
The memoir class

Peter Wilson

Abstract

The memoir class is essentially the book and report classes with lots of bells and whistles as it includes the functions of over 30 popularly used packages. It can also simulate the appearance of article class documents and provides a basis for producing the typewritten-like manuscripts which some publishers ask for.

1 Introduction

For nearly 20 years I was involved in using L^AT_EX to produce camera-ready copy of International Standards, in particular ISO 10303 (STEP). As the standard grew—it now consists of thousands of pages spread across many publications—and ISO and various committees kept changing their minds about what they wanted, I grew increasingly frustrated with having to keep modifying the internals of the class and packages that I had developed. Separately I became interested in book design and felt that there was a need for a class that would support layout experiments. I worked on this in a desultory fashion for several years and eventually produced something that I felt might be generally useful.

The memoir class was first released in 2001 and since then has proven to be reasonably popular. The class can be used as a replacement for the book and report classes, by default generating documents virtually indistinguishable from ones produced by those classes. The class includes options to produce documents with other appearances; for example an article class look or one that looks as though the document was produced on a typewriter with a single font, double spacing, no hyphenation, and so on. In the following I use the term ‘standard classes’ to denote the book and report classes and, when appropriate, the article class as well.

The memoir class includes the functionality of many packages, for instance the tocloft package for controlling the table of contents and methods similar to the fancyhdr package for designing your own headers. The built-in package functions are mainly related to document design and layout; memoir does not touch upon areas like those covered by the babel or hyperref packages or any related to typesetting mathematics. On the other hand it is easy to configure a work produced with memoir to meet a university’s thesis layout requirements.

memoir has improved substantially since it was first released — over 50 L^AT_EXers have provided code

or suggestions for improvements. The class is included in the T_EX user groups’ T_EX distributions and the latest version of the class and its supporting documentation is always available from CTAN at latex/contrib/memoir.

2 General considerations

The class is a large one consisting of about 10,000 lines of L^AT_EX code documented in a 400 page report; there is no need for most users to look at this [4]. There is a separate comprehensive User Manual [3] which runs to about 300 pages and from time to time an Addendum [5] is released noting extensions to the class; at the moment this runs to about 70 pages. There is also the memexsupp package by Lars Madsen [2] which provides some extra facilities for the class.

Altogether the documentation for memoir runs to some 800 pages and it is impossible to cover everything in a short, or even a long, article.

The first part of the Manual discusses some aspects of book design and typography in general, something that I haven’t come across in the usual L^AT_EX books and manuals. This is intended to provide a little background for when you design your own printed documents.

memoir provides slightly enhanced facilities for creating title pages but in my view it is better to create your own layout for a title page. To aid in this I have created some 25 examples of title pages that can be used as a starting point for your design [6]. These were produced using regular L^AT_EX facilities and are not dependent on memoir.

The standard classes provide point options of 10, 11, or 12 points for the main body font. memoir extends this by also providing 9, 14, and 17 point options. The width of the text block is automatically adjusted according to the selected point size to try and keep within generally accepted typographical limits for line lengths; you can override this if you wish. The class also provides easy methods for specifying the page layout parameters such as the margins—both the side margins and those at the top and bottom of the page; the methods are similar to those of the geometry package.

The page layout facilities also include methods, like those provided by the fancyhdr package, for defining your own header and footer styles, and you can have as many different ones as you wish. In fact the class provides seven styles to choose from before you have to resort to creating your own. The styles are all displayed in the Manual.

Sometimes it is useful, or even required, to place trimming marks on each page showing the desired

size of the final page with respect to the sheet of paper that is used in the printer. This is provided by the `showtrims` option. A variety of trim marks are provided and you can define your own if you need some other kind.

2.1 Sectioning styles

Handles are provided for designing and using your own styles for chapter titles and such. The class comes with over 20 predefined chapter styles ranging from the default look to a style that mimics that used in the *Companion* series of L^AT_EX books. There is even one which uses words instead of numerals for chapter numbers. The Manual shows examples of at least six of these styles and about 30 are shown in Lars Madsen's collection [1].

For those who like putting quotations near chapter titles the `epigraph` environment can be used.

The options for changing `\section` and lower level titles are more constrained, but generally speaking document design, unless for advertisements or other eye-catching ephemera, should be constrained.

Sometimes, but particularly in novels, a sectional division is indicated by just leaving a blank line or two between a pair of paragraphs, or there might be some decorative item like three or four asterisks. Commands are available for typesetting such anonymous divisions.

In the standard classes, sectioning commands have an optional argument which can be used to put a short version of the section title into the table of contents and the page header. `memoir` extends this with a second optional argument so you can specify one short version for the contents and an even shorter one for page headers where space is at a premium.

2.2 Captions

`memoir` incorporates the code from my `ccaption` package which lets you easily modify the appearance of figure and table captions; bilingual captions are available if required, as are captions placed at the side of a figure or table or continuation captions from, say, one illustration to another. Captioning can also be applied to 'non-floating' illustrations or as legends (i.e., unnumbered captions) to the regular floats. The captioning system also supports subfigures and subtables along the lines of the `subfig` package, plus letting you define your own new kinds of floats together with the corresponding 'List of ...'.

3 Tables

Code from the `array`, `dcolumn`, `delarray` and `tabularx` packages is integrated within the class. To improve

the appearance of rules in tabular material the `booktabs` package is also included.

Multipage tabulations are often set with the `longtable` or `xtable` packages, which can of course be used with the class. For simple tabulations that may continue from one page to the next, `memoir` offers a 'continuous tabular' environment. This doesn't have all the flexibility provided by the packages but can often serve instead of using them.

More interestingly, but more limited, the class provides 'automatic tabulars'. For these you provide a list of simple entries, like a set of names, and a number of columns and the entries are automatically put into the appropriate column. You choose whether the entries should be added row-by-row, like this with the `\autorows` command:

```
\autorows{c}{5}{1}{one, two, three, four,
five, six, seven, eight, nine, ten,
eleven, twelve, thirteen }
```

one	two	three	four	five
six	seven	eight	nine	ten
eleven	twelve	thirteen		

Or if you use the `\autocol`s command the entries are listed column-by-column, like this:

```
\autocols{c}{5}{1}{one, two, three, four,
five, six, seven, eight, nine, ten,
eleven, twelve, thirteen }
```

one	four	seven	ten	thirteen
two	five	eight	eleven	
three	six	nine	twelve	

4 Verse

The standard classes provide a very simple `verse` environment for typesetting poetry. This is greatly extended in `memoir`. For example in the standard classes the verse stanzas are at a fixed indentation from the left margin whereas `memoir` lets you control the amount of indentation so that you can make a poem appear optically centered within the textwidth.

Stanzas may be numbered, as can individual lines within a poem. There is a special environment for stanzas where lines are alternately indented. Also you can define an indentation pattern for stanzas when this is not regular as, for example, in a limerick where the 3rd and 4th of the five lines are indented with respect to the other three as shown below.

```
\indentpattern{00110}
\begin{verse}
\begin{patverse}
There was a young man of Quebec \\\
```

```
Who was frozen in snow to his neck. \\
When asked: 'Are you friz?' \\
He replied: 'Yes, I is, \\
But we don't call this cold in Quebec.'
\end{patverse}
\end{verse}
```

<p>There was a young man of Quebec Who was frozen in snow to his neck. When asked: 'Are you friz?' He replied: 'Yes, I is, But we don't call this cold in Quebec.'</p>
--

It is not always possible to fit a line into the available space and you can specify the particular indentation to be used when a 'logical' verse line spills over the available textwidth, thus forming two or more typeset 'physical' lines. On other occasions where there are two half lines the poet might want the second half line to start where the first one finished, like this:

```
\begin{verse}
Come away with me. \\
\vinphantom{Come away with me.} Impossible!
\end{verse}
```

<p>Come away with me. Impossible!</p>
--

5 End matter

Normally appendices come after the main body of a book. The class provides various methods for introducing appendices at the end, or you can place one or more appendices at the end of selected chapters if that suits you better.

`memoir` also lets you have more than one index and an index can be set in either the normal double column style or as a single column which would be more appropriate, say, for an index of first lines in a book of poetry. The titles of any bibliography or indexes are added to the table of contents, but you can prevent this if you wish.

The class provides a set of tools for making glossaries or lists of symbols, the appearance of which can, of course, be easily altered. The `makeindex` program is used to sort the entries. An example is shown in the current version of the Addendum. A recent addition to the class provides configurable end notes which can be used as well as, or instead of, footnotes.

6 Miscellaneous

As already noted, the Manual for `memoir` runs to some 300 pages and we cannot cover everything in a

short article. Suffice it to say that hooks and macros are provided for most aspects of document layout; for instance, footnotes can be as normal, typeset in two or three columns, or all run into a single paragraph. There is a `\sidenote` macro which is a non-floating `\marginpar` as well as the `\sidebar` macro for typesetting sidebars in the margin, starting at the top of the text block. You can create new verbatim-like environments, read and write information in external files, design your own style of `\maketitle`, convert numbers to words, reserve space at the bottom of a page, and so on and so forth.

7 Packages

Most packages work with the `memoir` class, the main exception being the `hyperref` package. This package modifies many of the internals of the standard classes but does not cater for all of the differences between `memoir` and the standard ones. If you wish to use `hyperref` with `memoir` then you must use the `memhfixc` package¹ after using `hyperref`. For example:

```
\documentclass[...]{memoir}
...
\usepackage[...]{hyperref}
\usepackage{memhfixc}
...
\begin{document}
```

However, if you have a version of `hyperref` dated 2006/11/15 or after, `hyperref` will automatically call in `memhfixc` so that you don't have to do anything.

The `memoir` class includes code either equivalent to, or extensions of, the following packages; that is, the set of commands and environments provided by `memoir` is at least that of these packages: `abstract`, `appendix`, `array`, `booktabs`, `ccaption`, `chngcntr`, `chngpage`, `dcolumn`, `delarray`, `enumerate`, `epigraph`, `framed`, `ifmtarg`, `ifpdf`, `index`, `makeidx`, `moreverb`, `needspace`, `newfile`, `nextpage`, `parskip`, `patchcmd`, `setspace`, `shortvrb`, `showidx`, `tabularx`, `titlerf`, `titling`, `tocbibind`, `tocloft`, `verbatim`, `verse`.

The `memoir` class ignores any `\usepackage` or `\RequirePackage` related to these. However, if you want to specifically use one of these packages rather than the integrated version then you can do so. For argument's sake, suppose you really want to use the `titling` package; you can do this:

```
\documentclass[...]{memoir}
\DisemulatePackage{titling}
\usepackage{titling}
```

The `memoir` class incorporates a version of the `setspace` package, albeit using different names for the

¹ `memhfixc` is supplied as part of the `memoir` distribution.

macros. The package enables documents to be set double spaced but leaves some document elements, like captions for example, single spaced. To do this it has to make some assumptions about how the document class works. I felt that this kind of capability should be part of the class and not depend on assumptions. In the particular case of the `setspace` package, even with the standard classes, there can be some unexpected spacing around displayed material; this has not occurred with `memoir`'s implementation.

The class also provides functionality similar to those provided by the following packages, although the commands are different: `crop`, `fancyhdr`, `geometry`, `sidecap`, `subfigure`, `titlesec`. You can use these packages if you wish, or just use the capabilities of the `memoir` class.

Sometimes a class or package may define a command that is also, differently, defined by a succeeding package. As an example, assume that you want to use the `memoir` class together with the `pack` package but they have both defined `\amacro`. There are several ways of dealing with this.

1. Discard the class's definition:

```
\documentclass[...]{memoir}
% kill the class definition
\let\amacro\undefined% or \relax
\usepackage{pack}
```

and `pack`'s version of `\amacro` is used from now on.

2. Discard the package's definition:

```
\documentclass[...]{memoir}
% save the class definition
\let\memamacro\amacro
\let\amacro\undefined
\usepackage{pack}
% restore the class definition
\let\amacro\memamacro
```

and `memoir`'s version of `\amacro` is used from now on.

3. Keep both definitions:

```
\documentclass[...]{memoir}
\let\memamacro\amacro
\let\amacro\undefined
\usepackage{pack}
```

and after this use `\memamacro` for `memoir`'s version and `\amacro` for `pack`'s definition. But this solution doesn't always work, as you might not know when the particular versions must be used, or it is impossible to partition the uses.

A last resort is to ask the authors that one or the other macro names be changed; however, for good reasons, neither may be willing to do this.

References

- [1] Lars Madsen. Various chapter styles for the `memoir` class, July 2006. <http://mirror.ctan.org/info/latex-samples/MemoirChapStyles/MemoirChapStyles.pdf>.
- [2] Lars Madsen. The `Memoir Experimental Support Package`, February 2007. <http://mirror.ctan.org/latex/macros/contrib/memexsupp>.
- [3] Peter Wilson. `The Memoir Class for Configurable Typesetting: User Guide`, 2004. <http://mirror.ctan.org/latex/macros/contrib/memoir/memman.pdf>.
- [4] Peter Wilson. `The LATEX memoir class for configurable book typesetting: Source code`, 2005. <http://mirror.ctan.org/latex/macros/contrib/memoir/memoir.pdf>.
- [5] Peter Wilson. `Addendum: The Memoir Class for Configurable Typesetting: User Guide`, 2007. <http://mirror.ctan.org/latex/macros/contrib/memoir/memmanadd.pdf>.
- [6] Peter Wilson. `Some Examples of Title Pages`, 2007. <http://mirror.ctan.org/info/latex-samples/titlepages.pdf>.

◇ Peter Wilson
18912 8th Ave. SW
Normandy Park, WA 98166
USA
herries dot press (at)
earthlink dot net