The \texttt{showkeys} package\textsuperscript{*}

David Carlisle        Morten Høgholm
2014/10/28

This file is maintained by the \LaTeX{} Project team. Bug reports can be opened (category \texttt{tools}) at \url{https://latex-project.org/bugs.html}.

1 Introduction

\texttt{showkeys.sty} modifies the \texttt{\label}, \texttt{\ref}, \texttt{\pageref}, \texttt{\cite}, and \texttt{\bibitem} commands so that the ‘internal’ key is printed. The package tries hard to position these labels so that the formatting of the rest of the document is unchanged. \texttt{\label} and \texttt{\bibitem} cause the key to appear in a box either in the margin, or in a \TeX{} box of zero width, which may possibly over-print other text. The \texttt{\ref}, \texttt{\pageref} and \texttt{\cite} commands print their arguments in small type, raised just above the line, like this: \texttt{sec:intro}. This package works with the \texttt{fleqn} option, the packages in the AMS-\LaTeX{} collection, and the \texttt{varioref}, \texttt{natbib} and \texttt{harvard} packages.

2 Package Options

Some people have commented that the printing of the \texttt{\ref} and \texttt{\cite} keys is less useful than the printing of the \texttt{\label} keys and so \texttt{showkeys} now supports two options that can be given in the \texttt{\usepackage} command:

\begin{itemize}
  \item \texttt{notref} to stop the redefinition of \texttt{\ref} and \texttt{\pageref}, and related commands from the \texttt{varioref} package.
  \item \texttt{notcite} to stop the redefinition of \texttt{\cite} and related commands from the \texttt{harvard} and \texttt{natbib} packages.
\end{itemize}

So if the package is loaded with \texttt{\usepackage[notref]{showkeys}} then \texttt{\ref} will have its standard definition, but \texttt{\label} will print its key argument (usually in the margin).

If you find the printed keys distracting, but don’t want to use the above options to stop them altogether you may use:

\begin{itemize}
  \item \texttt{color} Print the keys in a distinguishing colour. The default value is a light grey.
\end{itemize}

\textsuperscript{*}This file has version number v3.17, last revised 2014/10/28.
The colours may be changed by redefining the following two colours after the package is loaded. \texttt{refkey} (also used for \texttt{\cite}) and \texttt{labelkey} (also used for \texttt{\bibitem}). The defaults are:

\begin{verbatim}
\definecolor{refkey}{gray}{.75}
\definecolor{labelkey}{gray}{.75}
\end{verbatim}

If this option is used the \texttt{color} package will be loaded.

The package accepts two further options.

\texttt{final} to suppress the action of this package, for ‘final’ versions.

\texttt{draft} the normal behaviour of this package.

Clearly there is not much point in entering the \texttt{final} option directly in the \texttt{\usepackage} command, as just not loading this package would have the same effect, and execute more quickly, however the \texttt{final} option may be useful as it may be used once in the \texttt{documentclass} command to affect any number of packages that may be loaded. The \texttt{draft} option does not do anything, but is there to honour an informal convention that packages have these options in pairs.

You can also control the appearance of the typeset label with the command \texttt{\showkeyslabelformat}, which takes one argument. The default is

\begin{verbatim}
\providecommand*{\showkeyslabelformat}[1]{% 
  \fbox{\normalfont\small\ttfamily#1}}
\end{verbatim}

The command is called inside a group so you can put in local modifications of \texttt{\fboxsep}, for instance, without them leaking to the rest of the document.

\section{More Examples}

The only other similar package that I could find in the macro index, \texttt{mimi}\cite{4}, was \texttt{showlabels.sty}\cite{1}. After the first draft of this package was written, I found \texttt{anon:sk}\cite{2} on my local installation! I think the current package is more robust than \cite{2}, but I thought that \texttt{showkeys} was rather a good name, so I have stolen it for this file.

1. This has \texttt{\label} immediately after \texttt{\item}.
2. This has the \texttt{\label} at the end.

A minipage:-

\begin{verbatim}
\begin{minipage}{\textwidth}
\texttt{\mke} 1. This has \texttt{\label} immediately after \texttt{\item}.
2. This has the \texttt{\label} at the end.
\end{minipage}
\end{verbatim}

Displayed math (without \texttt{equation} counter).

\begin{verbatim}
0 = 0^{\texttt{disp}}
\end{verbatim}
Some text referring to the maths on page 1, and the item 11.

If showkeys thinks that the current environment is going to produce an “equation number”, then it does not show the label where the \label command occurs, but tries to put it in the margin, as shown with equation 1. The package ‘knows’ about the standard equation and eqnarray environments, and also all the numbered alignment environments offered by the AMS\LaTeX package, amsmath.

\begin{align}
1 &= 1 & (1) \quad \text{eq:xx} \\
2 &= 2 & (2) \quad \text{eqnara:a} \\
3 &= 3 \\
4 &= 4 & (3) \quad \text{eqnar:b}
\end{align}

Within a figure environment, the \label must not come before the \caption command. If you place \label inside the argument of \caption the label will be shown like this:

Figure 1: Within the caption argument. \text{cap:a}

If you place \label immediately after the \caption command it will be shown like this:

Figure 2: Immediately after the caption argument. \text{cap:b}

If you place the \label command at some random point after the \caption command, it may be shown like:

Figure 3: In vertical mode not immediately after a box. \text{cap:c}

References

[1] Gil Neiger, showlabels.sty, Undated package, similar to this one, but shows labels inline, affecting the formatting of the document.


[3] David M. Jones, \LaTeX Macro Index, A catalogue of \LaTeX macros, including \LaTeX packages, available from all good \LaTeX archives.

4 The Macros

1 (*package)

First we handle the options. Normally all related commands are defined to show their ‘keys’. But since v3.03 one can specify:

\footnote{Actually \marginpar is not used at all in this package now.}
notref to stop the redefinition of \ref (and \pageref, and related commands from varioref package).
notcite to stop the redefinition of \cite and related commands from the harvard and natbib packages.

\DeclareOption{notref}{\let\SK@ref\@empty}
\DeclareOption{notcite}{\let\SK@cite\@empty}

\SK@refcolor Colour commands. Normally no-op.
\SK@labelcolor Color option loads the color package and defines the colours. Delayed to the end of the package as package loading not allowed in this option section.
\let\SK@refcolor\relax
\let\SK@labelcolor\relax

\DeclareOption{color}{\AtEndOfPackage{%
\RequirePackage{color}%
\definecolor{refkey}{gray}{.75}%
\definecolor{labelkey}{gray}{.75}%
\def\SK@refcolor{\color{refkey}}%
\def\SK@labelcolor{\color{labelkey}}}}

Allow final to be specified in the document class options to suppress the loading of this package.
\DeclareOption{final}{%
\providecommand\showkeyslabelformat[1]{%%
\endinput}}
\DeclareOption{draft}{%}
\ProcessOptions

\SK@label The saved original definitions
\SK@bibitem The new definition, print the argument, and then do the old definition.
\let\SK@label\label
\let\SK@bibitem\@bibitem
\let\SK@lbibitem\@lbibitem

\label For \bibitem, position the showkeys code as for a standard list with \item and \label.
\let\SK@\relax
\def\SK@#1#2{\protected@edef\@tempa{#2}\expandafter#1\meaning\@tempa\SK@}

\SK@ Grab hold of #2 via \meaning so characters like & and ^ do not cause problems later, and pass the result on to the command #1.
\def\SK@#1#2{%
\protected@edef\@tempa{#2}\protected@edef\@tempb{#2}%
\expandafter\@tempa\meaning\@tempb\SK@}
\showkeyslabelformat

\providecommand*{\showkeyslabelformat[1]}{\fbox{\normalfont\small\ttfamily#1}}

\SK@label Strip off the initial segment of the \texttt{meaning} output, and then put the rest either in a \texttt{marginpar} or in a box of size 0pt, hopefully not disturbing the surrounding text.

\def\SK@label#1>{\SK@labelcolor\showkeyslabelformat{#2}}%
\ifvmode
  \if@inlabel
    If the \texttt{label} is straight after \texttt{item} (\bibitem is handled by this case as well) then the item label has not been added to the page yet. It is hanging around in the box \texttt{@labels} waiting for the paragraph to start. So just need to attach the label to this box.
  \global\setbox\@labels\hbox{\llap{\SK@lab\SK@lab@relax\kern\@totalleftmargin\kern\marginparsep}}%
  \else
    If we insert a box into the main vertical list, do not want to change \texttt{\prevdepth} as that would affect vertical spacing in the document. (The box itself should not cause any difference in break points as there is a node there anyway coming from the \texttt{\write} to the aux file.
    \dimen@\prevdepth
    \nointerlineskip
    The inner vertical mode cases are mainly designed to do the right thing with float captions, but seem to work OK in other cases as well.
    \ifinner
      \skip@\lastskip\unskip
    \else
      \skip@\lastskip\unskip
    \fi
  \else
    In outer vertical mode, previously used a \texttt{\vadjust} at the start of the next paragraph (and before that used \texttt{\marginpar}). These methods sometimes cause extra
space, e.g., if paragraph starts with a math display, so now just insert the box directly, taking care not to change `\prevdepth`.

\llap{\SK@lab\SK@lab@relax\kern\marginparsep}\
\fi

Restore `\prevdepth`.
\prevdepth\dimen@
\fi
\else

If we are in an numbered equation-style environment, do nothing as the code to print the number will also print the label, otherwise just stick the label at the current point, in a box of zero dimensions.
\csname SK@\@currenvir\endcsname
\ifSK@equation\else
\ifmmode
\SK@labx
\else
\SK@labx
\else
\else

Inner horizontal mode. Not much we can do, just stick it here.
\ifinner
\rlap{SK@lab}
\else
In outer horizontal mode use `\vadjust` to get to the margin.
\vadjust{\llap{\SK@lab\kern\marginparsep}%%%%%%%%
\SK@lab@relax
\fi
\fi
\fi}

Firstly we grab `\@eqnnum`.
\AtBeginDocument{%}
\let\SK@eqnnum\@eqnnum

Then check for `amsmath` where we grab the internal commands `\tagform@` and `\maketag@@@`. Redefine them and redefine `\@eqnnum` as well.
\@ifpackageloaded{amsmath}{%
\let\SK@tagform@\tagform@
\let\SK@maketag@@@\maketag@@@
\iftagsleft@
\def\tagform@#1{%
\ifx\df@label\@empty
\SK@lab@relax
\else
\expandafter\SK@@label\meaning\df@label\SK@
\fi
\else
\expandafter\SK@@label\meaning\df@label\SK@
\fi
\else
\expandafter\SK@@label\meaning\df@label\SK@
\fi
\else
\expandafter\SK@@label\meaning\df@label\SK@
\fi
\fi
Almost the same for tags on the right, except we use `\rlap` and typeset it after the tag.

```
def \tagform@#1{% 
  \ifx \df@label \@empty 
    \else 
        \expandafter \SK@@label \meaning \df@label \SK@ 
  \fi
  \SK@tagform@{#1} %
  \rlap {\kern \marginparsep \SK@lab} \SK@lab@relax}
```

If `amsmath` wasn’t loaded we check explicitly if the `leqno` option was used in `\documentclass` and redefine accordingly.

```
{% 
  \@ifundefined {ver@leqno.clo}{% 
    \def \@eqnnum \SK@eqnnum \rlap {\kern \marginparsep \SK@lab} \SK@lab@relax\% 
  }{% 
    \def \@eqnnum {\llap {\SK@lab \kern \displaywidth \kern \marginparsep} \SK@lab@relax \SK@eqnnum\}%
  }%
}\def \SK@labx {Print the label, and then globally reset the print command to `\relax`.}
\def \SK@labx {\rlap {\SK@lab \global \let \SK@lab \relax}
\def \SK@labx {Clear the label.
The following environments print an equation number, so \texttt{\textbackslash label} should not print its argument at the point where it appears. Note this will fail to show the label if you are in an \texttt{eqnarray} environment, and use \texttt{\textbackslash label} together with \texttt{\textbackslash no\textbackslash number}

This might just about make sense if you are going to use \texttt{\textbackslash pageref}, but that is too bad...

\begin{verbatim}
newif\ifSK@equation
\let\SK@equation\SK@equationtrue
\let\SK@eqnarray\SK@equationtrue
\end{verbatim}

When the AMS packages are loaded showkeys assumes environments work ‘The AMS way’ However \texttt{eqnarray} (unlike \texttt{equation}) is not redefined, so here we need to remove some of the AMS hacks.

\begin{verbatim}
toks@expandafter{eqnarray}
edef\eqnarray{\let\noexpand\tagform@\noexpand\SK@tagform@\the\toks@}
\end{verbatim}

The AMS environments

\begin{verbatim}
\let\SK@align\SK@equationtrue
\let\SK@alignat\SK@equationtrue
\let\SK@xalignat\SK@equationtrue
\let\SK@xxalignat\SK@equationtrue
\let\SK@gather\SK@equationtrue
\let\SK@multline\SK@equationtrue
\let\SK@flalign\SK@equationtrue
\end{verbatim}

Starred versions of the AMS environments.

\begin{verbatim}
\expandafter\let\csname SK@align*\endcsname\SK@equationtrue
\expandafter\let\csname SK@alignat*\endcsname\SK@equationtrue
\expandafter\let\csname SK@gather*\endcsname\SK@equationtrue
\expandafter\let\csname SK@multline*\endcsname\SK@equationtrue
\expandafter\let\csname SK@equation*\endcsname\SK@equationtrue
\end{verbatim}

This macro redefines a command \#1. The new definition can make use of the old definition as \texttt{\SK@old\textbackslash name}. If \#1 is really a \texttt{\protect\textbackslash ed} command with the real definition in a ‘space’ command then the ‘space’ version is used as the old definition. Need to test this for each command as some package may have changed the status of a command to being ‘protected’. The new definition is made as if with \texttt{\textbackslash DeclareRobustCommand}, but with \texttt{\textbackslash def} syntax for the argument specification.

\begin{verbatim}
def\SK@def#1{\edef\@tempa{\expandafter\@gobble\string#1}\
\@ifundefined{\@tempa\space}{\expandafter\let\csname SK@\@tempa\endcsname#1}
{\expandafter\let\csname\@tempa\space\endcsname}\
\expandafter\def\expandafter#1\expandafter{\expandafter\protect\csname\@tempa\space\endcsname}
\expandafter\def\csname\@tempa\space\endcsname}
\end{verbatim}

The next section redefines \texttt{\textbackslash ref} and \texttt{\textbackslash pageref} (unless the \texttt{notref} option was given).

\begin{verbatim}
\ifx\SK@ref@empty
\end{verbatim}

8
Even if `notref` option is used, need to fudge the `varioref` commands as they use `\label` internally.

\begin{verbatim}
\@ifpackageloaded{varioref}{%
  \let\label\SK@label
  \let\ref\SK@ref
  \let\pageref\SK@pageref
  \leavevmode
  \unskip
  \SK@@@@\ref{#1}{#2}{#3}
}
\end{verbatim}

Save the redefinition to `\begin{document}` so that this package can work with packages that redefine `\cite`. Tested with harvard and natbib packages. Also add code at this point to support varioref.

\begin{verbatim}
\@ifpackageloaded{varioref}{%
  \let\label\SK@label
  \let\ref\SK@ref
  \let\pageref\SK@pageref
  \leavevmode
  \unskip
  \SK@@@@\ref{#3}{#1}{#2}
}
\end{verbatim}

Now redefine `\cite` unless `notcite` option given.

\begin{verbatim}
\ifx\SK@cite\@empty
  \AtBeginDocument{%
    \ifx\HAR@checkdef\@undefined
      \expandafter\let\expandafter
      \SK@HAR@bi\csname\string\harvarditem\endcsname
      \expandafter\def\csname\string\harvarditem\endcsname[#1]#2#3#4{% #1}{#2}{#3}{#4}
    \fi
  }
\else
  \AtBeginDocument{%
    \ifx\HAR@checkdef\@undefined
      \expandafter\let\expandafter
      \SK@HAR@bi\csname\string\harvarditem\endcsname
      \expandafter\def\csname\string\harvarditem\endcsname[#1]#2#3#4{% #1}{#2}{#3}{#4}
    \fi
  }
\fi
\end{verbatim}

Standard (non-harvard) support, including extra cite commands from `natbib` and `cite`.
If `cite` or `overcite` is being used, redefine \citen rather than \cite so as not to spoil the space and punctuation calculations done by those packages.

```latex
\ifx\citen\@undefined
\SK@def\@citex[#1]{#2}{\SK@citex[{#1}]{#2}}%
\else
\SK@def\citen#1{\SK@\SK@@ref{#1}\SK@citen{#1}}%
\fi
```

In the harvard style do not redefine individual cite commands. Just redefine one internal command that is used in all the citation forms.

```latex
\def\SK@citex[#1]{\SK@\SK@@ref{#2}\SK@@citex[{#1}]{#2}}
```

\SK@@ref This is much simpler than the printing of the label, as we know that we can be in horizontal mode. Note extra group for colour safety.

```latex
\def\SK@citex[#1]{\SK@\SK@ref[#2]\SK@citex[#1]{#2}}
```

\SK@ref