Addendum

ADDENDUM

The Memoir Class

for

Configurable Typesetting

User Guide

Peter Wilson

THP
The Herries Press

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 10 9 8 7 6 5 4 3 2 1

First edition: November 2002

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

Contents

Introduction

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 fifth edition of the *User Manual* which describes version 1.2 of the memoir class. The class is currently at version 1.3.

The main extensions and changes include:

- There is more flexibility in typesetting the titles of unnumbered chapters;
- 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;
- As usual, minor glitches have been removed from the code.

One

Document divisions

1.1 Introduction

This chapter describes the changes and extensions to the commands for producing section heads.

1.2 Chapter headings

There is one new parameter for controlling the layout of `\chapter` and `\chapter*` titles. In the standard classes the title of an unnumbered chapter is typeset at the same position on the page as the word 'Chapter' for numbered chapters.

```
\printchapternonum
```

The macro `\printchapternonum` is called just before an unnumbered chapter title text is typeset. By default this does nothing but you can use `\renewcommand` to change this. For example, if you wished the title text for both numbered and unnumbered chapters to be at the same height on the page then you could redefine `\printchapternonum` to insert the amount of vertical space taken by any 'Chapter N' line.

```
\chapterprecishere{⟨text⟩}
\prechapterprecis
\postchapterprecis
```

The `\chapterprecishere` macro is intended for use immediately after a `\chapter`. The ⟨*text*⟩ argument is typeset in italics in a `quote` environment. The macro is not new (it is part of the `\chapterprecis` macro) but its definition has been changed to:

```
\newcommand{\chapterprecishere}[1]{%
  \prechapterprecis #1\postchapterprecis}
```

where `\prechapterprecis` and `\postchapterprecis` are defined as:

```
\newcommand{\prechapterprecis}{%
  \vspace*{-2\baselineskip}%
```

```
    \begin{quote}\normalfont\itshape}
  \newcommand{\postchapterprecis}{\end{quote}}
```

The `\prechapterprecis` and `\postchapterprecis` macros can be changed if another style of typesetting is required.

1.3 Moved headings

In the standard classes a `\section` or other divisional heading that is too close to the bottom of a page is moved to the top of the following page. If this happens and `\flushbottom` is in effect, the contents of the short page are stretched to make the last line flush with the bottom of the typeblock.

```
\raggedbottomsectiontrue
\raggedbottomsectionfalse
\bottomsectionskip
```

The `\raggedbottomsectiontrue` declaration will typeset any pages that are short because of a moved divisional header as though `\raggedbottom` was in effect for the short page; other pages are not affected. The length `\bottomsectionskip` controls the amount of stretch on the short page. Setting it to zero allows the last line to be flush with the bottom of the typeblock. The default setting of 10mm appears to remove any stretch.

The declaration `\raggedbottomsectionfalse`, which is the default, cancels any previous `\raggedbottomsectiontrue` declaration.

* * *

The `\plainfancybreak` macro inserts a plain break in the middle of a page or if the break would come at the bottom or top of a page it inserts a fancy break instead.

```
\pfbreak \pfbreak*
\pfbreakskip
\pfbreakdisplay{⟨text⟩}
```

The `\pfbreak` macro is an alternate for `\plainfancybreak` that may be more convenient to use. The gap for the plain break is given by the length `\pfbreakskip` which is initialised to produce two blank lines. The fancy break, which takes the same vertical space, is given by the ⟨*text*⟩ argument of `\pfbreakdisplay`. The default definition typesets three asterisks, as shown a few lines before this.

♣ ♢ ♣

You can change the definition of `\pfbreakdisplay` for a different style if you wish. The fancy break just before this was produced via:

```
\renewcommand{\pfbreakdisplay}{%
  $\clubsuit$\quad\$\diamondsuit$\quad$\clubsuit$}
\fancybreak{\pfbreakdisplay}
```

I used `\fancybreak` as I'm not sure where the break will come on the page and the simple `\pfbreak` macro might just have produced a couple of blank lines instead of the fancy display.

The paragraph following `\pfbreak` is not indented. If you want it indented use the `\pfbreak*` starred version.

Two

Tops and tails

2.1 Introduction

Some small conveniences have been added for the ToC and friends. There are major enhancements for indexing and indexes.

2.2 Table of contents

It is now possible to use `\tableofcontents`, `\listoffigures`, etc., multiple times in a document.

`\changetocdepth{`⟨*num*⟩`}`

The `\settocdepth` macro puts the `\changetocdepth` into the `toc` file and normally `\changetocdepth` resets the `tocdepth` within the ToC itself.

`\precistoctext{`⟨*text*⟩`}`
`\precistocfont`

The `\chapterprecistoc` macro puts `\precistoctext` into the `toc` file which then typesets its argument using the `\precistocfont` (default `\itshape`).

If you are setting both a short and a long ToC, for the short ToC you may wish to temporarily make `\changetocdepth` and `\precistoctext` swallow their arguments without doing anything else.

`\partnumberline{`⟨*num*⟩`}`
`\chapternumberline{`⟨*num*⟩`}`

In the ToC, the macros `\partnumberline` and `\chapternumberline` are responsible respectively for typesetting the `\part` and `\chapter` numbers. If you do not want, say, the `\chapter` number to appear you can do:

```
\renewcommand{\chapternumberline}[1]{}
```

NOTE: If you use the hyperref package you will also have to use the memhfixc package after the hyperref package:

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

The memhfixc package provides a patch for the hyperref package because it does not understand the `\partnumberline` and `\chapternumberline` commands.

```
\cftchapterbreak
```

When `\l@chapter` starts to typeset a `\chapter` entry in the ToC the first thing it does is to call the macro `\cftchapterbreak`. This is defined as:

```
\newcommand{\cftchapterbreak}{\addpenalty{-\@highpenalty}}
```

which encourages a page break before rather than after the entry. As usual, you can change `\cftchapterbreak` to do other things that you feel might be useful.

2.3 Indexing

The indexing commands have been significantly enhanced and include the functionality provided by the makeidx, showidx and index packages; these packages should not be used.

In the standard classes the index is set in two columns.

```
\onecolindextrue
\onecolindexfalse
```

The declaration `\onecolindexfalse`, which is the default, causes any indexes to be set in two columns. The declaration `\onecolindextrue` causes any following indexes to be set in one column. This can be useful if, for example, you need an index of the first lines of poems.

```
\makeindex[⟨file⟩]
\printindex[⟨file⟩]
```

The macro `\makeindex`, which must be put in the preamble if it is used, opens an `idx` file, which by default is called `jobname.idx`, where `jobname` is the name of the main LaTeX source file. If the optional ⟨*file*⟩ argument is given then a file called `file.idx` will be opened. The macro `\printindex` reads an `ind` file called `jobname.ind`, which should contain an `theindex` environment and the indexed items. If the optional ⟨*file*⟩ argument is given then the `file.ind` file will be read. The MAKEINDEX program is often used to convert an `idx` file to an `ind` file.

```
\index[⟨file⟩]{⟨item⟩}
\specialindex{⟨file⟩}{⟨counter⟩}{⟨item⟩}
```

The macro \index writes its ⟨*item*⟩ argument to an idx file. If the optional ⟨*file*⟩ argument is given then it will write to file.idx otherwise it writes to jobname.idx. The page for the ⟨*item*⟩ is also written to the idx file. The \specialindex macro writes its ⟨*item*⟩ argument to the file.idx and also writes the page number (in parentheses) and the value of the ⟨*counter*⟩. This means that indexing can be with respect to something other than page numbers. However, if the hyperref package is used the special index links will be to pages even though they appear to be with respect to the ⟨*counter*⟩; for example, if figure numbers are used as the index reference the hyperref link will be to the page where the figure appears and not the figure itself.

\see{⟨*item*⟩} \seename \seealso{⟨*items*⟩} \alsoname

The macro \see can be used in an \index command to tell the reader to 'see ⟨*item*⟩' instead of printing a page number. Likewise the \seealso macro directs the reader to 'see also ⟨*items*⟩'. For example:

```
\index{Alf|see{Alfred}}
\index{Frederick|seealso{Fred, Rick}}
```

The actual values for 'see' and 'see also' are given by the \seename and \alsoname macros whose default definitions are:

```
\newcommand{\seename}{see}
\newcommand{\alsoname}{see also}
```

\reportnoidxfilefalse \reportnoidxfiletrue

Following the declaration \reportnoidxfilefalse, which is the default, LaTeX will silently pass over attempts to use an idx file which has not been declared via \makeindex. After the declaration \reportnoidxfiletrue LaTeX will whinge about any attempts to write to an unopened file.

\showindexmarktrue \showindexmarkfalse

After the declaration \showindexmarktrue (practically) all \index and \specialindex ⟨*item*⟩ arguments are listed in the margin of the page on which the index command is issued. The default is \showindexmarkfalse.

Indexing and the natbib package

The natbib package will make an index of citations if \citeindextrue is put in the preamble after the natbib package is called for.

\citeindexfile

The name of the file for the citation index is stored in the macro \citeindexfile. This is initially defined as:

```
\newcommand{\citeindexfile}{\jobname}
```

That is, the citation entries will be written to the default `idx` file. This may be not what you want so you can change this, for example to:

```
\renewcommand{\citeindexfile}{names}
```

If you do change `\citeindexfile` then you have to put

```
\makeindex[\citeindex]
```

before

```
\usepackage[...]{natbib}
```

So, there are effectively two choices, either along the lines of

```
\renewcommand{\citeindexfile}{authors} % write to authors.idx
\makeindex[\citeindexfile]
\usepackage{natbib}
\citeindextrue
...
\renewcommand{\indexname}{Index of citations}
\printindex[\citeindexfile]
```

or along the lines of

```
\usepackage{natbib}
\citeindextrue
\makeindex
...
\printindex
```

POPULATING THE IDX FILE

In the standard classes, indexed items are written directly to an `idx` file. With the class, however, the indexed items are written to the `aux` file and then on the next LaTeX run the indexed items in the `aux` file are written to the designated `idx` file.

The disadvantage of this two stage process is that after any change to the indexed items LaTeX has to be run twice to ensure that the change is propagated to the `idx` file. Then, of course, a new `ind` will have to be created and LaTeX run one more time. However, this is what you have to do if you are using BibTeX.

The advantage of the approach is that indexed items from `\include` files that are not processed on a particular run are not lost. The standard direct write to an `idx` file loses any 'non-inluded' indexed items.

Three

Trim marks

3.1 Introduction

When the memoir class showtrims option is used, trim marks can be placed on each page, usually to indicate where the stock should be trimmed to obtain the planned page size.

Peter Heslin (p.j.heslin@durham.ac.uk) asked me to extend the simple trim mark provided by the memoir class as it appeared unlikely that the author of the crop package would take account of the class (see the thread titled *Incompatibility of memoir.cls and crop.sty*, October 2002 on ctt for details).

3.2 Marks

Trim marks can be placed at each corner of the (trimmed) page, and also at the middle of each side of the page.

```
\trimXmarks
\trimLmarks
\trimFrame
\trimNone
```

Some predefined trimming stayles are provided. After the declaration `\trimXmarks` marks in the shape of a cross are placed at the four corners of the page. The declaration `\trimLmarks` calls for corner marks in the shape of an 'L', in various orientations depending on the particular corner. After `\trimFrame` a frame will be drawn around each page, coinciding with the page boundaries. The declaration `\trimNone` disables all kinds of trim marking.

```
\trimmarks
\tmarktl \tmarktr \tmarkbr \tmarkbl
\tmarktm \tmarkmr \tmarkbm \tmarkml
```

The macro `\trimmarks` is responsible for drawing upto 8 marks. The marks are defined as zero sized pictures which are placed strategically around the borders of the page.

`\trimmarks` places the pictures `\tmarktl`, `\tmarktr`, `\tmarkbl`, and `\tmarkbr` are placed at the top left, top right, bottom right and bottom left corners of the page. The pictures `\tmarktm`, `\tmarkmr`, `\tmarkbm`, and `\tmarkml` are placed at the top middle, middle right, bottom middle and middle left of the edges of the page. All these `\tmark..` macros should expand to zero sized pictures.

For example, to draw short lines marking the half-height of the page, try this:

```
\providecommand{\tmarkml}{}
\renewcommand{\tmarkml}{%
  \begin{picture}(0,0){%
    \unitlength 1mm
    \thinlines
    \put(-2,0){\line(-1,0){10mm}}
  \end{picture}}
\providecommand{\tmarkmr}{}
\renewcommand{\tmarkmr}{%
  \begin{picture}(0,0){%
    \unitlength 1mm
    \thinlines
    \put(2,0){\line(1,0){10mm}}
  \end{picture}}
```

Thin horizontal lines of length 10mm will be drawn at the middle left and middle right of the page, starting 2mm outside the page boundary.

3.3 Sheet numbering

One application of trim marks is to show a printer where the stock should be trimmed. In this application in can be useful to also note the sheet number on each page, where the shhet number is 1 for the first page and increases by 1 for each page thereafter. The sheet number is independent of the page number.

```
\thesheetsequence
```

The macro `\thesheetsequence` typesets the current sheet sequence number and is analagous the `\thepage`.

```
lastsheet
lastpage
```

The counter `lastsheet` holds the number of sheets processed during the *previous* run of LaTeX. Similarly, the counter `lastpage` holds the number of the last page processed during the previous run. Note that the last page number is not necessarily the same as the last sheet number. For example:

In this document this is sheet 18 of 22 sheets, and page 8 of 12.

The previous sentence was the result of processing the following code

```
\textit{In this document this is
        sheet \thesheetsequence\ of \thelastsheet\ sheets,
```

```
        and page \thepage\ of \thelastpage.}
```

You may wish to use the sheet and/or page numbers as part of some trim marks. The following will note the sheet numbers above the page.

```
\newcommand{\trimseqpage}{%
  \begin{picture}(0,0)
    \unitlength 1mm
    \put(0,2){\makebox(0,0)[b]{Sheet: \thesheetsequence\ of \thelastsheet}}
  \end{picture}}
\let\tmarktm\trimseqpage
```

Index

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