

paralist<sup>\*</sup>  
扩展列表环境

Bernd Schandl  
schandl@gmx.net

中译版编译日期：2014 年 11 月 16 日  
译者：黄旭华<sup>†</sup>

---

<sup>\*</sup>宏包版本：v2.4，发布日期：2013/06/09。

<sup>†</sup>参考了 [CTEX论坛](#) [vincentme](#) 2010年12月6日的中译本。

## 摘 要

该宏包提供几个新的列表环境。常规列表 (Itemized list) 和排序列表 (enumerated list) 均可以更紧凑的格式排版成段落。大多数环境有格式化标签的可选参数。此外, L<sup>A</sup>T<sub>E</sub>X 的 `itemize` 环境和 `enumerate` 环境也可使用类似的可选参数。

## 目 录

1 介绍	4
2 宏包选项	4
3 格式化标签	5
4 默认的标签和边距	7
5 新列表环境	8
5.1 排序列表环境 . . . . .	8
5.2 常规列表环境 . . . . .	9
5.3 解说列表环境 . . . . .	9
5.4 空白列表环境 . . . . .	10
6 嵌套列表环境	10
7 精细调整	11
8 漏洞和希望	12
9 感谢	12

## 1 介绍

1998年5月，有人在 `comp.text.tex` 发了一篇贴子，询问是否可能扩展排序列表，使之 (a) 能在段落内使用，(b) 自动编号以及(c) 条目能被引用。另一篇贴子提道 `theapa` 宏包可以解决此问题。我在寻找这类环境并发现了这些贴子，看了 `theapa` 宏包后，决定取出和列表环境相关的代码，然后作了点修改。

过段时间后，紧凑版的 `enumerate`、`itemize` 和 `description` 环境以及能在大多数环境里格式化标签的可选参数都添加进来了。

## 2 宏包选项

只有选择了相应的选项后才具有宏包的特定功能。

`newitem/olditem` 使用 `newitem` 选项时（这是默认的）， $\text{\LaTeX}$  的 `itemize` 环境将被重定义，具有格式化标签的可选参数。请参阅第3节。使用 `olditem` 选项将保持 `itemize` 的原始定义。

`newenum/oldenum` 使用 `newenum` 选项时（这是默认的）， $\text{\LaTeX}$  的 `enumerate` 环境将被重定义，具有格式化标签的可选参数。请参阅第3节。使用 `oldenum` 选项将保持 `enumerate` 的原始定义。

`alwaysadjust` 环境 `compactenum`、`enumerate`、`compactitem` 和 `itemize` 的标签宽度将被调整到实际标签宽度。对于默认的标签，这意味着其宽度是减小的。

`neveradjust` 永不调整标签宽度，即使使用可选参数手动指定了标签宽度。该选项在选择了 `alwaysadjust` 选项后将被忽略。

`neverdecrease` 如果调整了标签宽度，该选项会忽略标签宽度的减小。这里有个例子说明此选项的意义。如果没有指定 `...adjust` 选项，`\begin{enumerate}` 与 `\begin{enumerate}[1.]` 中 `\item` 的缩进就会不相同，即使它们具有相同的标签。

`defblank` 定义 `inparablanck` 和 `asparablanck` 这两个新环境。请参阅第5.4节。

`pointlessenum` 排序列表的条目显示和引用时为“1”，“1.1”，“1.1.1”和“1.1.1.1”。请参阅第3节。

`pointedenum` 排序列表的条目显示时为“1.”，“1.1.”，“1.1.1.”和“1.1.1.1.”，引用时则没有末尾的点。请参阅第3节。如果选择了 `pointlessenum` 选项，则 `pointedenum` 选项将被忽略。

`flushright` 该选项将上述四种列表的标签右对齐。这是  $\text{\LaTeX}$  的默认设置，也是该宏包的默认设置。

`flushleft` 该选项将上述四种列表的标签左对齐。

`cfg` 如果存在配置文件 `paralist.cfg`，则加载它。（这是默认的）

`nocfg` 不加载配置文件。

不再支持 `increaseonly` 选项；用 `neverdecrease` 选项代替。

### 3 格式化标签

所有的常规列表和排序列表均有用于格式化标签的可选参数。下面的例子只在加载了 `paralist` 宏包并且没有选择 `olditem` 和 `oldenum` 这两个选项时才能运行。

使用  $\text{\LaTeX}$  的标准文档类时，`itemize` 用下面的符号表示列表的四个层次：• – \* ∙。如果想改变某个环境的符号，如下地使用可选参数

```
\begin{itemize}[$\star$]
```

那么  $\star$  号就会取代默认的符号。

排序列表的可选参数有点复杂，但用过 David Carlisle 的 `enumerate` 宏包可跳过本节剩余部分，因为使用了相同的语法（还有几乎一样的代码）。

A、a、I、i 和 1 这些记号可以用在可选参数中，得到 `\Alph`、`\alph`、`\Roman`、`\roman` 和 `\arabic` 样式的计数器。<sup>1</sup> 这些记号可以包围在任何字符中，也可以包围在  $\text{\TeX}$  表达式中。但是，如果不是指定为记号，A a I i 1 必须用 `{ }` 包围。以下是几个例子。

```
\begin{enumerate}[(i)]
```

生成 (i), (ii), (iii) ...

---

<sup>1</sup> 记号集合可以扩展。请在代码这一节查寻 `\pl@hook`。

```
\begin{enumerate}[\example a)]
```

生成 example a), example b), example c) ...

```
\begin{enumerate}[\A-1]
```

生成 A-1, A-2, A-3 ...

```
\begin{enumerate}[\bfseries {Item} I]
```

生成 **Item I**, **Item II**, **Item III** ...

注意, 最后一个例子写成 `[\textbf{Item I}]` 是错误的, 因为记号 **I** 写在了 `{ }` 里面。

`\ref` 命令只产生计数器, 而不包含周围的文字, 所以, 上面的例子引用第一项时会分别生成 `i`, `a`, `1` 和 `I`。

如果标签使用了罗马数字 (如上述的第一个例子), 则会尝试判断最大的标签, 假定最大的条目数不超过 7。如果这与实际不符, 实际的标签比这大, 则可以使用第二个可选参数指定更大的标签数, 例如:

```
\begin{enumerate}[(i)][8]
```

生成标签 (i), (ii), ..., (vii), (viii), (ix) 等, 直到 (xvi) 符合您的要求。<sup>2</sup>

`\pointedenum` 有两个宏包选项和两个对应的宏, 用来格式化排序列表的标签和引用。  
`\pointlessenum` `pointedenum` 选项和 `\pointedenum` 宏设置标签为 “1.”, “1.1.”, “1.1.1.” 和 “1.1.1.1.”, 引用时没有末尾的点。 `pointlessenum` 选项和 `\pointlessenum` 宏在标签里也没有末尾的点。

宏包选项会产生全局的影响, 而宏则产生局部影响或定义特殊环境, 例如:

```
\newenvironment{myenum}%  
  {\pointedenum\begin{enumerate}}%  
  {\end{enumerate}}
```

注意: `\begin{enumerate}` 需放在 `\pointedenum` 之后, 否则, `enumerate` 的可选参数不会生效 (也许您想在同一环境中使用它们, 而该环境不是真的有意义)。

`\paradescriptionlabel` 在文档类中, `description` 环境定义的标签格式为:

```
\newcommand*\descriptionlabel[1]{%
```

---

<sup>2</sup>该宏包的 2.4 版本中介绍了这个功能, 如果要在便携式文档中使用该功能, 在加载宏包时, 请指定 `\usepackage{paralist}` [2013/06/09]

```
\hspace\labelsep\normalfont\bfseries #1}.
```

这也被 `compactdesc` 使用了。对于 `inparadesc` 环境和 `asparadesc` 环境，有一个单独的名为 `\paradescriptionlabel` 的宏，（几乎）也是这样定义的：

```
\newcommand*\paradescriptionlabel[1]{%  
  \normalfont\bfseries #1}.
```

## 4 默认的标签和边距

如果要在整个文档里使用与  $\text{\LaTeX}$  默认格式不同的列表，每次使用环境的可选参数是很麻烦的。因此提供了三个可以定义默认列表环境标签和左边距的宏。

注意，调整标签的宏并不调整列表的左边距，因为这可能导致不想要的效果。如果要调整列表的左边距，可以使用 `\setdefaultleftmargin`。

如果下述三个宏的任何一个宏的参数为空，则不改变相应标签或边距。

`\setdefaultitem` 常规列表的默认标签可以通过宏 `\setdefaultitem` 来设置，它需要四个参数。例如如下可定义  $\text{\LaTeX}$  默认的标签：

```
\setdefaultitem{\textbullet}%  
  {\normalfont\bfseries \textendash}%  
  {\textasteriskcentered}{\textperiodcentered}
```

（这当然是很愚蠢的，如果要默认的效果，不需改动任何地方）。如要要用三角形（▷）代替第二级的短线，请使用：

```
\setdefaultitem{}{\$ \triangleright \$}{\{}}.
```

`\setdefaultenum` 使用第 3 节中介绍的语法，宏 `\setdefaultenum` 可以格式化排序列表的标签。可以通过下述方式定义  $\text{\LaTeX}$  默认的标签格式：

```
\setdefaultenum{1.}{(a)}{i.}{A.}.
```

如要在第三级使用大写罗马字母，请这样设置：

```
\setdefaultenum{}{}{I.}{I.}.
```

`\setdefaultleftmargin` 使用 `\setdefaultleftmargin` 调整列表的左边距。长度 `\leftmargin n` 指第  $n$  级的项目相对于  $n - 1$  级或周围文本（ $n = 1$  时）的缩进距离。使

用 `\leftmargin` $n$  的环境（至少）有 `enumerate`、`compactenum`、`itemize` 和 `compactitem`（可能还有些我没注意到）。 $\text{\LaTeX}$  设置可以这样定义：

```
\setdefaultleftmargin{2.5em}{2.2em}{1.87em}{1.7em}{1em}{1em}.
```

在双栏模式中， $\text{\LaTeX}$  使用较小的第一级、第五级、第六级缩进长度可以这样定义：

```
\setdefaultleftmargin{2em}{}{}{}{.5em}{.5em}.
```

宏 `\defaultitem`、`\defaultenum` 和 `\defaultleftmargin` 不应再使用了。它们的存在只是为了与  $< 2.1$  版本的文档兼容。

如果要对每个使用了 `paralist` 宏包的文档作修改，可以将这些定义放在 `paralist.cfg` 文件里。如果该文件存在，将在读取宏包结束时载入它（除非选择了 `nocfg` 选项）。

## 5 新列表环境

### 5.1 排序列表环境

`asparaenum` `asparaenum` 环境生成每一条目均为独立段落的排序列表。

举个例子，下面是使用 `asparaenum` 环境生成的两个段落。

1. 每个 `\item` 是一个独立的段落。第二行并不缩进（这是此环境的特征而非漏洞）。

2. 第二个 `\item` 看起就是这样的，并且带有标签。

这个例子是由下述的代码生成的：

```
\begin{asparaenum}
  \item 每个 ...
  \item 第二个 ... \label{p11}
\end{asparaenum}
```

使用 `\ref{p11}` 可以得到 2。

`inparaenum` `inparaenum` 环境可以在一个段落里生成排序列表，就像第一节“介绍”中的那样。

第一节“介绍”中的那个例子是由下述的代码生成的：

... 扩展排序列表，使之



```

\begin{inparaenum}[(a)]
  \item 能在段落内使用,
  \item 自动编号以及
  \item 条目能被引用。 \label{p12}
\end{inparaenum}
另一篇贴子提到 ...

```

使用 `\ref{p12}` 可以得到 c。

`compactenum`      `compactenum` 环境是标准的 `enumerate` 环境的紧凑版。所有垂直距离都设为零（实际上是可以调整的，请参阅第 7 节）。

## 5.2 常规列表环境

`asparaitem`      `asparaitem` 环境与 `asparaenum` 环境非常相似。它是用符号代替条目的编号。此环境有一个可选参数用来指定条目的符号。第 6 节有个例子。

`inparaitem`      与 `inparaenum` 环境类似地，`inparaitem` 环境也有一个可选参数。为了对称完整我添加了该可选参数，但不知是否会有人使用它。

`compactitem`      `compactitem` 环境是标准 `itemize` 环境的紧凑版，所有垂直距离设为零。所以使用这个环境

- 可以节省空间和
- 指定符号。
- 让我添加一个长一点的条目好让你看清楚这个环境的缩进与 `asparaitem` 环境的缩进是不同的。

上面这个例子的代码如下：

```

\begin{compactitem}[$\circ$]
  \item 可以节省空间和
  \item 指定符号。
  \item 让我添加 ...
\end{compactitem}

```

## 5.3 解说列表环境

`asparadesc`      `asparadesc` 环境与 `asparaenum` 环境也非常类似，仅仅是使用 `\item`

的可选参数作为段落的“介绍”。

`inparaitem` 和 `inparaenum` 环境类似，我增加了 `inparadesc` 环境，同样为了对称完整，但不知是否会有人使用它。

`compactdesc` `compactdesc` 环境是从 L<sup>A</sup>T<sub>E</sub>X 标准文档类中复制过来的，所有垂直距离设为零。顺便问一下，有人知道为什么 `description` 是在标准文档类中定义而不是在 `ltlists.dtx` 中定义的？

## 5.4 空白列表环境

有人需要没有标签的列表环境。可能是为了在 LyX 中输入结构化数据方便些。因为并非所有人需要这些（奇怪的）环境，所以仅当选择了 `defblank` 选项时才加载该环境。由于没有标签可指定格式，下述的两个环境没有可选参数。

`asparablank` 每一条目都像正常段落那样排版。若要使用 `\item` 的可选参数，需要在最后加上一些空白，因为 `\labelsep` 被设为零了。请使用这样的代码：

```
\item[\textbullet\hspace{.5em}]
```

`inparablank` 所有条目设定为：就像正常文本那样排版。上一段提到的“空白问题”可以自动解决。如果我不告诉您，您可能不知道这个段落是用下述的代码生成的：

```
... 设定为：
\begin{inparablank}
  \item 就像 ...
  \item 上一 ...
  \item 可以 ...
\end{inparablank}
如果我不 ...
```

## 6 嵌套列表环境

就像标准列表那样，所有列表环境都可以嵌套使用，虽然结果可能出乎意料。例如，不要在 `inpara...` 环境里嵌套另一个列表环境，但为什

么有人要这样做呢？与  $\text{\LaTeX}$  的标准列表环境相同，最大嵌套层数为6层（4个同样类型的）。

本段是

★ 是 `asparaitem` 环境及其可选参数的例子，

★ 以及示范 (i) 可以在 `asparaitem` 列表里使用 `inparaenum`，(ii) 而且仍然可以引用它。

引用来自子条目(ii)。

本例的代码为：

```
\begin{asparaitem}[$\star$]
  \item 是...的例子，
  \item 以及示范
    \begin{inparaenum}[(i)]
      \item 可以在 ...
      \item 而且仍然可以 ... \label{p13}
    \end{inparaenum}
\end{asparaitem}
引用来自子条目(\ref{p13})。
```

## 7 精细调整

好吧，我已经听到有人说“你的压缩列表是个好主意，但我喜欢稍微宽松点的”。这里有解决方案。下面这些长度能使用 `\setlength` 命令来调整并影响 `compact...` 环境。这里的长度名称和  $\text{\LaTeX}$  的长度名称相似，所以我将它从 `ltlists.dtx` 里复制过来。

`\pltopsep`: 列表的第一个条目与其之前的非列表段落之间的距离。

`\plpartopsep`: 当列表为新段落的开始时会在 `\topsep` 加上额外的距离（称为 `vmode`）。

`\plitemsep`: 同一列表的条目之间的距离（同一个列表前一条目的最后一个段落与后一条目的第一个段落之间的距离）。

`\plparsep`: 条目内段落之间的距离（同一条目内段落与段落之间的距离）  
– 本列表的 `\parskip`。

实际上，会在列表前后各添加一个 `...topsep`。

这些长度的默认值均为 0pt。当需要指定非零值时，可能根据字体大小比较好，例如使用 `ex` 或 `em` 这样的单位。

## 8 漏洞和希望

目前没有漏洞 ...。

但是，如果使用 `babel` 宏包并选择 `acadian`、`canadien`、`francais`、`frenchb` 或 `french` 中的一个选项（这些其实是一样的），这时会出现一个问题。因为它在 `\start{document}` 重定义了常规列表，`paralist` 宏包定义的 `itemize` 会丢失。这个问题有三种解决办法：

1. 接受没有可选参数的 `itemize` 环境。：(
2. 加载 `babel` 宏包后使用 `\FrenchItemizeSpacingfalse` 可以阻止 `babel` 宏包对 `itemize` 的重定义。：/
3. 找出合并 `babel` 宏包和 `paralist` 宏包代码的方法，并发给我。：)

任何有关本宏包及其文档的问题、建议和期望都请让我知道。赞扬当然更加欢迎；-) 本宏包的最新版本可以在 CTAN 或 <http://schandl.gmxhome.de/paralist/> 找到。

## 9 感谢

我感谢所有帮我指出漏洞和提出建议的用户。非常感谢 David Carlisle，没有他的 `enumerate` 宏包就不会有排序列表的可选参数。部分 `inpara...` 环境代码受 Mogens Lemvig Hansen 的 `shortlst` 宏包启发。