conv-xkv: Convert xkeyval format style

D. P. Story
Email: dpstory@uakron.edu
processed March 20, 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 \LaTeX notation is ⟨key⟩=⟨value⟩. To change to the JavaScript object style of key-values ⟨key⟩:⟨value⟩, use \cxkvsetkeys as you would \setkeys:

\cxkvsetkeys{⟨family⟩}{⟨KV-pairs⟩}

to convert ⟨key⟩:⟨value⟩ to ⟨key⟩=⟨value⟩ 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 ⟨key⟩ from the ⟨value⟩.

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 \texttt{\setkeys} is used:

\begin{verbatim}
\setkeys*[\langle prefix \rangle]{\langle families \rangle}[\langle na \rangle]{\langle keys \rangle}
\end{verbatim}

The syntax for \texttt{\cxkvsetkeys} shall be

\begin{verbatim}
\cxkvsetkeys{\langle name \rangle}*{\langle prefix \rangle}{\langle families \rangle}[\langle na \rangle]{\langle keys \rangle}
\end{verbatim}

The \texttt{conv-xkv} package does nothing with \texttt{xkeyval} arguments *{\langle prefix \rangle} and [\langle na \rangle] other than to collect them and pass them on to \texttt{\setkeys} at the appropriate time. The \texttt{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, \texttt{conv-xkv} simply passes everything on to \texttt{\setkeys}. What this means is that, for example, both \texttt{\cxkvsetkeys{myfam}{fname:Don,lname:Story}} works as does \texttt{\cxkvsetkeys{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 \texttt{convert2xkeyval.tex}, use it to explore the possibilities and is found in the \texttt{examples} folder of this distribution.

## 2 Preliminaries

We require the \texttt{xkeyval} package.

\begin{verbatim}
\\RequirePackage{xkeyval}
\end{verbatim}

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

\begin{verbatim}
\begingroup
\@makeother\%
\@makeother\=%
\edef\x{%
\edef\noexpand\x{%
\\endgroup
\noexpand\toks@{%
\catcode\%=
oexpand\the\catcode\%
\relax
\\catcode\!=
oexpand\the\catcode\!
\relax
)}%
\noexpand\x
)%
\noexpand\x
)%
\@makeother`
\@makeother=
\def\ckv@SetCatcodes{%
\end{verbatim}

2
\begingroup
\def\x#1{\catcode'\noexpand#1=\the\catcode'#1\relax}\def\ckv@RestoreCatcodes{\the\toks@\x\˜\x\$\x\&\x\ˆ\x\_\x\|\x:\x\:%\x\(\x\)\x\[\x\]\x\=/x\!\x\<\x\>\x\.:\x\;\x\+\x\-%\x\"\x\'}\begingroup
\def\x#1{\catcode'\noexpand#1=\the\catcode'#1\relax}\def\ckv@RestoreCatcodes{\the\toks@\x\˜\x\$\x\&\x\ˆ\x\_\x\|\x\:%\x\(\x\)\x\[\x\]\x\=/x\!\x\<\x\>\x\.:\x\;\x\+\x\-%\x\"\x\'}\endgroup
3 Core commands for this package

The default delimiter is the colon (\):

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

\textbf{usekvdelim} Use \texttt{usekvdelim} to display delimiter, as associated with the argument \texttt{#1}.

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

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

```
\def\DeclareDelimiter{\ckv@SetCatcodes\DeclareDelimiter@i}
\def\DeclareDelimiter@i#1#2{\@ifundefined{kvdelim-#1}
{\csarg\def{kvdelim-#1}{#2}\ckv@RestoreCatcodes\cxkvSetup{#1}}
{\ckv@RestoreCatcodes}}
```

\texttt{cxkv@tmptoks} is used to hold the converted key-values, the contents of this token register is passed to \texttt{\setkeys} in \texttt{\cxkv@cnvrtDelimniEquali}

```
\newtoks\cxkv@tmptoks \cxkv@tmptoks={}
\def\cxkv@tmptoks{\immediate\write18{\cxkv@tmptoks}}
\def\cxkv@dummy{dummy}
\def\cxkv@dummyc{dummy,}
\bgroup
\catcode`\#=12\relax\gdef\cxkvarg{#}
\obeyspaces\gdef\cxkv@TAB{ }
\egroup
```

\textbf{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 \texttt{\DeclareDelimiter} command in the preamble. The syntax is

```
\cxkvsetkeys[\langle name\rangle]{\langle family\rangle}{\langle KV-pairs\rangle}
```

where \langle KV-pairs\rangle are the key-value pairs using the declared delimiter.

```
\cxkvsetkeys{myfam}{\langle family\rangle}{\langle KV-pairs\rangle}
```

The family \texttt{myfam} and keys \texttt{fname} and \texttt{lname} must have been defined earlier: If the optional argument is not specified, then it is assumed the \langle name\rangle argument is \texttt{colon}, a reserved word for this package for this argument.

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

```
\def\cxkv@colon{colon}
```
The general form for \setkeys is
\setkeys*[{⟨prefix⟩}]{⟨families⟩}[⟨na⟩]{⟨keys⟩}

\cxkvsetkeys The syntax for \cxkvsetkeys shall be
\cxkvsetkeys⟨⟨name⟩⟩*[⟨prefix⟩]{⟨families⟩}[⟨na⟩]{⟨keys⟩}

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

\newcommand\cxkvsetkeys{%\@ifnextchar({\cxkvsetkeys@i}{\cxkvsetkeys@i(colon)}}
\def\cxkvsetkeys@i(#1){\cxkvsetkeys@ii{#1}}
\def\cxkvsetkeys@ii#1{\def\cxkv@delimname{#1}\@ifstar{\def\cxkv@skOpts{*}{\cxkvsetkeys@iii}{\cxkvsetkeys@iv}}{\def\cxkv@skOpts{}{\cxkvsetkeys@iii}{\cxkvsetkeys@iv}}}
\newcommand\cxkvsetkeys@iii[2][]{\def\@rgi{#1}\ifx\@rgi\@empty\expandafter\def\expandafter\cxkv@skOpts\expandafter{\cxkv@skOpts{#2}}\else\expandafter\def\expandafter\cxkv@skOpts\expandafter{\cxkv@skOpts[#1]{#2}}\fi\def\thisxkvF@mily{#2}\cxkvsetkeys@iv}
\newcommand\cxkvsetkeys@iv[2][]{\def\@rgi{#1}\ifx\@rgi\@empty\else\expandafter\def\expandafter\cxkv@skOpts\expandafter{\cxkv@skOpts[#1]}\fi\expandafter\cxkvsetkeys@v\expandafter{\thisxkvF@mily}{#2}}
\def\cxkvsetkeys@v#1#2{\cxkv@skipfalse\ifx\cxkv@delimname\cxkv@colon\else\InputIfFileExists{xkv-\cxkv@delimname.cut}{\PackageInfo{conv-xkv}{Inputting xkv-\cxkv@delimname.cut}}{\PackageInfo{conv-xkv}{Cannot find xkv-\cxkv@delimname.cut}}\fi\@nameuse{cxkvsetkeys-\cxkv@delimname}{#1}{#2}}
\csarg\def{cxkvsetkeys-colon}#1#2{\def\thisxkvF@mily{#1}\def\thisxkvV@lues{#2}\def\cxkv@scratch{}\cxkv@tmptoks={}\@nameuse{cxkv@cnvrtDelimniiEqual-colon}#2,dummy:dummy,\@nil}
\csarg\def{cxkv@cnvrtDelimniiEqual-colon}#1:#2,#3\@nil{\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.
\newcommand\cxkvSetup#1{\bgroup\IfFileExists{xkv-#1.cut}{\PackageInfo{conv-xkv}{xkv-#1.cut already exists, \MessageBreak will not create another one}}{%\PackageInfo{conv-xkv}{Creating the file xkv-#1.cut containing \MessageBreak required definitions}}\newwrite \cxkv@write\uccode'c='\%\def\w{#1}\def\x{cxkvsetkeys-#1}\def\y{cxkv@cnvrtDelimniiEqual{-}#1}\PackageInfo{conv-xkv}{\MessageBreak Writing \cxkv@write to conv-xkv}
\def\z{kvdelim-#1}\
immediate\openout \cxkv@write xkv-#1.cut\
immediate\write\cxkv@write{\string\makeatletter}\
\uppercase{\immediate\write\cxkv@write{\string\csarg\string\def{\y}\cxkvarg1\@nameuse{\z}\
\cxkvvar2,\cxkvvar3\string\@nil{c\textasciitilde J}\cxkv@TAB\
\string\cxkv@cnvrtDelimiiEquali{\w}{\cxkvvar1}\%
\{\cxkvvar2}{\cxkvvar3}\}}}
\uppercase{\immediate\write\cxkv@write{\string\csarg\string\def\x\cxkvarg1\cxkvarg2{c\textasciitilde J}\cxkv@TAB\
\string\def\thisxkvF@mily{\cxkvarg1}\string\def\thisxkvV@lues{\cxkvarg2}\string\
\let\cxkv@scratch\string\@empty\string\
\cxkv@dummy\@nameuse{\z}\cxkv@dummy,\string\@nil\}}}
\immediate\write\cxkv@write{\string\makeatother}\
\immediate\closeout \cxkv@write
\egroup}
\cxkv@cnvrtDelimiiEquali continues \cxkv@cnvrtDelimiiEqual. It is the part that does not need to be redefined.
\newif\ifcxkv@keyonly \cxkv@keyonlyfalse
\def\cxkv@comma{,}
\def\cxkv@removecomma#1,\@nil{\def\cxkv@key{#1}}
\def\cxkv@parsecomma#1,#2\@nil{\def\@rgi{#1}\def\@rgii{#2}\%
\ifx\@rgii\@empty\cxkv@keyonlyfalse\else\cxkv@keyonlytrue\cxkv@removecomma#2\@nil\fi}
\newif\ifcxkv@skip \cxkv@skipfalse
\def\cxkv@cnvrtDelimiiEquali#1#2#3#4{\%
\ifx\thisxkvV@lues\@empty\else
\ifx\cxkv@rgii\@empty\cxkv@keyonlyfalse\else
\cxkv@skiptrue
\fi
\fi
\let\thisxkvV@lues\@empty
\ifcxkv@skip\else
\ifx\cxkv@rgiii\cxkv@dummy\cxkv@dummy\cxkv@dummy\@nil
\ifcxkv@keyonly
\edef\cxkv@next{\noexpand\setkeys\cxkv@skOpts{\thisxkvV@lues}}\%
\cxkv@skiptrue
\fi
\fi
\let\thisxkvV@lues\@empty
\ifcxkv@skip\else
\ifx\cxkv@rgiii\cxkv@dummy\cxkv@dummy\cxkv@dummy\@nil
\ifcxkv@keyonly
\edef\cxkv@next{\noexpand\setkeys\cxkv@skOpts\{\the\cxkv@tmptoks,\@rgi\}}\%
\edef\cxkv@next{\noexpand\setkeys\cxkv@skOpts\{\the\cxkv@tmptoks}}\%
\edef\cxkv@next{\noexpand\setkeys\cxkv@skOpts\{\the\cxkv@tmptoks}}\%
\edef\cxkv@next{\noexpand\setkeys\cxkv@skOpts\{\the\cxkv@tmptoks}}\%

\edef\cxkv@next\noexpand\setkeys\cxkv@skOpts{\the\cxkv@tmptoks}\
\else\
\edef\cxkv@next\noexpand\setkeys\cxkv@skOpts{\the\cxkv@tmptoks}\
\fi\
\else\
\cxkv@parsecomma#2,\@nil\
\ifcxkv@keyonly\
\edef\cxkv@tmp\noexpand{\the\cxkv@tmptoks,\@rgi}\
\cxkv@tmptoks=\expandafter{\cxkv@tmp}\
\edef\cxkv@scratch\noexpand{\the\cxkv@tmptoks}\
\edef\cxkv@next\noexpand\@nameuse{cxkv@cnvrtDelimniiEqual-#1}\cxkv@key\
\@nameuse{kvdelim-#1}\@nil%#3\noexpand\@nil}\
\else\
\edef\cxkv@tmp\noexpand{\the\cxkv@tmptoks,#2=\{#3\}}\
\cxkv@tmptoks=\expandafter{\cxkv@tmp,\@rgi}\
\edef\cxkv@scratch\noexpand{\the\cxkv@tmptoks}\
\edef\cxkv@next\noexpand\@nameuse{cxkv@cnvrtDelimniiEqual-#1}\@nil%#4\noexpand\@nil}\
\fi\fi\cxkv@next
\ckv@RestoreCatcodes

⟨/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

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Page Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>!</code></td>
<td>34, 60</td>
</tr>
<tr>
<td><code>#</code></td>
<td>84</td>
</tr>
<tr>
<td><code>$</code></td>
<td>23, 49</td>
</tr>
<tr>
<td><code>%</code></td>
<td>122</td>
</tr>
<tr>
<td><code>&amp;</code></td>
<td>24, 50</td>
</tr>
<tr>
<td><code>:</code></td>
<td>28, 54</td>
</tr>
<tr>
<td><code>;</code></td>
<td>38, 64</td>
</tr>
<tr>
<td><code>=</code></td>
<td>5, 11, 18, 21</td>
</tr>
<tr>
<td><code>\makeother</code></td>
<td>4, 5, 17, 18, 20-22, 27-42</td>
</tr>
<tr>
<td><code>\onlypreamble</code></td>
<td>79</td>
</tr>
<tr>
<td><code>\orgi</code></td>
<td>94, 100, 148, 166, 178</td>
</tr>
<tr>
<td><code>\orgii</code></td>
<td>148, 149</td>
</tr>
<tr>
<td><code>\</code></td>
<td>25, 51</td>
</tr>
<tr>
<td><code>-</code></td>
<td>26, 52</td>
</tr>
<tr>
<td><code>-</code></td>
<td>4, 10, 17, 20</td>
</tr>
<tr>
<td><code>!</code></td>
<td>27, 53</td>
</tr>
<tr>
<td><code>^</code></td>
<td>22, 48</td>
</tr>
</tbody>
</table>

## B

`\bgroup` | 83, 116

## C

`\ckv@RestoreCatcodes` | 46, 77, 78, 191
`\ckv@SetCatcodes` | 19, 71, 75
`\csarg` | 72, 73, 77, 110, 114, 129, 139
`\ckv@cnvrtDelimniiEquali` | 115, 131, 152
`\ckv@colon` | 87, 105
`\ckv@comma` | 146
`\ckv@delimname` | 91, 105-109
`\ckv@dummy` | 81, 140, 163
`\ckv@dummyyc` | 82
`\ckv@key` | 147, 182
`\ckv@keyonlyfalse` | 145, 149
`\ckv@keyonlytrue` | 150
`\ckv@next` | 156, 169, 172, 181, 187, 189
`\ckv@parsecasma` | 148, 164, 176
`\ckv@removecomma` | 147, 150
`\ckv@rgiii` | 153, 163
`\ckv@rgiv` | 153, 155
`\ckv@scratch` | 111, 137, 168, 180, 185, 186
`\ckv@skipfalse` | 104, 151
`\ckv@skiptrue` | 158
`\ckv@skOptts` | 92, 93, 95, 96, 98, 101, 102, 157, 170, 173
`\ckv@TAB` | 85, 130, 134, 138

## D

`\DeclareDelimiter` | 75
`\DeclareDelimiter@i` | 75, 76

## E

`\egroup` | 86, 144

## I

`\iftcxv@keyonly` | 145, 165, 177
`\iftcxv@skip` | 151, 162
`\IfFileExists` | 117
`\InputIfFileExists` | 106

## M

`\makeatletter` | 127
`\makeatother` | 141

## N

`\newwrite` | 121

## O

`\obeyspaces` | 85
`\openout` | 126

## P

`\PackageInfo` | 107, 108, 117, 119

## R

`\RequirePackage` | 2

## T

`\thisxkvF@mily` | 99, 103, 111, 135
`\thisxkvV@lues` | 111, 136, 154, 157, 161
5 Change History

v1.0 (2016/12/20)
  General: Date of first upload to CTAN .......... 2

v1.1 (2017/01/03)
  General: Try to detect if the expected delimiter
  is present at all .................................. 6

v1.1a (2017/01/03)
  General: Change in syntax, use parentheses
  rather than brackets .............................. 2

v1.1c (2017/02/17)
  General: Enclose #3 in braces .................. 7