

Addendum

ADDENDUM
The Memoir Class
for
Configurable Typesetting
User Guide
Peter Wilson

HP
The Herries Press

© 2002–2006, Peter R. Wilson
All rights reserved

The Herries Press, Normandy Park, WA.

Printed in the World

The paper used in this publication may meet the minimum requirements of the American National Standard for Information Sciences — Permanence of Paper for Printed Library Materials, ANSI Z39.48–1984.

10 09 08 07 06 05 04 03 02 20 19 18 17 16 15

First edition:	November 2002
Second impression, with minor additions	December 2002
Third impression, with additions	January 2003
Fourth impression, with additions	January 2003
Fifth impression, with minor additions	February 2003
Sixth impression, with minor additions	February 2003
Seventh impression, with minor additions	April 2003
Eighth impression, with minor additions	June 2003
Ninth impression, with minor additions	July 2003
Tenth impression, with minor additions	September 2003
Eleventh impression, with minor additions	November 2003
Second edition:	September 2005
Twelfth impression, with minor additions	October 2005
Thirteenth impression, with minor additions	November 2005
Fourteenth impression, with minor additions	January 2006
Fifteenth impression, with minor additions	August 2006

addendum, *n.* [L., gerundive of *addere*: see ADD] 1. a thing added or to be added 2. an appendix or supplement to a book, etc. 3. the part of a gear tooth that projects beyond the pitch circle, or the distance that it projects

Webster's New World Dictionary, Second College Edition.

memoir, *n.* [Fr. *mémoire*, masc., a memorandum, memoir, fem., memory < L. *memoria*, MEMORY] 1. a biography or biographical notice, usually written by a relative or personal friend of the subject 2. [*pl.*] an autobiography, usually a full or highly personal account 3. [*pl.*] a report or record of important events based on the writer's personal observation, special knowledge, etc. 4. a report or record of a scholarly investigation, scientific study, etc. 5. [*pl.*] the record of the proceedings of a learned society

Webster's New World Dictionary, Second College Edition.

Short contents

Short contents	· vii
Contents	· viii
List of Figures	· x
List of Tables	· xi
Introduction to Edition 1	· xiii
Introduction to Edition 2	· xv
Acknowledgements	· xvii
1 Corrections and omissions	· 1
2 Page layout	· 3
3 Document divisions	· 7
4 Front and rear	· 15
5 Boxes and environments	· 31
6 Captions	· 35
7 Miscellaneous	· 39
Command summary	· 47
Bibliography	· 57
Index	· 59

Contents

Short contents	vii
Contents	viii
List of Figures	x
List of Tables	xi
Introduction to Edition 1	xiii
Introduction to Edition 2	xv
Acknowledgements	xvii
1 Corrections and omissions	1
1.1 Pagination	1
1.2 Folding paper	1
1.3 Part headings	1
1.4 Appendices	1
1.5 Writing out verbatim	2
1.6 Subcaptions	2
1.7 New list entries	2
2 Page layout	3
2.1 Text height	3
2.2 Side margins	4
2.3 Overfull lines	5
3 Document divisions	7
3.1 Book headings	7
<i>Book heading layout 8</i>	
3.2 Part headings	9
3.3 Chapter styles	10
<i>Reparticle 10, Southall 10, Chappell 11, Demo2 11</i>	

3.4	Poem Titles	12
	<i>Main Poem Title layout parameters 13, Detailed Poem Title layout parameters 13</i>	
4	Front and rear	15
4.1	Title page	15
4.2	The ToC	15
	<i>Book ToC entry 17</i>	
4.3	The index	18
4.4	Glossaries	19
4.5	Controlling the glossary	20
	<i>Setting up makeindex 20, Raw input data 23, The listing 24, The glossary for this document 25</i>	
4.6	Endnotes	26
	<i>Changing the appearance 29</i>	
5	Boxes and environments	31
5.1	Sidebars	31
5.2	Side notes	31
5.3	Boxed verbatims	31
5.4	Verse	32
5.5	Lists	33
6	Captions	35
6.1	Side captions	35
	<i>Tweaks 36</i>	
6.2	Caption title text	38
7	Miscellaneous	39
7.1	General	39
	<i>Font commands 39, Footnotes 39, Floats 39, Two simple macros 41, Trim marks 41, Vertical centering 41</i>	
7.2	For package writers	42
	<i>Emulating packages 42, Extending a macro 42, Inserting code before and after a file, package or class 43</i>	
7.3	Heading hooks	44
	Command summary	47
	Bibliography	57
	Index	59

List of Figures

4.1 Example endnote listing 28

List of Tables

2.1	Results from sample <code>\textheight</code> adjustments	4
3.1	Division levels	7
4.1	MakeIndex configuration file input parameters	21
4.2	MakeIndex configuration file output parameters	22
6.1	Permitted arguments for some sidecaption related commands	37

Introduction to Edition 1

At the request of users I keep extending the memoir class. The *User Manual* has some 250 or so pages and it is a burden to the author to keep changing it and also for the readers to keep getting new copies, especially when a change can be as small as a sentence or paragraph. Hence I trust that this addendum will suffice until there is enough material to warrant a new edition of the manual.

This addendum applies to the fifth edition of the *User Manual* which describes version 1.2 of the memoir class. The class is currently at version 1.3a with patch version 1.9 or later.

The main extensions and changes to the class and manual include:

- There is more flexibility in typesetting the titles of unnumbered chapters;
- Major extensions for typesetting footnotes;
- Major extensions for indexing, including one column and multiple indexes;
- Major extensions to cropmarks;
- Ability to use `\tableofcontents` and friends multiple times;
- Sheet numbers in addition to page numbers, plus access to the numbers of the last sheet and last page;
- Various methods for formatting numbers;
- Better cooperation with the `chapterbib` and `natbib` packages when they use their `sectionbib` option;
- Sectioning commands can take a second optional argument for header text;
- Section titles, as well as numbers, may be referenced;
- Extra ‘need space’ macros;
- New macros for ‘slashed’ fractions (fractions like $\frac{6}{29}$);
- Extensions to framed boxes;
- Odd page checking extended to apply to non-arabic numbered pages;
- Means of setting ‘optimum’ textwidth;
- More intuitive effects of `\mainmatter` and `\backmatter` when the `article` option is used;
- Control of the spacing of items in the bibliography;
- A ‘fixed’ version of `\marginpar`;
- Extensions for typesetting arrays and tabulars, including continuous tabulars and automatic tabulation;
- As usual, minor glitches have been removed from the code.

Introduction to Edition 2

January 2004 saw a major extension of the memoir class to version 1.6; all the then current patches were folded into the class itself. Edition 6 of the User Manual was released which included the contents of the November 2003 Addendum. Thus, at that time there was no need for either a patch file or the Addendum.

Since then new patches have been made for memoir, fixing problems but adding no new commands. However, in September 2005 a new version, v1.618, of memoir was released which did include some new functions, but not significant enough to warrant a new edition of the User Manual. Edition 2 of the Addendum applies to Edition 6 of the User Manual and covers memoir v1.618 extensions. Further impressions may be released if there are later extensions.

The more major changes or extensions noted in this edition include:

- New part-like pages
- Improved control over higher level ToC entries
- New macros for typesetting the titles of poems
- New macros for making and typesetting glossaries
- Minor extensions for sidebars, boxed verbatims and verses
- The `\em` command is no longer deprecated and minor extension to `\emph`
- Side captions
- New 'book' document division
- Minor extension to numbering verse lines

The following extensions are more for package writers than general authors:

- New macros for specifying emulated packages
- New macro for extending an existing macro
- Hooks into sectioning, captioning, etc., commands

Acknowledgements

The Acknowledgements list in the User Manual is somewhat out of date. I am very grateful to the following who have improved memoir by noting or fixing errors, or providing ideas, suggestions or code: Paul Abrahams, William Adams, Tim Arnold, Donald Arseneau, Stephan von Bechtolsheim, Jens Berger, Karl Berry, Javier Bezos, Sven Bovin, Ignasi Furió Caldenty, Ezequiel Martín Camara, David Carlisle, Gustavo Cevolani, Jean-Côme Charpentier, Michael A. Cleverly, Steven Douglas Cochran, Frederic Connes, Žarko F. Čučej, Michael W. Daniels, Michael Downes, Christopher Dutchyn, Thomas Dye, Victor Eijkhout, Danie Els, Robin Fairbairns, Simon Fear, Kai von Fintel, Ulrike Fischer, Matthew Ford, Musa Furber, Daniel Richard G., Ignacio Fernández Galván, Romano Giannetti, Donald Goodman, Gabriel Guernik, Matthias Haldiman, Kathryn Hargreaves, Sven Hartrumpf, hazydirk, Carsten Heinz, Florence Henry, Peter Heslin, Lars Hoemke, Urs Hofer, Morten Høgholm, Hendrik Holm, Vladimir Ivanovic, Martin Jørgensen, Stefan Kahrs, Marcus Kohm, Jørgen Larsen, Kevin Lin, Matthew Lovell, Daniel Luecking, Lars Madsen, Vittorio De Martino, Frank Mittelbach, Vilar Camara Neto, Rolf Niepraschk, Patrik Nyman, Heiko Oberdiek, Scott Pakin, Adriano Pascoletti, Paul, Steve Peter, Erik Quaeghebeur, Aaron Rendahl, Chris Rowley, Bernd Raichle, René, Robert Schlicht, Dirk Schlimm, Arnaud Schmittbuhl, Rainer Schöpf, Paul Stanley, James Szinger, Jens Taprogge, Reuben Thomas, Bastiaan Niels Veelo, Emanuele Vicentini, Jörg Vogt, Jürgen Vollmer, and David Wilson. If I have inadvertently left anyone out please let me know¹ and I will make the correction.

Along those lines, if you have any questions please direct them to the `comp.text.tex` newsgroup instead of directly to me as you are more likely to get a satisfactory and timely response.

¹I am currently reachable via email at `herries dot press at earthlink dot net`.

One

Corrections and omissions

1.1 Pagination

In section 1.2 of the manual I say that pagination usually starts with the Table of Contents page. This is wrong! *All* pages take part in the pagination, including the Half-title and Title pages. It is the folios (i.e., the printed page numbers) that usually start on the Table of Contents page.

1.2 Folding paper

In section 2.1 of the manual I say that you cannot fold a piece of paper more than six times. This is incorrect. A high school student, Britney Gallivan, has managed to fold a sheet of paper in half twelve times (see, for example, <http://mathworld.wolfram.com/Folding.html>).

1.3 Part headings

The manual claims that `\midpartskip` is a length; it is not, it is a macro and cannot be altered by a `\setlength`.

```
\beforepartskip \midpartskip \afterpartskip
```

The definitions of these three macros, which control the vertical spaces in a `\part` heading, are

```
\newcommand*{\beforepartskip}{\null\vfil}  
\newcommand{\midpartskip}{\par\vskip 20pt}  
\newcommand*{\afterpartskip}{\vfil\newpage}
```

1.4 Appendices

```
\ifanappendix
```

When memoir is processing an appendix `\ifanappendix` is set TRUE. otherwise it is FALSE.

1.5 Writing out verbatim

The class provides two methods for writing out verbatim material to a file. Unfortunately only one was described in the manual. Here are the two methods.

```
\begin{verbatimoutput}{\file} material \end{verbatimoutput}
\begin{writeverbatim}{\stream} material \end{writeverbatim}
```

The material in these environments is written out verbatim. Material from the `verbatimoutput` environment is written to the `\file` file. On the other hand, material from the `writeverbatim` environment is written to whatever file is currently associated with the `\stream` output stream.

1.6 Subcaptions

```
\subcaptionref{\key}
\subcaptionref*{\key}
\subcaptionlabelfont{\fontspec}
```

To refer to a subcaption you can use either `\subcaptionref` or the starred version `\subcaptionref*`. In the latter case the normal font is used and in the former the reference is typeset using the `\subcaptionlabelfont`.

1.7 New list entries

In section 10.4 *New list of... and entries* of the manual I gave an example of creating a new list entry for a 'subanswer' as:

```
\newlistentry[answer]{subanswer}{1}
```

Unfortunately `\newlistentry` takes one optional and three required arguments.

```
\newlistentry[\within]{\ctr}{\ext}{\level-1}
```

The example should have read:

```
\newlistentry[answer]{subanswer}{ans}{1}
```

Two

Page layout

2.1 Text height

After specifying the page layout you want, the `\checkandfixthelayout` command has to be called to implement your specification.

```
\checkandfixthelayout [algorithm]  
\checkthelayout [algorithm]
```

The `\checkandfixthelayout [algorithm]` macro uses `\checkthelayout [algorithm]` to check the page layout specification you have given, and then implements it. The `\checkthelayout [algorithm]` macro checks the layout, modifying the `\textheight` according to *algorithm*, but does not implement the layout.

Following a suggestion by Morten Høgholm, if you use the `calc` package then you can now use constructs like the following in a page layout specification:

```
\setlength{\textheight}{41\baselineskip + \topskip}  
\settypeblocksize{41\baselineskip + \topskip}{33pc}{*}
```

When using `\flushbottom` LaTeX expects that the `\textheight` is such that an integral number of text lines in the body font will fit exactly into the height. If not, then it issues ‘underfull vbox’ messages. More precisely, if the *b* is the `\baselineskip` and *t* is the `\topskip`, *N* is an integer (the number of lines in the textblock), and *T* is the `\textheight` then to avoid underfull vboxes the following relationship must hold

$$T = (N - 1)b + t \tag{2.1}$$

By default `\checkthelayout` ensures that the final `\textheight` meets this criterion. The optional *algorithm* argument lets you control just how it does this. In the following *H* is your requested value for the `\textheight` and the other symbols are as before, with *T* as the adjusted value, and using integer arithmetic.¹ The permissible values for *algorithm* are:

fixed The `\textheight` is not altered.

$$T = H \tag{2.2}$$

If you use this option you may find that underfull vboxes are reported for `\flushbottom` pages.

¹In this context ‘integer arithmetic’ means that the result of a division will be rounded down. For example 99/10 in ‘real arithmetic’ results in 9.9, whereas with integer arithmetic the result is 9, not 10.

Table 2.1: Results from sample `\textheight` adjustments

Requested height	Algorithm			
	fixed	classic	lines	nearest
	adjusted height in pts, (lines)			
10.0\baselineskip	120.0pt, (10)	130pt, (11)	118pt, (10)	118pt, (10)
10.2\baselineskip	122.4pt, (10)	130pt, (11)	118pt, (10)	118pt, (10)
10.4\baselineskip	124.8pt, (10)	130pt, (11)	118pt, (10)	130pt, (11)
10.6\baselineskip	127.2pt, (10)	130pt, (11)	118pt, (10)	130pt, (11)
10.8\baselineskip	129.6pt, (10)	130pt, (11)	118pt, (10)	130pt, (11)
11.0\baselineskip	132.0pt, (11)	142pt, (12)	130pt, (11)	130pt, (11)

`classic` This is the default and is the one used by the standard classes.

$$T = b\lfloor H/b \rfloor + t \quad (2.3)$$

The relationship (2.1) is maintained.

`lines` This is similar to `classic`, but results in a smaller final value.

$$T = b\lfloor (H - b)/b \rfloor + t \quad (2.4)$$

The relationship (2.1) is maintained.

`nearest` The calculated value is the nearest to the given value while still maintaining the relationship (2.1).

$$T = b\lfloor (H - t + b/2)/b \rfloor + t \quad (2.5)$$

Table 2.1 shows the results from the various `\textheight` adjustment calculations² where the `\baselineskip` is 12pt and the `\topskip` is 10pt, which are the normal values for a Computer Modern 10pt font. In all cases the `fixed` algorithm resulted in underfull vboxes. If you know the number of lines that you want, say 42, then requesting

```
%% setting equivalent to \setlength{\textheight}{42\baselineskip}
\checkandfixthelayout[lines]
```

will result in the most appropriate `\textheight`.

2.2 Side margins

In twoside printing the spine margin is normally the same on both recto and verso pages and, unless the spine and foreedge margins are the same, the textblock is shifted side to side when printing the recto and verso pages. Additionally you can have different headers and footers for the recto and verso pages. In oneside printing the textblock is not moved and the headers and footers are the same for both odd and even pages.

Some documents are designed to have, say, a very wide righthand margin in which to put illustrations; this leads to needing the spine margin on verso pages to be much larger than the spine margin on recto pages. This can be done with the `oneside` option. However,

²For comparison the optimum heights from equation 2.1 for 10, 11 and 12 lines are respectively 118pt, 130pt and 142pt.

different headers and footers are required for the recto and verso pages, which can only be done with the twoside option. The way to get the desired effects is like this (twoside is the default):

```
\documentclass{memoir}
%% set up the recto page layout
\checkandfixthelayout% or perhaps \checkandfixthelayout[lines]
\setlength{\evensidemargin}{\oddsidemargin}% after \checkandfix...
...
```

2.3 Overfull lines

TeX tries very hard to keep text lines justified while keeping the interword spacing as constant as possible, but sometimes fails complaining about an overfull hbox.

```
\fussy \sloppy
\begin{sloppypar} ... \end{sloppypar}
\midsloppy
\begin{midsloppypar} ... \end{midsloppypar}
```

The default mode for LaTeX typesetting is `\fussy` where the (variation of) interword spacing in justified text is kept to a minimum. Following the `\sloppy` declaration there may be a much looser setting of justified text. The `sloppypar` environment is equivalent to:

```
{\par \sloppy ... \par}
```

Additionally the class provides the `\midsloppy` declaration (and the `midsloppypar` environment) which allows a setting somewhere between `\fussy` and `\sloppy`. Using `\midsloppy` you will get fewer overfull lines compared with `\fussy` and fewer obvious large interword spaces than with `\sloppy`. I have used `\midsloppy` for this manual; it hasn't prevented overfull lines or noticeably different interword spaces, but has markedly reduced them compared with `\fussy` and `\sloppy` respectively.

Three

Document divisions

3.1 Book headings

Frederic Connes has told me that in French typography there is often a document division above the `\part` level. This is also sometimes the case with English typography — the *Chicago Manual of Style* [CMS93, p21] shows an example. Based on code that he kindly sent me, a new document division, Book, is provided that is above Part in the division hierarchy. There are now six levels of document division and these are given in Table 3.1.

```
\book [<toc-title>] {<title>}  
\book* {<title>}
```

The `\book` command is like the `\part` command and produces a similar looking title page in the document.

```
\bookmark {<title>}
```

The `\book` code includes `\bookmark {<title>}` for storing the title of the book if it is going to be used, for example, in page headers. Its default definition is simply:
`\newcommand*{\bookmark}[1]{}`

Table 3.1: Division levels

Division	Level
<code>\book</code>	-2
<code>\part</code>	-1
<code>\chapter</code>	0
<code>\section</code>	1
<code>\subsection</code>	2
<code>\subsubsection</code>	3
<code>\paragraph</code>	4
<code>\subparagraph</code>	5

3. DOCUMENT DIVISIONS

```
\bookblankpage \nobookblankpage
```

A page with a `\book` title is normally followed by a blank page. If you do not want this then use the `\nobookblankpage` declaration. The `\bookblankpage` declaration reverts the behaviour to the normal blank page.

The pagestyle *book* is applied to a book division title page and the pagestyle *afterbook*, which defaults to *empty*, is applied to the blank page, if any, following a `\book` page.

BOOK HEADING LAYOUT

The layout for a `\book` page is very similar to that for a `\part` page, although there are separate controls for each.

The code for typesetting the page is in essence like this.

```
\newcommand{\book}[1]{%
  \cleardoublepage
  \thispagestyle{book}%
  \beforebookskip
  \printbookname\booknumnum{\booknumfont\thebook}%
  \midbookskip
  \printparttitle{#1}%
  \bookmark{#1}%
  \afterpartskip}
```

```
\beforebookskip \midbookskip \afterbookskip
```

These commands are called before any part of the title is set, between typesetting the number (if any) and the title, and after the title has been set.

```
\bookname
\printbookname \booknamefont \booknumnum
```

`\bookname`, defined as `\newcommand*{\bookname}{Book}` is the name for a book. `\printbookname`, prints the `\bookname` using the `\booknamefont`, which is defined as: `\newcommand*{\booknamefont}{\normalfont\huge\bfseries}`. This is followed by `\booknumnum`, which is defined to be `\space`.

```
\booknumfont \printbooknum
```

`\printbooknum` prints the book number (i.e., `\thebook`) using the `\booknumfont`, whose default definition is the same as `\booknamefont`.

```
\booktitlefont \printbooktitle{<title>}
```

`\printbooktitle{<title>}` prints the book title using the `\booktitlefont`, which is defined as: `\newcommand*{\booktitlefont}{\normalfont\Huge\bfseries}`

3.2 Part headings

```
\partmark{<title>}
```

The `\part` code now includes `\partmark{<title>}` for storing the title of the part if it is going to be used, for example, in page headers. Its default definition is simply:

```
\newcommand*{\partmark}[1]{}
```

```
\partblankpage \nopartblankpage
```

A page with a `\part` title is normally followed by a blank page. If you do not want this then use the `\nopartblankpage` declaration. The `\partblankpage` declaration reverts the behaviour to the normal blank page.

A new pagestyle *afterpart*, which defaults to *empty*, is applied to the blank page, if any, following a `\part` page.

```
\newleadpage[<page-style>]{<cmdname>}{<title>}
\newleadpage*[<page-style>]{<cmdname>}{<title>}
\renewleadpage[<page-style>]{<cmdname>}{<title>}
\renewleadpage*[<page-style>]{<cmdname>}{<title>}
```

`\newleadpage` and associates are variants of the `\newcommand` and companions; the original suggestion and code was from Danie Els. The `\newleadpage` macro defines a macro `\cmdname` that when called typeset an Appendixpage-like page with a title `<title>` using the `<page-style>` for the page. The default `<page-style>` is empty. The macro `\renewleadpage` redefines an existing leadpage command.

For example,

```
\newleadpage{plates}{Picture Gallery}
```

creates the new command `\plates` which when called generates an unnumbered part-like page with the title **Picture Gallery**.

```
\leadpagetoclevel
```

When `\(re)newleadpage` is used the resulting command adds `<title>` to the ToC as though it was an unnumbered chapter. More precisely it will be added as though it were an unnumbered `\leadpagetoclevel` entry, whose default definition is:

```
\newcommand*{\leadpagetoclevel}{chapter}
```

When the starred versions `\(re)newleadpage*` are used the resulting command will not add `<title>` to the ToC.

Internally the resulting commands use `\partmark` for storing the `<title>` for possible later use by you.

3.3 Chapter styles

Lars Madsen has collected a wide variety of chapter styles. These, and the code to produce them, are available at <http://www.imf.au.dk/system/latex/artikler/MemoirChapStyles>.

REPARTICLE

When the article option is used the default chapter and section styles are close to, but not identical, the corresponding division headings in the article class.

The *reparticle* chapterstyle makes a `\chapter` replicate the appearance of a `\section` in the article class.

```
\reparticle
```

The `\reparticle` command makes chapter and lower level division heads replicate those of the article class. You can use it, for example, like:

```
\ifartopt
  \reparticle
\fi
```

to call it if the article option has been requested.

SOUTHALL

On 2006/01/08 Thomas Dye posted his *southall* chapterstyle on `comp.text.tex` and kindly gave me permission to include it here. It is based on the headings in a Cambridge Press book¹ by Aidan Southall. It produces a simple numbered heading with the title set as a block paragraph, and with a horizontal rule underneath. His original code called for lining figures for the number but I have commented out that bit.

```
% Thomas Dye's southall chapter style
\newlength{\headindent}
\newlength{\rightblock}
\makechapterstyle{southall}{%
  \setlength{\headindent}{36pt}
  \setlength{\rightblock}{\textwidth}
  \addtolength{\rightblock}{-\headindent}
  \setlength{\beforechapskip}{2\baselineskip}
  \setlength{\afterchapskip}{5\baselineskip}
  \setlength{\midchapskip}{0pt}
  \renewcommand{\chaptitelfont}{\huge\rmfamily\raggedright}
  \renewcommand{\chapnumfont}{\chaptitelfont}
  \renewcommand{\printchaptername}{}
  \renewcommand{\chapternamenum}{}
  \renewcommand{\afterchapternum}{}
  \renewcommand{\printchapternum}{%
    \begin{minipage}[t][\baselineskip][b]{\headindent}
```

¹Which I haven't seen

```

    {\vspace{0pt}\chapnumfont%%\figureversion{lining}
      \thechapter}
  \end{minipage}}
\renewcommand{\printchaptertitle}[1]{%
  \hfill\begin{minipage}[t]{\rightblock}
    {\vspace{0pt}\chaptitelfont ##1\par}\end{minipage}}
\renewcommand{\afterchaptertitle}{%
  \par\vspace{\baselineskip}%
  \hrulefill \par\nobreak\noindent \vskip\afterchapskip}}

```

CHAPPELL

Another style that includes rules is one I have called *chappell*, which is based on the chapter heads in [CB99]. The style can easily form the basis for general heads in non-technical books.

```

\makechapterstyle{chappell}{%
  \setlength{\beforechapskip}{0pt}
  \renewcommand*{\chapnamefont}{\large\centering}
  \renewcommand*{\chapnumfont}{\large}
  \renewcommand*{\printchapternonum}{%
    \vphantom{\printchaptername}%
    \vphantom{\chapnumfont 1}%
    \afterchapternum
    \vskip -\onelineskip}
  \renewcommand*{\chaptitelfont}{\Large\itshape}
  \renewcommand*{\printchaptertitle}[1]{%
    \hrule\vskip\onelineskip \centering\chaptitelfont ##1}}

```

This style centers the chapter number, draws a rule across the page under it, and below that comes the title, again centered. All the fiddling in the `\printchapternonum` macro is to try and ensure that the rule above the title is at the same height whether or not the chapter is numbered (the ToC being an example of an unnumbered heading).

DEMO2

This is the original code for the *demo* chapterstyle.

```

\makechapterstyle{demo}{%
  \renewcommand*{\printchaptername}{\centering}
  \renewcommand*{\printchapternum}{\chapnumfont \numtoName{\c@chapter}}
  \renewcommand*{\chaptitelfont}{\normalfont\Huge\sffamily}
  \renewcommand*{\printchaptertitle}[1]{%
    \hrule\vskip\onelineskip \raggedleft \chaptitelfont ##1}
  \renewcommand*{\afterchaptertitle}{%
    {\vskip\onelineskip \hrule\vskip \afterchapskip}
  }% end demo

```

This has one serious failing and what I am now thinking is a poor design decision. The failing is that if you have any appendices that use the *demo* chapterstyle then they are numbered instead of being lettered. The poor design is that the position of the title with

respect to the top of the page is not the same for numbered and unnumbered chapters. The *demo2* chapterstyle below fixes both of these.

```
\makechapterstyle{demo2}{%
  \renewcommand*{\printchaptername}{\centering}
  \renewcommand*{\printchapternum}{\chapnumfont
    \ifanappendix \thechapter \else \numtoName{\c@chapter}\fi}
  \renewcommand*{\chaptitelfont}{\normalfont\Huge\sffamily}
  \renewcommand*{\printchaptertitle}[1]{%
    \hrule\vskip\onelineskip \raggedleft \chaptitelfont ##1}
  \renewcommand*{\afterchaptertitle}{%
    \vskip\onelineskip \hrule\vskip \afterchapskip}
  \setlength{\beforechapskip}{3\baselineskip}
  \renewcommand*{\printchapternonum}{%
    \vphantom{\chapnumfont One}
    \afterchapternum%
    \vskip\topskip}
  \setlength{\beforechapskip}{2\onelineskip}
}% end{demo2}
```

You may find it instructive to compare the code for the *demo* and *demo2* chapterstyles.

Starting in November 2002 this document used the *demo* chapterstyle. From July 2006 onwards it uses the *demo2* chapterstyle.

3.4 Poem Titles

The new command `\PoemTitle` provides more flexibility in typesetting titles of poems than the original `poemtitle` does.

```
\PoemTitle[<fortoc>][<forhead>]{<title>}
\NumberPoemTitle
\PlainPoemTitle
\thepoem
\poemtitlemark{<forhead>}
\poemtitlepstyle
```

The `\PoemTitle` command takes the same arguments as the `\chapter` command; it typesets the title for a poem and adds it to the ToC. Following the declaration `\NumberPoemTitle` the title is numbered but there is no numbering after the `\PlainPoemTitle` declaration.

The macro `\poemtitlemark` is called with the argument *<forhead>* so that it may be used to set marks for use in a page header via the normal mark process. The `\poemtitlepstyle` macro, which by default does nothing, is provided as a hook so that, for example, it can be redefined to specify a particular pagestyle that should be used. For example:

```
\renewcommand*{\poemtitlemark}[1]{\markboth{#1}{#1}}
\renewcommand*{\poemtitlepstyle}{%
  \pagestyle{headings}%
  \thispagestyle{empty}}
```

```
\PoemTitle*[{forhead}] {title}
\poemtitlestarmark{forhead}
\poemtitlestarpstyle
```

The `\PoemTitle*` command produces an unnumbered title that is not added to the ToC. Apart from that it operates in the same manner as the unstarred version. The `\poemtitlestarmark` and `\poemtitlestarpstyle` can be redefined to set marks and pagestyles.

MAIN POEM TITLE LAYOUT PARAMETERS

```
\PoemTitleheadstart
\printPoemTitlenonum
\printPoemTitlenum
\afterPoemTitlenum
\printPoemTitletitle{title}
\afterPoemTitle
```

The essential of the code used to typeset a numbered *title* from a `PoemTitle` is:

```
\PoemTitleheadstart
\printPoemTitlenum
\afterPoemTitlenum
\printPoemTitletitle{title}
\afterPoemTitle
```

If the title is unnumbered then `\printPoemTitlenonum` is used instead of the `\printPoemTitlenum` and `\afterPoemTitlenum` pair of macros.

The various elements of this can be modified to change the layout. By default the number is centered above the title, which is also typeset centered, and all in a `\large` font.

The elements are detailed in the next section.

DETAILED POEM TITLE LAYOUT PARAMETERS

```
\beforePoemTitleskip
\PoemTitlenumfont
\midPoemTitleskip
\PoemTitlefont
\afterPoemTitleskip
```

As defined, `\PoemTitleheadstart` inserts vertical space before a poem title. The default definition is:

```
\def\PoemTitleheadstart{\vspace{\beforePoemTitleskip}}
\newlength{\beforePoemTitleskip}
\setlength{\beforePoemTitleskip}{1\onelineskip}
```

`\printPoemTitlenum` typesets the number for a poem title. The default definition, below, prints the number centered and in a large font.

```
\def\printPoemTitlenum{\PoemTitlenumfont \thepoem}
\newcommand*{\PoemTitlenumfont}{\normalfont\large\centering}
```

3. DOCUMENT DIVISIONS

The definition of `\printPoemTitlenonum`, which is used when there is no number, is simply

```
\def\printPoemTitlenonum{}
```

`\afterPoemTitlenum` is called between setting the number and the title. It ends a paragraph (thus making sure any previous `\centering` is used) and then may add some vertical space. The default definition is:

```
\def\afterPoemTitlenum{\par\nobreak\vskip \midPoemTitleskip}
```

```
\newlength{\midPoemTitleskip}
```

```
\setlength{\midPoemTitleskip}{0pt}
```

The default definition of `\printPoemTitletitle` is below. It typesets the title centered and in a large font.

```
\def\printPoemTitletitle#1{\PoemTitlefont #1}
```

```
\newcommand*{\PoemTitlefont}{\normalfont\large\centering}
```

The macro `\afterPoemTitle` finishes off the title typesetting. The default definition is:

```
\def\afterPoemTitle{\par\nobreak\vskip \afterPoemTitleskip}
```

```
\newlength{\afterPoemTitleskip}
```

```
\setlength{\afterPoemTitleskip}{1\onelineskip}
```


Four

Front and rear

4.1 Title page

In the manual I recommended that you did not use `\maketitle` and the associated titling commands for title pages — `\maketitle` and friends do a reasonable job for titles of articles or a quick report but are too inflexible for more prestigious work. In other words, design your own title page and use the general LaTeX commands for vertical and horizontal spacing to lay it out, or perhaps use a `picture` environment.

Vincent Zoonekynd has a collection of a range of example title pages. These, and the code used to produce them, are available at http://zoonek.free.fr/LaTeX/LaTeX_samples_title/0.html.

4.2 The ToC

There have been requests for a ToC layout along the following lines:

```
PART I Title of part ....
Chapter 1. Title of chapter ...
  1.1 A section ...
...
Appendix A. Title of appendix
  A.1 Another section
...
```

Danie Els suggested that this could be accomplished with a few new commands.¹

<pre>\cftpartname \cftchaptername \cftappendixname</pre>
--

These commands form part of the code for `\part` and `\chapter` entries in the ToC, coming before the relevant number. Their default definitions are empty. To get the above ToC you can do the following:

¹Plus the recoding of some internal macros.

```
\renewcommand*\cftpartname{PART~}  
\renewcommand*\cftchaptername{\chaptername~}  
\renewcommand*\cftappendixname{\appendixname~}  
\renewcommand*\cftchapteraftersnum{.} % dot after number  
\setlength{\cftchapternumwidth}{2em} % allow more space
```

Lars Madsen further suggested that the scheme be extended to include names for figures and tables as several academic institutions like each figure or table entry in a LoF or LoT to be redundantly labeled as a figure or table.

```
\cftsectionname \cftsubsectionname \cftsubsubsectionname  
\cftparagraphname \cftsubparagraphname  
\cftfigurename \cfttablename
```

They are created by the `\newlistentry` macro and by default the `\cft...name` macros have no content.

However, for example, if they required you to put ‘Fig’ before the number of each entry in the LoF, then

```
\renewcommand*\cftfigurename{Fig\enspace}
```

could be a suitable response in your preamble.

If you happen to use the `\newsfloat{<floatenv>}` command, for instance

```
\newsfloat{figure}
```

to enable subfigures, then there will also be a `\cftsubfloatfigurename` created for you to use if the powers that be really push you.

Another query has been how to get the titles in the ToC to be set raggedright instead of the usual flushright. Assuming that there are more than 100 pages in the document:

```
\setrmarg{3.55em plus 1fil}
```

where the last four characters before the closing brace are: digit 1, lowercase F, lowercase I, and lowercase L.

You may have noticed that there are two ToCs for this document — a short one and a long one. This is how they were produced.

The code below was used to produce the short ToC.

```
%% Short contents and Different ToC style  
\renewcommand{\contentsname}{Short contents}  
\let\oldchangetocdepth\changetocdepth  
\let\oldcftchapterfillnum\cftchapterfillnum  
\renewcommand{\changetocdepth}[1]{  
\setcounter{tocdepth}{0} % chapters  
\renewcommand{\cftchapterfont}{\hfill\sffamily}  
\renewcommand{\cftchapterleader}{\textperiodcentered\space}  
\renewcommand{\cftchapterafterpnum}{\cftparfillskip}  
\setpnumwidth{0em}  
\setrmarg{0.3\textwidth}  
\tableofcontents  
\clearpage
```

The above sets the page numbers as though they were left adjusted in the page number box but with LaTeX reporting overfull hboxes. Changing the width of the box for the page number

```
\setpnumwidth{1.5em}
```

results in the page numbers being right adjusted in the 1.5em wide box, which I think does not look as good. To get the left adjusted effect with no complaints try:

```
\renewcommand*{\cftchapterfillnum}[1]{%
  {\cftchapterleader}\nobreak
  \hbox to 1.5em{\cftchapterpagefont #1\hfil}\cftchapterafterpnum\par}
```

The next piece of code will typeset a group of subsections in the ToC as a paragraph.

```
%% have subsections as a paragraph in the ToC
\makeatletter
\let\oldnumberline\numberline
\renewcommand{\cftsubsectionfont}{\itshape}
\renewcommand{\cftsubsectionpagefont}{\itshape}
\renewcommand{\l@section}[2]{\relax
  \def\numberline##1{\textit{##1}}%
  \leftskip=\cftsubsectionindent
  \rightskip=\@toCRMarg
  % \advance\rightskip \z@ plus \hsize % uncomment this for raggedright
  % \advance\rightskip \z@ plus 2em % uncomment this for semi-raggedright
  \parfillskip=\fill
  \ifhmode ,\ \else\noindent\fi
  \ignorespaces {\cftsubsectionfont #1}~{\cftsubsectionpagefont #2}%
  \let\numberline\oldnumberline\ignorespaces
}
\AtEndDocument{\addtocontents{toc}{\par}}
```

```
\makeatother
```

The final piece of code below sets up the rest of the second ToC.

```
%% Default contents
\renewcommand{\contentsname}{Contents}
\let\changetocdepth\oldchangetocdepth
\let\cftchapterfillnum\oldcftchapterfillnum
\renewcommand{\cftchapterfont}{\normalfont\sffamily}
\renewcommand{\cftchapterleader}{\sffamily\cftdotfill{\cftchapterdotsep}}
\renewcommand{\cftchapterafterpnum}{}
\makeatletter
\renewcommand{\cftchapterbreak}{\par\addpenalty{-\@highpenalty}}
\makeatother
\setpnumwidth{2.55em}
\setrmarg{3.55em}
\setcounter{tocdepth}{2}
\tableofcontents
```

BOOK TOC ENTRY

There are the usual kinds of commands for configuring the appearance of \book entries in the ToC.

```
\cftbookname
```

This is called before the book number is set in the ToC. The default definition is empty.

```
\cftbeforebookskip \cftbookindent \cftbooknumwidth
```

These lengths control the vertical space before a book entry, the indentation of the entry from the left margin, and the space for typesetting the number.

```
\cftbookfont \cftbookpagefont
```

These specify the fonts to be used for typesetting the number and title of a book entry in the ToC, and the page number.

```
\cftbookpresnum \cftbookaftersnum \cftbookaftersnumb
```

The book number is typeset in a box. `\cftbookpresnum` and `\cftbookaftersnum` are called within the box before and after the number. `\cftbookaftersnumb` is called after the box has been typeset. By default these commands do nothing.

```
\cftbookleader \cftbookdotsep
```

`\cftbookleader` typesets the leader between a book title and the page number in the ToC. By default this is a dotted leader with `\cftbookdotsep` between the dots. By default this is set for a book entry to produce no dots.

4.3 The index

The internal indexing code has been changed slightly to make things more efficient. Now any changes to indexed items will be immediately reflected in the `idx` file. Previously it took two LaTeX runs to achieve this.

The `MakIndex` indexing program can handle a memoir hyperindex, but the `xindy` program cannot, as indicated by the following extract from an email sent to me by Frederic Connes:

```
... You use "|hyperspindexpage(\thepage)", which xindy doesn't
recognize as a valid markup-locref. And I don't see how to add
it, because xindy only accepts one argument in markup-locref
(if the number is not a page number, it will still point to a
page with that number), so replacing it with "|hyperpage" won't
work.
```

```
...It would be nice to be able to deactivate the
"|hyperspindexpage" part, as it causes xindy to crash.
```

I don't use `xindy`, but I have provided something that addresses the problem.

```
\memhyperindexfalse
```

Putting `\memhyperindexfalse` into the preamble will prevent any hyperindexing no matter what you have instructed the `hyperref` package to do.

Since then Frederic Connes has provided some code for when `xindy` will be used.

```
\xindyindex
```

Put the `\xindyindex` declaration in the pramble when you will be using `xindy` to process the raw index; there is no need to use `\memhyperindexfalse` as well as this.

4.4 Glossaries

Unlike for indexes, LaTeX provides less than minimal support for glossaries. It provides a `\makeglossary` command for initiating a glossary and a `\glossary` command which puts its argument, plus the page number, into a `glo` file, and that's it. `memoir`, combined with the `MakelIndex` program [CH88], enables you to generate and print a glossary in your document. The commands for creating a glossary are similar to those for indexes.

```
\makeglossary[file]
```

You have to put `\makeglossary` in your preamble if you want a glossary. This opens a file called by default `\jobname.glo`. If you use the optional `file` argument the file `file.glo` will be opened. A glossary `glo` file is analagous to an index `idx` file.

```
\printglossary[file]
```

To print a glossary call `\printglossary` which will print the glossary from file `\jobname.gls`, or from `file.gls` if the optional argument is used. A glossary `gls` file is analagous to an index `ind` file.

```
\glossary[file](key){term}{desc}
```

Use the `\glossary` command to add a `term` and its description, `desc`, to a glossary file. By default this will be `\jobname.glo` but if the optional `file` argument is given then the information will be written to `file.glo`. The `(key)` argument is optional. If present then `key` will be added to the file to act as a sort key for the `term`, otherwise `term` will be used as the sort key.

By using the optional `file` arguments you can have several glossaries, subject to TeX's limitations on the number of files that can be open at any one time.

A simple glossary entry might be:

```
\glossary{glossary}{A list of terms and their descriptions.}
```

The glossary facilities are designed so that the `MakelIndex` program can be used to convert the raw glossary data in a `glo` file into the printable glossary in a `gls` file.

```
\begin{theglossary} entry list \end{theglossary}
```

Glossary entries are typeset in a `theglossary` environment. It is assumed that a `gls` file will contain a complete `theglossary` environment, from `\begin{theglossary}` all the way through to `\end{theglossary}`.

`\glossitem{<term>}{<desc>}{<ref>}{<num>}`

A `\glossitem` is a glossary entry within a `theglossary` environment for a `<term>` with `<description>`. The `<num>` argument is the page or section where the corresponding `\glossary` was issued. The `<ref>` argument, if not empty, might be the section or page number corresponding to the `<num>` page or section number. The default definition is

```
\newcommand{\glossitem}[4]{#1 #2 #3 #4}
```

which is not very exciting. You may well prefer to use your own definition.

4.5 Controlling the glossary

SETTING UP MAKEINDEX

If you just run `MakelIndex` on a `glo` file you will get lots of errors; `MakelIndex` has to be configured to read a `glo` file and generate a useful `gls` file as by default it expects to read an index `idx` file and produce an index `ind` file. A configuration file like an index `ist` file will be needed. There is no recommended extension for such a file but I have come to favour `gst`. The command line for `MakelIndex` to create a sorted glossary from the raw data in a `glo` file, say `fred.glo`, using a configuration file called, say `basic.gst`, is

```
makeindex -s basic.gst -o fred.gls fred.glo
```

For other jobs just change the file names appropriately.

So, what is in a `gst` file? The potential contents are given by Chen & Harrison [CH88] and also in the *Companion* [MG⁺04, Chap. 11]. At a minimum you need this:

```
%% basic.gst basic makindex glossary style file
%% Output style parameters
preamble "\\begin{theglossary}"
postamble "\\n\\end{theglossary}\\n"
item_0    "\\n\\glossitem"
delim_0   "{\\memglonum{"
encap_suffix "}}}"
%% Input style parameters
keyword "\\glossaryentry"
```

The keyword line says that each entry in an input (`glo`) file will be of the form:

```
\glossaryentry{entry text}{number}
```

and by a miracle of coding, this is what `memoir` will put in a `glo` file for each `\glossary` command.

The preamble and postamble lines tell the program to start and end its output file with `\begin{theglossary}` and `\end{theglossary}`, respectively. The `item_0` tells the program to start each output entry with `\glossitem`. The `delim_0` says that `{\memglonum{` should be put between the end of the entry text and the (page) number. Finally `encap_suffix` requests `}}}` to be put after any ‘encapsulated’ (page) number.

Table 4.1: MakeIndex configuration file input parameters

Keyword	Default	Description
keyword (<i>s</i>)	"\\indexentry"	The argument to this command is a MakeIndex index entry
arg_open (<i>c</i>)	'{'	Argument start delimiter
arg_close (<i>c</i>)	'}'	Argument end delimiter
range_open (<i>c</i>)	'('	Start of an explicit page range
range_close (<i>c</i>)	')'	End of an explicit page range
level (<i>c</i>)	'!'	Character denoting a new subitem level
actual (<i>c</i>)	'@'	Character denoting that the following text is to appear in the actual index file
encap (<i>c</i>)	' '	Character denoting that the rest of the argument is to be used as an encapsulating command for the page number
quote (<i>c</i>)	'"'	Character that escapes the following character
escape (<i>c</i>)	'\\'	Symbol with no special meaning unless followed by the quote character, when both characters will be printed. The quote and escape characters must be different.
page_compositor (<i>s</i>)	"-"	Composite number separator

(*s*) of type string, (*c*) of type character

A complete listing of the possible entries in a configuration file, also called a style file, for MakeIndex is in Table 4.1 and 4.2 with the exception of the output file page number setting keywords.

The `gst` file I have used for this document has a few more items than the basic one.

```
%% memman.gst makindex glossary style file for memman and friends
%% Output style parameters
preamble "\\begin{theglossary}"
postamble "\\end{theglossary}\\n"
group_skip "\\n\\glossaryspace\\n"
item_0 "\\n\\glossitem"
delim_0 "{\\memglonum{"
encap_suffix "}}}"
indent_space "\\t"
indent_length 2
%% Input style parameters
keyword "\\glossaryentry"
actual '?'
page_compositor "-"
```

The `group_skip` line asks that `\\glossaryspace` be put between the last entry for one letter and the first for the next letter. The `indent_space` and `indent_length` give a smaller indent for continuation lines in the output than the default.

Table 4.2: MakeIndex configuration file output parameters

Keyword	Default	Description
<code>preamble (s)</code>	<code>"\begin{theindex}\n"</code>	Text for the start of the output file
<code>postamble (s)</code>	<code>"\n\n\end{theindex}\n"</code>	Text at the end of the output file
<code>group_skip (s)</code>	<code>"\n\n\indexspace\n"</code>	Vertical space before a new letter group
<code>heading_prefix (s)</code>	<code>"</code>	Prefix for heading for a new letter group
<code>heading_suffix (s)</code>	<code>"</code>	Suffix for heading for a new letter group
<code>headings_flag (n)</code>	<code>0</code>	A value = 0 inserts nothing between letter groups. A value > 0 includes an uppercase instance of the new symbol, while a value < 0 includes a lowercase instance, all within <code>heading_prefix</code> and <code>heading_suffix</code>
<code>item_0 (s)</code>	<code>"\n\item "</code>	Command inserted in front of a level 0 entry
<code>item_1 (s)</code>	<code>"\n \subitem "</code>	As above for a level 1 entry
<code>item_2 (s)</code>	<code>"\n \subsubitem "</code>	As above for a level 2 entry
<code>item_01 (s)</code>	<code>"\n \subitem "</code>	Command inserted in front of a level 1 entry starting at level 0
<code>item_12 (s)</code>	<code>"\n \subsubitem "</code>	Command inserted in front of a level 2 entry starting at level 1
<code>item_x1 (s)</code>	<code>"\n \subitem "</code>	Command inserted in front of a level 1 entry when the parent level has no page numbers
<code>item_x2 (s)</code>	<code>"\n \subitem "</code>	As above for a level 2 entry
<code>delim_0 (s)</code>	<code>", "</code>	Delimiter between level 0 entry and first page number
<code>delim_1 (s)</code>	<code>", "</code>	As above for level 1 entry
<code>delim_2 (s)</code>	<code>", "</code>	As above for level 2 entry
<code>delim_n (s)</code>	<code>", "</code>	Delimiter between page numbers
<code>delim_r (s)</code>	<code>"-"</code>	Designator for a page range
<code>encap_prefix (s)</code>	<code>"\\"</code>	Prefix in front of a page encapsulator
<code>encap_infix (s)</code>	<code>"{"</code>	Infix for a page encapsulator
<code>encap_suffix (s)</code>	<code>"}"</code>	Suffix for a page encapsulator
<code>page_precedence (s)</code>	<code>"rnaRA"</code>	Page number precedence for sorting. <code>r</code> and <code>R</code> are lower- and uppercase roman; <code>a</code> and <code>A</code> are lower- and uppercase alphabetic; <code>n</code> is numeric
<code>line_max (n)</code>	<code>"72"</code>	Maximum length of an output line
<code>indent_space (s)</code>	<code>"\t\t"</code>	Indentation commands for wrapped lines
<code>indent_length (n)</code>	<code>"16"</code>	Indentation length for wrapped lines

(*s*) of type string, (*n*) of type number, `"\n"` and `"\t"` are newline and tab.

The actual entry says that the input file will use ? instead of the default @ as the flag for separating a key from the start of the real entry. The `page_compositor` indicates that any compound numbers will be like 1.2.3 instead of the default 1-2-3.

RAW INPUT DATA

```
\@@wrglom@m{<file>}{<key>}{<term>}{<desc>}{<ref>}{<num>}
```

The `\glossary` macro writes its arguments to the aux file in the form of arguments to the `\@@wrglom@m` internal macro. In turn this calls a series of other macros that eventually write the data to the `<file>` glo file in the format (where @ is the actual flag):

```
\glossaryentry{key@{\memgloterm{term}} {\memglodesc{desc}}{\memglorefer{ref}}
|memglonumf}{num}
```

which MakeIndex then effectively converts into

```
\glossitem{\memgloterm{term}}{\memglodesc{desc}}{\memglorefer{ref}}
{\memglonum{\memglonumf{num}}}
```

```
\memgloterm{<term>}
\memglodesc{<desc>}
\memglorefer{<ref>}
\memglonum{<num>}
```

These macros can be redefined to format the various parts of a glossary entry. Their default definitions are simply

```
\newcommand{\memgloterm}[1]{#1}
\newcommand{\memglodesc}[1]{#1}
\newcommand{\memglorefer}[1]{#1}
\newcommand{\memglonum}[1]{#1}
```

For example, if you wanted the term in bold, the description in italics, and no numbers:

```
\renewcommand{\memgloterm}[1]{\textbf{#1}}
\renewcommand{\memglodesc}[1]{\textit{#1}}
\renewcommand{\memglonum}[1]{}
```

There are several macros that effect a glossary entry but which must not be directly modified (the `\memglonumf` shown above as part of the `\glossaryentry` is one of these). Each of the following `\changeGLOSS...` macros takes an optional `<file>` argument. The changes to the underlying macro apply only to the glossary of that particular `<file>` (or the `\jobname` file if the argument is not present).

```
\changeGLOSSactual [ <file> ] { <char> }
\changeGLOSSref [ <file> ] { <thecounter> }
\changeGLOSSnum [ <file> ] { <thecounter> }
\changeGLOSSnumformat [ <file> ] { <format> }
```

`\changeGLOSSactual` sets `<char>` as the actual character for the `<file>` glossary. It is initially @. This must match with the actual specified for the `gst` file that will be applied.

`\changeGLOSSref` specifies that `<thecounter>` should be used to generate the `<ref>` for the `<file>` glossary. It is initially nothing.

`\changeGLOSSnum` specifies that $\langle thecounter \rangle$ should be used to generate the $\langle num \rangle$ for the $\langle file \rangle$ glossary. It is initially `\thepage`.

`\changeGLOSSnumformat` specifies that $\langle format \rangle$ should be used to format the $\langle num \rangle$ for the $\langle file \rangle$ glossary. The format of $\langle format \rangle$ is `|form`, where `|` is the encap character specified in the `gst` file, and `form` is a formatting command, taking one argument (the number), without any backslash. For example

```
\changeGLOSSnumformat{|textbf}
```

to get bold numbers. It is initially set as `|memjustarg`, where this is defined as:

```
\newcommand{\memjustarg}[1]{#1}
```

There must be a format defined for the $\langle num \rangle$ otherwise the arguments to `\glossitem` will not be set correctly.

The `\makeGLOSSary` command uses the `\change...` commands to define the initial versions, so only use the `\change...` macros *after* `\makeGLOSSary`. In this document an early version of the glossary was set up by

```
\makeGLOSSary
\changeGLOSSactual{?}
\makeatletter
\changeGLOSSnum{\@currentlabel}
\makeatother
\changeGLOSSnum{\thepage}
```

The first call of `\changeGLOSSnum` makes the number the current numbered chapter, or numbered section, or numbered I didn't like that when I tried it, so the second call resets the number to the page number.

THE LISTING

The final glossary data in the `gls` file is typeset in the `thegLOSSary` environment, which is much like the `theindex` and `thebibliography` environments.

The environment starts off with a chapter-style unnumbered title. There are several macros for specifying what happens after that.

```
\gLOSSaryname
\gLOSSarymark
\gLOSSaryintoc \nogLOSSaryintoc
```

The title for the glossary is `\gLOSSaryname` whose initial definition is

```
\newcommand*{\gLOSSaryname}{Glossary}
```

`\gLOSSarymark`, which by default does nothing, can be redefined to set marks for headers. The glossary title will be added to the ToC if the `\gLOSSaryintoc` declaration is in force, but will not be added to the ToC following the `\nogLOSSaryintoc`.

```
\pregLOSSaryhook
```

The macro `\pregLOSSaryhook` is called after the glossary title has been typeset. By default it does nothing, but you could redefine it to, for example, add some explanatory material before the entries start.

```
\onecolglossarytrue \onecolglossaryfalse
\glossarycolsep \glossaryrule
```

The glossary can be typeset in two columns (`\onecolglossaryfalse`) but by default (`\onecolglossarytrue`) it is set in one column. When two columns are used, the length `\glossarycolsep` is the distance between the columns and the length `\glossaryrule` is the width (default 0) of a vertical rule between the columns.

```
\begintheglossaryhook
\atendtheglossaryhook
```

The last thing that `\begin{theglossary}` does is call `\begintheglossaryhook`. Similarly, the first thing that is done at the end of the environment is to call `\atendtheglossaryhook`. By default these macros do nothing but you can redefine them.

For example, if you wanted the glossary in the form of a description list, the following will do that.

```
\renewcommand*\begintheglossaryhook{\begin{description}}
\renewcommand*\atendtheglossaryhook{\end{description}}
\renewcommand{\glossitem}[4]{\item[#1:] #2 #3 #4}
```

THE GLOSSARY FOR THIS DOCUMENT

The following is the code I have used to produce the glossary in this document.

This is the code that is in the preamble.

```
%%% in the preamble
\makeglossary
\changeGLOSSACTUAL{?}
\changeGLOSSNUM{\thepage}
\changeGLOSSNUMFORMAT{|hyperpage}%% for hyperlinks
\renewcommand*\GLOSSARYNAME{Command summary}
\renewcommand*\GLOSSARYMARK{\markboth{\GLOSSARYNAME}{}}
```

```
\makeatletter
\renewcommand{\glossitem}[4]{%
  \sbox\@tempboxa{#1 \space #2 #3 #4}%
  \par\hangindent 2em
  \ifdim\wd\@tempboxa<0.8\linewidth
    #1 \space #2 #3 \dotfill #4\relax
  \else
    #1 \dotfill #4\
    #2 #3
  \fi}
\makeatother
```

The redefinition of `\glossitem` works as follows (it is similar to code used in the setting of a `\caption`):

1. Put the whole entry into a temporary box.

2. Set up a hanging paragraph with 2em indentation after the first line.
3. Check if the length of the entry is less than 80% of the linewidth.
4. For a short entry set the name, description, and any reference then fill the remainder of the line with dots with the number at the right margin.
5. For a longer entry, set the title and number on a line, separated by a line of dots, then set the description (and reference) on the following lines.

In the document the raw data is collected by the `\glossary` commands in the body of the text. For instance, although I have not actually used the first two:

```
\glossary(cs)%
  {\cs{cs}\gmarg{name}}%
  {Typesets \texttt{name} as a macro name with preceding backslash,
   e.g., \cs{name}.}%
\glossary(gmarg)%
  {\cs{gmarg}\gmarg{arg}}%
  {Typesets \texttt{arg} as a required argument, e.g., \gmarg{arg}.}
\glossary(glossaryname)%
  {\cs{glossaryname}}%
  {Name for a glossary}%
\glossary(memgloterm)%
  {\cs{memgloterm}\gmarg{term}}%
  {Wrapper round a glossary term.}%
```

Any change to the glossary entries will be reflected in the `glo` produced from that LaTeX run. Then run `MakelIndex` on the `glo` file using the appropriate `gst` configuration file, and run LaTeX again to get the corrected, sorted and formatted result printed by `\printglossary`.

4.6 Endnotes

Endnotes are often used instead of footnotes so as not to interrupt the flow of the main text. Although endnotes are normally put at the end of the document, they may instead be put at the end of each chapter.

The endnotes package already uses the command `\endnote` for an endnote, so the class uses `\pagenote` for an endnote so as not to clash if you prefer to use the package. The following was originally supplied as the `pagenote` package.

```
\makepagenotes
\pagenote[id]{text}
\printpagenotes \printpagenotes*
```

The general principal is that notes are written out to a file which is then input at the place where the notes are to be printed. The note file has an `ent` extension, like the table of contents file has a `toc` extension.

You have to put `\printpagenotes` in your preamble if you want endnotes. This will open the `ent` note file which is called `\jobname.ent`.

In the body of the text use `\pagenote` to create an endnote, just as you would use `\footnote` to create a footnote. In the books that I have checked there are two common methods of identifying an endnote:

1. Like a footnote, put a number in the text at the location of the note and use the same number to identify the note when it finally gets printed.
2. Put no mark in the text, but when it is finally printed use a few words from the text to identify the origin of the note. The page number is often used as well with this method.

The $\langle text \rangle$ argument of `\pagenote` is the contents of the note and if the optional $\langle id \rangle$ argument is not used the result is similar to having used `\footnote` — a number in the main text and the corresponding number in the endnotes listing (as in 1 above). For the second reference style (2 above) use the optional $\langle id \rangle$ argument for the ‘few words’, and no mark will be put into the main text but $\langle id \rangle$ will be used as the identification in the listing.

For one set of endnotes covering the whole document put `\printpagenotes` where you want them printed, typically before any bibliography or index. The `\printpagenotes` macro inputs the `ent` endnote file for printing and then closes it to any further notes.

For notes at the end of each chapter put `\printpagenotes*`, which inputs the `ent` file for printing then empties it ready for more notes, at the end of each chapter.

The simple use is like this:

```
\documentclass[...]{memoir}
...
\makepagenote
...
\begin{document}
\chapter{One}
... \pagenote{An end note.} ...
... \pagenote{Fascinating information.}
...
\chapter{Last}% chapter 9
... \pagenote{Another note.}% 30th note
...
...
\printpagenotes
...
\end{document}
```

This will result in an endnote listing looking like Figure 4.1.

For notes at the end of each chapter:

```
\documentclass[...]{memoir}
...
\makepagenote
...
\begin{document}
\chapter{One}
... \pagenote{An end note.} ...
...
\printpagenotes*
\chapter{Last}
... \pagenote{Another note.} ...
...
```

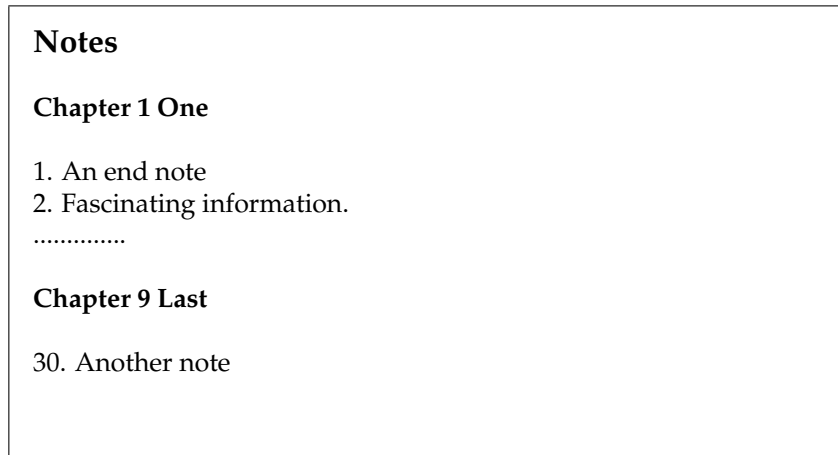


Figure 4.1: Example endnote listing

```
\printpagenotes*
%% no more chapters
...
\end{document}
```

```
\continuousnotenums
\notepageref
```

The `pagenote` counter is used for the notes. By default the endnotes are numbered per chapter. If you want the numbering to be continuous throughout the document use the `\continuousnotenums` declaration. Normally the information on which page a note was created is discarded but will be made available to notes in the endnote listing following the `\notepageref` declaration. These declarations should be put in your preamble.

Because of how TeX writes information to files, when the `\notepageref` declaration is used there must be no notes on the page where `\printnotes` or `\printnotes*` closes the ent file. If necessary, a `\clearpage` or similar must be used before the print command.

```
\notesname
\notedivision
```

When `\printnotes` (or `\printnotes*`) is called the first thing it does is call the macro `\notedivision`. By default this is defined as:

```
\newcommand*{\notesname}{Notes}
\newcommand*{\notedivision}{\chapter{\notesname}}
```

In other words, it will print out a heading for the notes that will be read from the ent file. `\print...` then closes the ent file for writing and after this `\inputs` it to get and process the notes.

CHANGING THE APPEARANCE

```
\notenumintext{<num>}
\notenuminnotes{<num>}
```

The `\pagenote` counter is used for pagenotes. The macro `\notenumintext` is called by `\pagenote` with the value of the `\pagenote` counter as the `<num>` argument to print the value of the `\pagenote` counter in the main text. By default it is printed as a superscript, but this can be changed, or even eliminated. In the note listing `\notenuminnotes` is used to print the number of a note. The default definitions are:

```
\newcommand*\notenumintext}[1]{\textsuperscript{#1}}
\newcommand*\notenuminnotes}[1]{\normalfont #1. }
```

```
\noteentry{<notenum>}{<id>}{<text>}{<pagenum>}
\prenoteinnotes
\postnoteinnotes
```

The `\pagenote` macro writes `\noteentry`, with the appropriate values for the arguments, to the ent file, where `<notenum>` is the note number (from the `\pagenote` counter), `<id>` and `<text>` are as supplied to `\pagenote`, and if the `\notepageref` declaration option is used, `<pagenum>` is the page number, otherwise it is empty. The `\noteentry` macro controls the typesetting of the note.

The default definition of `\noteentry` is

```
\newcommand{\noteentry}[4]{%
\prenoteinnotes
\noteidinnotes{#1}{#2}\pageinnotes{#4}\noteinnotes{#3}%
\postnoteinnotes}
```

and the definitions of other macros are:

```
\newcommand{\prenoteinnotes}{\par\noindent}
\newcommand{\postnoteinnotes}{\par}
```

so that (the first paragraph of) each note is printed as a non-indented paragraph.

If you would prefer, say, hanging paragraphs try:

```
\renewcommand{\prenoteinnotes}{\par\noindent\hangindent 2em}
```

```
\noteidinnotes{<notenum>}{<id>}
\idtextinnotes{<id>}
\notenuminnotes{<num>}
```

The `\noteidinnotes` calls `\idtextinnotes` to print the note id if it is given, otherwise it calls `\notenuminnotes` to print the note number. These are defined respectively as:

```
\newcommand*\idtextinnotes}[1]{[#1]\space}
\newcommand*\notenuminnotes}[1]{\normalfont #1.\space}
```

```
\pageinnotes{<pagenum>}
\printpageinnotes{<pagenum>}
```

The macro `\pageinnotes` controls the printing of a note's page reference. If the `\notepageref` declaration has been used it calls `\printpageinnotes` to do the actual printing. Its definition is:

```
\newcommand*\printpageinnotes}[1]{%  
  (\pagerefname\ #1)\space}
```

```
\noteinnotes{<text>}
```

The macro `\noteinnotes{<text>}` is simply:

```
\newcommand{\noteinnotes}[1]{#1}
```

and is used to print the text of a note.

```
\addtonotes{<text>}
```

The macro `\addtonotes` inserts `<text>` into the `ent` file. For example, before the first note in a chapter, `\addtonotes` is used to write the `\pagenotesubhead` command to the file.

Note: As the argument to `\pagenote` and `\addtonotes` is moving you may have to `\protect` any fragile commands. If you get strange error messages, try using `\protect` and see if they go away.

```
\pagenotesubhead{<chapapp>}{<num>}{<title>}  
\pnchap \pnschap
```

The macro `\pagenotesubhead` typesets the subheadings in an endnote list. It is inserted into the `ent` file via `\addtonotes`. The `<chapapp>` argument is normally `\chaptername` but if the notes are from an appendix then `\appendixname` is used instead. `<num>` is the number of the chapter, or blank if there is no number. Lastly, `<title>` is `\pnchap` for regular chapters which defaults to the ToC entry, or `\pnschap` for starred chapters which defaults to the normal title. The default definition of `\pagenotesubhead` is very simply:

```
\newcommand*\pagenotesubhead}[3]{%  
  \section*{#1 #2 #3}}
```

The scheme is set up under the assumption that notes will only be printed at the end of the document. If you intend to put them at the end of each chapter, then you will probably want to change the definitions of the `\notedivision` and `\pagenotesubhead` macros. For example:

```
\renewcommand*\notedivision{\section*{\notesname}}  
\renewcommand*\pagenotesubhead}[3]{}
```

and remember to use `\printnotes*` at each place you want the current set of notes to be printed.

```
\foottopagenote \pagetofootnote
```

You can have both footnotes and endnotes in the same document. However you may start with all footnotes and later decide you would have preferred endnotes instead, or *vice-versa*. The `\foottopagenote` declaration makes `\footnotes` behave as `\pagenotes`, and `\pagetofootnote` has the opposite effect. In either conversion the optional argument will be ignored as for `\pagenote` it can be arbitrary text whereas for `\footnote` it must be a number.

Five

Boxes and environments

5.1 Sidebars

It is now possible to control on which side of the page a sidebar gets placed.

```
\sidebarmargin{<margin>}
```

For onecolumn documents and with `\sidebaronesidefalse` you can use `\sidebarmargin` to specify which margin you want sidebars to be located. Possible values for *<margin>* are: `left`, `right`, `inner` or `outer` with the obvious meanings. (`\sidebarmargin{right}` and `\sidebaronesidetrue` are equivalent.) There are no positioning options for twocolumn documents.

5.2 Side notes

```
\sidepar{<left>}[<right>]  
\sideparvshift
```

`\sidepar` provides a fixed version of `\marginpar`. In the original version the marginal text could shift up or down depending on the presence or absence of ascenders and descenders in the marginal text and the main line text. The length `\sideparvshift` was provided to enable the shift to be compensated for, and its default value was `-2.08ex`. Dan Luecking has provided the new version of `\sidepar` whereby the marginal text does not shift and the default value of `\sideparvshift` is now `0pt`.

5.3 Boxed verbatims

```
\bvendofpage{<code>}
```

A boxed verbatim may extend across a page break and the `\bvendofpage` macro determines what happens at the bottom of the page just before the break. The default definitions is:

```
\newcommand{\bvendofpage}{\hrule\kern-.4pt}
```

which results in a horizontal rule being drawn. A `\hrule` takes `0.4pt` of vertical space and the `\kern` of `-0.4pt` backs up by `0.4pt` vertically, so as far as \TeX is concerned no space has been used. You may change the macro to something that better matches your needs if necessary.

5.4 Verse

There are a couple of small additions to the code for verse.

```
\vleftofline{<text>}
```

A verse line may start with something, for example open quote marks, where it is desirable that it be ignored as far as the alignment of the remainder of the line is concerned¹ — a sort of ‘hanging left punctuation’. When it is put at the start of a line in the `verse` environment the `<text>` of `\vleftofline` is typeset but ignored as far as horizontal indentation is concerned.

Compare the two settings below:

```
\noindent ‘No, this is what was spoken by the prophet Joel:
\vbegin{verse}
‘\,‘\,‘In the last days,’ God says, \
‘I will pour out my Spirit on all people. \
Your sons and daughters will prophesy, \
\ldots \
And everyone who calls \ldots ’\,’
\vend{verse}
```

“No, this is what was spoken by the prophet Joel:

```
““In the last days,” God says,
“I will pour out my Spirit on all people.
Your sons and daughters will prophesy,
...
And everyone who calls ...””
```

```
\noindent ‘No, this is what was spoken by the prophet Joel:
\vbegin{verse}
\vleftofline{‘\,‘\,‘}In the last days,’ God says, \
\vleftofline{‘}I will pour out my Spirit on all people. \
Your sons and daughters will prophesy, \
\ldots \
And everyone who calls \ldots ’\,’
\vend{verse}
```

“No, this is what was spoken by the prophet Joel:

```
““In the last days,” God says,
“I will pour out my Spirit on all people.
Your sons and daughters will prophesy,
...
And everyone who calls ...””
```

```
\vleftmargin
```

In the basic LaTeX `verse` environment the body of the verse is indented from the left of the text block by an amount `\leftmargini`, as is the text in many other environments based

¹The problem was presented to me by Matthew Ford who also provided the example text.

on the basic LaTeX `list` environment. For memoir's `verse` environment only, the default indent is set by the length `\vleftmargin` (which is initially set equal to `\leftmargini`). For poems with particularly long lines it could, for example, be advantageous to eliminate any indentation via:

```
\setlength{\vleftmargin}{0em}
```

If necessary the verse could even be moved into the left margin by giving `\vleftmargin` a negative length value, such as `-0.3em`.

Later, for poems with shorter lines the indentation can be reset to the default by:

```
\setlength{\vleftmargin}{\leftmargini}
```

```
\verselinenumbersright \verselinenumbersleft
```

Following the declaration `\verselinenumbersright`, which is the default, any verse line numbers will be set in the righthand margin. The `\verselinenumbersleft` declaration will set any subsequent line numbers to the left of the lines.

5.5 Lists

Normally within a `list` environment paragraphs are indented by `\listparindent`. By default this is set to `0pt`, which can be overridden in the second argument to the `list` environment (i.e. `\layout`).

```
\begin{list}{\label}{\layout} \item ... \end{list}
\listparindent
\everylistparindent
```

The class, though, initialises `\listparindent` to `\everylistparindent`, which in turn is initialized to `0pt`.

```
\defaultlists
```

The command `\defaultlists` sets the layout parameters for the `list` environment to their default values.

```
\tightlists \tightlists*
\firmlists \firmlists*
```

Following the `\tightlists` declaration the layout of `list` environments, such as `itemize`, `enumerate` or `description`, is tightened up by removing any internal vertical spacing within the list. If there is a blank line before the list, then a vertical space of half `\onelineskip` is put before and after the list. This space is not added when the `\tightlist*` declaration is given. In a tight list the paragraph indentation is set to the value of the regular `\parindent` paragraph indentation by setting `\everylistparindent` to `\parindent` (otherwise they would be block paragraphs with no interparagraph spaces).

The `\firmlists` declaration sets the vertical spacing in lists midway between their default values and those for `\tightlists`. The starred version, `\firmlists*` reduces the space before and after a list a little below that for the unstarred declaration.

5. BOXES AND ENVIRONMENTS

The regular list layout can be set at any time by issuing the `\defaultlists` declaration.

This document is set using `\tightlists`. If you are going to use any of the non-default list settings, then I suggest that this one is perhaps the most useful.

Six

Captions

6.1 Side captions

The class now provides for placing captions at the side of figures or tables, or other floats.

```
\begin{sidecaption} [/fortoc] {<title>} [<label>]  
the body of the float  
\end{sidecaption}
```

The `sidecaption` environment is used for a sidecaption rather than a macro. The body of the float is put inside the environment. For example:

```
\begin{figure}  
  \begin{sidecaption}{An illustration}[fig:ill]  
    \centering  
    \includegraphics{...}  
  \end{sidecaption}  
\end{figure}
```

whereby the caption, ‘Figure N: An illustration’, will be placed in the margin alongside the graphic, and for reference purposes will be given given the `\label fig:ill`.

```
\sidecapwidth \sidecapsep  
\sidecapmargin{margin}  
\ifscapmargleft \scapmarglefttrue \scapmargleftfalse
```

The caption is set in a box `\sidecapwidth` wide (the default is `\marginparwidth`) offset `\sidecapsep` (default `\marginparsep`) into the margin.

If the float is a single column float in a twocolumn document then the caption is always¹ placed in the adjacent margin, otherwise the `\sidecapmargin` command controls the margin where the sidecaption will be placed. The possible values for `<margin>` are one of: `left`, `right`, `inner`, or `outer`. If `left` or `right` is specified the caption will go into the left or right margin. If `inner` or `outer` is specified then in a two sided document the caption will be on different sides of the textblock according to whether it is a recto or verso page; in a one sided document the caption margin is fixed. The left margin is the default.

¹Well, nearly always. See the `\overridescapmargin` command later.

6. CAPTIONS

When the caption is to be set in the left margin, `\ifscapmargleft` is set `true`, and for a right margin it is set `false`.

```
\setsidecappos{<pos>}
```

By default a sidecaption is vertically centered with respect to the float it is captioning. This can be altered by using the `\setsidecappos` declaration. The allowed values for `<pos>` are:

- t — the top of the caption is aligned with the top of the float
- c — (the default) the center of the caption is aligned with the center of the float
- b — the bottom of the caption is aligned with the bottom of the float

The other kinds of simple captions can also be put at the side of a float. The positioning and styling commands for these are exactly those for `sidecaption`. Bilingual captions can only be placed above or below the float.

```
\begin{sidecontcaption}{<title>}[<label>]
the body of the float
\end{sidecontcaption}
```

Sidecaptions may be continued with the `sidecontcaption` environment.

```
\begin{sidenamedlegend}[<fortoc>]{<title>}
the body of the float
\end{sidenamedlegend}
```

Named legends may be set at the side with the `sidenamedlegend` environment.

```
\begin{sidelegend}{<title>}
the body of the float
\end{sidelegend}
```

Legends may be set at the side with the `sidelegend` environment.

TWEAKS

```
\sidecapstyle
```

Just before the caption is set, the `\sidecapstyle` command is called. This may be used to set the styling for the particular caption. By default it sets captions that are in the left margin `raggedleft`, and those that are in the right margin are set `raggedright`. The default definition is:

```
\newcommand*{\sidecapstyle}{%
%% \captionnamefont{\bfseries}%
\ifscapmargleft
  \captionstyle{\raggedleft}%
\else
  \captionstyle{\raggedright}%
\fi}
```

Table 6.1: Permitted arguments for some sidecaption related commands

<code>\sidecapmargin</code>	<code>\overridescapmargin</code>
left	left
right	right
inner	
outer	

You can change the command to suit your purposes; for example, uncommenting the `\captionnamefont` line would result in the caption's float name being set in a bold font. See the manual for more information on what you can do to change the appearance of a caption.

```
\overridescapmargin{<margin>}
\sidecapraise
```

Sometimes the caption may not be placed exactly where you want it — it may be in the wrong margin or at the wrong height.

The command `\overridescapmargin` will force the following caption into the `<margin>` you specify which can only be `left` or `right`. In a twosided document where `\sidecapmargin` is `inner` or `outer` and the caption goes in the wrong margin, it is likely that the declaration `\strictpagechecktrue` will solve the problem. The wrong margin might be chosen in a twocolumn document where the float is in the second column; use `\overridescapmargin{right}` to fix this.

The caption may not be at quite the height you want with respect to the float. The caption will be raised by the length `\sidecapraise` in addition to the calculated movement (or lowered if `\sidecapraise` is negative).

```
\sidecapfloatwidth{<length>}
```

The float is set in a minipage with width `sidecapfloatwidth`, whose default definition is

```
\newcommand*{\sidecapfloatwidth}{\linewidth}
```

That is, the normal width is the same as the current `\linewidth`. For a narrow table, say, you may want to reduce this, for example to half by

```
\renewcommand*{\sidecapfloatwidth}{0.5\linewidth}
```

Note that `\sidecapfloatwidth` is a macro, not a length, so it must be altered by using a `\renewcommand*`, *not* by `\setlength`.

If you do reduce the `\sidecapfloatwidth` you may notice that the sidecaption is actually placed a distance `\sidecapsep` with respect to the float's minipage, not with respect to the text block.

Table 6.1 was created by the following code.

```
\newlength{\mylength}
\setlength{\mylength}{\linewidth}
\addtolength{\mylength}{-\sidecapsep}
\addtolength{\mylength}{-\sidecapwidth}
```

```
\begin{table}
  \sidecapmargin{left}%
  \renewcommand*{\sidecapfloatwidth}{\mylength}%
  \raggedleft
  \begin{sidecaption}{%
    Permitted arguments for some sidecaption related commands}[scap:one]
  \centering
  \begin{tabular}{cc} \toprule
    \cs{sidecapmargin} & \cs{overridescapmargin} \\ \midrule
    \texttt{left}      & \texttt{left}      \\
    \texttt{right}     & \texttt{right}     \\
    \texttt{inner}     & \\
    \texttt{outer}    & \\ \bottomrule
  \end{tabular}
  \end{sidecaption}
\end{table}
```

The calculations on the `\mylength` length are so that the sidecaption and float will just fit inside the textblock.

Note that the `\raggedleft` command before the sidecaption environment makes the float's minipage be placed raggedleft (i.e., moved across to the right hand edge of the textblock) while the `\centering` centers the tabular within the minipage. You can get a variety of horizontal placements by judicious use of `\raggedright`, `\centering` and `\raggedleft` commands. If you do move the float sideways to leave space for the caption make sure that the caption will go to the side you want. In the example code I 'moved' the float to the right so I made sure that the caption would go on the left by explicitly setting

```
\sidecapmargin{left}
```

As far as TeX is concerned a sidecaption takes no horizontal space. If you use a sidecaption in a wrapped float from, say, the `wrapfig` package, make sure that the sidecaption gets placed where it won't be overlaid by the main text.

6.2 Caption title text

```
\captiontitlefinal{<stuff>}
```

`\captiontitlefinal{<stuff>}` will put `<stuff>` immediately at the end of a caption's title, but `<stuff>` will not appear in the LoF or LoT. The default is

```
\captiontitlefinal{}
```

but it could be called instead as, say

```
\captiontitlefinal{.}
```

to put a period (full stop) after the title.

Seven

Miscellaneous

7.1 General

FONT COMMANDS

The `\em` command is no longer deprecated (it was a misreading on my part to deprecate it in the first place).

```
\eminershape{shape}
```

If the `\emph` command is used within italic text then the newly emphasized text will be typeset using the `\eminershape` font shape. The default definition is:

```
\newcommand*\eminershape{\upshape}
```

which you can change if you wish.

FOOTNOTES

There was a question on CTT asking how to ensure that footnotes were at the bottom of the page when `\raggedbottom` was in effect. The solution is to add a `\vfill` to the `\footnoterule` macro, as below.

```
\renewcommand*\footnoterule{\kern-3pt\vfill
  \hrule width 0.4\columnwidth \kern 2.6pt}
```

FLOATS

A page which consists of floats (e.g., `table`, `figure`) with no body text is called a *floatpage*. By default, floats on a floatpage are centered vertically. To move the floats to the top of the page try

```
\makeatletter
  \setlength{\@fptop}{0pt}
  \setlength{\@dblftop}{0pt}
\makeatother
```

To move the floats to the bottom of the page, replace `\@fptop` and `\@dblftop` by `\@fpbot` and `\@dblfpbot` respectively.

One author thought it would be nice to be able to have different page headings according to whether the page was a floatpage, or there was a float at the top of the page, or a float at the bottom of a page or there was text at the top and bottom.

This, I think, is not a common requirement and, further, that to provide this involves changing parts of the LaTeX output routine — something only to be tackled by the bravest of the brave. If it were to be done then were best done in a package that could be easily ignored. The following is an outline of what might be done; I do not recommend it and if you try this and all your work disappears then on your own head be it.

```
% notefloat.sty
\newif\iffloatattop
\floatattopfalse
\newif\iffloatatbot
\floatatbotfalse

\renewcommand*{\@addtotoporbot}{%
\@getfpsbit \tw@
\ifodd \@tempcnta
\@flsetnum \@topnum
\ifnum \@topnum>\z@
\@tempwafalse
\@flcheckspace \@toproom \@toplist
\if@tempswa
\@bitor\@currtype{\@midlist\@botlist}%
\if@test
\else
\@flupdates \@topnum \@toproom \@toplist
\@inserttrue
\global\floatattoptrue
\fi
\fi
\fi
\if@insert
\else
\@addtobot
\fi}

\renewcommand*{\@addtobot}{%
\@getfpsbit 4\relax
\ifodd \@tempcnta
\@flsetnum \@botnum
\ifnum \@botnum>\z@
\@tempwafalse
\@flcheckspace \@botroom \@botlist
\if@tempswa
\global \maxdepth \z@
\@flupdates \@botnum \@botroom \@botlist
```

```

        \@inserttrue
\global\floatatbottrue
        \fi
        \fi
\fi}

\let\p@wold@output\@outputpage
\renewcommand*{\@outputpage}{%
        \p@wold@output
\global\floatattopfalse
\global\floatatbotfalse}

```

\endinput

\floatatop is probably set TRUE if there is a float at the top of the page and \floatatbot is probably set TRUE if there is a float at the bottom of the page.

TWO SIMPLE MACROS

There are two trivial macros that can be generally useful.

```

\memjustarg{<text>}
\memgobble{<text>}

```

The \memjustarg macro just uses its argument and is defined as:

```
\newcommand*\memjustarg[1]{#1}
```

The \memgobble macro gobbles down and swallows its argument. Its definition is:

```
\newcommand\memgobble[1]{}

```

Do *not* redefine either \memjustarg or \memgobble; if you do various pieces of code will behave in unexpected ways that you will not like.

TRIM MARKS

```
\showtrimsoff \showtrimson
```

If the showtrims option has been used then the \showtrimsoff switches off the trim marks; the \showtrimson declaration, which is the default, switches on the trim marks. These declarations do nothing if the showtrims option is not used.

VERTICAL CENTERING

The manual describes one method for centering text vertically. The new vplace environment provides a simpler way.

```
\begin{vplace}[<num>] text \end{vplace}
```

The contents of the vplace environment are vertically centered. The optional <num> argument can be used to specify the ratio of the upper space to the lower space. You can put other text on the page above or below the centered text. The environment may be useful for title pages.

7.2 For package writers

The facilities described in this section are for anyone to use but I suspect that they may be most useful to package developers.

EMULATING PACKAGES

```
\EmulatedPackage{package}[date]  
\EmulatedPackageWithOptions{optionlist}{package}[date]
```

These commands are for package writers; they are based on a conversation with Donald Arseneau on CTT. They fool L^AT_EX into thinking that the *package* has already been loaded so it won't try loading it again. These are probably only useful if your package includes the actual code for *package*.

memoir does include code from several packages and uses a similar internal command to ensure that the packages are not loaded following some later `\usepackage` command. The names of the emulated packages are written to the log file. At the time of writing the emulated packages are: abstract, appendix, array, booktabs, ccaption, chngcntr, crop, dcolumn, delarray, enumerate, epigraph, framed, ifmtarg, ifpdf, index, makeidx, moreverb, needspace, newfile, nextpage, pagenote, patchcmd, shortvrb, showidx, tabularx, titleref, tocibind, tocloft, verbatim, and verse. As well as the emulated packages memoir provides functions equivalent to those in the following packages, although the class does not prevent you from using them: fancyhdr, geometry, sidecap, subfigure, and titlesec.

```
\DisemulatePackage{package}
```

This command undoes any prior `\EmulatedPackage` or `\EmulatedPackageWithOptions` for the *package* package. For example, if you wish to use the index package instead of memoir's emulation then put

```
\DisemulatePackage{index}  
\usepackage{index}  
in your preamble.
```

EXTENDING A MACRO

```
\patchcommand{macro}{start-code}{end-code}
```

The `\patchcommand` is from the late Michael Downes' patchcmd package [Dow00]. It inserts the *start-code* at the start of the current definition of the macro *macro*, and inserts *end-code* at the end of its current definition. The *macro* can have zero to nine parameters. If *macro* uses `\futurelet` (e.g., it is a starred command or takes an optional argument) only *start-code* is useful — *end-code* must be empty otherwise things get messed up. If *macro* has any delimited arguments then `\patchcommand` cannot be used.

INSERTING CODE BEFORE AND AFTER A FILE, PACKAGE OR CLASS

The kernel provides two commands, `\AtBeginDocument` and `\AtEndDocument` which can only be used in the preamble, for inserting code at the start and end of the document environment.

The kernel also provides the macros `\AtEndOfPackage{<code>}` and `\AtEndOfClass{<code>}` for inserting code at the end of the current package or class. More precisely, these macros call the `<code>` after the package or class file has been input via `\InputIfFileExists`.

The class provides a more comprehensive set of macros for code insertions, which should be used before the relevant file is called for.

```
\AtBeginFile{<file>}{<code>}
\AtEndFile{<file>}{<code>}
```

The `\AtBeginFile` macro inserts `<code>` just before the `<file>` file is `\input` (or `\included`, etc.). Similarly `\AtEndFile` inserts the `<code>` immediately after the `<file>`. The `<file>` argument must be the same as used in the corresponding `\input` command. If `<file>` includes an extension, for example `fred.def`, then that is taken as the complete name, otherwise if there is no extension, for instance `fred`, then the `.tex` extension is automatically appended making the full name `fred.tex`.

The `\At...File` commands must be issued *before* the corresponding `<file>` is input otherwise nothing will happen.

```
\AtBeginPackage{<pack>}{<code>}
\AtEndPackage{<pack>}{<code>}
\RequireAtEndPackage{<pack>}{<code>}
```

The `\AtBeginPackage` command will insert `<code>` just before the `<pack>` package is used. Similarly `\AtEndPackage` will insert the `<code>` immediately after the `<pack>`. The `<pack>` argument must be the same as used in the corresponding `\usepackage` command, that is, without any extension. The `\At...Package` commands must be issued *before* the corresponding `<pack>` is used otherwise nothing will happen.

The `\RequireAtEndPackage` command will, like `\AtEndPackage`, insert `<code>` at the end of the `<pack>` package if it has not yet been used. If the package has already been used then the `<code>` is called immediately.

```
\AtBeginClass{<class>}{<code>}
\AtEndClass{<class>}{<code>}
\RequireAtEndClass{<class>}{<code>}
```

The `\AtBeginClass` command will insert `<code>` just before the `<class>` class is used. Similarly `\AtEndClass` will insert the `<code>` immediately after the `<class>`. The `<class>` argument must be the same as used in the corresponding `\LoadClass` command, that is, without any extension. The `\At...Class` commands must be issued *before* the corresponding `<class>` is used otherwise nothing will happen.

The `\RequireAtEndClass` command will, like `\AtEndClass`, insert *code* at the end of the *class* class if it has not yet been used. If the class has already been used then the *code* is called immediately.

There is an unfortunate interaction between the kernel's `\AtEndOfPackage` and the class's `\AtEndPackage`, and similarly for the `\AtEndOfClass` and `\AtEndClass`. I discovered this when I tried to automate using the `memhfixc` package if `hyperref` was being used by putting the following into the memoir code

```
\AtEndPackage{hyperref}{\usepackage{memhfixc}}
```

which caused all sorts of problems.

The kernel scheme looks like this:

```
\newcommand{\usepackage}[1]{%  
  ...  
  \InputIfFileExists{#1}  
<AtEndOfPackage code>}
```

The basic mechanism for implementing the class macros is by modifying the kernel's `\InputIfFileExists` macro, which internally uses a form of `\input` to read in the file, so that the inserted *code* comes immediately before and after the `\input`, somewhat like:

```
\renewcommand{\InputIfFileExists}[1]{%  
  ...  
  <before code> \input{#1} <after code>}
```

If `\AtEndPackage` is applied to a package that has an internal `\AtEndOfPackage` then the result can be sketched as:

```
\newcommand{\usepackage}[1]{%  
  ...  
  <before code>  
  \input{#1}  
  <after code>  
  <AtEndOfPackage code>  
}
```

In other words the body of the package is read in, the `\AtEndPackage` code is called, and then *after* that the `\AtEndOfPackage` code is called.

The `hyperref` package internally uses `\AtEndOfPackage` to read some files and `memhfixc` had to be input after these. A way to automate `memhfixc` after `hyperref` is:

```
\AtEndPackage{hyperref}{%  
  \AtBeginDocument{\usepackage{memhfixc}}}
```

but this seems more trouble than it's worth.

7.3 Heading hooks

On 2nd September 2005 I posted two messages to the `comp.text.tex` newsgroup saying that I was creating a new version of memoir and that I would consider inserting hooks into the class code that package writers might find useful. I got no requests for any hooks or anything else from package writers. I therefore assume that no package author sees any problems if a memoir class document author uses the package.

However, I have provided macros that that may be useful for those who want to do things with the contents of section headings, captions, and the like. The macros are called within the relevant heading or caption code, and by default are defined to do nothing.

Hooks for the `\book` and `\book*` commands.

```
\membookinfo{<thebook>}{<fortoc>}{<title>}
\membookstarinfo{<title>}
```

Hooks for the `\part` and `\part*` commands.

```
\mempartinfo{<thepart>}{<fortoc>}{<title>}
\mempartstarinfo{<title>}
```

In many cases a `\mem...info` macro includes an argument related to the heading's number (`<thepart>` for `\mempartinfo`). In certain circumstances, such as a `\chapter` in the `\frontmatter`, there might not be a number even though the normal unstarred version of the command is used. In these cases the number argument (`<thechapter>` in the case of `\memchapinfo`) is left empty.

Hooks for the `\chapter` and `\chapter*` commands. Note that regular chapters and those as appendices are treated differently.

```
\memchapinfo{<thechapter>}{<fortoc>}{<forhead>}{<title>}
\memchapstarinfo{<fortoc>}{<title>}
\memappchapinfo{<thechapter>}{<fortoc>}{<forhead>}{<title>}
\memappchapstarinfo{<fortoc>}{<title>}
```

Hooks for `\section`, `\subsection`, etc., and their starred versions. `<name>` is the type of section (e.g., `section`, or `subsection`, or `subsubsection` or ...

```
\memsecinfo{<name>}{<thename>}{<fortoc>}{<forhead>}{<title>}
\memsecstarinfo{<name>}{<title>}
```

Hooks for appendix-like page headings.

```
\memapppageinfo{<title>}
\memapppagestarinfo{<title>}
\memleadpageinfo{<pstyle>}{<cmdname>}{<title>}
\memleadpagestarinfo{<pstyle>}{<cmdname>}{<title>}
```

Hooks for `\poemtitle`, `\PoemTitle`, and their starred versions.

```
\mempoeminfo{<title>}
\mempoemstarinfo{<title>}
\memPoemTitleinfo{<thepoem>}{<fortoc>}{<forhead>}{<title>}
\memPoemTitlestarinfo{<fortoc>}{<title>}
```

Hooks for the several kinds of `\caption` and `\legend` commands.

```
\memcaptioninfo{<type>}{<thetype>}{<fortoc>}{<title>}
\memlegendinfo{<title>}
\memnamedlegendinfo{<fortoc>}{<title>}
\membitwonumcaptioninfo{<type>}{<thetype>}{<fortoc1>}{<title1>}
    {<name2>}{<fortoc2>}{<title2>}
\membionenumcaptioninfo{<type>}{<thetype>}{<fortoc1>}{<title1>}
    {<name2>}{<fortoc2>}{<title2>}
\membicaptioninfo{<type>}{<thetype>}{<fortoc1>}{<title1>}{<name2>}{<title2>}
```

Command summary

<code>\addtonotes{⟨text⟩}</code>	Inserts <code>⟨text⟩</code> into the endnotes ent file.	30
<code>afterbook</code>	8
	Page style applied to the blank page, if any, following a book title page. Defaults to <i>empty</i> .	
<code>\afterbookskip</code>	Macro called after setting a book title	8
<code>afterpart</code>	9
	Page style applied to the blank page, if any, following a part title page. Defaults to <i>empty</i> .	
<code>\afterpartskip</code>	Vertical space after a part heading	1
<code>\afterPoemTitle</code>	13
	Called after printing the title of a <code>\PoemTitle</code> .	
<code>\afterPoemTitlenum</code>	13
	Called after printing the number of a <code>\PoemTitle</code> .	
<code>\afterPoemTitleskip</code>	Vertical space after a poem title	13
<code>\AtBeginClass{⟨pack⟩}{⟨code⟩}</code>	43
	Inserts <code>⟨code⟩</code> just before the <code>⟨class⟩</code> class is used.	
<code>\AtBeginFile{⟨file⟩}{⟨code⟩}</code>	43
	Inserts <code>⟨code⟩</code> just before the <code>⟨file⟩</code> is input (or included, etc.).	
<code>\AtBeginPackage{⟨pack⟩}{⟨code⟩}</code>	43
	Inserts <code>⟨code⟩</code> just before the <code>⟨pack⟩</code> package is used.	
<code>\AtEndClass{⟨class⟩}{⟨code⟩}</code>	43
	Inserts <code>⟨code⟩</code> just after the <code>⟨class⟩</code> class is used.	
<code>\AtEndFile{⟨file⟩}{⟨code⟩}</code>	43
	Inserts <code>⟨code⟩</code> just after the <code>⟨file⟩</code> is input (or included, etc.).	
<code>\AtEndPackage{⟨pack⟩}{⟨code⟩}</code>	43
	Inserts <code>⟨code⟩</code> just after the <code>⟨pack⟩</code> package is used.	
<code>\atendtheglossaryhook</code>	25
	Vacuous macro called as the first thing by <code>\end{theglossary}</code> .	
<code>\beforebookskip</code>	8
	Macro called before setting any part of a book page	
<code>\beforepartskip</code>	Vertical space before a part heading.	1
<code>\beforePoemTitleskip</code>	Vertical space before a poem title.	13

<code>\begintheglossaryhook</code>	25
Vacuous macro called as the last thing by <code>\begin{theglossary}</code> .	
<code>\book[<i><toc-title></i>]{<i><title></i>}</code>	7
Typsets a numbered book title and adds the number and title to the ToC.	
<code>\book*{<i><title></i>}</code>	7
Typsets an unnumbered book title and puts nothing in the ToC.	
<code>\bookblankpage</code>	8
Declaration for a blank page to be output after a book title page. This is the default.	
<code>\bookmark{<i><forhead></i>}</code>	7
For setting any marks containing the text of a header for a <code>\book</code> .	
<code>\bookname</code> The name for a book division (default Book).	8
<code>\booknamenum</code>	8
Macro called after <code>\printbookname</code> and before <code>\printbooknum</code> . Defaults to a space.	
<code>\booknamefont</code>	8
Font used for printing a book name. Defaults to a <code>\huge</code> bold font.	
<code>\booknumfont</code>	8
Font used for printing a book number. Defaults to a <code>\huge</code> bold font.	
<code>book</code> Page style applied to the book title page. Defaults to <i>empty</i>	8
<code>\booktitlefont</code>	8
Font used for printing a book title. Defaults to a <code>\Huge</code> bold font.	
<code>\bvendofpage{<i><code></i>}</code>	31
Controls the appearance of a page break in a boxed verbatim.	
<code>\cftappendixname</code>	15
In the ToC, called before the number of a chapter forming an appendix.	
<code>\cftbeforebookskip</code> Space before a book entry in the ToC.	18
<code>\cftbookaftersnum</code>	18
Macro called after a book number in the ToC in the number box.	
<code>\cftbookaftersnumb</code>	18
Macro called after a book number's box in the ToC.	
<code>\cftbookdotsep</code>	18
Separation between the dots on a book leader in the ToC.	
<code>\cftbookfont</code>	18
Font for typesetting a book number and title in the ToC.	
<code>\cftbookindent</code>	18
Indent of a ToC book entry from the left margin.	
<code>\cftbookleader</code>	18
Typesets the leader between a book title and page number in the ToC.	
<code>\cftbookname</code> In the ToC, called before the number of a book.	18
<code>\cftbooknumwidth</code>	18
Space allowed for a book entry's number in the ToC.	
<code>\cftbookpagefont</code>	18
Font for typesetting a book's page number in the ToC.	
<code>\cftbookpresnum</code>	18
Macro called before a book number in the ToC in the number box.	

<code>\cftchaptername</code>	15
In the ToC, called before the number of a chapter.	
<code>\cftfigurename</code> In the LoF, called before the number of a figure.	16
<code>\cftparagraphname</code>	16
In the ToC, called before the number of a paragraph.	
<code>\cftpartname</code> In the ToC, called before the number of a part.	15
<code>\cftsectionname</code>	16
In the ToC, called before the number of a section.	
<code>\cftsubparagraphname</code>	16
In the ToC, called before the number of a subparagraph.	
<code>\cftsubsectionname</code>	16
In the ToC, called before the number of a subsection.	
<code>\cftsubsubsectionname</code>	16
In the ToC, called before the number of a subsubsection.	
<code>\cfttablename</code> In the LoT, called before the number of a table.	16
<code>\changeGLOSSACTUAL</code> [<i>file</i>] { <i>char</i> }	23
Specifies <i>char</i> as the actual character for glossary <i>file</i> .	
<code>\changeGLOSSNUM</code> [<i>file</i>] { <i>thecounter</i> }	23
Specifies <i>thecounter</i> as the <i>num</i> for glossary <i>file</i> .	
<code>\changeGLOSSNUMFORMAT</code> [<i>file</i>] { <i>format</i> }	23
Specifies <i>format</i> as the format for <i>num</i> for glossary <i>file</i> .	
<code>\changeGLOSSREF</code> [<i>file</i>] { <i>thecounter</i> }	23
Specifies <i>thecounter</i> as the <i>ref</i> for glossary <i>file</i> .	
<code>\checkandfixthelayout</code> [<i>algorithm</i>]	3
Command to check and implement the page layout specifications, adjusting the <code>\textheight</code> using <i>algorithm</i> (classic, fixed, lines, or nearest, the default being classic) for the calculation.	
<code>\checkthelayout</code> [<i>algorithm</i>]	3
Command to check the page layout specifications, adjusting the <code>\textheight</code> using <i>algorithm</i> (classic, fixed, lines, or nearest, the default being classic) for the calculation.	
<code>\continuousnotenums</code>	28
Declaration to make the numbering of endnotes continuous throughout the document.	
<code>\defaultlists</code>	33
Command setting list layout parameters to their default values.	
<code>\DisemulatePackage</code> { <i>package</i> }	42
Undo a previous <code>\EmulatedPackage</code> or <code>\EmulatedPackageWithOptions</code> for the <i>package</i> package.	
<code>\eminnershape</code> { <i>shape</i> }	39
Font shape for emphasized text within emphasized text.	
<code>\EmulatedPackage</code> { <i>package</i> }[<i>date</i>]	42
Claim that the <i>package</i> package has been loaded.	
<code>\EmulatedPackageWithOptions</code> { <i>optionlist</i> }{ <i>package</i> }[<i>date</i>]	42
Claim that the <i>package</i> package has been loaded with options <i>optionlist</i> .	

<code>\everylistparindent</code>	33
Default value for <code>\listparindent</code> — the paragraph indentation in a <code>list</code> environment.	
<code>\firmlists</code>	33
Declaration for decreased vertical spacing in <code>list</code> -based environments. Some additional space may be added before and after.	
<code>\firmlists*</code>	33
Declaration for decreased vertical spacing in <code>list</code> -based environments. There is no additional space before or after.	
<code>\foottopagenote</code>	30
Declaration which turns <code>\footnotes</code> into <code>\pagenotes</code> .	
<code>\fussy</code>	5
Declaration for TeX to minimise interword space variations in justified text lines.	
<code>\glossary</code> [<i>file</i>] (<i>key</i>) { <i>term</i> } { <i>description</i> }	19
Adds <i>term</i> and its description, <i>desc</i> , to a glossary file — <code>\jobname.glo</code> by default or to <code>\file.glo</code> . The optional argument <i>key</i> can be used to provide a different sort key for <i>term</i> .	
<code>\glossarycolsep</code> Columns separation in a two column glossary.	25
<code>\glossaryintoc</code> Declaration to add glossary title to the ToC.	24
<code>\glossarymark</code> Redefine to specify marks for headers.	24
<code>\glossaryname</code> Name for a glossary.	24
<code>\glossaryrule</code>	25
Width of inter-column rule in a two column glossary.	
<code>\glossitem</code> { <i>term</i> }{ <i>desc</i> }{ <i>ref</i> }{ <i>num</i> }	20
Glossary entry used in a <code>theglossary</code> environment	
<code>\idtextinnotes</code> { <i>id</i> } Prints an endnote's id text	29
<code>\ifanappendix</code>	1
TRUE when processing an appendix, otherwise FALSE.	
<code>\ifsidecapleft</code>	35
true if sidecaptions will be set in the left margin, otherwise they will be set in the right margin.	
<code>\leadpagetoclevel</code> { <i>sec</i> }	9
The ToC style for an entry from a <code>\newleadpage</code> generated command.	
<code>\begin</code> { <code>list</code> }{ <i>label</i> }{ <i>layout</i> }	33
List environment with entries introduced by <code>\item</code> . <i>label</i> is the default label for <code>\item</code> and <code>{<i>layout</i>}</code> specifies changes to the default layout.	
<code>\listparindent</code> Paragraph indentation in a <code>list</code> environment.	33
<code>\makeglossary</code> [<i>file</i>]	19
Opens file <code>\jobname.glo</code> , or <code>\file.glo</code> , for glossary entries	
<code>\makepagenotes</code>	26
Preamble command for enabling page/end notes.	

<code>\memappchapinfo{<thechapter>}{<fortoc>}{<forhead>}{<title>}</code>	45
Code hook into an appendix <code>\chapter</code>	
<code>\memappchapstarinfo{<fortoc>}{<title>}</code>	45
Code hook into an appendix <code>\chapter*</code>	
<code>\memapppageinfo{<title>}</code> Code hook into <code>\appendixpage</code>	45
<code>\memapppagestarinfo{<title>}</code>	45
Code hook into <code>\appendixpage*</code> .	
<code>\membicaptioninfo{<type>}{<thetype>}{<fortoc1>}{<title1>}{<name2>}{<title2>}</code> ..	46
Code hook into <code>\membicaption</code> .	
<code>\membionenumcaptioninfo{<type>}{<thetype>}{<fortoc1>}{<title1>}{<name2>}{<fortoc2>}{<title2>}</code>	46
Code hook into <code>\membionenumcaption</code> .	
<code>\membitwonumcaptioninfo{<type>}{<thetype>}{<fortoc1>}{<title1>}{<name2>}{<fortoc2>}{<title2>}</code>	46
Code hook into <code>\membitwonumcaption</code> .	
<code>\membookinfo{<thebook>}{<fortoc>}{<title>}</code>	45
Code hook into <code>\book</code>	
<code>\membookstarinfo{<title>}</code> Code hook into <code>\book*</code>	45
<code>\memcaptioninfo{<type>}{<thetype>}{<fortoc>}{<title>}</code>	46
Code hook into <code>\caption</code>	
<code>\memchapinfo{<thechapter>}{<fortoc>}{<forhead>}{<title>}</code>	45
Code hook into <code>\chapter</code>	
<code>\memchapstarinfo{<fortoc>}{<title>}</code> Code hook into <code>\chapter*</code>	45
<code>\memglodesc{<desc>}</code> Wrapper round a glossary description.	23
<code>\memglonum{<num>}</code> Wrapper round glossary numbers.	23
<code>\memgloref{<ref>}</code> Wrapper round a glossary ref.	23
<code>\memgloterm{<term>}</code> Wrapper round a glossary term.	23
<code>\memgobble{<text>}</code> Gobbles its argument. Do <i>not</i> redefine it.	41
<code>\memhyperindexfalse</code>	19
Preamble command to prevent hyperindexing.	
<code>\memjustarg{<text>}</code> Definition is just <code><text></code> . Do <i>not</i> redefine it.	41
<code>\memleadpageinfo{<pstyle>}{<cmdname>}{<title>}</code>	45
Code hook into <code>\newleadpage</code> and <code>\renewleadpage</code> .	
<code>\memleadpageinfo{<pstyle>}{<cmdname>}{<title>}</code>	45
Code hook into <code>\newleadpage*</code> and <code>\renewleadpage*</code> .	
<code>\memlegendinfo{<title>}</code> Code hook into <code>\legend</code>	46
<code>\memnamedlegendinfo{<fortoc>}{<title>}</code>	46
Code hook into <code>\namedlegend</code>	
<code>\mempartinfo{<thepart>}{<fortoc>}{<title>}</code> Code hook into <code>\part</code>	45
<code>\mempartstarinfo{<title>}</code> Code hook into <code>\part*</code>	45
<code>\mempoeminfo{<title>}</code> Code hook into <code>\poemtitle</code>	45
<code>\mempoemstarinfo{<title>}</code> Code hook into <code>\poemtitle*</code>	45
<code>\memPoemTitleinfo{<thepoem>}{<fortoc>}{<forhead>}{<title>}</code>	45
Code hook into <code>\PoemTitle</code>	
<code>\memPoemTitlestarinfo{<fortoc>}{<title>}</code>	45
Code hook into <code>\PoemTitle*</code>	

<code>\memsecinfo{⟨name⟩}{⟨thename⟩}{⟨fortoc⟩}{⟨forhead⟩}{⟨title⟩}</code>	45
Code hook into the <code>\name</code> section command	
<code>\memsecstarinfo{⟨name⟩}{⟨title⟩}</code>	45
Code hook into the <code>\name*</code> section command	
<code>\midbookskip</code>	8
Macro called between setting a book number and the title	
<code>\midpartskip</code>	1
Vertical space between a part number and the part title.	
<code>\midPoemTitleskip</code>	13
Vertical space between the number and text of a poem title.	
<code>\midsloppy</code>	5
Declaration for TeX to allow moderate interword space variations in justified text lines.	
<code>\begin{midsloppy}</code>	5
Typeset contents of the enclosed paragraph(s) using <code>\midsloppy</code> .	
<code>\newleadpage [⟨page-style⟩]{⟨cmdname⟩}{⟨title⟩}</code>	9
Creates a command <code>\cmdname</code> which typesets a part-like page with <code>⟨title⟩</code> and the <code>⟨page-style⟩</code> pagestyle, if given, otherwise it uses the empty pagestyle. The <code>⟨title⟩</code> is added to the ToC.	
<code>\newleadpage* [⟨page-style⟩]{⟨cmdname⟩}{⟨title⟩}</code>	9
Creates a command <code>\cmdname*</code> which typesets a part-like page with <code>⟨title⟩</code> and the <code>⟨page-style⟩</code> pagestyle, if given, otherwise it uses the empty pagestyle.	
<code>\newlistentry [⟨within⟩]{⟨cntr⟩}{⟨ext⟩}{⟨level-1⟩}</code>	2
Creates the commands necessary for typesetting an entry in the ‘List of ... for the file with extension <code>⟨ext⟩</code> . The entry uses the <code>⟨cntr⟩</code> (possibly <code>⟨within⟩</code>) counter and <code>⟨level-1⟩</code> is the entry’s level minus one.	
<code>\nobookblankpage</code>	8
Declaration for no blank page to be output after a book title page.	
<code>\noglossaryintoc</code>	24
Declaration to prohibit adding glossary title to the ToC.	
<code>\nopartblankpage</code>	9
Declaration for not following a part page with a blank one.	
<code>\notedivision</code>	28
Heading printed by the <code>\printnotes</code> and <code>\printnotes*</code> macros.	
<code>\noteentry{⟨notenum⟩}{⟨id⟩}{⟨text⟩}{⟨pagenum⟩}</code>	29
Typesets a pagenote with number <code>⟨notenum⟩</code> , identifier <code>⟨id⟩</code> , contents <code>⟨text⟩</code> , created on page <code>⟨pagenum⟩</code> .	
<code>\noteidinnotes{⟨notenum⟩}{⟨id⟩}</code>	29
Prints an endnote’s number or id in the endnote listing.	
<code>\noteinnotes{⟨text⟩}</code> Prints the text of an endnote.	30
<code>\notenuminnotes{⟨num⟩}</code> Prints an endnote’s number	29
<code>\notenuminnotes{⟨num⟩}</code>	29
Typesets the number <code>⟨num⟩</code> of a page note in the note listing.	
<code>\notenumintext{⟨num⟩}</code>	29
Typesets the number <code>⟨num⟩</code> of a page note in the main text.	

<code>\notepageref</code>	28
Declaration that page numbers are available to notes in the endnote listing.	
<code>\notesname</code> Name for endnotes, default Notes.	28
<code>\NumberPoemTitle</code>	12
Declaration for <code>\PoemTitle</code> to be numbered.	
<code>\onecolglossaryfalse</code> Declaration for a two column glossary.	25
<code>\onecolglossarytrue</code> Declaration for a single column glossary.	25
<code>\overridescapmargin{<margin>}</code>	37
A one-time override of <code>\sidecapmargin</code> .	
<code>\pageinnotes{<pagenum>}</code>	29
Controls the printing of an endnote's page reference number.	
<code>\pagenotesubhead{<chapapp>}{<num>}{<title>}</code>	30
Typesets a subheading for notes from chapter or appendix <code><chapapp></code> <code><num></code> called <code><title></code> .	
<code>\pagetofootnote</code>	30
Declaration which turns <code>\pagenotes</code> into <code>\footnotes</code> .	
<code>\partblankpage</code>	9
Declaration for following a part page with a blank one.	
<code>\partmark{<forhead>}</code>	9
For setting any marks containing the text of a header for a <code>\part</code> .	
<code>\patchcommand{<macro>}{<start-code>}{<end-code>}</code>	42
Inserts <code><start-code></code> before the current definition of the <code><macro></code> and <code>{<end-code>}</code> at the end of the current definition.	
<code>\PlainPoemTitle</code>	12
Declaration for <code>\PoemTitle</code> to be unnumbered.	
<code>\pnchap</code>	30
Chapter title for <code>\pagenotesubhead</code> . Defaults to the ToC entry.	
<code>\pnschap</code>	30
Starred chapter title for <code>\pagenotesubhead</code> . Defaults to the regular title.	
<code>\PoemTitle</code> [<code><fortoc></code>] [<code><forhead></code>] <code>{<title>}</code>	12
Typesets the title for a poem and puts it into the ToC.	
<code>\PoemTitle</code> [<code><fortoc></code>] [<code><forhead></code>] <code>{<title>}</code>	13
Typesets an unnumbered title for a poem but does not add it to the ToC.	
<code>\PoemTitlefont</code> Font for the text of a poem title	13
<code>\PoemTitleheadstart</code>	13
Called at the start of typesetting a <code>\PoemTitle</code> .	
<code>\poemtitlemark{<forhead>}</code>	12
Used to set marks for a <code>\PoemTitle</code> .	
<code>\PoemTitlenumfont</code> Font for the number of a poem title	13
<code>\poemtitlepstyle</code> Page style for a <code>\PoemTitle</code>	12
<code>\poemtitlestarmark{<forhead>}</code>	13
Used to set marks for a <code>\PoemTitle*</code> .	
<code>\poemtitlestarpstyle</code> Page style for a <code>\PoemTitle*</code>	13
<code>\postnoteinnotes</code>	29
Called by <code>\noteentry</code> to finish the printing of an endnote.	

<code>\preglossaryhook</code>	24
Vacuous macro called after a glossary title is typeset.	
<code>\prenoteinnotes</code>	29
Called by <code>\noteentry</code> to initialise the printing of an endnote.	
<code>\printbookname</code> Prints the <code>\bookname</code> using the <code>\booknamefont</code>	8
<code>\printbooknum</code> Prints the book number using the <code>\booknumfont</code>	8
<code>\printbooktitle{<title>}</code>	8
Prints the book title using the <code>\booktitlefont</code> .	
<code>\printglossary[<file>]</code>	19
Prints the glossary from file <code>\jobname.gls</code> , or <code>\file.gls</code>	
<code>\printpageinnotes{<pagenum>}</code>	29
Prints an endnote's page reference number.	
<code>\printpagenotes</code>	26
Input the pagenote ent file for printing, then close it to any more notes.	
<code>\printpagenotes*</code>	26
Input the pagenote ent file for printing, then empty it ready for further notes.	
<code>\printPoemTitlenonum</code>	13
Used instead of <code>\printPoemTitlenum</code> for an unnumbered <code>\PoemTitle</code> .	
<code>\printPoemTitlenum</code> Typesets the number for a <code>\PoemTitle</code>	13
<code>\printPoemTitletitle{<title>}</code>	13
Typesets the title of a <code>\PoemTitle</code> .	
<code>\renewleadpage[<page-style>]{<cmdname>}{<title>}</code>	9
Redefines the <code>\cmdname</code> (see <code>\newleadpage</code>).	
<code>\renewleadpage*[<page-style>]{<cmdname>}{<title>}</code>	9
Redefines the <code>\cmdname*</code> (see <code>\newleadpage*</code>).	
<code>\reparticle</code>	10
Makes chapter and lower level headings replicate the equivalent article class headings.	
<code>reparticle</code> Chapter style replicating a section in an article.	10
<code>\RequireAtEndClass{<class>}{<code>}</code>	43
Inserts <code><code></code> just after the <code><class></code> class is used, or immediately if <code><class></code> has already been used.	
<code>\RequireAtEndPackage{<pack>}{<code>}</code>	43
Inserts <code><code></code> just after the <code><pack></code> package is used, or immediately if <code><pack></code> has already been used.	
<code>\setsidcappos{<pos>}</code>	36
Declaration of the vertical position of a sidecaption with respect to the float.	
<code>\showtrimsoff</code>	41
If the <code>showtrims</code> option has been used this declaration switches the trim marks off.	
<code>\showtrimsoff</code>	41
If the <code>showtrims</code> option has been used this declaration switches the trim marks on. It is the default.	
<code>\sidebarmargin{<margin>}</code> Set the margin(s) for sidebars.	31
<code>\sidecapfloatwidth{<length>}</code>	37
Macro holding the width of a float with a sidecaption.	

<code>\sidecapmargin{<margin>}</code>	Sets the the margin for sidecaptions.	35
<code>\sidecapraise</code>	Vertical distance added to the default vertical placement of a sidecaption.	37
<code>\sidecapsep</code>	Length specifying the horizontal separation between a sidecaption and the float.	35
<code>\sidecapstyle</code>	Style settings for a sidecaption.	36
<code>\begin{sidecaption}[<fortoc>]{<title>}[<label>]</code>	Environment for setting a sidecaption.	35
<code>\sidecapwidth</code>	Length specifying the maximum width of a sidecaption.	35
<code>\begin{sidecontcaption}{<title>}[<label>]</code>	Environment for setting a continued sidecaption.	36
<code>\begin{sidelegend}{<title>}[<label>]</code>	Environment for setting a legend kind of sidecaption.	36
<code>\begin{sidenamedlegend}{<title>}[<label>]</code>	Environment for setting a named legend kind of sidecaption.	36
<code>\sidepar[<left>]{<right>}</code>	Typeset <i><right></i> in the outer margin or, if given, <i><left></i> in the left of verso pages.	31
<code>\sideparvshift</code>	Vertical adjustment for <code>\sidepar</code> marginal text.	31
<code>\sloppy</code>	Declaration for TeX to allow large interword space variations in justified text lines.	5
<code>\begin{sloppypar}</code>	Typeset contents of the enclosed paragraph(s) using <code>\sloppy</code>	5
<code>\subcaptionlabelfont{<fontspec>}</code>	The font for the identifier of a subcaption.	2
<code>\subcaptionref{<key>}</code>	Typesets a reference to the subcaption labelled with <i><key></i> using the <code>\subcaptionlabel</code> font.	2
<code>\subcaptionref*{<key>}</code>	Typesets a reference to the subcaption labelled with <i><key></i> using the normal font.	2
<code>\textheight</code>	The height of the textblock	3
<code>\begin{theglossary}</code>	Environment for typesetting a glossary.	20
<code>\thepoem</code>	Typeset the current Poem Title number	12
<code>\tightlists</code>	Declaration for removing vertical spacing from list-based environments. Half a <code>\onelineskip</code> may be added before and after.	33
<code>\tightlists*</code>	Declaration for removing vertical spacing from list-based environments. There is no additional space before or after.	33
<code>\topskip</code>	The height of the first line of text on a page. This is usually less than the <code>\baselineskip</code>	3
<code>\begin{verbatimoutput}{<file>}</code>	Environment whose contents will be written verbatim to the <i><file></i> file.	2

<code>\verselinenumbersleft</code>	33
Declaration to have any verse line numbers set to the left of the lines.	
<code>\verselinenumbersright</code>	33
Declaration to have any verse line numbers set in the righthand margin. This is the default.	
<code>\vleftmargin</code> Minimum left margin for verse.	32
<code>\vleftofline{<text>}</code> Puts <text> at the left of a line of verse.	32
<code>\begin{vplace} [<num>]</code>	41
The contents of this environment are centered vertically. The optional <num> argument can be used to specify the ratio of the upper space to the lower space.	
<code>\begin{writeverbatim}{<stream>}</code>	2
Environment whose contents will be written verbatim to the <stream> output stream.	
<code>\xindyindex</code>	19
Declaration to be put in the preamble when the xindy program will be used to process the raw index.	

Bibliography

- [CB99] Warren Chappell and Robert Bringhurst. *A Short History of the Printed Word*. Hartley & Marks, 1999, ISBN 0-88179-154-7.
- [CH88] Pehong Chen and Michael A. Harrison. *Index Preparation and processing*. September 1988. (Available from CTAN in `indexing/makeindex/paper`)
- [Dow00] Michael J. Downes. *The patchcmd package*. July 2000. (Available from CTAN in `macros/latex/contrib/patchcmd`)
- [CMS93] John Grossman (Managing Editor). *The Chicago Manual of Style*. Fourteenth Edition. University of Chicago Press, 1993.
- [MG⁺04] Frank Mittelbach, Michael Goossens, et al. *The LaTeX companion: Second Edition*. Addison-Wesley, 2004. ISBN 0-201-36299-6.

Index

The first page number is usually, but not always, the primary reference to the indexed topic.

`\@wrglom@`, 23
`\@dblfpbot`, 39
`\@dblftop`, 39
`\@fpbot`, 39
`\@fptop`, 39

`abstract` (package), 42
`\addtonotes`, 30
`afterbook` (pagestyle), 8
`\afterbookskip`, 8
`afterpart` (pagestyle), 9
`\afterpartskip`, 1
`\afterPoemTitle`, 13, 14
`\afterPoemTitlenum`, 13, 14
`\afterPoemTitleskip` (length), 13
`appendix` (package), 42
`\appendixname`, 30
`array` (package), 42
Arseneau, Donald, 42
`article` (class), 10
`article` (option), xiii, 10
`\AtBeginClass`, 43
`\AtBeginDocument`, 43
`\AtBeginFile`, 43
`\AtBeginPackage`, 43
`\AtEndClass`, 43, 44
`\AtEndDocument`, 43
`\AtEndFile`, 43
`\AtEndOfClass`, 43, 44
`\AtEndOfPackage`, 43, 44
`\AtEndPackage`, 43, 44

`\atendtheglossaryhook`, 25

`\backmatter`, xiii
`\baselineskip` (length), 3, 4
`\beforebookskip`, 8
`\beforepartskip`, 1
`\beforePoemTitleskip` (length), 13
`\begintheglossaryhook`, 25
`\book`, 7, 8, 17, 45
`book` (pagestyle), 8
`\book*`, 7, 45
`\bookblankpage`, 8
`\bookmark`, 7
`\bookname`, 8
`\booknamefont`, 8
`\booknamenum`, 8
`\booknumfont`, 8
`booktabs` (package), 42
`\booktitlefont`, 8
`\bvendofpage`, 31

`calc` (package), 3
`\caption`, 25, 45
`\captionnamefont`, 37
`\captiontitlefinal`, 38
`ccaption` (package), 42
`\centering`, 14, 38
`\cftappendixname`, 15
`\cftbeforebookskip` (length), 18
`\cftbookaftersnum`, 18
`\cftbookaftersnumb`, 18

- `\cftbookdotsep`, 18
- `\cftbookfont`, 18
- `\cftbookindent (length)`, 18
- `\cftbookleader`, 18
- `\cftbookname`, 17
- `\cftbooknumwidth (length)`, 18
- `\cftbookpagefont`, 18
- `\cftbookpresnum`, 18
- `\cftchaptername`, 15
- `\cftfigurename`, 16
- `\cftparagraphname`, 16
- `\cftpartname`, 15
- `\cftsectionname`, 16
- `\cftsubparagraphname`, 16
- `\cftsubsectionname`, 16
- `\cftsubsubsectionname`, 16
- `\cfttablename`, 16
- `\changeGLOSSACTUAL`, 23
- `\changeGLOSSNUM`, 23, 24
- `\changeGLOSSNUMFORMAT`, 23, 24
- `\changeGLOSSREF`, 23
- chappell* (chapterstyle), 11
- `\chapter`, 10, 12, 15, 45
- `\chapter*`, 45
- chapterbib (package), xiii
- `\chaptername`, 30
- chapterstyle,
 - chappell*, 11
 - demo*, 11, 12
 - demo2*, 12
 - reparticle*, 10
 - southall*, 10
- `\checkandfixthelayout`, 3
- `\checkthelayout`, 3
- chngcntr (package), 42
- class, xiii
 - article, 10
 - memoir, xiii, xv, xvii
- Connes, Frederic, 7, 18
- `\continuousnotenums`, 28
- counter,
 - pagenote, 28, 29
- crop (package), 42
- dcolumn (package), 42
- addtonotes, 30
- `\defaultlists`, 33, 34
- delarray (package), 42
- demo* (chapterstyle), 11, 12
- demo2* (chapterstyle), 12
- description (environment), 33
- `\DisemulatePackage`, 42
- document (environment), 43
- Downes, Michael, 42
- Dye, Thomas, 10
- Els, Danie, 9, 15
- `\em`, xv, 39
- `\emindershape`, 39
- `\emph`, xv, 39
- empty* (pagestyle), 8, 9
- `\EmulatedPackage`, 42
- `\EmulatedPackageWithOptions`, 42
- `\endnote`, 26
- endnotes (package), 26
- ent (file), 26–30
- enumerate (environment), 33
- enumerate (package), 42
- environment,
 - description, 33
 - document, 43
 - enumerate, 33
 - figure, 39
 - itemize, 33
 - list, 33
 - midsloppypar, 5
 - minipage, 37, 38
 - picture, 15
 - sidecaption, 35, 36, 38
 - sidecontcaption, 36
 - sidelegend, 36
 - sidenamedlegend, 36
 - sloppypar, 5
 - table, 39
 - tabular, 38
 - thebibliography, 24
 - theglossary, 20, 24
 - theindex, 24
 - verbatimoutput, 2
 - verse, 32, 33
 - vplace, 41
 - writeverbatim, 2
- epigraph (package), 42
- `\everylistparindent (length)`, 33

- fancyhdr (package), 42
figure (environment), 39
file,
 ent, 26–30
 log, 42
 toc, 26
\firmlists, 33
\firmlists*, 33
\flushbottom, 3
\footnote, 26, 27, 30
\footnoterule, 39
\foottopagenote, 30
Ford, Matthew, 32
framed (package), 42
\frontmatter, 45
\fussy, 5
\futurelet, 42

geometry (package), 42
\glossary, 19, 20, 23, 26
\glossarycolsep (length), 25
\glossaryentry, 23
\glossaryintoc, 24
\glossarymark, 24
\glossaryname, 24
\glossaryrule (length), 25
\glossitem, 20, 24, 25

\hrule, 31
hyperref (package), 19, 44
Høgholm, Morten, 3

sidecapfloatwidth, 37
\idtextinnotes, 29
\ifanappendix, 1
ifmtarg (package), 42
ifpdf (package), 42
\ifscapmargleft, 35, 36
\include, 43
index (package), 42

\InputIfFileExists, 44
itemize (environment), 33

\kern, 31

\large, 13
\leadpagetoclevel, 9

\leftmargini (length), 32, 33
\legend, 45
length,
 \afterPoemTitleskip, 13
 \baselineskip, 3, 4
 \beforePoemTitleskip, 13
 \cftbeforebookskip, 18
 \cftbookindent, 18
 \cftbooknumwidth, 18
 \everylistparindent, 33
 \glossarycolsep, 25
 \glossaryrule, 25
 \leftmargini, 32, 33
 \linewidth, 37
 \listparindent, 33
 \marginparsep, 35
 \marginparwidth, 35
 \midPoemTitleskip, 13
 \onelineskip, 33
 \parindent, 33
 \sidecapraise, 37
 \sidecapsep, 35, 37
 \sidecapwidth, 35
 \sideparvshift, 31
 \textheight, 3, 4
 \topskip, 3, 4
 \vleftmargin, 32, 33
\linewidth (length), 37
list (environment), 33
\listparindent (length), 33
\LoadClass, 43
log (file), 42
Luecking, Daniel, 31

Madsen, Lars, 10, 16
\mainmatter, xiii
\makeglossary, 19, 24
makeidx (package), 42
MakeIndex (program), 18–21, 23, 26
\makepagenotes, 26
\maketitle, 15
\marginpar, xiii, 31
\marginparsep (length), 35
\marginparwidth (length), 35
\memappchapinfo, 45
\memappchapstarinfo, 45
\memapppageinfo, 45

\memappagestarinfo, 45
\membicaptioninfo, 46
\membionenumcaptioninfo, 46
\membitwonumcaptioninfo, 46
\membookinfo, 45
\membookstarinfo, 45
\memcaptioninfo, 46
\memchapinfo, 45
\memchapstarinfo, 45
\memglodesc, 23
\memglonum, 23
\memgloref, 23
\memgloterm, 23
\memgobble, 41
memhfixc (package), 44
\memhyperindexfalse, 18, 19
\memjustarg, 41
\memleadpageinfo, 45
\memleadpagestarinfo, 45
\memlegendinfo, 46
\memnamedlegendinfo, 46
memoir (class), xiii, xv, xvii
\mempartinfo, 45
\mempartstarinfo, 45
\mempoeminfo, 45
\mempoemstarinfo, 45
\memPoemTitleinfo, 45
\memPoemTitlestarinfo, 45
\memsecinfo, 45
\memsecstarinfo, 45
\midbookskip, 8
\midpartskip, 1
\midPoemTitleskip (length), 13
\midslippy, 5
midsloppypar (environment), 5
minipage (environment), 37, 38
moreverb (package), 42

natbib (package), xiii
needspace (package), 42
\newcommand, 9
newfile (package), 42
\newleadpage, 9
\newleadpage*, 9
\newlistentry, 2, 16
\newsubfloat, 16
nextpage (package), 42

\nobookblankpage, 8
\noglossaryintoc, 24
\nopartblankpage, 9
\notedivision, 28, 30
\noteentry, 29
\noteidinnotes, 29
\noteinnotes, 30
\notenuminnotes, 29
\notenumintext, 29
\notepageref, 28, 29
\notesname, 28
\NumberPoemTitle, 12

PoemTitle, 13
poemtitle, 12
\onecolglossaryfalse, 25
\onecolglossarytrue, 25
\onelineskip (length), 33
oneside (option), 4
option,
 article, xiii, 10
 oneside, 4
 sectionbib, xiii
 showtrims, 41
 twoside, 4, 5
\overridescapmargin, 37

package,
 abstract, 42
 appendix, 42
 array, 42
 booktabs, 42
 calc, 3
 ccaption, 42
 chapterbib, xiii
 chngcntr, 42
 crop, 42
 dcolumn, 42
 delarray, 42
 endnotes, 26
 enumerate, 42
 epigraph, 42
 fancyhdr, 42
 framed, 42
 geometry, 42
 hyperref, 19, 44
 ifmtarg, 42

ifpdf, 42
 index, 42
 makeidx, 42
 memhfixc, 44
 moreverb, 42
 natbib, xiii
 needspace, 42
 newfile, 42
 nextpage, 42
 pagenote, 26, 42
 patchcmd, 42
 shortvrb, 42
 showidx, 42
 sidecap, 42
 subfigure, 42
 tabularx, 42
 titleref, 42
 titlesec, 42
 tocbibind, 42
 tocloft, 42
 verbatim, 42
 verse, 42
 wrapfig, 38
 \pageinnotes, 29
 \pagenote, 26, 27, 29, 30
 pagenote (counter), 28, 29
 pagenote (package), 26, 42
 \pagenotesubhead, 30
 pagestyle,
 afterbook, 8
 afterpart, 9
 book, 8
 empty, 8, 9
 \pagetofootnote, 30
 \parindent (length), 33
 \part, 7–9, 15, 45
 \part*, 45
 \partblankpage, 9
 \partmark, 9
 patchcmd (package), 42
 \patchcommand, 42
 picture (environment), 15
 \PlainPoemTitle, 12
 \pnchap, 30
 \pnschap, 30
 \PoemTitle, 12, 45
 \poemtitle, 45
 \PoemTitle*, 13
 \PoemTitlefont, 13
 \PoemTitleheadstart, 13
 \poemtitlemark, 12
 \PoemTitlenumfont, 13
 \poemtitlepstyle, 12
 \poemtitlestarmark, 13
 \poemtitlestarpstyle, 13
 \postnoteinnotes, 29
 \preglossaryhook, 24
 \prenoteinnotes, 29
 \printbookname, 8
 \printbooknum, 8
 \printbooktitle, 8
 \printglossary, 19, 26
 \printnotes, 28
 \printnotes*, 28, 30
 \printpageinnotes, 29
 \printpagenotes, 26, 27
 \printpagenotes*, 26, 27
 \printPoemTitlenonum, 13, 14
 \printPoemTitlenum, 13
 \printPoemTitledtitle, 13, 14
 program,
 MakelIndex, 18–21, 23, 26
 xindy, 18, 19
 \protect, 30
 \raggedbottom, 39
 \raggedleft, 38
 \raggedright, 38
 \renewcommand*, 37
 \renewleadpage, 9
 \renewleadpage*, 9
 \reparticle, 10
reparticle (chapterstyle), 10
 \RequireAtEndClass, 43, 44
 \RequireAtEndPackage, 43
 \scapmargleftfalse, 35
 \scapmarglefttrue, 35
 \section, 10, 45
 sectionbib (option), xiii
 \setlength, 37
 \setsidecappos, 36
 shortvrb (package), 42
 showidx (package), 42

showtrims (option), 41
 \showtrimsoff, 41
 \showtrimson, 41
 \sidebarmargin, 31
 \sidebaronesidefalse, 31
 \sidebaronesidetrue, 31
 sidecap (package), 42
 \sidecapfloatwidth, 37
 \sidecapmargin, 35, 37
 \sidecapraise (length), 37
 \sidecapsep (length), 35, 37
 \sidecapstyle, 36
 sidecaption (environment), 35, 36, 38
 \sidecapwidth (length), 35
 sidecontcaption (environment), 36
 sidelegend (environment), 36
 sidenamedlegend (environment), 36
 \sidepar, 31
 \sideparvshift (length), 31
 \sloppy, 5
 sloppypar (environment), 5
 southall (chapterstyle), 10
 \space, 8
 \strictpagechecktrue, 37
 \subcaptionlabelfont, 2
 \subcaptionref, 2
 \subcaptionref*, 2
 subfigure (package), 42
 \subsection, 45

 table (environment), 39
 \tableofcontents, xiii
 tabular (environment), 38
 tabularx (package), 42
 \textheight (length), 3, 4
 thebibliography (environment), 24
 \thebook, 8
 theglossary (environment), 20, 24
 theindex (environment), 24
 \thepoem, 12
 \tightlist*, 33
 \tightlists, 33, 34
 \tightlists*, 33
 titleref (package), 42
 titlesec (package), 42
 toc (file), 26
 tocbibind (package), 42

 tocloft (package), 42
 \topskip (length), 3, 4
 twoside (option), 4, 5

 \usepackage, 42, 43

 verbatim (package), 42
 verbatimoutput (environment), 2
 verse (environment), 32, 33
 verse (package), 42
 \verselinenumbersleft, 33
 \verselinenumbersright, 33
 \vfill, 39
 \vleftmargin (length), 32, 33
 \vleftofline, 32
 vplace (environment), 41

 wrapfig (package), 38
 writeverbatim (environment), 2

 xindy (program), 18, 19
 \xindyindex, 19

 Zoonekynd, Vincent, 15