Semi-Manual Grid Setting Using gridset *

Markus Kohm

2024-08-17 v0.4

Abstract

Grid setting — also known as strict in-register setting — is something, that should be done for a lot of documents but is not easy using LaTeX. Package gridset helps to get the information needed for grid setting. It does not implement auto grid setting, but there is a command `\vskipnextgrid`, that moves to the next grid position. This may be enough under some circumstances. In other circumstances it may fail. So gridset is only one more step for grid setting not a complete solution.

Important Note: This package should have been never released, because it was only a very quick implementation of an idea. You should not use it for any productive purpose. It has been made for testing only. I would prefer to retire it from any distribution. Nevertheless I know few persons using the package. So it will be still there but without any support!

Contents

1 User Manual

`\gridinterval` This macro contains a number without unit! The number is the distance between two grid lines in unit 'scale points' (sp). You may set it so another value using, e.g.

```
\newlength{\⟨name of your length⟩}
\setlength{\⟨name of your length⟩}{⟨new length value⟩}
\newcounter{\⟨name of your counter⟩}
\setcounter{\⟨name of your counter⟩}{\⟨name of your length⟩}
\edef\gridinterval{\⟨name of your length⟩}
```

`\gridbase` This macro contains an integer number, that represents the y-coordinate of the

*This is an alpha version! Don't use it! Only test it! There's no support and everything may change!

upper start of the grid. If you want to change it, just save a position and `\edef` the `\gridbase` to the y-pos of that position.

Most time you don't need to change `\gridinterval` and `\gridposition`, because they are initialized to a base line grid at start of first page. Because of this, it doesn't matter, that changing them is not really user friendly.

`\SavePos` `\savepos` `\SavePos{`⟨*unique name*⟩`}` saves informations about the current position to the aux-file. These informations are read at next LaTeX run and may be used (see `\the...` commands below) then. The ⟨*unique name*⟩ has to be a position name, that is unique for all saved position informations of the current document. Note that the command has been renamed from `\savepos` to `\SavePos` in release 0.2, because LuaTeX since 0.85 uses `\savepos` as a new primitive. On other engines the old name `\savepos` is still available.

`\vskipnextgrid` This command moves to the next grid position. To achieve this, a position information is saved at this and used at next LaTeX run. The used name of the position information is `vb!`⟨*number of skip*⟩. ⟨*number of skip*⟩ is the number of the current `\vskipnextgrid` usage. Counter `gridcnt` is used to number the usage of `\vskipnextgrid`.

`\thegridinfo` `\thegridinfo{`⟨*name*⟩`}` outputs

- arabic page number of the named position,
- grid base, that was valid saving the information of the named position,
- grid interval, that was valid saving the information of the named position,
- x-coordinate of the named position,
- y-coordinate of the named position.

The coordinates and intervals are numbers without units. The unit is 'scale points' (sp).

`\theposinfo` `\theposinfo{`⟨*name*⟩`}` outputs

- y-coordinate of the named position,
- grid line number (first is 0) of the next grid position,
- offset of the next grid position from grid base,
- distance to the next grid position.

The coordinates, offsets and distances are numbers without units. The unit is 'scale points' (sp).

`\theypos` `\theypos{`⟨*name*⟩`}` outputs the y-coordinate of the named position.

2 Implementation

`gridset@luaorpdf` `\gridset@pageheight` `\gridset@pagewidth` `\gridset@savepos` `\gridset@lastxpos` `\gridset@lastypos` We need some locals because LuaTeX changed the names of several primitives inherited from PDFTeX.

```
\newcommand*{\gridset@luaorpdf}[1]{%
  \expandafter\newcommand\csname gridset@#1\endcsname{}%
  \ifcsname pdf#1\endcsname
    \expandafter\let\csname gridset@#1\expandafter\endcsname
```

```
    \csname pdf#1\endcsname
  \else
    \ifcsname #1\endcsname
      \expandafter\let\csname gridset@#1\expandafter\endcsname
      \csname #1\endcsname
    \else
      \PackageError{gridset}{%
        neither \expandafter\string\csname #1\endcsname\space
        nor \xpandafter\string\csname pdf#1\endcsname\space
        defined%
      }{This package needs either PDFTeX or LuaTeX or XeTeX.}%
    \fi
  \fi
}
\gridset@luaorpdf{pageheight}
\gridset@luaorpdf{pagewidth}
\gridset@luaorpdf{savepos}
\gridset@luaorpdf{lastxpos}
\gridset@luaorpdf{lastypos}
```

`\gridbase` `\gridinterval` These contain the grid information. `\gridbase` is a integer number representing the absolute y coordinate of the upper end of the grid relative to the same reference point `\pdfsavepos` uses. `\gridinterval` is a integer number representing the distance of two grid lines. The unit is 'scaled point' (sp) both time.

```
\newcommand*{\gridbase}{}
\newcommand*{\gridinterval}{}
```

`\gridbase` and `\gridinterval` need to be initialized at the start of the first page (fixme: shouldn't this be done at the start of every page?). We use this occasion to also initialize `\pdfpageheight` and `\pdfpagewidth` if this hasn't been done already.

```
\AtBeginDocument{%
  \ifdim\gridset@pageheight=\z@
    \gridset@pageheight=\paperheight
  \fi
  \ifdim\gridset@pagewidth=\z@
    \gridset@pagewidth=\paperwidth
  \fi
  \begingroup
    \@tempdima=\dimexpr \gridset@pageheight - \topmargin - 1in
                        - \headheight - \headsep
                        - \topskip \relax
    \@tempcnta=\@tempdima
    \xdef\gridbase{\the\@tempcnta}%
    \@tempcnta=\baselineskip
    \xdef\gridinterval{\the\@tempcnta}%
  \endgroup
}
```

`\savepos` `\SavePos` Save current position on the page to the `aux`-file. The argument is a unique name for the position. The saved informations are:

- the name of the position,
- the arabic page number of the page with the position,

- the grid base, that was valid for this position,
- the grid interval, that was valid for this position,
- the x-coordinate of the absolute position,
- the y-coordinate of the absolute position.

Todo: Currently we use our own command `\newpos` to store the position information in the aux-file. But with the extended reference mechanism of LaTeX since 2023-11-01 this should be replaced by the new properties.

```
\newcommand*{\SavePos}[1]{%
  \begingroup
    \gridset@savepos
    \protected@write\@auxout{}{%
      \protect\newpos{#1}{\the\count\z@}{\gridbase}{\gridinterval}{%
        \noexpand\number\gridset@lastxpos
      }{%
        \noexpand\number\gridset@lastypos
      }%
    }%
  \endgroup
}
\ifx\savepos\gridset@savepos
  \PackageInfo{gridset}{LuaTeX detected.\MessageBreak
    Note, gridset command is \string\SavePos\MessageBreak
    but not \string\savepos, which is\MessageBreak
    a LuaTeX primitive
  }%
\else
  \PackageInfo{gridset}{\string\savepos\space defined as an alias of
    \string\SavePos}%
  \newcommand*{\savepos}{\SavePos}%
\fi
```

`\newpos` This is the command, that has been written to the aux-file. Reading the aux-file it defines several position dependant macros to store the position information. Reading the aux-file while `\begindocument` a double definition test is done. Reading the aux-file while `\enddocument` a test is done, if the position has been changed and notes the user about needed additional LaTeX runs. (fixme: shouldn't the test be done with the x- and the y-coordinate instead of the vskip only?) The defined macros are:

`\pos@`⟨*position name*⟩`@page` the arabic page number of the position

`\pos@`⟨*position name*⟩`@base` the valid grid base while saving the position

`\pos@`⟨*position name*⟩`@interval` the valid grid interval while saving the position

`\pos@`⟨*position name*⟩`@x` the x-coordinate of the position

`\pos@`⟨*position name*⟩`@y` the y-coordinate of the postion

`\pos@⟨position name⟩@line` the number of the next grid line for the position (first grid line has number 0)

`\pos@⟨position name⟩@offset` distance of the next grid line from the grid base

`\pos@⟨position name⟩@vskip` distance to the next grid line for the position

All values are integers. The unit to all values is 'scaled points' (sp). See `\pdfsavepos` at the pdfTeX user manual for more information.

```
\newcommand*{\newpos}[6]{%
  \grid@unique@test{#1}{#2}%
  \expandafter\global\@namedef{pos@#1@page}{#2}%
  \expandafter\global\@namedef{pos@#1@base}{#3}%
  \expandafter\global\@namedef{pos@#1@interval}{#4}%
  \expandafter\global\@namedef{pos@#1@x}{#5}%
  \expandafter\global\@namedef{pos@#1@y}{#6}%
  \begingroup
    \@tempcnta=\numexpr \@nameuse{pos@#1@base} - \@nameuse{pos@#1@y}\relax
    \@tempcnta=\numexpr \@tempcnta + \@nameuse{pos@#1@interval} - 1\relax
    \divide\@tempcnta by\@nameuse{pos@#1@interval}\relax
    \expandafter\xdef\csname pos@#1@line\endcsname{\the\@tempcnta}%
    \@tempcnta=\numexpr \@tempcnta * \@nameuse{pos@#1@interval}\relax
    \expandafter\xdef\csname pos@#1@offset\endcsname{\the\@tempcnta}%
    \@tempcnta=\numexpr \@nameuse{pos@#1@y}
                        - ( \@nameuse{pos@#1@base} - \@tempcnta )\relax
    \expandafter\let\expandafter\@tempa\csname pos@#1@vskip\endcsname%
    \expandafter\xdef\csname pos@#1@vskip\endcsname{\the\@tempcnta}%
    \expandafter\ifx\csname pos@#1@vskip\endcsname\@tempa\else
      \grid@ReRunMessage
    \fi
  \endgroup
}
```

`\grid@unique@test` A very simple test to warn if a position name isn't unique.

```
\newcommand*{\grid@unique@test}[2]{%
  \expandafter\ifx\csname pos@#1@page\endcsname\relax\else
    \PackageError{gridset}{position `#1' is not unique.\@gobble}{%
      You have used the position name `#1' you are using on page
      `#2'\MessageBreak
      already on page `\csname pos@#1@page\endcsname'.\MessageBreak
      You should stop processing, remove the aux-files and correct the
      names.\MessageBreak
      If you'd continue, this will result in grid position
      failures,\MessageBreak
      that won't be reported!}%
  \fi
}
```

The test should be done only, when reading the `aux` file at `\begin{document}`, but not later (e.g. at `\end{document}`). So we deactivate the test after reading the `aux` file in `\begin{document}`.

```
\AtBeginDocument{%
  \global\let\grid@unique@test\@gobbletwo
}
```

\grid@ReRunMessage The change test will be done for each \newpos but one user information at the end of the document should be enough. So we use a message macro, that destroys itself after first usage.

```
\newcommand*\grid@ReRunMessage{}
\AtBeginDocument{%
  \renewcommand*\grid@ReRunMessage{%
    \PackageWarningNoLine{gridset}{Grid position labels may have
      changed.\MessageBreak
      Rerun to get grid positions right}%
    \global\let\grid@ReRunMessage\relax
  }%
}
```

\vskipnextgrid gridcnt Move to next grid position. The counter gridcnt is used to give every move to position a unique position name. The names are 'vp!⟨*number of the move to position*⟩'. You may use this to get informations e.g. about the last move to position.

```
\newcounter{gridcnt}
\newcommand*{\vskipnextgrid}{%
  \begingroup
    \stepcounter{gridcnt}\edef\@tempa{vp!\thegridcnt}%
    \ifvmode
```

\pdfsavepos in vertical mode is a problem, because the base line alignment will be done at least at paragraph breaking. Because of this, we have to leave the vertical mode and do it then. But remark: If you change the base line skip e.g. changing the font size, the next line would not be grid aligned!

```
      \leavevmode\SavePos{\@tempa}%
      \expandafter\ifx\csname pos@\@tempa @vskip\endcsname\relax
      \else
        \expandafter\ifnum \csname pos@\@tempa @vskip\endcsname =\z@\else
          \PackageInfo{gridset}{%
            vmode \string\vskip\csname pos@\@tempa @vskip\endcsname sp%
          }%
          \vskip -\parskip\vskip -\baselineskip
          \expandafter\vskip\csname pos@\@tempa @vskip\endcsname sp\relax
        \fi
      \fi
    \else
```

\pdfsavepos in horizontal mode is a problem too, because we have to enter the vertical mode to do vertical skips. Because of this, the remark is the same like the vertical mode remark.

```
      \parskip=\z@
      \SavePos{vp!\thegridcnt}%
      \expandafter\ifx\csname pos@\@tempa @vskip\endcsname\relax
      \else
        \expandafter\ifnum \csname pos@\@tempa @vskip\endcsname =\z@\else
          \PackageInfo{gridset}{%
            hmode \string\vskip\csname pos@\@tempa @vskip\endcsname sp%
          }%
          \vskip \dimexpr -\baselineskip
                         + \csname pos@\@tempa @vskip\endcsname sp\relax
```

In twocolumn mode we have to take care that in the second column we reduce the horizontal movement by the width of the first column plus the column separation. Note: This may fail, because the column information may be wrong outside the output routine. Maybe we should add this information to `\SavePos` or correct the x-pos there.

```
          \leavevmode
          \if@twoside
            \expandafter\ifodd\csname pos@\@tempa @page\endcsname\relax
              \hskip \dimexpr -1in - \oddsidemargin - \parindent
                             \if@twocolumn\if@firstcolumn\else
                               - \columnwidth - \columnsep
                             \fi\fi
                               + \csname pos@\@tempa @x\endcsname sp\relax
            \else
              \hskip \dimexpr -1in - \evensidemargin - \parindent
                             \if@twocolumn\if@firstcolumn\else
                               - \columnwidth - \columnsep
                             \fi\fi
                               + \csname pos@\@tempa @x\endcsname sp\relax
            \fi
          \else
            \hskip \dimexpr -1in - \oddsidemargin - \parindent
                             \if@twocolumn\if@firstcolumn\else
                               - \columnwidth - \columnsep
                             \fi\fi
                             + \csname pos@\@tempa @x\endcsname sp\relax
          \fi
        \fi
      \fi
    \fi
  \endgroup
}
```

(fixme: A better solution would be to first move and then set the position. But that solution needs some more tests and maybe some more ideas, because after moving the position is on grid and so the saved x-pos would be on grid.)

`\thegridinfo` `\theposinfo` `\theypos` Some informations about the grid (valid for a position) or the position.

```
\newcommand*{\thegridinfo}[1]{%
  page=\@nameuse{pos@#1@page},
  base=\@nameuse{pos@#1@base},
  interval=\@nameuse{pos@#1@interval},
  x=\@nameuse{pos@#1@x},
  y=\@nameuse{pos@#1@y}%
}
\newcommand*{\theposinfo}[1]{%
  y=\@nameuse{pos@#1@y},
  gridline=\@nameuse{pos@#1@line},
  gridoffset=\@nameuse{pos@#1@offset},
  movedown=\@nameuse{pos@#1@vskip}%
}
\newcommand*{\theypos}[1]{\@nameuse{pos@#1@y}}
```

3 Example

You may try the following example document. You have to do several LaTeX runs until no new rerun warning occurs.

```
\documentclass[a4paper,12pt]{article}
\usepackage{gridset}
\usepackage{blindtext}
\raggedbottom

\pagestyle{myheadings}

\begin{document}
\markright{gridbase=\gridbase, gridinterval=\gridinterval\ without move down}%
\newcounter{Zeile}%
\makeatletter
\@whilenum \value{Zeile}<40\do {%
  \stepcounter{Zeile}%
  \theZeile. Zeile:
  \SavePos{\thepage.\theZeile}\thegridinfo{\thepage.\theZeile}\par
}%
\makeatother
\clearpage
\setcounter{Zeile}{0}
\makeatletter
\@whilenum \value{Zeile}<20\do {%
  \stepcounter{Zeile}%
  \theZeile. Zeile:
  \SavePos{\thepage.\theZeile}\theposinfo{\thepage.\theZeile}\par
}%
\makeatother
\clearpage
\parskip=.5\baselineskip
\setcounter{Zeile}{0}
\makeatletter
\@whilenum \value{Zeile}<20\do {%
  \stepcounter{Zeile}%
  \theZeile. Zeile:
  \SavePos{\thepage.\theZeile}\theposinfo{\thepage.\theZeile}\par
}%
\makeatother
\clearpage
\markright{gridbase=\gridbase, gridinterval=\gridinterval\ with real move down
at vmode}%
\parskip=.5\baselineskip
\setcounter{Zeile}{0}
\makeatletter
\@whilenum \value{Zeile}<25\do {%
  \stepcounter{Zeile}%
  \vskipnextgrid\theZeile. Zeile: \theposinfo{vp!\thegridcnt}\par
}%
\makeatother
\clearpage
\markright{gridbase=\gridbase, gridinterval=\gridinterval\ with real move down
at hmode}%
```

```
\parskip=.5\baselineskip
\setcounter{Zeile}{0}
\makeatletter
\@whilenum \value{Zeile}<25\do {%
  \stepcounter{Zeile}%
  \theZeile. Zeile: \vskipnextgrid\theposinfo{vp!\thegridcnt}\par
}%
\makeatother
\clearpage
\parskip=0pt
\blindtext
\begin{itemize}
\item Test
\item Test
\end{itemize}
\vskipnextgrid\theposinfo{vp!\thegridcnt}\blindtext

\end{document}
```

Change History