

# Customizing lists with the **enumitem** package

Javier Bezos\*

Version 2.1  
2007-06-30

## 1 Introduction

When I began to use L<sup>A</sup>T<sub>E</sub>X several year ago, two particular points annoyed me because I found customizing them was very complicated —headlines/footlines and lists. A new way to redefine the former is accomplished in my own *titlesec* package, but none was available to customize the latter except:

- **enumerate**, which just allows to change the label and it does it pretty well (or almost, as in Spanish a standard label is *a*) which cannot be set with `\emph{a}`) and one must write `\itshape a\/\upshape`).
- **mdwlist**, which only “provides some vaguely useful list-related commands and environments,” as its manual states, and not a coherent way of handling lists.
- **paralist**, which provides lists within a paragraph (the original purpose of this package), a few other hard-wired specific changes and the optional argument of **enumerate**. Note **enumitem** does not allow grouping items within a paragraph yet.

One of the main drawbacks of the standard **list** is its weird parameters, whose meaning is not always obvious. In order to provide a cleaner interface two approaches were possible: either defining new lists, or introducing a new syntax making the standard lists easier to customize. For marks I took the first approach in *titlesec*, just because I did not manage to find a satisfactory solution with the L<sup>A</sup>T<sub>E</sub>X internal macros, but since lists are in some sense more “complete” than sections and marks, I have taken here the second approach.

In the interface a sort of “inheritance” is used. You can set globally the behaviour of lists and then override several parameters of, say, **enumerate** and then in turn override a few parameters of a particular instance. The values will be searched in the hierarchy.

This is basically an advance of a project which I am working on which will add this kind of inheritance to other structures of L<sup>A</sup>T<sub>E</sub>X.

## 2 The package

This package is intended to ease customizing the three basic list environments: **enumerate**, **itemize** and **description**. It extends their syntax to allow an optional argument where a set of parameters in the form **key=value** are available:

---

\*For bug reports, comments and suggestions go to <http://www.texytipografia.com/contact.html>. English is not my strong point, so contact me when you find mistakes in the manual. Other packages by the same author: *gloss* (with José Luis Díaz), *accents*, *tensind*, *esindex*, *dotlessi*, *titlesec*, *titletoc*.

- Vertical spacing:
  - `topsep`
  - `partopsep`
  - `parsep`
  - `itemsep`
- Horizontal spacing:
  - `leftmargin`
  - `rightmargin`
  - `listparindent`
  - `labelwidth`
  - `labelsep`
  - `itemindent`

These are equivalent to the well known list parameters, but two of them, namely `leftmargin` and `labelsep`, can have in addition the special value of `*` as described below.

The following new parameters are provided:

- `font`. Sets the label font. Useful when the label is changed with the optional argument of `\item`.
- `label`. Sets the label to be used in the current level. A set of starred versions of `\alph`, `\Alph`, `\arabic`, `\roman` and `\Roman`, without argument stand for the current counter in `enumerate`.<sup>1</sup> Thus

```
\begin{enumerate}[label=\emph{\alph*}]]
```

prints *a*), *b*), and so on (this is a standard style in Spanish).

A fancier example (which looks ugly, but it is intended only to illustrate what is possible; requires `color` and `pifont`):

```
\begin{enumerate}[label=\protect\fcolorbox{blue}{yellow}{\ding{\value*}},
start=172]
```

- `label*` is like `label` but its value is appended to the current label. For example, the following defines a `legal` list (1., 1.1., 1.1.1., and so on):

```
\newlist{legal}{enumerate}{10}
\setlist[legal]{label*=\arabic*.
```

- `ref`. By default, `label` sets also the form of cross references and `\the...`, but you can define a different format. For example, to remove the right parenthesis:

```
\begin{enumerate}[label=\emph{\alph*}),ref=\emph{\alph*}]
```

In both `label` and `ref`, the counters can be used as usual:

```
\begin{enumerate}[label=\thenumi.\arabic*.]
```

---

<sup>1</sup>Actually, the asterisk is currently the argument but things may change. Consider them as starred variants and follow the corresponding syntax.

or

```
\begin{enumerate}[label=\arabic{enumi}.\arabic*.]
```

(provided the current level is the second one).

Note the `labels` are *not* accumulated to form the reference. If you want, say, something like 1.a from 1) as first level and a) as second level, you must set it with `ref`. You may use `\ref{level1}.\ref{level2}` with appropriate `ref` settings, but as Robin Fairbairns points out in the T<sub>E</sub>X FAQ

... [that] would be both tedious and error-prone. What is more, it would be undesirable, since you would be constructing a visual representation which is inflexible (you could not change all the references to elements of a list at one fell swoop).

This is sensible and I recommend to follow the advice, but sometimes you might want something like:

```
... subitem \ref{level2} of item \ref{level1} ...
```

Both `label` and `ref` are moving arguments, and fragile commands must be protected *except* the counters.

- `leftmargin=*`. Sets the label width to the width of the current label, using the default value of 0 in `\arabic*`, `viii` in `\roman*`, `m` in `\alph*` and similarly in uppercase forms. The left margin is then set to `labelwidth + labelsep`. You can give a value to the key `widest`, and it will be used to set the margin. For example:

```
\begin{itemize}[label=\textbullet, leftmargin=*]  
\begin{enumerate}[label=\roman*], leftmargin=*, widest=iii]
```

Put in in other words, the label starts at the same place as the surrounding text (but see `labelindent` below), then comes the label, the space after it as set by `labelsep` and the item body.

- `labelsep=*`. Similar to the previous one but here the label sep is set to `leftmargin - labelwidth`.

```
\begin{itemize}[label=\textbullet,  
leftmargin=2pc, labelsep=*]
```

As before, the label starts at the same place as the surrounding text, then comes the label, the space after it and the item body. The difference between `labelsep=*` and `leftmargin=*` is that in the former the item body begins at a fixed place (namely, `leftmargin`), while the latter begins at a variable place depending on the label (but always the same within a list, of course). Most of times, what you would want is `leftmargin=*`.

Unfortunately, `labelsep` has not a default value to be applied to all lists—simply the current value is used. If you nest a `leftmargin=*` within a `labelsep=*`, the value to compute the left margin is not what you could expect. The way L<sup>A</sup>T<sub>E</sub>X assigns values to the list parameters is one of the things which makes redefining lists a mess, but with `enumitem` you can set default values for every list, as described below. In this case, all you need is:

```
\setlist{itemsep=.5em}
```

- **widest.** To be used in conjunction with any of the two previous keys, if desired. It overrides the default values for the label width. Sometimes, if lists are not very long, a value of **a** for **\alph** is more sensible than **m**:

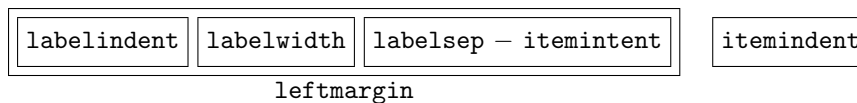
```
\begin{enumerate}[leftmargin=*,widest=a] % Assume standard 2nd level
```

- **labelindent.** Blank space before the label with **leftmargin=\*** and **labelsep=\***, and in description. Example:

```
\begin{enumerate}[label=\arabic*., leftmargin=2\parindent,
                    labelindent=\parindent, labelsep=*]
```

Since **\parindent** is not used as such inside lists, but instead is set internally to either **itemindent** or **listparindent**, when used as the value of a parameter **enumitem** returns the global value, i. e., the value it has outside the outermost list.

The horizontal space in the left margin of the current level is distributed in the following way:



Remember **labelsep** spans part of **leftmargin** and **itemindent** if the latter is not zero. This is often somewhat confusing, so remember that if you want a separation between label and item *body* of, say, 6pt, and the item indent is 8pt, then **labelsep** should be 6 + 8. The foregoing discussion about how the **\*** values are computed presumes **itemindent** is 0 pt, but any other value it is taken into account.

- **align.** How the label is aligned. Two values are possible: **left** and the default **right**. The parameters controlling the label spacing should be properly set, either by hand or more conveniently with the **\*** settings:

```
\begin{enumerate}[label=\Roman*., align=left, leftmargin=*]
```

- **start.** Sets the number of the first item.
- **resume.** Makes the counter to be continued from the previous **enumerate**, instead of resetting it to 1.

```
\begin{enumerate}
\item First item.
\item Second item.
\end{enumerate}
Text.
\begin{enumerate}[resume]
\item Third item
\end{enumerate}
```

- **resume\*** is like **resume** but the options from the previous list are used, too. This option must be restricted to the optional argument in a environment (this is the only place where it makes sense) and further keys are not allowed. It should be used sparingly—if you are using it often, then very likely you want to define a new list (see below).

- `beginpenalty`, `midpenalty` and `endpenalty` set the penalty at the beginning of a list, between items and at the end of the list. Please, refer to your  $\text{\LaTeX}$  or  $\text{\TeX}$  manual about how penalties control page breaks. Unlike other parameters, when a list starts their values are not reset to the default, thus they apply to the enclosed lists, too.
- `before` and `before*` executes code before the list starts (more precisely, in the second argument of the `list` environment used to define them). The former sets the code to be executed, overriding any previous value, while the starred form adds the code to the existing one (in the setting hierarchy, see below, *not* with relation to the enclosing list). It can contain, say, rules and text, but this has not been extensively tested.
- `after` and `after*` is the same but just before the list ends.

A key are available in `description`

- `style`. Sets the *style*, with the following values:
  - `nextline`: if the label does not fit in the margin, the text continues in the next line, otherwise it is placed in a box of width `leftmargin`, ie, the item body never sticks into the left margin.
  - `same`line: like `nextline` but if the label does not fit in the margin the text continues in the same line.
  - `multiline`: the label is placed in a parbox whose width is `leftmargin`, with several lines if necessary.
  - `unboxed`: much like the standard `description`, but the label is not boxed to avoid uneven spacing and unbroken labels if they are long.

A few predefined valueless keys might be useful:

- `noitemsep` kills the space between items and paragraphs (ie, `itemsep=0pt` and `parsep=0pt`).
- `nolistsep` kills all vertical spacing.
- `fullwidth` is still somewhat experimental. The `leftmargin` is null and the label is part of the text. Here `labelsep` sets the separation between the label and the first word. You may want to set `itemindent` and `listparindent` to `\parindent`, too.

Global changes, to be applied to all of these list, are also possible:

- `\setenumerate[level]{format}`
- `\setitemize[level]{format}`
- `\setdescription{format}`
- `\setlist[level]{format}`

Where `level` is the list level in `list`, and the corresponding level in `enumerate` and `itemize`. Without `level`, the format applies to all levels. Here list does not mean any list but only the three ones handled by this package. Example:

```
\setlist{noitemsep}
\setlist[1]{\labelindent=\parindent} % << Usually a good idea
\setitemize{leftmargin=*}
\setitemize[1]{label=$\triangleleft$}
\setenumerate{labelsep=*, leftmargin=1.5pc}
```

```

\setenumerate[1]{label=\arabic*., ref=\arabic*}
\setenumerate[2]{label=\emph{\alph*}},
    ref=\theenumi.\emph{\alph*}}
\setenumerate[3]{label=\roman*}, ref=\theenumii.\roman*}
\setdescription{font=\sffamily\bfseries}

```

These settings are read in the following order: list, list at the current level, enumerate/itemize/description, and enumerate/itemize at the current level; if a key has several values, the last one, i.e. the most specific one, is applied. Finally, the optional argument, if any, is applied. L<sup>A</sup>T<sub>E</sub>X provides a set of macros to change many of these parameters, but setting them with the package is more consistent and sometimes more flexible at the cost of being more “explicit” (and verbose).

No attempt has been made (yet) to discriminate the font size (`\normalsize`, `\small...`). Suggestions are most welcome.

## 2.1 enumerate-like labels

With the package option `shortlabels` you can use an `enumerate`-like syntax, where `A`, `a`, `I`, `i` and `1` stand for `\Alph*`, `\alph*`, `\Roman*`, `\roman*` and `\arabic*`. This is intended mainly as a sort of compatibility mode with the `enumerate` package, and therefore the following special rule applies: if the very first option (at any level) is not recognized as a valid key, then it will be considered a label with the `enumerate`-like syntax. For example:

```

\begin{enumerate}[i], labelindent=\parindent]
...
\end{enumerate}

```

Although perhaps not so useful, you can omit `label=` in the `itemize` environment under similar conditions, too:

```

\begin{itemize}[\textbullet]
...
\end{itemize}

```

In addition, you can define new keys (or redefine them), which is particularly useful for `enumerate` to be adapted to specific typographical rules or to extend it for non-Latin scripts:

```

\SetEnumerateShortLabel{<key>}{<replacement>}

```

where `<replacement>` contains one of the starred versions of counters. For example:

```

\SetEnumerateShortLabel{i}{\textsc{\roman*}}

```

redefines `i` so that items using this key are numbered with small caps roman numerals.

## 2.2 Cloning the three basic lists

As of version 2.0 the three lists can be cloned so that you can define “logical” environments behaving like them. To define a new list (or redefine an existing one), use `\newlist` (or `\renewlist`), which has the following syntax:

```

\newlist{<name>}{<type>}{<max-depth>}

```

where `<type>` is `enumerate`, `itemize` or `description`. Then you can (in fact you must, at least the label) set the new list with `\setlist`:

```

\newlist{ingredients}{itemize}{1}
\setlist[ingredients]{label=\textbullet}
\newlist{steps}{enumerate}{2}
\setlist[steps,1]{label=(\arabic*)}
\setlist[steps,2]{label=(\arabic*)}

```

The optional argument of `\setlist` has an extended syntax—names say which lists applies the settings to, and numbers the level. Do not use this syntax in the three basic lists, but that above, except if you redefine them.

## 2.3 New counters

With

```
\AddEnumerateCounter{<LaTeX counter>}{<TeX count>}{<widest label>}
```

a counter representation is “registered” so that `enumitem` recognizes it. Intended mainly for non Latin scripts, but also useful in Latin scripts. For example:

```

\def\c@text#1{\expandafter\@c@text\csname c@#1\endcsname}
\def\@c@text#1{\ifcase#1\or First\or Second\or Third\or
Fourth\or Fifth\or Sixth\fi}
\AddEnumerateCounter{\c@text}{\@c@text}{Second}

```

## 2.4 Package option loadonly

With this package option the package is loaded but the three lists are not redefined. You can create your own lists, yet, or even redefine the existing ones.

## 3 Three patterns

Three list layouts could be considered very frequent. Let us apply the parameters above to define them.

The first pattern aligns the label with the surrounding `\parindent` while the item body is indented depending on the label and a fixed `labelsep`:

```

labelindent=\parindent,
leftmargin=*

```

A fairly frequent variant is when the label is aligned with the surrounding text:

```
leftmargin=*
```

The former looks better in the first level while the latter seems preferable in subsequent ones. That can be easily set with

```

\setlist{leftmargin=*}
\setlist[1]{labelindent=\parindent} % Only the level 1

```

The second pattern aligns the item body with the surrounding `\parindent`. In this case:

```
leftmargin=\parindent
```

A third pattern would be to align the label with `\parindent` and the item body with `2\parindent`:

```

labelindent=\parindent,
leftmargin=2\parindent,
itemsep=*

```

Again, a variant would be to align the label with the surrounding text and the itembody with `\parindent`:

```
leftmargin=\parindent,
itemsep=*
```

Note here `\parindent` means the global value applied to normal paragraphs.

## 4 The trivlist issue

$\text{\LaTeX}$  uses a simplified version of `list` named `trivlist` to set displayed material, like `center`, `quote`, etc., even if conceptually they are not lists. Unfortunately, `trivlist` uses the current list settings, which has the odd side effect that changing the vertical spacing of lists also changes sometimes the spacing in these environments.

This package modifies `trivlist` so that the default settings for the current level (ie, those set by the corresponding `clo` files) are set again. In standard  $\text{\LaTeX}$  that is usually redundant, but if we want to fine tune lists, not resetting the default values could be a real issue (particularly if you use the `nolistsep` option).

A minimal control of vertical spacing has been made possible with

- `\setdisplay[level]{format}`

but `trivlist` itself, which is not used directly very often, does not accept an optional argument. This feature is not intended as a full-fledge `trivlist` formatter.

If for some reason you do not want to change `trivlist` and preserve the original definition, you can use the package option `ignoredisplayed`.

## 5 Samples

In these samples we set `\setlist{noitemsep}`

---

```
En un lugar de la Mancha, de cuyo nombre no quiero acordarme,
no ha mucho tiempo que viv\'\{i}a un hidalgo de los de
\begin{enumerate}[labelindent=\parindent,leftmargin=*]
  \item lanza en astillero,
  \item adarna antigua,
  \item roc\'\{i}n flaco, y
  \item galgo corredor.
\end{enumerate}
Una olla de algo m\'\{a}s vaca que carnero, salpic\'\{o}n las m\'\{a}s
noches, duelos y quebrantos los s\'\{a}bados...
```

En un lugar de la Mancha, de cuyo nombre no quiero acordarme, no ha mucho tiempo que vivía un hidalgo de los de

1. lanza en astillero,
2. adarna antigua,
3. rocín flaco, y
4. galgo corredor.

Una olla de algo más vaca que carnero, salpicón las más noches, duelos y quebrantos los sábados...

---

```
With \begin{enumerate}[leftmargin=*]
```

En un lugar de la Mancha, de cuyo nombre no quiero acordarme, no ha mucho tiempo que vivía un hidalgo de los de

1. lanza en astillero,



2. adarna antigua,
3. rocín flaco, y
4. galgo corredor.

Una olla de algo más vaca que carnero, salpicón las más noches, duelos y quebrantos los sábados...

With `\begin{enumerate}[leftmargin=\parindent]`

En un lugar de la Mancha, de cuyo nombre no quiero acordarme, no ha mucho tiempo que vivía un hidalgo de los de

1. lanza en astillero,
2. adarna antigua,
3. rocín flaco, y
4. galgo corredor.

Una olla de algo más vaca que carnero, salpicón las más noches, duelos y quebrantos los sábados...

With

`\begin{enumerate}[labelindent=\parindent, leftmargin=*,  
label=\Roman*., widest=IV, align=left]`

En un lugar de la Mancha, de cuyo nombre no quiero acordarme, no ha mucho tiempo que vivía un hidalgo de los de

- I. lanza en astillero,
- II. adarna antigua,
- III. rocín flaco, y
- IV. galgo corredor.

Una olla de algo más vaca que carnero, salpicón las más noches, duelos y quebrantos los sábados...

With `\begin{enumerate}[label=\fbox{\arabic*}]`. A reference to the first item is 1

En un lugar de la Mancha, de cuyo nombre no quiero acordarme, no ha mucho tiempo que vivía un hidalgo de los de

- 1 lanza en astillero,
- 2 adarna antigua,
- 3 rocín flaco, y
- 4 galgo corredor.

Una olla de algo más vaca que carnero, salpicón las más noches, duelos y quebrantos los sábados...

With nested lists.

En un lugar de la Mancha, de cuyo nombre no quiero acordarme, no ha mucho tiempo que viv\''{\i}a un hidalgo de los de

```
\begin{enumerate}[label=\alph*), labelindent=\parindent,  
leftmargin=*, start=12]  
  \item lanza en astillero,  
  \begin{enumerate}[label=\alph{enumi}.\roman*), leftmargin=*, start=8]  
    \item adarna antigua,  
  \end{enumerate}  
  \item roc\''{\i}n flaco, y  
  \begin{enumerate}[label=\alph{enumi}.\roman*), leftmargin=*, resume]  
    \item galgo corredor.  
  \end{enumerate}  
\end{enumerate}
```

Una olla de algo m\''{a}s vaca que carnero, salpic\''{o}n las m\''{a}s  
noches, duelos y quebrantos los s\''{a}bados...

En un lugar de la Mancha, de cuyo nombre no quiero acordarme, no ha mucho tiempo que vivía un hidalgo de los de

- l) lanza en astillero,
- l.vii) adarna antigua,
- m) rocín flaco, y
- m.viii) galgo corredor.

Una olla de algo más vaca que carnero, salpicón las más noches, duelos y quebrantos los sábados...

---

En un lugar de la Mancha, de cuyo nombre no quiero acordarme, no ha mucho tiempo que viv\'\i}a un hidalgo de los de

```
\begin{description}[font=\sffamily\bfseries, leftmargin=3cm,
style=nextline]
\item[Lo primero que ten\'\i}a el Quijote] lanza en astillero,
\item[Lo segundo] adarna antigua,
\item[Lo tercero] roc\'\i}n flaco, y
\item[Y por \'\u}ltimo, lo cuarto] galgo corredor.
\end{description}
```

Una olla de algo m\'\a}s vaca que carnero, salpic\'\o}n las m\'\a}s  
noches, duelos y quebrantos los s\'\a}bados...

En un lugar de la Mancha, de cuyo nombre no quiero acordarme, no ha mucho tiempo que vivía un hidalgo de los de

**Lo primero que tenía el Quijote**

lanza en astillero,

**Lo segundo**

adarna antigua,

**Lo tercero**

rocín flaco, y

**Y por último, lo cuarto**

galgo corredor.

Una olla de algo más vaca que carnero, salpicón las más noches, duelos y quebrantos los sábados...

---

Same, but with **sameline**.

En un lugar de la Mancha, de cuyo nombre no quiero acordarme, no ha mucho tiempo que vivía un hidalgo de los de

**Lo primero que tenía el Quijote** lanza en astillero,

**Lo segundo**

adarna antigua,

**Lo tercero**

rocín flaco, y

**Y por último, lo cuarto** galgo corredor.

Una olla de algo más vaca que carnero, salpicón las más noches, duelos y quebrantos los sábados...

---

Same, but with **multiline**. Note the text overlaps if the item body is too short.

En un lugar de la Mancha, de cuyo nombre no quiero acordarme, no ha mucho tiempo que vivía un hidalgo de los de

**Lo primero que** lanza en astillero,

**tenía el Quijote**

adarna antigua,

**Lo tercero**

rocín flaco, y

**Y por último, lo**

galgo corredor.

**cuarto**

Una olla de algo más vaca que carnero, salpicón las más noches, duelos y quebrantos los sábados...

## 6 Afterword

### 6.1 L<sup>A</sup>T<sub>E</sub>X lists

As it is well known, L<sup>A</sup>T<sub>E</sub>X predefines three lists: enumerate, itemize and description. This is a very frequent classification which one can also find in, say, HTML. However, there is a more general model based in three fields—namely, label, title, and body—, so that enumerate and itemize has label (numbered and unnumbered) but no title, while description has title but no label. In this model, one can have a description with entries marked with labels, as for example:

```

\newcommand\litem[1]{\item{\bfseries #1,\enspace}}
\begin{itemize}[label=\textbullet]
\litem{Lo primero que ten\'}{\i}a el Quijote} lanza en astillero,
... etc.

```

---

En un lugar de la Mancha, de cuyo nombre no quiero acordarme, no ha mucho tiempo que vivía un hidalgo de los de

- **Lo primero que tenía el Quijote,** lanza en astillero,
- **Lo segundo,** adarna antigua,
- **Lo tercero,** rocín flaco, y
- **Y por último, lo cuarto,** galgo corredor.

---

This format is not infrequent at all and I think any future development of this package has to take it into account.

## 6.2 Known issues

Description styles other than the default one don't group labels if they are nested.

You must use the  $\LaTeX$  syntax for environments, with `\begin` and `\end`.

## 6.3 Acknowledgements

I wish to thank particularly the comments and suggestions from Lars Madsen, which has found some bugs, too.