Package **listparskip** v. 1.0 Implementation

Conrad Kosowsky

December 2024

`kosowsky.latex@gmail.com`

Overview

The listparskip package modifies list environments such that they add `\parskip` and `\partopsep` before or after a list if and only if the environment follows or precedes, respectively, a blank line (i.e. a `\par`).

This file documents the code for the listparskip package. It is not a user guide! If you are looking for discussion of how listparskip will change the layout of your documents, see `listparskip_user_guide.pdf` and `listparskip_example.pdf`, which are included with the listparskip installation and are available on CTAN.

Spacing in list environments is subtle, and it will be helpful to summarize how LaTeX implements vertical space in lists. Every list environment comes from either a `\list` or `\trivlist` macro, and `\list` is responsible for the standard list structure that shows up in enumerate, itemize, and description environments. A `\trivlist` is a list without any indentation or space for item markers, and this command comes up, for example, in quotations or some verbatim environments where fine control over vertical spacing is important. Both user-level commands serve as wrappers for the internal macro `\@trivlist`, which translates the definition of the current environment into internal list control sequences for processing later in the document. Default dimensions for different list levels come from the class file in the `\@listi` through `\@listiv` macros.

When the user enters a list environment, for example with `\begin{enumerate}`, TeX will at some point encounter `\@trivlist`. When that happens, TeX prepares to add `\topsep` and `\parskip` to the main vertical list, and if TeX is in V mode, it prepares to add an extra `\partopsep`. The environment call itself doesn't add space or penalties, and instead, the first `\item` inside the list environment is responsible for these changes. Future uses of `\item` add an `\itempenalty` and `\itemsep` to the main vertical list, and any `\parsep` space additions happen automatically because LaTeX sets `\parskip` to `\parsep` inside `\list` immediately after `\@trivlist`. At the end of the list environment, TeX adds `\topsep` and, if it was present at the beginning of the environment, `\partopsep` to the main vertical list. LaTeX does not instruct TeX to manually add `\parskip`, so if the user includes, for example, an `\hbox` right after `\end{enumerate}`, TeX will not add `\parskip` there. (TeX is always in V mode after it begins or ends a list environment.)

Changing this behavior at the beginning of lists is straightforward—listparskip patches `\@trivlist` by treating the extra `\parskip` the same way LaTeX normally treats `\partopsep`. Dealing with after-list space is more complicated. Normally `\@endparenv` adds `\topsep` and possibly `\partopsep` directly to the main vertical list, but we want a more tailored approach. Instead, listparskip redefines `\@endparenv` to add `\topsep` to the main vertical list and (globally) save `\topsep` plus `\partopsep` in `\@tempskipa`. Immediately outside the list environment, listparskip sets `\parskip` to 0pt and redefines `\par` to remove the previous `\topsep` and replace it with `\@tempskipa`. To ensure that (a) the extra `\partopsep` happens only if a blank line immediately follows the list and (b) TeX avoids inserting `\parskip` only if h mode material immediately follows the list, listparskip makes `\par`, `\everypar`, and `\item` restore `\par` and `\parskip` to their previous (local) definitions. Using creative groupings or manually redefining `\par` or `\everypar` right after a list may break things, so please be careful!

Sometimes the user will want to include non-horizontal material, such as a box, immediately after a list environment without worrying about an extra or missing `\parskip` or `\partopsep`. In the implementation outlined above, listparskip adds or removes this space whenever TeX first encounters a `\par` or h mode material after a list. A box is neither of these things, so any after-list adjustments will show up after the box instead of before. The easiest way to avoid this problem is to transition to h mode and then back to v mode before adding the box, and the macro `\nullline` does this.

* * *

First, the package should declare itself. There are no package options.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{listparskip}[2024/12/17 v. 1.0 Package listparskip]
\newdimen\@saveparskip
```

We begin with spacing at the start of list environments. We patch `\@trivlist` such that `\@topsepadd` increases by `\parskip` only when TeX is already in v mode.

```
\long\def\patch@trivlist#1\ifvmode#2\else#3%
  \advance\@topsep\parskip#4\@nil{%
    #1%
    \ifvmode
      \advance\@topsepadd\partopsep
      \advance\@topsepadd\parskip
    \else
    #3#4}
\toks@\expandafter\expandafter\expandafter
  {\expandafter\patch@trivlist\@trivlist\@nil}
\edef\@trivlist{\the\toks@}
```

We change `\@endparenv` so that it adds only `\topsep`. We set `\@tempskipa` globally in order to access its value in `\@doendpe`, which is outside the current group.

```
\def\@endparenv{%
  \addpenalty\@endparpenalty
  \addvspace\topsep
  \global\@tempskipa\glueexpr\topsep+\partopsep\relax
  \@endpetrue}
```

Normally `\@restoreparskip` is `\relax`, but we change it in `\@doendpe` to return `\parskip` back to its previous value, which we have stored in `\@saveparskip`. This makes `\parskip\z@` a one-op. (Or a no-op if TeX sees a `\par` or `\item` after the list and doesn't enter h mode.)

```
\let\@restoreparskip\relax
\def\@doendpe{%
  \@endpetrue
  \@saveparskip\parskip
  \parskip\z@
  \def\@restoreparskip{%
    \parskip\@saveparskip
    \let\@restoreparskip\relax}
```

We add `\@restoreparskip` to `\par`, and the `\addvspace` will replace the previous addition of `\topsep` from `\@endparenv` with `\topsep` plus `\partopsep`. We put everything inside an `\edef` so we can expand `\the\@tempskipa` now and not worry about a redefinition of `\@tempskipa` before the next `\par`. (Unlikely, but you never know.)

```
  \edef\par{%
    \unexpanded{\clubpenalty\@clubpenalty
      \everypar{}%
      \@restorepar\par
      \@restoreparskip
      \@endpefalse}%
    \noexpand\addvspace{\the\@tempskipa}}%
```

The contents of `\everypar` here are the same as the LaTeX default in this situation except that we add `\@restorepar` and `\@restoreparskip`.

```
  \everypar{\@restorepar
    \@restoreparskip
    {\setbox\z@\lastbox}%
    \everypar{}%
    \@endpefalse}}
```

Next we add `\@restorepar` to the beginning of `\item`. Calling `\patch@item` on a control sequence `#1` will insert `\@restorepar` and `\@restoreparskip` into `#1`'s definition.

```
\def\patch@item#1{%
  \edef#1{\noexpand\@restorepar
    \noexpand\@restoreparskip
    \expandafter\unexpanded\expandafter{#1}}}
```

The hyperref package sometimes turns `\item` into a wrapper for itself. In that case, hyperref will have renamed `\item` to `\H@item`, so we patch that instead.

```
\@ifpackageloaded{hyperref}{%
  \ifHy@implicit
    \patch@item\H@item
  \else
    \patch@item\item
  \fi}{\patch@item\item}
```

Finally, create the command that forces a transition from v mode to h mode and back without (directly) adding any space.

```
\def\nullline{\ifvmode\nointerlineskip\leavevmode\par\fi}
```

All done!