conv-xkv: Convert xkeyval format style

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

processed March 20, 2017

Contents

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 ⟨*name*⟩ is now delimited by **parentheses**, rather than the standard brackets. This is to be able to detect ⟨*name*⟩ when the full syntax of `\setkeys` is used:

`\setkeys*[`⟨*prefix*⟩`]{`⟨*families*⟩`}[`⟨*na*⟩`]{`⟨*keys*⟩`}`

The syntax for `\cxkvsetkeys` shall be

`\cxkvsetkeys(`⟨*name*⟩`)*[`⟨*prefix*⟩`]{`⟨*families*⟩`}[`⟨*na*⟩`]{`⟨*keys*⟩`}`

The conv-xkv package does nothing with xkeyval arguments `*[`⟨*prefix*⟩`]` and `[`⟨*na*⟩`]` 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 ⟨*key*⟩⟨*delim*⟩⟨*value*⟩ to ⟨*key*⟩=⟨*value*⟩.

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 `\cxkvsetkeys{myfam}{fname:Don,lname:Story}` works as does `\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 `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.

```
\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.

```
\begingroup
  \@makeother\‘%
  \@makeother\=%
  \edef\x{%
    \edef\noexpand\x{%
      \endgroup
      \noexpand\toks@{%
        \catcode 96=\noexpand\the\catcode‘\noexpand\‘\relax
        \catcode 61=\noexpand\the\catcode‘\noexpand\=\relax
      }%
    }%
    \noexpand\x
  }%
\x
\@makeother\‘
\@makeother\=
\def\ckv@SetCatcodes{%
```

```
  \@makeother\‘%
  \@makeother\=%
  \@makeother\~%
  \catcode‘\$=3 %
  \catcode‘\&=4 %
  \catcode‘\^=7 %
  \catcode‘\_=8 %
  \@makeother\|%
  \@makeother\:%
  \@makeother\(%
  \@makeother\)%
  \@makeother\[%
  \@makeother\]%
  \@makeother\/%
  \@makeother\!%
  \@makeother\<%
  \@makeother\>%
  \@makeother\.%
  \@makeother\;%
  \@makeother\+%
  \@makeother\-%
  \@makeother\"%
  \@makeother\’%
}
\begingroup
  \def\x#1{\catcode‘\noexpand#1=\the\catcode‘#1\relax}%
  \xdef\ckv@RestoreCatcodes{%
    \the\toks@
    \x\~%
    \x\$%
    \x\&%
    \x\^%
    \x\_%
    \x\|%
    \x\:%
    \x\(%
    \x\)%
    \x\[%
    \x\]%
    \x\/%
    \x\!%
    \x\<%
    \x\>%
    \x\.%
    \x\;%
    \x\+%
    \x\-%
    \x\"%
    \x\’%
  }%
```

```
\endgroup
\ckv@SetCatcodes
```

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}{:}
```

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

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

`\DeclareDelimiter` 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}}
\@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`

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

```
\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}
    {\def\cxkv@skOpts{}\cxkvsetkeys@iii}}
\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.

```
\def\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}%
```

```
    \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}%
        \cxkvarg2,\cxkvarg3\string\@nil{c^^J\cxkv@TAB
        \string\cxkv@cnvrtDelimniiEquali{\w}{\cxkvarg1}%
        {\cxkvarg2}{\cxkvarg3}}}}
    \uppercase{\immediate\write\cxkv@write{\string\csarg\string\def
        {\x}\cxkvarg1\cxkvarg2{c^^J\cxkv@TAB
        \string\def\string\thisxkvF@mily{\cxkvarg1}\string
        \def\string\thisxkvV@lues{\cxkvarg2}\string
        \let\string\cxkv@scratch\string\@empty\string
        \cxkv@tmptoks={}c^^J\cxkv@TAB
        \string\@nameuse{\y}\cxkvarg2,%
        \cxkv@dummy\@nameuse{\z}\cxkv@dummy,\string\@nil}}}
    \immediate\write\cxkv@write{\string\makeatother}%
    \immediate\closeout \cxkv@write
}%
\egroup}
```

`\cxkv@cnvrtDelimniiEquali` continues `\cxkv@cnvrtDelimniiEqual`. 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@cnvrtDelimniiEquali#1#2#3#4{%
    \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`.

```
    \ifx\thisxkvV@lues\@empty\else
        \ifx\cxkv@rgiv\@empty
            \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@parsecomma#2,\@nil
        \ifcxkv@keyonly
            \edef\cxkv@tmp{\the\cxkv@tmptoks,\@rgi}%
            \cxkv@tmptoks=\expandafter{\cxkv@tmp}%
            \edef\cxkv@scratch{\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{\the\cxkv@tmptoks,\@rgi}%
            \cxkv@tmptoks=\expandafter{\cxkv@tmp}%
            \edef\cxkv@scratch{\the\cxkv@tmptoks}%
            \edef\cxkv@next{\noexpand
                \@nameuse{cxkv@cnvrtDelimniiEqual-#1}\cxkv@key
                \@nameuse{kvdelim-#1}#3,#4\noexpand\@nil}
        \else
```

(2017/02/17) Enclose `#3` in braces

```
            \cxkv@tmptoks=\expandafter{\cxkv@scratch,#2={#3}}%
            \edef\cxkv@scratch{\the\cxkv@tmptoks}%
            \def\cxkv@next{%
                \@nameuse{cxkv@cnvrtDelimniiEqual-#1}#4\@nil}\fi
    \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.

5 Change History