

# The `perpage` package

## Version 1.12

David Kastrup\*

2006/07/15

### 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<sup>A</sup>T<sub>E</sub>X 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.<sup>1</sup> It could be a good idea to redefine `\cnterr` if you use a format with limited range: at the first pass, footnotes are not reset across pages and things like `\fnsymbol` will quickly run out of characters to use.

`\theperpage`

If you want to label things also on a per page base, for example with

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

you’ll have the problem that `lis` is updated asynchronously with the real page, since T<sub>E</sub>X does not know which page the figure will end up. If you have used the `perpage` package for modifying the figure counter, however, at the point where 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`

---

\*[dak@gnu.org](mailto:dak@gnu.org)

<sup>1</sup>This unfortunately means that you can’t just use `\alph` in order to get figures on page 10 numbered as “10”, “10a”, “10b”.

`\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<sup>A</sup>T<sub>E</sub>X 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<sup>A</sup>T<sub>E</sub>X, 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.<sup>1</sup> 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 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`).

---

<sup>1</sup>Note the use of `\theperpage` here, see above.