



The Treasure Chest

Package tours from CTAN

Give us the tools, and we will finish the job.

Churchill, in a broadcast dated February 1941

Well, everyone's a Churchill these days—we all want good tools to help, not hinder, us as we get on with our jobs, albeit more peaceful ones than in 1941. Hopefully some of these packages will provide just the tools *you* need.

The selection focuses on tools to achieve certain layout effects for authors:¹ itemized lists shaped as paragraphs; managing the use and definition of acronyms; epigraphs—those ‘pithy quotations often found at the start (or end) of a chapter’ (or the start of a column, in this case); and two types of ‘hanging’—paragraphs and punctuation. So go fetch the files and have some fun!

All packages described in this column are available in the following location on CTAN:²

`macros/latex/contrib/supported`

Only the specific subdirectory will be identified in individual descriptions, to reduce the repetition—and the line length of such paths!

Another commonality: many of these packages use two mechanisms afforded by L^AT_EX 2_ε—`.ins` and `.dtx` files.

The `.ins` file is run through L^AT_EX in order to generate the style file; the `.dtx` file is usually the main source of documentation and should therefore also be processed. Different T_EX installations store such files differently; check your documentation or your machine on where other such files are located, and add these to the same places.

Note that, on occasion, it may be necessary to process the documentation file two or even three times, in order to resolve all cross-references. Processing of these files sometimes generates additional files which the authors have found useful to include, along with the usual `.log` and `.aux` files.

¹ I would like to thank the authors—Bernd Schandl, Tobias Oetiker, and Peter Wilson—who were so cooperative and helpful as I worked through their package instructions, the cumulative package interferences, and so on. It's been fun putting this column together—and it's been a pleasure working with these authors.

² Either via the web at www.tug.org (follow the links) or via ftp to [ctan.tug.org/tex-archive](ftp://ctan.tug.org/tex-archive).

1 paralist

1.1 Quick Tour

Package: `paralist`

This is version 1.8, dated 29 Apr. 1999.

Author: Bernd Schandl

`bschand@math.clemson.edu`

Compatible with: L^AT_EX 2_ε

Keywords:

in-paragraph list environments, customizable labels

Purpose:

`paralist.sty` provides some new list environments. Itemized and enumerated lists can be typeset within paragraphs, as paragraphs and in a compact version. Most environments have optional arguments to format the labels. Additionally, the default L^AT_EX environments `itemize` and `enumerate` can be extended to use a similar optional argument.

Explanation of the name: “lists in paragraph shape.”

Location on CTAN: `/paralist`

Files to fetch:

`README`
`paralist.ins`
`paralist.dtx`
`license.txt`.³

How to install:

Put the `.dtx` and `.ins` files with the other class and style files on your system. The `README` file provides processing information.

Files generated:

`paralist.sty` (the style file)
`paralist.dvi` (documentation)
`paralist.drv` (more detailed documentation with code + index)

Documentation:

Processing the `.dtx` file yields 7 pages. Additional documentation can also be generated, following instructions in the `README` file.

Table of Contents:

1. Introduction
2. Package Options
3. Formatting the Labels
4. Defaults for Labels and Margins

³ The new license text (LPPL): “This package can be redistributed and/or modified under the terms of the L^AT_EX Project Public License distributed from CTAN archives in the directory `macros/latex/base/lppl.txt`; either version 1 of the license, or (at your option) any later version.” The regular user can just delete the file after downloading, because a pointer to the license is included in every file anyway.

5. New Environments
 - (a) Enumerated
 - (b) Itemized
 - (c) Descriptive
 - (d) Blank
6. Nesting Environments
7. Fine-tuning
8. Bugs and Wishes
9. Acknowledgments

1.2 Detailed tour

The package grew out of `theapa`,⁴ and also owes a great deal to David Carlisle's `enumerate` package, as noted by the author.

The documentation details four package options:

1. Two of these allow the redefinition of the standard environments `enumerate` and `itemize`, so that they have an optional argument to format the labels.
2. Another option addresses the space required for labels.
3. The final option is to define some list environments without labels, an odd but sometimes useful option.

The above demonstrates one of the two new environments: (1) `asparaenum`, whereby each `\item` is set as a separate paragraph, and (2) `inparaenum`, which sets its `\items` into one continuous paragraph. This paragraph, by the way, demonstrates the use of `inparaenum`, with the addition of a label option, `[\bfseries (1)]`, as opposed to the default \LaTeX strategy, which requires customized options for each instance of `\item`.

There are matching environments for the 'itemize' type of list: `asparaitem` and `inparaitem`, also with options to modify their labels. And, as mentioned in the 'Purpose' section, there are compact versions of these environments—where all the vertical skips have been set to zero. It was used above, to set the Table of Contents list.

1.3 Final notes

Here follows a whimsical example of resetting the labels for the `compactitem` environment. Note that since there is no 'counting' going on, `compactenum` would not work.⁵

⁴ A \LaTeX /AMS- \LaTeX document style option (supported) for use with the `theapa` Bib \TeX bibliography style (`theapa.bst`), by Young U. Ryu `ryoung@utdallas.edu`. Latest version is v2.5.1 (May 1992). See also `theapa.tex`, `theapa.bst`.

⁵ Actually, I originally had used `compactenum`, not thinking about this counting being the essential difference between

```
\defaultitem{ $\clubsuit$ }
           { $\diamondsuit$ }
           { $\heartsuit$ }
           { $\spadesuit$ }
```

```
 $\clubsuit$  something first
   $\diamond$  something first
     $\heartsuit$  something first
       $\spadesuit$  something first
       $\spadesuit$  something second
     $\heartsuit$  something second
   $\diamond$  something second
 $\clubsuit$  something second
```

The author has one query (p. 5 in the documentation):

... does anybody know why `description` has to be defined by the document class and is not defined in `ltlists.dtx`?

Send your replies directly to Bernd, and hopefully he can come back with an answer!

The author also has two comments on additional features to look for:

1. Using `\defaultenum` to redefine the counter styles throughout the whole document is pretty cool, but maybe I am a little biased here.
2. With a little extra work (by redefining `\pl@hook`) it is possible to use basically everything (foot-note symbols, Greek characters ...) as counters for enumerated lists.

The flexibility the package affords, while taking a few tries to sort out, cannot fail to intrigue and please writers of long documents—or rather, of multi-part paragraphs! Lawyers come to mind, for some reason ;-)

2 acronym

2.1 Quick Tour

Package: `acronym`

This is version 1.3, dated 1996/09/19.

Author: Tobias Oetiker

`oetiker@ee.ethz.ch`

Compatible with: $\LaTeX 2_{\epsilon}$

Keywords:

acronyms, mottos, topical quotations; full and short versions; auto-generation of acronym listing

Purpose:

The `acronym` package ensures that all acronyms

the two environments. So now that I've made that mistake for you ... !

used in the text are spelled out in full at least once. The package provides several commands and one environment for dealing with acronyms.

Location on CTAN: /acronym

Files to fetch:

acronym.ins

acronym.dtx

How to install:

Process both the .ins and .dtx files and then store where other such files are kept.

Files generated:

acronym.sty (the style file)

acronym.dvi (documentation)

acrotest.tex (sample file)

Documentation:

The .dtx documentation runs to 5 pages; no separate `readme` is provided. Instructions on processing the .dtx and test file are in the .ins file, but no explicit instructions that the .ins file be run through $\text{\LaTeX}2_{\epsilon}$ are provided.

Table of Contents:

1. Introduction
2. The user interface
 - (a) Acronyms in the Text
 - (b) Defining Acronyms
3. An example file
4. The implementation

2.2 The detailed tour

This package covers such a small corner of the writing/editing/typesetting and yet, it will probably become invaluable for those whose writings require that they manage and control large numbers of acronyms.

The writing convention of using the complete name or title ('Full Name'), followed by the acronym in parentheses, for first appearance, and then only the acronym ('short name') for subsequent instances, is handled via four macros for the various treatments.

In addition, should one want to print up the acronyms in an appendix or other section, with both full and short names, along with any additional description, an `acronym` environment is available.

The documentation provides the verbatim input code for the sample file, to demonstrate the coding. A final section, 'The implementation', provides details on how the macros work.

2.3 Sample file

In the early nineties, GSM was deployed in many European countries. Global System for Mobile com-

munication (GSM) offered for the first time international roaming for mobile subscribers. The GSM's use of Time Division Multiple Access (TDMA) as its communication standard was debated at length. And every now and then there are big discussions whether Code Division Multiple Access (CDMA) should have been chosen over TDMA.

If you want to know more about Global System for Mobile communication (GSM), Time Division Multiple Access (TDMA), Code Division Multiple Access (CDMA) and other acronyms (oa), just read a book about mobile communication.

2.3.1 Acronyms Used

GSM Global System for Mobile communication.

GSM is the new standard for digital cellular communication in Europe.

TDMA Time Division Multiple Access. Some would say that this is not as good as CDMA.

CDMA Code Division Multiple Access. The spread spectrum modulation used in the Qualcomm system.

2.4 Final notes

Incorporating the `acronym` package into *TUGboat's* collection of macros has led to a clash in control sequence names: we both use `\acro!` In *TUGboat*, this is used to set ALLCAPS acronyms into the next smaller font: ALLCAPS. However, it's also a very logical macro to use for acronym macros. Our solution? After loading the *TUGboat* macros, but before loading the `acronym` package, the following lines were inserted in the preamble:

```
\let\TBacro\acro
\let\acro\relax
```

And then all instances of `\acro` for ALLCAPS were changed to as `\TBacro`.

One more point. As any editor knows, the first thing we do is go in and change things! Well, there is one small part which can't be changed—yet. In testing, I found that the first argument for `\acro`, where the initials are put, won't accept changes in font size or style. The author has agreed that this would be a worthwhile feature and will incorporate it in the next upgrade. We will begin to carry upgrade information as it becomes available for this, and any other package found in the 'Treasure Chest'.

3 epigraph

3.1 Quick Tour

Package: epigraph

This is version 1.1, dated 1998/11/29.

Author: Peter R. Wilson

`peter.r.wilson@boeing.com`

Compatible with: L^AT_EX 2_ε

Keywords:

epigraphs, epigram, quotation

Purpose:

The `epigraph` package is designed for typesetting epigraphs—pithy quotations often found at the start (or end) of a chapter. It provides commands for typesetting a single epigraph and environments for typesetting a list of epigraphs. Epigraphs can be typeset at either the left, the center, or the right of the typeblock. Similarly, the source line can appear in different positions.

Location on CTAN: /epigraph

Files to fetch:

README

epigraph.ins

epigraph.dtx

How to install:

The README file provide instructions for two levels of detail. The usual sequence of processing the `.ins` file and then the `.dtx` file (run this latter three times, to resolve all cross-references) yield the style file and documentation. Store these where other such files are kept.

For additional processing of an index `.idx` file, there are instructions for using `makeindex`. But it is not necessary to do the second level in order to begin installing and using the package.

Files generated:

epigraph.sty (the style file)

epigraph.dvi (documentation)

epigraph.toc and epigraph.idx (for `makeindex` documentation)

Documentation:

The documentation runs to 8 pages, with the macros nicely set off in the left margin, as is common with `.dtx` material.

Table of Contents:

1. Contents
2. Introduction
3. The `epigraph` package
 - (a) The `epigraph` command
 - (b) The `epigraphs` environment
 - (c) Epigraphs before chapter headings
4. General
5. The package code
 - (a) Epigraphs before a chapter title
6. References

3.2 The detailed tour

The documentation seems to cover almost all possible combinations and situations one would normally

expect to run into—however, see the first epigraph, below, to explain the unanticipated uses!

The macros are very simple, the explanations clear, with sufficient verbatim samples. I like it!

Below are the epigraphs used in last year's issues of *TUGboat*:

Book design no less than ship design or any other kind of design is a matter of looking backward and forward at the same time. The grandeur of a volume's illustrations may so dominate its design that portability and readability become subsumed in the dominant objective.

Petroski (1997), in *TUGboat*, 19,1

No matter how many palettes of buttons and how many menu options are offered, users of a program will always want to do something the author has not foreseen. Adding still more buttons and menus is not the answer.

Hayes (1995), in *TUGboat*, 19,2

[proceedings issues use this space to provide details on the conference]

TUGboat, 19,3

Certainly, if I were a publishing house, if I were in the publishing business myself, I would have probably had ten different versions of T_EX by now for ten different complicated projects that had come in. They would all look almost the same as T_EX, but no one else would have this program—they wouldn't need it, they're not doing exactly the book that my publishing house was doing.

Knuth (1996), in *TUGboat*, 19,4

I added `\itemsep=1.5pc` to get some space between each one. Also, by setting the following parameter (line-wrapped to fit the column), the font size/style can be changed for the entire environment:

```
\renewcommand{\epigraphsize}%
{\bfseries\footnotesize}
```

The documentation doesn't mention the bit about font style being changeable via this command, but it does work. Also, each of the two arguments—one for the text, one for the source—will accept the usual L^AT_EX font commands.

3.3 Final notes— and a query

The author, Peter Wilson, offered up the following epigraphs for the opening of this issue of the Treasure Chest. And while I like my own little discovery,

his are too good to leave out. Notice the variations in position for the source line(s).

The epigraph is among the most delightful of scholarly habits.

MARY-CLAIRE VAN LEUNEN,
A Handbook for Scholars, 1978

I hate quotations. Tell me what you know.

Ralph Waldo Emerson, *Journals*, May 1849

QUOTATION, n. The act of repeating erroneously the words of another. The words erroneously repeated.

Ambrose Bierce,
The Devil's Dictionary, 1906

There is no amount of praise which a man and an author cannot bear with equanimity. Some authors can even stand flattery.

Maurice Baring, *Dead Letters*, 1910

The query is from the author:

As a result of someone's posting to the newsgroup `comp.text.tex` I am considering adding a facility in the `epigraph` package for putting typeset material on the (verso) page immediately before the (recto) page on which the chapter heading is typeset. *If* I do this (and it would be helpful to know if this was a common need), then it would be in a new version of the package.

4 hanging

4.1 Quick Tour

Package: hanging

This is version 1.0, dated 1998/11/29.

Author: Peter R. Wilson

`peter.r.wilson@boeing.com`

Compatible with: L^AT_EX 2_ε

Keywords:

hanging paragraphs, hanging punctuation

Purpose:

The `hanging` package provides facilities for defining hanging paragraphs and hanging punctuation.

Location on CTAN: /hanging

Files to fetch:

README

`hanging.ins`

`hanging.dtx`

How to install:

The README file provides the by-now familiar instructions on handling the two other files.

Files generated:

`hanging.sty` (the style file)

`hanging.dvi` (documentation)

`hanging.toc` and `hanging.idx` (for `makeindex` documentation)

Documentation:

Documentation is very clear and spare: 5 pages in all, including the documented source code. The README file provides processing information.

Table of Contents:

1. Contents
2. Introduction
3. The `hanging` package
 - (a) Hanging paragraphs
 - (b) Hanging punctuation
4. The package code
 - (a) Hanging paragraphs
 - (b) Hanging punctuation
5. References

4.2 The detailed tour

Very nifty little set of macros! Very simple to understand and then use. For hanging paragraphs, you can choose between one macro for one paragraph, or an environment for several paragraphs. Both commands take two arguments: one to specify the left indent, the second to specify how many lines to print before starting the hanging indent. For example,

```
\hangpara{3pc}{1}
```

yields a 3pc indent from the left, and only one full line of text before the indent comes into play. Now, by setting that first argument to a negative value (`-3pc`), the indent switches from the left margin to the right, as done with this paragraph. The potential for playing around should be obvious!

The `hangparas` environment can be contrasted with the effects of L^AT_EX's `description`, mainly in terms of ease of control of the `\leftmargin` value. Below are the same two texts, first done with `hangparas` (with a 1pc indent), then with `description`.

`paralist`: provides some new list environments. Itemized and enumerated lists can be typeset within paragraphs, as paragraphs and in a compact version. Most environments have optional arguments to format the labels.

`acronym`: ensures that all acronyms used in the text are spelled out in full at least once. The package provides several commands and one environment for dealing with acronyms.

epigraph: is designed for typesetting epigraphs — the pithy quotations often found at the start (or end) of a chapter. It provides commands for typesetting a single epigraph and environments for typesetting a list of epigraphs. Epigraphs can be typeset at either the left, the center, or the right of the typeblock.

hanging: provides facilities for defining hanging paragraphs and hanging punctuation.

paralist: provides some new list environments. Itemized and enumerated lists can be typeset within paragraphs, as paragraphs and in a compact version. Most environments have optional arguments to format the labels.

acronym: ensures that all acronyms used in the text are spelled out in full at least once. The package provides several commands and one environment for dealing with acronyms.

epigraph: is designed for typesetting epigraphs — the pithy quotations often found at the start (or end) of a chapter. It provides commands for typesetting a single epigraph and environments for typesetting a list of epigraphs. Epigraphs can be typeset at either the left, the center, or the right of the typeblock.

hanging: provides facilities for defining hanging paragraphs and hanging punctuation.

As for hanging punctuation (look carefully at the margins of this paragraph), while the author views it as an interesting — if less-than-practical — facility, it does work! There is a `hangpunct` environment, within which no special coding of the following punctuation marks is required: ‘ ’ . ! ? : ; , . Where the characters ‘ ’ . are required in an ‘unhung’ version, alternate macros are provided.

4.3 Final notes

In working with so many packages in one file, there are bound to be some small interference hiccups. One has already been mentioned: the sharing of the `\acro` command (see Section 2).

Another has arisen with `hanging`: a command not sufficiently localized provoked the following error message:

```
LaTeX Error: Command \allowhyphens
already defined.
```

The quick solution offered by the author:

Because of the time constraints [to get this issue out], edit the generated `hanging.sty` file and change each occurrence of `\allowhyphens` to `h@ngallowhyphens`. It is an internal command in the package, and to be consistent I should have called it `\h@ngallowhyphens` in the first place.

It is fortunate that the `\allowhyphens` command was specified using `\newcommand`, as advised for package writers, rather than using the `\def` command. In the latter case the original definitions would have been silently clobbered and probably cause the *TUGboat* editors much undeserved grief. But as it was, warnings were given of the problem so that a more robust solution could be developed.

That is, the *TUGboat* command is intended for users, so it should *not* be restricted. That’s what we’ve done, and it did the trick. When the next upgrade comes on-line, it will be reported in this column.

◇ Christina Thiele
15 Wiltshire Circle
Nepean, Ontario
K2J 4K9 Canada
cthiele@ccs.carleton.ca

Helpful hint for finding files on the CTAN archives via ftp

Don’t know where to find the package you want? The following shows how to use the `quote site index` command, to quickly locate packages. Our example is for `paralist`.

```
ftp> quote site index paralist
200-index paralist
200-NOTE. This index shows at most 20 lines. for a full list of files,
200-retrieve /tex-archive/FILES.byname
200-1999/04/01 | 2456 | macros/latex/contrib/supported/paralist/README
200-1999/04/01 | 620 | macros/latex/contrib/supported/paralist/license.txt
200-1999/04/01 | 37787 | macros/latex/contrib/supported/paralist/paralist.dtx
200-1999/04/01 | 1092 | macros/latex/contrib/supported/paralist/paralist.ins
200 (end of 'index paralist')
```