pairs}`

The case of using a colon (:) for the delimiter is already defined, its name is ‘colon’ and need not be declared.

Important change in syntax With version dated 2017/01/03 or later, the optional argument $\langle name \rangle$ is now delimited by **parentheses**, rather than the standard brackets. This is to be able to detect $\langle name \rangle$ when the full syntax of `\setkeys` is used:

```
\setkeys*[\langle prefix \rangle]{\langle families \rangle}[\langle na \rangle]{\langle keys \rangle}
```

The syntax for `\ckvsetkeys` shall be

```
\ckvsetkeys(\langle name \rangle)*[\langle prefix \rangle]{\langle families \rangle}[\langle na \rangle]{\langle keys \rangle}
```

The `conv-xkv` package does nothing with `xkeyval` arguments `*[\langle prefix \rangle]` and `[\langle na \rangle]` other than to collect them and pass them on to `\setkeys` at the appropriate time. The `conv-xkv` is concerned only with converting a new notation $\langle key \rangle \langle delim \rangle \langle value \rangle$ to $\langle key \rangle = \langle value \rangle$.

If the key-values do not contain the designated delimiter, `conv-xkv` simply passes everything on to `\setkeys`. What this means is that, for example, both `\ckvsetkeys{myfam}{fname:Don,lname:Story}` works as does `\ckvsetkeys{myfam}{fname=Don,lname=Story}`. One then has the option of using the standard notation or an alternate notation.

Demo file The example file is `convert2xkeyval.tex`, use it to explore the possibilities and is found in the `examples` folder of this distribution.

2 Preliminaries

We require the `xkeyval` package.

```
2 \RequirePackage{xkeyval}
```

The code below is taken from `hyperref`, and `set` and `restore` commands are renamed. This hopefully makes a number of special characters available to act as a delimiter.

```
3 \begingroup
4   \@makeother\'%
5   \@makeother\=%
6   \edef\x{%
7     \edef\noexpand\x{%
8       \endgroup
9       \noexpand\toks@{%
10        \catcode 96=\noexpand\the\catcode'\noexpand\'\relax
11        \catcode 61=\noexpand\the\catcode'\noexpand\=\relax
12      }%
13    }%
14    \noexpand\x
15  }%
16 \x
17 \@makeother\%
18 \@makeother\=
19 \def\ckv@SetCatcodes{%
```

```

20 \makeother\'%
21 \makeother\=%
22 \makeother\~%
23 \catcode'\$=3 %
24 \catcode'\&=4 %
25 \catcode'\^=7 %
26 \catcode'\_ =8 %
27 \makeother\| %
28 \makeother\: %
29 \makeother\( %
30 \makeother\) %
31 \makeother\[ %
32 \makeother\] %
33 \makeother\/ %
34 \makeother\! %
35 \makeother\< %
36 \makeother\> %
37 \makeother\. %
38 \makeother\; %
39 \makeother\+ %
40 \makeother\- %
41 \makeother\" %
42 \makeother\' %
43 }
44 \begingroup
45 \def\x#1{\catcode'\noexpand#1=\the\catcode'#1\relax}%
46 \xdef\ckv@RestoreCatcodes{%
47   \the\toks@
48   \x\~%
49   \x\$%
50   \x\&%
51   \x\^%
52   \x\_%
53   \x\| %
54   \x\: %
55   \x\( %
56   \x\) %
57   \x\[ %
58   \x\] %
59   \x\/ %
60   \x\! %
61   \x\<%
62   \x\>%
63   \x\. %
64   \x\; %
65   \x\+ %
66   \x\- %
67   \x\" %
68   \x\' %
69 }%

```

```

70 \endgroup
71 \ckv@SetCatcodes

```

3 Core commands for this package

The default delimiter is the colon (:).

```

72 \def\csarg#1#2{\expandafter#1\csname#2\endcsname}
73 \csarg\def{kvdelim-colon}{:}

```

`\usekvdelim` Use `\usekvdelim` to display delimiter, as associated with the argument #1.

```

74 \def\usekvdelim#1{\@nameuse{kvdelim-#1}}

```

`\DeclareDelimiter` In the preamble, we declare the delimiter to be used. The command takes on argument, which is the delimiter to be used, for example ‘:’ or ‘->’. If this declaration does not appear in the preamble, the delimited is taken to be ‘:’.

```

75 \def\DeclareDelimiter{\ckv@SetCatcodes\DeclareDelimiter@i}
76 \def\DeclareDelimiter@i#1#2{\@ifundefined{kvdelim-#1}
77   {\csarg\def{kvdelim-#1}{#2}\ckv@RestoreCatcodes\cxkvSetup{#1}}
78   {\ckv@RestoreCatcodes}}
79 \@onlypreamble\DeclareDelimiter

```

`\cxkv@tmptoks` is used to hold the converted key-values, the contents of this token register is passed to `\setkeys` in `\cxkv@cnvrtDelimniiEquali`

```

80 \newtoks\cxkv@tmptoks \cxkv@tmptoks={}
81 \def\cxkv@dummys{dummy}
82 \def\cxkv@dummysc{dummy,}
83 \bgroup
84   \catcode'\#=12\relax\gdef\cxkvarg{#}
85   \obeyspaces\gdef\cxkv@TAB{   }
86 \egroup

```

`\cxkvsetkeys` This is the default definition, setup for using the colon (:) as the key-value delimiter. But these next two commands are redefined by the `\DeclareDelimiter` command in the preamble. The syntax is

```
\cxkvsetkeys[<name>]{<family>}{<KV-pairs>}
```

where *<KV-pairs>* are the key-value pairs using the declared delimiter.

```
\cxkvsetkeys{myfam}{fname: Fred,lname: Flintstone}
```

The family `myfam` and keys `fname` and `lname` must have been defined earlier: If the optional argument is not specified, then it is assumed the *<name>* argument is `colon`, a reserved word for this package for this argument.

```

\define@key{myfam}{\def\fname{#1}}
\define@key{myfam}{\def\lname{#1}}

87 \def\cxkv@colon{colon}

```

The general form for `\setkeys` is

`\setkeys*[\langle prefix \rangle]{\langle families \rangle}[\langle na \rangle]{\langle keys \rangle}`

`\cxkvsetkeys` The syntax for `\cxkvsetkeys` shall be

`\cxkvsetkeys(\langle name \rangle)*[\langle prefix \rangle]{\langle families \rangle}[\langle na \rangle]{\langle keys \rangle}`

The process to pick up the full parameter set of `\setkeys` is lengthy.

```

88 \newcommand\cxkvsetkeys{%
89   \ifnextchar({\cxkvsetkeys@i}{\cxkvsetkeys@i(colon)})}
90 \def\cxkvsetkeys@i(#1){\cxkvsetkeys@ii{#1}}
91 \def\cxkvsetkeys@ii#1{\def\cxkv@delimname{#1}\ifstar
92   {\def\cxkv@skOpts{*}\cxkvsetkeys@iii}
93   {\def\cxkv@skOpts{}\cxkvsetkeys@iii}}
94 \newcommand\cxkvsetkeys@iii[2][\def\@rgi{#1}\ifx\@rgi\@empty
95   \expandafter\def\expandafter\cxkv@skOpts
96     \expandafter{\cxkv@skOpts{#2}}\else
97   \expandafter\def\expandafter
98     \cxkv@skOpts\expandafter{\cxkv@skOpts[#1]{#2}}\fi
99   \def\thisxkvF@mily{#2}\cxkvsetkeys@iv}
100 \newcommand\cxkvsetkeys@iv[2][\def\@rgi{#1}\ifx\@rgi\@empty\else
101   \expandafter\def\expandafter\cxkv@skOpts
102     \expandafter{\cxkv@skOpts[#1]}\fi
103   \expandafter\cxkvsetkeys@v\expandafter{\thisxkvF@mily}{#2}}
104 \def\cxkvsetkeys@v#1#2{\cxkv@skipfalse
105   \ifx\cxkv@delimname\cxkv@colon\else
106     \InputIfFileExists{xkv-\cxkv@delimname.cut}
107     {\PackageInfo{conv-xkv}{Inputting xkv-\cxkv@delimname.cut}}
108     {\PackageInfo{conv-xkv}{Cannot find xkv-\cxkv@delimname.cut}}\fi
109   \@nameuse{cxkvsetkeys-\cxkv@delimname}{#1}{#2}}

110 \csarg\def{cxkvsetkeys-colon}#1#2{%
111   \def\thisxkvF@mily{#1}\def\thisxkvV@lues{#2}\def\cxkv@scratch{}}%
112   \cxkv@tmptoks={}%
113   \@nameuse{cxkv@cnvrtDelimniiEqual-colon}#2,dummy:dummy,\@nil}
114 \csarg\def{cxkv@cnvrtDelimniiEqual-colon}#1:#2,#3\@nil{%
115   \cxkv@cnvrtDelimniiEquali{colon}{#1}{#2}{#3}}

```

`\cxkvSetup` Write the definitions of `\cxkvsetkeys` and `\cxkv@cnvrtDelimniiEqual` to the file `conv-xkv.cut` then input this file back in.

```

116 \def\cxkvSetup#1{\bgroup
117 \IfFileExists{xkv-#1.cut}{\PackageInfo{conv-xkv}{xkv-#1.cut
118   already exists,\MessageBreak will not create another one}}{%
119   \PackageInfo{conv-xkv}{Creating the file xkv-#1.cut
120     containing\MessageBreak required definitions}%
121   \newwrite \cxkv@write
122   \uccode'c='}%
123   \def\w{#1}\def\x{cxkvsetkeys-#1}%
124   \def\y{cxkv@cnvrtDelimniiEqual-#1}%

```

```

125 \def\z{kvdelim-#1}%
126 \immediate\openout \cxkv@write xkv-#1.cut
127 \immediate\write\cxkv@write{\string\makeatletter}%
128 \uppercase{\immediate\write\cxkv@write{\string
129 \csarg\string\def{y}\cxkvarg1\@nameuse{z}%
130 \cxkvarg2,\cxkvarg3\string\@nil{c^J\cxkv@TAB
131 \string\cxkv@cnvrtDelimniiEquali{w}{\cxkvarg1}%
132 {\cxkvarg2}{\cxkvarg3}}}}
133 \uppercase{\immediate\write\cxkv@write{\string\csarg\string\def
134 {x}\cxkvarg1\cxkvarg2{c^J\cxkv@TAB
135 \string\def\string\thisxkvF@mily{\cxkvarg1}\string
136 \def\string\thisxkvV@lues{\cxkvarg2}\string
137 \let\string\cxkv@scratch\string\@empty\string
138 \cxkv@tmptoks={}c^J\cxkv@TAB
139 \string\@nameuse{y}\cxkvarg2,%
140 \cxkv@dumny\@nameuse{z}\cxkv@dumny,\string\@nil}}}}
141 \immediate\write\cxkv@write{\string\makeatother}%
142 \immediate\closeout \cxkv@write
143 }%
144 \egroup}

```

\cxkv@cnvrtDelimniiEquali continues \cxkv@cnvrtDelimniiEqual. It is the part that does not need to be redefined.

```

145 \newif\ifcxkv@keyonly \cxkv@keyonlyfalse
146 \def\cxkv@comma{,}
147 \def\cxkv@removecomma#1,\@nil{\def\cxkv@key{#1}}
148 \def\cxkv@parsecomma#1,#2\@nil{\def\@rgi{#1}\def\@rgii{#2}%
149 \ifx\@rgii\@empty\cxkv@keyonlyfalse\else
150 \cxkv@keyonlytrue\cxkv@removecomma#2\@nil\fi}
151 \newif\ifcxkv@skip \cxkv@skipfalse
152 \def\cxkv@cnvrtDelimniiEquali#1#2#3#4{%
153 \def\cxkv@rgiii{#3}\def\cxkv@rgiv{#4}%

```

If the fourth argument is empty, that means there were no delimiters in the argument, so we pass the original argument \thisxkvF@mily to \setkeys.

```

154 \ifx\thisxkvV@lues\@empty\else
155 \ifx\cxkv@rgiv\@empty
156 \edef\cxkv@next{\noexpand
157 \setkeys\cxkv@skOpts{\thisxkvV@lues}}%
158 \cxkv@skiptrue
159 \fi
160 \fi
161 \let\thisxkvV@lues\@empty
162 \ifcxkv@skip\else
163 \ifx\cxkv@rgiii\cxkv@dumny
164 \cxkv@parsecomma#2,\@nil
165 \ifcxkv@keyonly
166 \edef\cxkv@tmp{\the\cxkv@tmptoks,\@rgi}%
167 \cxkv@tmptoks=\expandafter{\cxkv@tmp}%
168 \edef\cxkv@scratch{\the\cxkv@tmptoks}%

```

```

169         \edef\cxkv@next{\noexpand
170             \setkeys\cxkv@skOpts{\the\cxkv@tmptoks}}%
171     \else
172         \edef\cxkv@next{\noexpand
173             \setkeys\cxkv@skOpts{\the\cxkv@tmptoks}}%
174     \fi
175 \else
176     \cxkv@parsecomma#2,\@nil
177     \ifcxkv@keyonly
178         \edef\cxkv@tmp{\the\cxkv@tmptoks,\@rgi}%
179         \cxkv@tmptoks=\expandafter{\cxkv@tmp}%
180         \edef\cxkv@scratch{\the\cxkv@tmptoks}%
181         \edef\cxkv@next{\noexpand
182             \@nameuse{cxkv@cnvrtDelimniiEqual-#1}\cxkv@key
183             \@nameuse{kvdelim-#1}#3,#4\noexpand\@nil}
184     \else
185         \cxkv@tmptoks=\expandafter{\cxkv@scratch,#2=#3}%
186         \edef\cxkv@scratch{\the\cxkv@tmptoks}%
187         \def\cxkv@next{%
188             \@nameuse{cxkv@cnvrtDelimniiEqual-#1}#4\@nil}\fi
189     \fi\fi\cxkv@next
190 }
191 \ckv@RestoreCatcodes
192 </package>

```

4 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			
\!	34, 60	\cxkv@tmp	166, 167, 178, 179
\#	84	\cxkv@tmptoks	80, 112, 138, 166–168, 170, 173, 178–180, 185, 186
\\$	23, 49	\cxkv@write	121, 126–128, 133, 141, 142
\%	122	\cxkvarg	84, 129–132, 134–136, 139
\&	24, 50	\cxkvsetkeys	87, 88
\:	28, 54	\cxkvsetkeys@i	89, 90
\;	38, 64	\cxkvsetkeys@ii	90, 91
\=	5, 11, 18, 21	\cxkvsetkeys@iii	92–94
\@makeother	4, 5, 17, 18, 20–22, 27–42	\cxkvsetkeys@iv	99, 100
\@onlypreamble	79	\cxkvsetkeys@v	103, 104
\@rgi	94, 100, 148, 166, 178	\cxkvSetup	77, <u>116</u>
\@rgii	148, 149		
\^	25, 51	D	
_	26, 52	\DeclareDelimiter	<u>75</u>
\‘	4, 10, 17, 20	\DeclareDelimiter@i	75, 76
\	27, 53	E	
\~	22, 48	\egroup	86, 144
B		I	
\bgroup	83, 116	\ifcxkv@keyonly	145, 165, 177
C		\ifcxkv@skip	151, 162
\ckv@RestoreCatcodes	46, 77, 78, 191	\IfFileExists	117
\ckv@SetCatcodes	19, 71, 75	\InputIfFileExists	106
\csarg	72, 73, 77, 110, 114, 129, 133	M	
\cxkv@cnvrtDelimniiEquali	115, 131, 152	\makeatletter	127
\cxkv@colon	87, 105	\makeatother	141
\cxkv@comma	146	N	
\cxkv@delimname	91, 105–109	\newwrite	121
\cxkv@dummy	81, 140, 163	O	
\cxkv@dummys	82	\obeyspaces	85
\cxkv@key	147, 182	\openout	126
\cxkv@keyonlyfalse	145, 149	P	
\cxkv@keyonlytrue	150	\PackageInfo	107, 108, 117, 119
\cxkv@next	156, 169, 172, 181, 187, 189	R	
\cxkv@parsecomma	148, 164, 176	\RequirePackage	2
\cxkv@removecomma	147, 150	T	
\cxkv@rgiii	153, 163	\thisxkvF@mily	99, 103, 111, 135
\cxkv@rgiv	153, 155	\thisxkvV@lues	111, 136, 154, 157, 161
\cxkv@scratch	111, 137, 168, 180, 185, 186		
\cxkv@skipfalse	104, 151		
\cxkv@skipttrue	158		
\cxkv@skOpts	92, 93, 95, 96, 98, 101, 102, 157, 170, 173		
\cxkv@TAB	85, 130, 134, 138		

	U		<code>\usekvdelim</code>	4, 74
<code>\uccode</code>	122		W	
<code>\uppercase</code>	128, 133	<code>\write</code>	127, 128, 133, 141	

5 Change History

v1.0 (2016/12/20)		is present at all	6
General: Date of first upload to CTAN	2	v1.1a (2017/01/03)	
v1.1 (2017/01/03)		General: Change in syntax, use parentheses	
General: Try to detect if the expected delimiter		rather than brackets	2