

The **perpage** package

Version 2.0

David Kastrup*

2014/10/25

1 Description

The **perpage** package adds the ability to reset counters per page and/or keep their occurrences sorted in order of appearance on the page.

It works by attaching itself to the code for `\stepcounter` and will then modify the given counter according to information written to the `.aux` file, which means that multiple passes may be needed. Since it uses the internals of the `\label` mechanism, the need for additional passes will get announced by L^AT_EX as “labels may have changed”.

`\MakePerPage`

`\MakePerPage[2]{footnote}`

will start footnote numbers with 2 on each page (the optional argument defaults to 1). 2 might be a strange number, unless you have used something like

`\renewcommand\thefootnote{\fnsymbol{footnote}}`

and want to start off with a dagger. The starting value must not be less than 1 so that the counter logic can detect the reset of a counter reliably.¹ It could be a good idea to redefine `\@cnterr`

the counter is incremented, the macro `\theperpage` will be set to the correct value corresponding to the actual page location. Note that this macro is shared between all counters, so advancing a different counter under control of `perpage` will render `\thefigure` incorrect.

`\MakeSorted`

`\MakeSorted{figure}`

will make the `figure` counter get ‘sorted’: this means that counter values will be assigned in order of appearance in the output, not in order of appearance in the source code. For example, the order of interspersed one- and two-column figures might get mixed up by L^AT_EX in the output. Making the counter sorted will fix the order to match the order of appearance. A similar problem is when ordinary footnotes are present in floating material (this does not work in standard L^AT_EX, but might do so when using `manyfoot.sty` or `bigfoot.sty`): this might jumble their order in the output, and making their counter sorted will make things appear fine again.

While this would not fix the order in the table of figures, fortunately the respective entries actually get written out in order of appearance in the output anyway, so this indeed fixes the problem.

Manually setting the counter does not lead to reliable results in general; as a special case, however, resetting it to zero is recognized (this can also happen automatically when the counter is dependent on some other counter). The point where it is reset in the source code separates ‘count groups’: everything in the source before that point is assigned sorted numbers separately from everything appearing behind it, and the sequence numbers start again with 1 with the first item appearing in the output (not the source) from the new count group.

`\MakeSortedPerPage`

`\MakeSortedPerPage[2]{table}`

will make the table numbers restart at 2 on each page *and* will keep them sorted, to boot. Introducing new count groups by resetting the counter to 0 manually will not work, as it is not clear how to handle count groups scattered between pages. You will usually want to use something like

`\renewcommand{\thefigure}{\theperpage-\arabic{figure}}`

to go along with a page-wise figure number.¹ Note that it would be quite silly to start the ranges with 2: this is just an example for the optional argument in case that you ever need it.

`\AddAbsoluteCounter`

`\AddAbsoluteCounter{equation}`

will create a counter `absequation` that will advance together with the counter `equation` but will not get reset along with it. This is not sorted into output

¹Note the use of `\theperpage` here, see above.

order, but just runs along with the sequence in the source file. As a special case, the counter `abspage` is created in this manner and `\theabspage` is defined as an arabic number that works in the same contexts as `\page` (namely, gets properly deferred by `\protected@write`).

2 The documentation driver

This is the default driver for typesetting the docs. Running it through as a separate file will include the code section. Running the original `.dtx` file through will omit the code.

```

1 <*driver>
2 \documentclass{ltxdoc}
3 \usepackage{perpage}
4 \MakePerPage{footnote}
5 \begin{document}
6 \OnlyDescription
7 <driver> \AlsoImplementation
8 \DocInput{perpage.dtx}
9 \end{document}
10 </driver>

```

3 The package interfaces

First identification.

```

11 <*style>
12 \NeedsTeXFormat{LaTeX2e}
13 \ProvidesPackage{perpage}[2014/10/25 2.0 Reset/sort counters per page]

```

`\pp@cl@begin` These macros are considerable tricky. They are called as artificial ‘dependent’ counters when the counter they are hooked into is advanced. The way in which those counters are called are one of the following:

```
\def\@stpelt#1{\global\csname c@#1\endcsname \z@}
```

which is the default way of resetting a subordinate counter used in L^AT_EX, or

```
\def\@stpelt#1{\global\csname c@#1\endcsname \m@ne\stepcounter{#1}}
```

which is a little present from `fixltx2e.sty` as of 2014/05/01, quite complicating this feat.

The startup code swallows either `\global \advance` or `\global`.

```
14 \def\pp@cl@begin{\z@\z@ \begingroup}
```

The command used for ending our fake counters checks for the `\m@ne` condition. We don’t want to bump our auxiliary counters twice, so we remove the following `\stepcounter` command. Things will go haywire if there is none, of course.


```

133 \pp@fetchctr{#1}%
134 \ifx\pp@page\@empty
135 \else \setcounter{#1}{\pp@label}\fi
136 \pp@writectr\pp@ppagectr{#1}{\noexpand\theabspage}}

```

\pp@ppagectr

```

137 \def\pp@ppagectr#1#2#3#4{\@ifundefined{c@pp@a@#1}{\%
138 \def\next{#3}%
139 \expandafter\ifx\csname pp@page@#1\endcsname\next
140 \addtocounter{pp@a@#1}\@ne
141 \else
142 \setcounter{pp@a@#1}{\value{pp@r@#1}}%
143 \fi
144 \global\expandafter\let\csname pp@page@#1\endcsname\next
145 \expandafter\pp@labeldef\expandafter
146 {\number\value{pp@a@#1}}{#1}{#2}{#3}{#4}}

```

\@testdef L^AT_EX's current (2007) definition of this macro causes save stack overflow. We fix this by an additional grouping. Delay to the beginning of document to keep Babel happy.

```

147 \AtBeginDocument{%
148 \begingroup
149 \@testdef{}{undefined}{}%
150 \expandafter
151 \endgroup
152 \ifx\@undefined\relax
153 \let\pp@@testdef\@testdef
154 \def\@testdef#1#2#3{\pp@@testdef{#1}{#2}{#3}%
155 \if@tempswa\aftergroup\@tempswatrue\fi}%
156 \fi}
157 \</style>

```