Le package showkeys*

David Carlisle

Traduction française par Jean-Pierre Drucbert†

1997/06/12

sec+intro

1 Introduction

Le package showkeys modifie les commandes `\label`, `\ref`, `\pageref`, `\cite` et `\bibitem` pour que la clé « interne » soit imprimée. Ce package fait de grands efforts pour placer ces labels de façon que la mise en page du reste de votre document reste inchangée. `\label` et `\bibitem` font que le label apparaîtra dans une boîte soit dans la marge, soit dans une boîte TEX de largeur nulle, qui peut éventuellement se surcharger à du texte. Les commandes `\ref`, `\pageref` et `\cite` impriment leurs arguments en petits caractères, juste au dessus de la ligne, comme ceci : sec+intro.

Ce package fonctionne aussi avec l'option de classe fleqn, les packages qui font partie d'AMSLATEX, et avec les packages varioref, natbib et harvard.

options

2 Options du package

Certaines personnes ont dit que l'impression des clés `\ref` et `\cite` était moins utile que celle des clés `\label` et donc maintenant le package showkeys accepte deux options pouvant être données dans la commande `\usepackage` :

notref pour empêcher la redéfinition de `\ref` et `\pageref`, et des commandes associées du package `varioref`.

notcite pour empêcher la redéfinition de la commande `\cite` et des commandes associées dans les packages `harvard` et `natbib`.

Donc si ce package est chargé par `\usepackage[notref]{showkeys}`, alors `\ref` aura sa définition standard, mais `\label` imprimera la clé donnée en argument (habituellement dans la marge).

Si vous estimez que l'impression des clés vous distrait, mais ne voulez pas utiliser les options ci-dessus pour la supprimer, vous pouvez aussi utiliser :

color Imprime les clés dans une couleur distincte. La couleur par défaut est un gris pâle.

Les couleurs peuvent être modifiées en redéfinissant les deux couleurs suivantes *après* chargement du package : `refkey` (également utilisée pour `\cite`) et `labelkey` (également utilisée pour `\bibitem`). Les valeurs par défaut sont :

*Ce fichier a le numéro de version v3.12, révisé le 1997/06/12.

†Dernière mise à jour le 20/01/2000

```
\definecolor{refkey}{gray}{.75}
\definecolor{labelkey}{gray}{.75}
```

Si cette option est utilisée, le package `color` sera automatiquement chargé. `showkeys` accepte deux autres options.

final pour supprimer l'action de ce package, pour la version « finale » du document.

draft pour un comportement normal de ce package.

Il est clair qu'il est sans intérêt de donner directement l'option `final` dans la commande `\usepackage`, car le fait de simplement ne pas charger le package aurait le même effet, et s'exécuterait plus rapidement ; cependant l'option `final` peut être utile si elle est utilisée une fois pour toutes dans la commande `\documentclass` afin d'affecter tous les packages qui seront chargés. L'option `draft` ne fait rien du tout, mais elle est là pour honorer une convention informelle qui veut que les packages aient ces options par paires.

3 Plus d'exemples

examples

Le seul autre package similaire que j'ai pu trouver dans l'index des commandes, [3], était `showlabels.sty`, [1]. Après avoir écrit la première épreuve de package, j'ai trouvé [2] dans mon installation locale ! Je pense que mon package est plus robuste que [2], mais comme j'ai trouvé que `showkeys` était plutôt un bon nom, je l'ai volé pour ce package.

e^1 1. Cet item a une commande `\label` immédiatement derrière `\item`.

e^2 2. Celui-ci a la commande `\label` à la fin.

Une `minipage` :-

À l'intérieur d'environnements tels que cette `minipage`, nous ne pouvous pas utiliser de notes marginales (`\marginpar`[1]), donc l'aspect est légèrement différent. Voici de nouveau cet environnement `enumerate` :

e^10 1. Cet item a une commande `\label` immédiatement derrière `\item`.

e^20 2. Celui-ci a la commande `\label` à la fin.

Mathématiques en hors-texte (sans compteur `equation`).

$$0 = 0 \quad \text{disp}$$

Du texte faisant référence à la page 2, et à l'item 1.

Si `showkeys` pense que l'environnement courant va produire un « numéro d'équation », alors il ne montre pas le label là où se trouve la commande `\label`, mais essaie de le placer dans la marge, comme cela se voit pour l'équation 1. Le package « connaît » les environnements standard `equation` et `eqnarray`, et connaît aussi tous les environnements d'alignement numérotés offerts par le package AMSLaTeX, `amsmath`.

[1] En réalité, `\marginpar` n'est actuellement plus utilisé dans ce package.

$$1 = 1 \tag{1}$$ eq+xx

$$\begin{aligned} 2 &= 2 \qquad (2) \\ 3 &= 3 \\ 4 &= 4 \qquad (3) \end{aligned}$$ eqnar+a eqnar+b

À l'intérieur d'un environnement `figure` ou `table`, le `\label` ne doit pas être placé avant la commande `\caption`. Si vous placez `\label` à l'intérieur de l'argument de `\caption`, le label aura cet aspect :

Fig. 1 – À l'intérieur de l'argument de la légende. cap+a

Si vous placez `\label` immédiatement après la commande `\caption`, il sera montré comme ceci :

Fig. 2 – Immédiatement après l'argument de la légende. cap+b

Si vous placez la commande `\label` n'importe où après la commande `\caption`, il pourra être montré comme ceci :

Fig. 3 – En mode vertical pas immédiatement derrière une boîte.

cap+c

Références

GN+sl [1] Gil Neiger, *showlabels.sty*, Un package, sans date, similaire à celui-ci, mais montrant les labels en ligne, ce qui affecte la mise en page du document.

anon+sk [2] Anonyme, *showkeys.sty*, Paquetage, daté du 14 mai 1988. Très similaire à celui-ci, utilise aussi `\marginpar` en mode vertical externe.

DMJ+mi [3] David M. Jones, *TEX Macro Index*, Un catalogue de macros TEX, comprenant des fichiers de packages LATEX, disponible sur toutes les bonnes archives TEX.

4 The Macros

1 ⟨*package⟩

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

`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.

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

\SK@refcolor
\SK@labelcolor

Colour commands. Normally no-op.

```
4 \let\SK@refcolor\relax
5 \let\SK@labelcolor\relax
```

`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.

```
\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 supress the loading of this package.

```
\DeclareOption{final}{\endinput}
\DeclareOption{draft}{}
\ProcessOptions
```

`\SK@label` `\SK@bibitem` `\SK@lbibitem` The saved original definitions

```
\let\SK@label\label
\let\SK@bibitem\@bibitem
\let\SK@lbibitem\@lbibitem
```

`\label` The new definition, print the argument, and then do the old definition.

```
\def\label#1{%
  \@bsphack
  \SK@\SK@@label{#1}%
  \begingroup
    \SK@label{#1}%
  \endgroup
  \@esphack}
```

`\@bibitem` `\@lbibitem` For `\bibitem`, position the showkeys code as for a standard list with `\item` and `\label`.

```
\def\@bibitem#1{%
  \SK@bibitem{#1}\SK@\SK@@label{#1}\ignorespaces}
\def\@lbibitem[#1]#2{%
  \SK@lbibitem[#1]{#2}\SK@\SK@@label{#2}\ignorespaces}
```

`\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}%
  \expandafter#1\meaning\@tempa\SK@}
```

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

```
\def\SK@@label#1>#2\SK@{%
```

Need to work globally as in some cases like alignments, and `fleqn`, the counter will be printed in a different group to the `\label` command.

```
  \gdef\SK@lab{\smash{\SK@labelcolor\fbox{%
                      \normalfont\small\ttfamily#2}}}%
  \ifvmode
    \if@inlabel
```

If the \label is straight after \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 \@labels waiting for the paragraph to start. So just need to attatch the label to this box.

```
      \global\setbox\@labels\hbox{%
        \llap{\SK@lab\SK@lab@relax
              \kern\@totalleftmargin\kern\marginparsep}%
        \box\@labels}%
    \else
```

If we insert a box into the main vertical list, do not want to change \prevdepth as that would afect 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 \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
```

In inner vertical mode, attach the label to the right of the immediately preceding box, if it is a box before the current point. Otherwise just put it in a box of zero dimensions, with no interline skip. (This may slightly move the surrounding text (but perhaps not now that \prevdepth is restored.)

```
        \advance\skip@\lastskip\unskip
        \setbox\z@\lastbox
        \ifvoid\z@
          \llap{\SK@lab\SK@lab@relax\kern\marginparsep}%
        \else
          \hbox{\box\z@\kern\marginparsep\SK@labx}%
        \fi
        \vskip\skip@
      \else
```

In outer vertical mode, previously used a \vadjust at the start of the next paragraph (and before that used \marginpar). These methods sometimes cause extra space, eg 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
```

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}}%
        \fi
        \SK@lab@relax
      \fi
    \fi
  \fi}
```

`\iftagsleft@` Make sure that this AMSLaTeX command really is an `\if..`

```
\AtBeginDocument{%
  \let\SK@eqnnum\@eqnnum
  \def\@tempa{\let\iftagsleft@\iffalse}%
  \ifx\iftagsleft@\undefined\@tempa\fi%
```

`\tag@form@` `\eqnnum` Perhaps if `leqno` is operative, I should define `\@eqnnum` with the 'left' version, but it does not really matter.

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

`\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}
```

\SK@lab@relax Clear the label.

```
\def\SK@lab@relax{\global\let\SK@lab\relax}\SK@lab@relax
```

\SK@equation
\SK@eqnarray The following environments print an equation number, so `\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 `eqnarray` environment, and use `\label` together with `\nonumber` This might just about make sense if you are going to use `\pageref`, but that is too bad...

```
\newif\ifSK@equation
\let\SK@equation\SK@equationtrue
\let\SK@eqnarray\SK@equationtrue
```

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

```
\toks@\expandafter{\eqnarray}
\edef\eqnarray{\let\noexpand\tagform@\noexpand\SK@tagform@\the\toks@}
```

\SK@align The AMS environments
\SK@alignat
\SK@xalignat
\SK@xxalignat
\SK@gather
\SK@multline
\SK@flalign

```
\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
```

\SK@def This macro redefines a command `#1`. The new definition can make use of the old definition as `\SK@`*old name*. If `#1` is really a `\protect`'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 `\DeclareRobustCommand`, but with `\def` syntax for the argument specification.

```
\def\SK@def#1{%
  \edef\@tempa{\expandafter\@gobble\string#1}%
  \@ifundefined{\@tempa\space}%
    {\expandafter\let\csname SK@\@tempa\endcsname#1}%
    {\expandafter\let\csname SK@\@tempa\expandafter\endcsname
                          \csname\@tempa\space\endcsname}%
  \expandafter\def\expandafter#1\expandafter{%
        \expandafter\protect\csname\@tempa\space\endcsname}%
  \expandafter\def\csname\@tempa\space\endcsname}
```

The next section redefines `\ref` and `\pageref` (unless the `notref` option was given).

```
\ifx\SK@ref\@empty
```

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

```
\AtBeginDocument{%
  \ifx\vpageref\@undefined\else
    \SK@def\@@vpageref#1[#2]#3{{%
```

```
      \let\label\SK@label
      \SK@@@vpageref#1[#2]{#3}}}%
    \DeclareRobustCommand\vref[1]{%
      \unskip~\ref{#1}%
      {\let\label\SK@label
       \SK@@@vpageref\unskip[\unskip\space]{#1}}}%
  \fi}
\else
```

\ref
\pageref

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.

```
\AtBeginDocument{%
  \SK@def\ref#1{\SK@\SK@@ref{#1}\SK@ref{#1}}%
  \SK@def\pageref#1{\SK@\SK@@ref{#1}\SK@pageref{#1}}%
  \ifx\vpageref\@undefined\else
```

varioref support.

```
    \SK@def\@@vpageref#1[#2]#3{{%
      \let\label\SK@label\let\ref\SK@ref\let\pageref\SK@pageref
      \leavevmode\unskip\SK@\SK@@ref{#3}\SK@@@vpageref#1[#2]{#3}}}%
    \DeclareRobustCommand\vref[1]{%
      \unskip~\ref{#1}%
      {\let\label\SK@label\let\ref\SK@ref\let\pageref\SK@pageref
       \SK@@@vpageref\unskip[\unskip\space]{#1}}}%
  \fi}
\fi
```

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

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

\cite

```
\AtBeginDocument{%
  \ifx\HAR@checkdef\@undefined
```

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.

```
    \ifx\citen\@undefined
      \SK@def\cite#1#{\SK@citea{#1}}%
    \else
      \SK@def\citen#1{\SK@\SK@@ref{#1}\SK@citen{#1}}%
    \fi
    \SK@def\citeauthor#1{\SK@\SK@@ref{#1}\SK@citeauthor{#1}}%
```

```
    \SK@def\citefullauthor#1{\SK@\SK@@ref{#1}\SK@citefullauthor{#1}}%
    \SK@def\citeyear#1{\SK@\SK@@ref{#1}\SK@citeyear{#1}}%
  \else
```

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

```
    \SK@def\HAR@checkdef#1#2{%
      \expandafter\SK@\expandafter\SK@@ref\expandafter{#1}%
      \SK@HAR@checkdef{#1}{#2}}%
      \expandafter\let\expandafter
         \SK@HAR@bi\csname\string\harvarditem\endcsname
      \expandafter\def\csname\string\harvarditem\endcsname[#1]#2#3#4{%
        \SK@HAR@bi[#1]{#2}{#3}{#4}\SK@\SK@@label{#4}}%
  \fi}
\def\SK@citea#1#2{%
  \SK@\SK@@ref{#2}\SK@cite#1{#2}}
\fi
```

`\SK@@ref` This is much simpler than the printing of the label, as we know that we can be in horizontal mode.

```
\def\SK@@ref#1>#2\SK@{%
  \leavevmode\vbox to\z@{%
    \vss
    \SK@refcolor
    \rlap{\vrule\raise .75em%
       \hbox{\underbar{\normalfont\footnotesize\ttfamily#2}}}}}
```

⟨/package⟩