

Addendum

ADDENDUM

The Memoir Class

for

Configurable Typesetting

User Guide

Peter Wilson

THP
The Herries Press

© 2002–2005, 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 16 15 14 13 12

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

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 ·	i
Contents ·	ii
List of Tables ·	iv
Introduction to Edition 1 ·	v
Introduction to Edition 2 ·	vii
Acknowledgements ·	ix
1 Corrections and omissions ·	1
2 Document divisions ·	3
3 Front and rear ·	7
4 Boxes and environments ·	17
5 Captions ·	21
6 Miscellaneous ·	25
Command summary ·	31
Bibliography ·	37
Index ·	38

Contents

Short contents	i
Contents	ii
List of Tables	iv
Introduction to Edition 1	v
Introduction to Edition 2	vii
Acknowledgements	ix
1 Corrections and omissions	1
1.1 Writing out verbatim	1
2 Document divisions	3
2.1 Part headings	3
2.2 Poem Titles	4
<i>Main Poem Title layout parameters 4, Detailed Poem Title layout parameters 5</i>	
3 Front and rear	7
3.1 The ToC	7
3.2 The index	9
3.3 Glossaries	9
3.4 Controlling the glossary	10
<i>Setting up makeindex 10, Raw input data 12, The listing 15, The glossary for this document 15</i>	
4 Boxes and environments	17
4.1 Sidebars	17
4.2 Boxed verbatims	17
4.3 Verse	17
5 Captions	21
5.1 Side captions	21
<i>Tweaks 22</i>	

6	Miscellaneous	25
6.1	General	25
	<i>Font commands 25, Footnotes 25, Floats 25, Two simple macros 27</i>	
6.2	For package writers	27
	<i>Emulating packages 27, Extending a macro 28</i>	
6.3	Heading hooks	28
	Command summary	31
	Bibliography	37
	Index	38

List of Tables

3.1	MakeIndex configuration file input parameters	12
3.2	MakeIndex configuration file output parameters	13
5.1	Permitted arguments for some sidecaption related commands	24

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

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, Donald Arseneau, Stephan von Bechtolsheim, Jens Berger, Karl Berry, Javier Bezos, Sven Bovin, Ignasi Furió Caldenty, Ezequiel Martín Camara, David Carlisle, Gustavo Cevolani, Michael A. Cleverly, Steven Douglas Cochran, Frederic Connes, Michael W. Daniels, Michael Downes, Victor Eijkhout, Danie Els, Robin Fairbairns, Simon Fear, Kai von Fintel, Matthew Ford, Musa Furber, Daniel Richard G., Ignacio Fernández Galván, Romano Giannetti, Gabriel Guernik, Kathryn Hargreaves, Sven Hartrumpf, hazydirk, Carsten Heinz, Florence Henry, Peter Heslin, Lars Hoemke, Urs Hofer, Morton Høgholm, Hendrik Holm, Vladimir Ivanovic, Stefan Kahrs, Jøgen 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, Steve Peter, Chris Rowley, Bernd Raichle, Robert Schlicht, Arnaud Schmittbuhl, Rainer Schöpf, Paul Stanley, James Szinger, Jens Taprogge, Reuben Thomas, Bastiaan Niels Veelo, Emanuele Vicentini, Jürgen Vollmer, and possibly others. 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 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.

<pre>\begin{verbatimoutput}{\langle file \rangle} material \end{verbatimoutput} \begin{writeverbatim}{\langle stream \rangle} material \end{writeverbatim}</pre>
--

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

Two

Document divisions

2.1 Part headings

`\partmark{<part-title>}`

The `\part` code now includes `\partmark` 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]{}
```

```
\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.

2.2 Poem Titles

The new command `\PoemTitle` provides more flexibility in typesetting titles of poems that 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}
```

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}
```

2. DOCUMENT DIVISIONS

Three

Front and rear

3.1 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>\cftpdbname \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:

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

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}
```

¹Plus the recoding of some internal macros.

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\oldchangelocdepth\changelocdepth
\let\oldcftchapterfillnum\cftchapterfillnum
\renewcommand{\changelocdepth}[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

```

3.2 The index

The MakeIndex 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.

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.

3.3 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 MakeIndex 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 descptions.}
```

The glossary facilities are designed so that the MakeIndex 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.

3.4 Controlling the glossary

SETTING UP MAKEINDEX

If you just run MakeIndex on a `glo` file you will get lots of errors; MakeIndex 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 `MakeIndex` 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{}` could 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.

A complete listing of the possible entries in a configuration file, also called a style file, for `MakeIndex` is in Table 3.1 and 3.2 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 "\\n\\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 3.1: 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

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[6]{<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}}{\memgloref{ref}}
{\memglonumf}{num}}
```

which MakeIndex then effectively converts into

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

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

Table 3.2: 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.

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{\memgloref}[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).

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

`\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 *<thecounter>* should be used to generate the *<num>* for the *<file>* glossary. It is initially `\thepage`.

`\changeGLOSSnumformat` specifies that *<format>* should be used to format the *<num>* for the *<file>* glossary. The format of *<format>* 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 *<num>* 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.

Firstly the code that is in the preamble.

```

%%% in the preamble
\makeglossary
\changeGlossActual{?}
\changeGlossNum{\thepage}
\changeGlossNumFormat{\hyperpage}%% for hyperlinks
\renewcommand*{\glossaryname}{Command summary}

```

```

\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
\renewcommand*{\glossarymark}{\markboth{\glossaryname}{}}

```

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}%

```

Any change to the glossary entries will be reflected in the `glo` produced from that LaTeX run. Then run `MakeIndex` 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`.

Four

Boxes and environments

4.1 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 `\sidebaronesidettrue` are equivalent.) There are no positioning options for twocolumn documents.

4.2 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.

4.3 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 $\langle text \rangle$ of `\vleftoffline` 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:
\begin{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 ’\,\,’
\end{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:
\begin{verse}
\vleftoffline{‘\,\,\,’} ‘‘In the last days,’’ God says, \\\
\vleftoffline{‘\,\,\,’} ‘‘I will pour out my Spirit on all people. \\\
Your sons and daughters will prophesy, \\\
\ldots \\\
And everyone who calls \ldots ’\,\,’
\end{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 ...’’”
```

<code>\vleftmargin</code>

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 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}
```

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

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

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

```
\setlength{\leftmargin}{\leftmargin}
```


Five

Captions

5.1 Side captions

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

```
\begin{sidecaption}[\langle fortoc \rangle]{\langle title \rangle}[\langle label \rangle]  
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{\langle margin \rangle}  
\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

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

5. CAPTIONS

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.

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}

```

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 5.1 was created by the following code.

```

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

```

Table 5.1: Permitted arguments for some sidecaption related commands

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

```
\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.

Six

Miscellaneous

6.1 General

FONT COMMANDS

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

`\emminnershape{<shape>}`

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

```
\newcommand*{\emminnershape}{\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 `\@dblfpptop` 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@
      \@tempswafalse
      \@flcheckspace \@toproom \@toplist
      \if@tempswa
        \@bitor\@currtype{\@midlist\@botlist}%
        \if@test
          \else
            \@flupdates \@topnum \@toproom \@toplist
            \@inserttrue
          \global\floatattoptrue
          \fi
        \fi
      \fi
    \fi
  \if@insert
    \else
      \@addtobot
    \fi}

\renewcommand*{\@addtobot}{%
  \@getfpsbit 4\relax
  \ifodd \@tempcnta
    \@flsetnum \@botnum
    \ifnum \@botnum>\z@
      \@tempswafalse
      \@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

\floatattop 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.

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

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.

6.2 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

<pre> \EmulatedPackage{<package>}[<date>] \EmulatedPackageWithOptions{<optionlist>}{<package>}[<date>] </pre>

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.)

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.

6.3 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 `\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>\afterPoemTitle</code>	Called after printing the title of a <code>\PoemTitle</code> .	5
<code>\afterPoemTitlenum</code>	Called after printing the number of a <code>\PoemTitle</code> .	5
<code>\afterPoemTitleskip</code>	Vertical space after a poem title	5
<code>\atendtheglossaryhook</code>	Vacuous macro called as the first thing by <code>\end{theglossary}</code> .	15
<code>\beforePoemTitleskip</code>	Vertical space before a poem title.	5
<code>\begintheglossaryhook</code>	Vacuous macro called as the last thing by <code>\begin{theglossary}</code> .	15
<code>\bvendofpage{<code>}</code>	Controls the appearance of a page break in a boxed verbatim.	17
<code>\cftappendixname</code>	In the ToC, called before the number of a chapter forming an appendix.	7
<code>\cftchaptername</code>	In the ToC, called before the number of a chapter.	7
<code>\cftpartname</code>	In the ToC, called before the number of a part.	7
<code>\changeglossactual[<file>]{<char>}</code>	Specifies <code><char></code> as the actual character for glossary <code><file></code> .	14
<code>\changeglossnum[<file>]{<thecounter>}</code>	Specifies <code><thecounter></code> as the <code><num></code> for glossary <code><file></code> .	14
<code>\changeglossnumformat[<file>]{<format>}</code>	Specifies <code><format></code> as the format for <code><num></code> for glossary <code><file></code> .	14
<code>\changeglossref[<file>]{<thecounter>}</code>	Specifies <code><thecounter></code> as the <code><ref></code> for glossary <code><file></code> .	14
<code>\emminnershape{<shape>}</code>	Font shape for emphasized text within emphasized text.	25
<code>\EmulatedPackage{<package>}[<date>]</code>	Claim that the <code><package></code> package has been loaded.	27
<code>\EmulatedPackageWithOptions{<optionlist>}{<package>}[<date>]</code>	Claim that the <code><package></code> package has been loaded with options <code><optionlist></code> .	27
<code>\glossary[<file>](<key>){<term>}{<description>}</code>	Adds <code><term></code> and its description, <code><desc></code> , to a glossary file — <code>\jobname.glo</code> by default	10

or to <code>\file.glo</code> . The optional argument <code><key></code> can be used to provide a different sort key for <code><term></code> .	
<code>\glossarycolsep</code> Columns separation in a two column glossary.	15
<code>\glossaryintoc</code> Declaration to add glossary title to the ToC.	15
<code>\glossarymark</code> Redefine to specify marks for headers.	15
<code>\glossaryname</code> Name for a glossary.	15
<code>\glossaryrule</code>	15
Width of inter-column rule in a two column glossary.	
<code>\glossitem{<term>}{<desc>}{<ref>}{<num>}</code>	10
Glossary entry used in a <code>theglossary</code> environment	
 <code>\ifsidecapleft</code>	 21
true if sidecaptions will be set in the left margin, otherwise they will be set in the right margin.	
 <code>\leadpagetoclevel{<sec>}</code>	 4
The ToC style for an entry from a <code>\newleadpage</code> generated command.	
 <code>\makeglossary[<file>]</code>	 10
Opens file <code>\jobname.glo</code> , or <code>\file.glo</code> , for glossary entries	
<code>\memappchapinfo{<thechapter>}{<fortoc>}{<forhead>}{<title>}</code>	28
Code hook into an appendix <code>\chapter</code>	
<code>\memappchapstarinfo{<fortoc>}{<title>}</code>	28
Code hook into an appendix <code>\chapter*</code>	
<code>\memapppageinfo{<title>}</code> Code hook into <code>\appendixpage</code>	29
<code>\memapppagestarinfo{<title>}</code>	29
Code hook into <code>\appendixpage*</code> .	
<code>\membicaptioninfo{<type>}{<thetype>}{<fortoc1>}{<title1>}{<name2>}{<title2>}</code> ..	29
Code hook into <code>\membicaption</code> .	
<code>\membionenumcaptioninfo{<type>}{<thetype>}{<fortoc1>}{<title1>}{<name2>}{<fortoc2>}{<title2>}</code>	29
Code hook into <code>\membionenumcaption</code> .	
<code>\membitwonumcaptioninfo{<type>}{<thetype>}{<fortoc1>}{<title1>}{<name2>}{<fortoc2>}{<title2>}</code>	29
Code hook into <code>\membitwonumcaption</code> .	
<code>\memcaptioninfo{<type>}{<thetype>}{<fortoc>}{<title>}</code>	29
Code hook into <code>\caption</code>	
<code>\memchapinfo{<thechapter>}{<fortoc>}{<forhead>}{<title>}</code>	28
Code hook into <code>\chapter</code>	
<code>\memchapstarinfo{<fortoc>}{<title>}</code> Code hook into <code>\chapter*</code>	28
<code>\memglodesc{<desc>}</code> Wrapper round a glossary description.	14
<code>\memglonum{<num>}</code> Wrapper round glossary numbers.	14
<code>\memgloref{<ref>}</code> Wrapper round a glossary ref.	14
<code>\memgloterm{<term>}</code> Wrapper round a glossary term.	14
<code>\memgobble{<text>}</code> Gobbles its argument. Do <i>not</i> redefine it.	27
<code>\memhyperindexfalse</code>	9
Preamble command to prevent hyperindexing.	

<code>\memjustarg{<text>}</code>	Definition is just <text>. Do not redefine it.	27
<code>\memleadpageinfo{<pstyle>}{<cmdname>}{<title>}</code>	Code hook into <code>\newleadpage</code> and <code>\renewleadpage</code> .	29
<code>\memleadpageinfo{<pstyle>}{<cmdname>}{<title>}</code>	Code hook into <code>\newleadpage*</code> and <code>\renewleadpage*</code> .	29
<code>\memlegendinfo{<title>}</code>	Code hook into <code>\legend</code>	29
<code>\memnamedlegendinfo{<fortoc>}{<title>}</code>	Code hook into <code>\namedlegend</code>	29
<code>\mempartinfo{<thepart>}{<fortoc>}{<title>}</code>	Code hook into <code>\part</code>	28
<code>\mempartatarinfo{<title>}</code>	Code hook into <code>\part*</code>	28
<code>\mempoeminfo{<title>}</code>	Code hook into <code>\poemtitle</code>	29
<code>\mempoemstarinfo{<title>}</code>	Code hook into <code>\poemtitle*</code>	29
<code>\memPoemTitleinfo{<thepoem>}{<fortoc>}{<forhead>}{<title>}</code>	Code hook into <code>\PoemTitle</code>	29
<code>\memPoemTitlestarinfo{<fortoc>}{<title>}</code>	Code hook into <code>\PoemTitle*</code>	29
<code>\memsecinfo{<name>}{<thename>}{<fortoc>}{<forhead>}{<title>}</code>	Code hook into the <code>\name</code> section command	28
<code>\memsecstarinfo{<name>}{<title>}</code>	Code hook into the <code>\name*</code> section command	28
<code>\midPoemTitleskip</code>	Vertical space between the number and text of a poem title.	5
<code>\newleadpage[<page-style>]{<cmdname>}{<title>}</code>	Creates a command <code>\cmdname</code> which typesets a part-like page with <title> and the <page-style> pagestyle, if given, otherwise it uses the empty pagestyle. The <title> is added to the ToC.	3
<code>\newleadpage*[<page-style>]{<cmdname>}{<title>}</code>	Creates a command <code>\cmdname*</code> which typesets a part-like page with <title> and the <page-style> pagestyle, if given, otherwise it uses the empty pagestyle.	3
<code>\noglossaryintoc</code>	Declaration to prohibit adding glossary title to the ToC.	15
<code>\NumberPoemTitle</code>	Declaration for <code>\PoemTitle</code> to be numbered.	4
<code>\onecolglossaryfalse</code>	Declaration for a two column glossary.	15
<code>\onecolglossarytrue</code>	Declaration for a single column glossary.	15
<code>\overridescapmargin{<margin>}</code>	A one-time override of <code>\sidecapmargin</code> .	23
<code>\partmark{<forhead>}</code>	For setting any marks containing the text of a header for a <code>\part</code> .	3
<code>\patchcommand{<macro>}{<start-code>}{<end-code>}</code>	Inserts <start-code> before the current definition of the <macro> and {<end-code>} at the end of the current definition.	28
<code>\PlainPoemTitle</code>	Declaration for <code>\PoemTitle</code> to be unnumbered.	4

<code>\PoemTitle[<i><fortoc></i>][<i><forhead></i>]{<i><title></i>}</code>	4
Typesets the title for a poem and puts it into the ToC.	
<code>\PoemTitle[<i><fortoc></i>][<i><forhead></i>]{<i><title></i>}</code>	4
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	5
<code>\PoemTitleheadstart</code>	5
Called at the start of typesetting a <code>\PoemTitle</code> .	
<code>\poemtitlemark{<i><forhead></i>}</code> Used to set marks for a <code>\PoemTitle</code>	4
<code>\PoemTitlenumfont</code> Font for the number of a poem title	5
<code>\poemtitlepstyle</code> Page style for a <code>\PoemTitle</code>	4
<code>\poemtitlestarmark{<i><forhead></i>}</code>	4
Used to set marks for a <code>\PoemTitle*</code> .	
<code>\poemtitlestarpstyle</code> Page style for a <code>\PoemTitle*</code>	4
<code>\preglossaryhook</code>	15
Vacuous macro called after a glossary title is typeset.	
<code>\printglossary[<i><file></i>]</code>	10
Prints the glossary from file <code>\jobname.gls</code> , or <code>\file.gls</code>	
<code>\printPoemTitlenonum</code>	5
Used instead of <code>\printPoemTitlenum</code> for an unnumbered <code>\PoemTitle</code> .	
<code>\printPoemTitlenum</code> Typesets the number for a <code>\PoemTitle</code>	5
<code>\printPoemTitletitle{<i><title></i>}</code>	5
Typesets the title of a <code>\PoemTitle</code> .	
 <code>\renewleadpage[<i><page-style></i>]{<i><cmdname></i>}{<i><title></i>}</code>	3
Redefines the <code>\cmdname</code> (see <code>\newleadpage</code>).	
<code>\renewleadpage*[<i><page-style></i>]{<i><cmdname></i>}{<i><title></i>}</code>	3
Redefines the <code>\cmdname*</code> (see <code>\newleadpage*</code>).	
 <code>\setsidcappos{<i><pos></i>}</code>	22
Declaration of the vertical position of a sidecaption with respect to the float.	
<code>\sidebarmargin{<i><margin></i>}</code> Set the margin(s) for sidebars.	17
<code>\sidecapfloatwidth{<i><length></i>}</code>	23
Macro holding the width of a float with a sidecaption.	
<code>\sidecapmargin{<i><margin></i>}</code> Sets the the margin for sidecaptions.	21
<code>\sidecapraise</code>	23
Vertical distance added to the default vertical placement of a sidecaption.	
<code>\sidecapsep</code>	21
Length specifying the horizontal separation between a sidecaption and the float.	
<code>\sidecapstyle</code> Style settings for a sidecaption.	22
<code>\begin{sidecaption}[<i><fortoc></i>]{<i><title></i>}[<i><label></i>]</code>	21
Environment for setting a sidecaption.	
<code>\sidecapwidth</code>	21
Length specifying the maximum width of a sidecaption.	
<code>\begin{sidecontcaption}{<i><title></i>}[<i><label></i>]</code>	22
Environment for setting a continued sidecaption.	
<code>\begin{sidelegend}{<i><title></i>}[<i><label></i>]</code>	22
Environment for setting a legend kind of sidecaption.	

<code>\begin{sidenamedlegend}{\langle title \rangle}[\langle label \rangle]</code>	22
Environment for setting a named legend kind of sidecaption.	
<code>\begin{theglossary}</code> Environment for typesetting a glossary.	10
<code>\thepoem</code> Typeset the current Poem Title number	4
<code>\begin{verbatimoutput}{\langle file \rangle}</code>	1
Environment whose contents will be written verbatim to the <i>\langle file \rangle</i> file.	
<code>\vleftmargin</code> Minimum left margin for verse.	18
<code>\vleftoffline{\langle text \rangle}</code> Puts <i>\langle text \rangle</i> at the left of a line of verse.	17
<code>\begin{writeverbatim}{\langle stream \rangle}</code>	1
Environment whose contents will be written verbatim to the <i>\langle stream \rangle</i> output stream.	

Bibliography

- [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`)
- [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@m, 12
- \@dblfpbot, 26
- \@dblfpbot, 26
- \@fpbot, 26
- \@fptop, 26

- \afterPoemTitle, 4, 5
- \afterPoemTitlenum, 4, 5
- \afterPoemTitleskip (length), 5
- Arseneau, Donald, 27
- article (option), v
- \atendtheglossaryhook, 15

- \backmatter, v
- \beforePoemTitleskip (length), 5
- \begintheglossaryhook, 15
- \bvendofpage, 17

- \caption, 16, 29
- \captionnamefont, 23
- \centering, 5, 24
- \cftappendixname, 7
- \cftchaptername, 7
- \cftpartname, 7
- \changeGLOSSactual, 14
- \changeGLOSSnum, 14
- \changeGLOSSnumformat, 14
- \changeGLOSSref, 14
- \chapter, 4, 7, 28
- \chapter*, 28
- chapterbib (package), v
- class, v
 - memoir, v, vii, ix, 9, 11, 18, 27, 28

- Connes, Frederic, 9

- Downes, Michael, 28

- Els, Danie, 7
- els, Danie, 3
- \em, vii, 25
- \emminnershape, 25
- \emph, vii, 25
- \EmulatedPackage, 27
- \EmulatedPackageWithOptions, 27
- environment,
 - figure, 25
 - list, 18
 - minipage, 23, 24
 - sidecaption, 21, 22, 24
 - sidecontcaption, 22
 - sidelegend, 22
 - sidenamedlegend, 22
 - table, 25
 - tabular, 24
 - thebibliography, 15
 - theglossary, 10, 15
 - theindex, 15
 - verbatimoutput, 1
 - verse, 18
 - writeverbatim, 1

- figure (environment), 25
- \footnoterule, 25
- Ford, Matthew, 18
- \frontmatter, 28
- \futurelet, 28

\glossary, 10–12, 16
 \glossarycolsep (length), 15
 \glossaryentry, 14
 \glossaryintoc, 15
 \glossarymark, 15
 \glossaryname, 15
 \glossaryrule (length), 15
 \glossitem, 10, 11, 14, 16

 \hrule, 17
 hyperref (package), 9

 sidecapfloatwidth, 23
 \ifscapmargleft, 21, 22

 \kern, 17

 \large, 5
 \leadpagetoclevel, 3
 \leftmargini (length), 18
 \legend, 29
 length,
 \afterPoemTitleskip, 5
 \beforePoemTitleskip, 5
 \glossarycolsep, 15
 \glossaryrule, 15
 \leftmargini, 18
 \linewidth, 23
 \marginparsep, 21
 \marginparwidth, 21
 \midPoemTitleskip, 5
 \sidecapraise, 23
 \sidecapsep, 21, 23
 \sidecapwidth, 21
 \vleftmargin, 18, 19
 \linewidth (length), 23
 list (environment), 18

 \mainmatter, v
 \makeglossary, 10, 14
 MakeIndex (program), 9–12, 16
 \marginpar, v
 \marginparsep (length), 21
 \marginparwidth (length), 21
 \memappchapinfo, 28
 \memappchapstarinfo, 28
 \memapppageinfo, 28
 \memapppagestarinfo, 28

 \membicaptioninfo, 29
 \membionenumcaptioninfo, 29
 \membitwonumcaptioninfo, 29
 \memcaptioninfo, 29
 \memchapinfo, 28
 \memchapstarinfo, 28
 \memglodesc, 12
 \memglonum, 12
 \memgloref, 12
 \memgloterm, 12
 \memgobble, 27
 \memhyperindexfalse, 9
 \memjustarg, 27
 \memleadpageinfo, 28
 \memleadpagestarinfo, 28
 \memlegendinfo, 29
 \memnamedlegendinfo, 29
 memoir (class), v, vii, ix, 9, 11, 18, 27, 28
 \mempartinfo, 28
 \mempartstarinfo, 28
 \mempoeminfo, 29
 \mempoemstarinfo, 29
 \memPoemTitleinfo, 29
 \memPoemTitlestarinfo, 29
 \memsecinfo, 28
 \memsecstarinfo, 28
 \midPoemTitleskip (length), 5
 minipage (environment), 23, 24

 natbib (package), v
 \newcommand, 3
 \newleadpage, 3
 \newleadpage*, 3
 \noglossaryintoc, 15
 \NumberPoemTitle, 4

 PoemTitle, 5
 poemtitle, 4
 \onecolglossaryfalse, 15
 \onecolglossarytrue, 15
 option,
 article, v
 sectionbib, v
 \overridescapmargin, 23

 package,
 chapterbib, v

hyperref, 9
 natbib, v
 patchcmd, 28
 wrapfig, 24
 pagestyle,
 \part, 3, 7, 28
 \part*, 28
 \partmark, 3, 4
 patchcmd (package), 28
 \patchcommand, 28
 \PlainPoemTitle, 4
 \PoemTitle, 4, 29
 \poemtitle, 29
 \PoemTitle*, 4
 \PoemTitlefont, 5
 \PoemTitleheadstart, 4, 5
 \poemtitlemark, 4
 \PoemTitlenumfont, 5
 \poemtitlepstyle, 4
 \poemtitlestarmark, 4
 \poemtitlestarpstyle, 4
 \preglossaryhook, 15
 \printglossary, 10, 16
 \printPoemTitlenonum, 4, 5
 \printPoemTitlenum, 4, 5
 \printPoemTitletitle, 4, 5
 program,
 MakelIndex, 9–12, 16
 xindy, 9

 \raggedbottom, 25
 \raggedleft, 24
 \raggedright, 24
 \renewcommand*, 23
 \renewleadpage, 3
 \renewleadpage*, 3

 \scapmargleftfalse, 21
 \scapmarglefttrue, 21
 \section, 28
 sectionbib (option), v
 \setlength, 23
 \setsidecappos, 22
 \sidebarmargin, 17
 \sidebaronesidefalse, 17
 \sidebaronesidettrue, 17
 \sidecapfloatwidth, 23

 \sidecapmargin, 21, 23
 \sidecapraise (length), 23
 \sidecapsep (length), 21, 23
 \sidecapstyle, 22
 sidecaption (environment), 21, 22, 24
 \sidecapwidth (length), 21
 sidecontcaption (environment), 22
 sidelegend (environment), 22
 sidenamedlegend (environment), 22
 \strictpagechecktrue, 23
 \subsection, 28

 table (environment), 25
 \tableofcontents, v
 tabular (environment), 24
 thebibliography (environment), 15
 theglossary (environment), 10, 15
 theindex (environment), 15
 \thepoem, 4

 \usepackage, 27

 verbatimoutput (environment), 1
 verse (environment), 18
 \vfill, 25
 \vleftmargin (length), 18, 19
 \vleftoffline, 17, 18

 wrapfig (package), 24
 writeverbatim (environment), 1

 xindy (program), 9