

The `admbmanual` class*

Dangerous Curve
`typesetting@dangerouscurve.org`

December 8, 2014

1 Introduction

The `admbmanual` class lets you write manuals in the style of the ADMB Project.

2 Usage for the Manual Writer

2.1 Get Started with a Sample Skeleton

You can use the file `admb-sample.tex`, which contains a simple skeleton example, as the starting point for your manual.

To compile your manual, you need to use the Makefile by running the command `make` on the command line. If your L^AT_EX's file name is `admb-sample.tex`, then type: `make filename=admb-sample`

If you want to clear out intermediate auxiliary files left around by L^AT_EX, type (again, for `admb-sample.tex`): `make clean filename=admb-sample`

2.2 Use Standard L^AT_EX

Insofar as we could, we embedded coding changes into already existing L^AT_EX commands. So, you can pretty much write your manuals using standard L^AT_EX commands for the book class. For instance, instead of making you put an extra command in front of the `\appendix` (which you'd have to do to get the page numbers to work properly), we embed that command in the `\appendix` command. Similarly, you do not have to explicitly ask for a license page: if there is a file called `license.tex` (hopefully, with the license in it) that can be seen by L^AT_EX, then you'll get a license page; if not, you'll get a blank page.

We recommend using L^AT_EX commands in favor of plain TeX commands. For one thing, it makes your document more "color-safe." For instance, use `\sbox` rather than `\setbox`, `\mbox` rather than `\hbox`, and `\parbox` or the minipage environment rather than `\vbox`. Use `\sbox` rather than `\setbox`. If you're going to define macros, instead of using `\def`, use `\newcommand` or `\renewcommand`. If you're

*This document corresponds to `admbmanual` v1.0, dated 2009/06/01.

going to define an environment, use `\newenvironment` or `\renewenvironment` instead `\def\myenv{...}` and `\def\myenv{...}`.¹ When typesetting mathematics, try to avoid using plain \TeX 's $\$ \$ \dots \$ \$$ for displays, and *do* avoid using `eqnarray` and `eqnarray*`.²

We also recommend installing the “cm-super” package. It is part of most common \LaTeX distribution, is invoked automatically, and will improve the appearance of large fonts.

2.3 Read Some Good Guides

While by no means an exhaustive list, we recommend the two aforementioned guides, along with *Making Tables with \LaTeX* ,³ which is just a sample from the great Amy Hendrickson's teaching guides, and (for those of you with 141 minutes to spare) *The Not So Short Introduction to $\text{\LaTeX} 2\epsilon$* by Tobias Oetiker, Hubert Partl, Irene Hyna, and Elisabeth Schlegl.

2.4 Use These Other Things We've Provided

Below are the commands we've defined for you to use. We don't list all the standard \LaTeX commands we've redefined to make this class do what it does, but if we've changed their behavior in any way about which you need to know, we do mention it.

2.4.1 Do Bold Math

`\bm{\langle math symbols in math mode \rangle}`

From the `bm` package, this command lets you write e.g., `\bm{$n^2 \alpha$}` to get bold math. (You can also use `\boldsymbol{n^2 \alpha}`.) Don't use this command in chapter/section/paragraph heads. They do what you want (that is, make bold the math you put in them, if the head is to be in bold), and besides, it will break them if you do.

2.4.2 All Caps Means Small Caps

All all-caps words or acronyms longer than two characters should be set in what are called “small caps.” This is because words such as these stick out like sore thumbs, especially within text, which is mostly in lowercase. Small caps are usually a couple of points smaller than the full-sized caps of a given size, so words made of small caps blend better with the text than do those made up of full-sized caps. The one place where we don't use small caps for all-cap words is in the manual's title, as there, these words are not surrounded by text.

To get small caps, e.g., ASCII, you type `\textsc{ascii}`. In general, you type `\textsc{\langle the otherwise all-cap word in all lowercase \rangle}`. Remember to make the

¹Thanks to the $\text{\LaTeX} 3$ Project's *$\text{\LaTeX} 2\epsilon$ for class and package writers*.

²Thanks to *Short Math Guide for \LaTeX* by Michael Downes.

³Found at <http://www.texnology.com/teachingsamp.pdf>

argument lowercase, or you'll just get the full-sized caps (and the macro will look as if it didn't do anything).

2.4.3 Tell the Footers Your Manual's Name

\manualname \manualname{\{*manual name in Upper/lowercase*\}}

This tells the typesetter what manual name to put in the footers. The convention so far at ADMB has been to put this in Upper/lowercase, e.g., Admb, Admb-Re, and Autodif.

2.4.4 Build the Title Page

The L^AT_EX book, etc., classes already define such title-page macros as \title, \author, and \date, which you set before calling \maketitle. We ignore the date (although you can set it, if you wish, without effect). We have an enhanced way of doing the title, so we tell you how to build one. We also introduce another macro: \publisher.

\maketitle You call this after setting at least the title and the author(s) of your manual. It is currently hardwired to use a certain logo. If that changes, we'll probably have to redo the scaling and spacing again, so we discourage you from substituting in another logo file. If there is a file *license.tex* somewhere where L^AT_EX can find it, this command also sets a license page on the *verso* (reverse side) of the title page.

Build the Title You can build a nice-looking title by combining inside \title at least \largeitlepart with any number of instances of \mediumtitlepart, and \smalltitlepart, or their nonslanted counterparts \mediumtitlepartnonslanted, and \smalltitlepartnonslanted.

\title{\{*Any combination of calls to various part commands below*\}}

The medium and small title parts' contents hierarchically relate to the large one. For instance, you might write:

```
\title{%
  \mediumtitlepart{Introduction to}
  \largeitlepart{MAP-ALL}
  \smalltitlepart{A Saavy Mapping Converter}}
```

In other words, you have to try out various combinations to see what looks good. If you can't tell, ask a typographer or graphic designer.

The Title Parts Each sized part is really a “cluster” of a sort. The arguments can be longer than can fit, or look nice, across the page, so you can add hand breaks (via a double backslash: \\). The hand-broken lines of each cluster will have regular vertical space (called “primary leading”) between them, but between clusters, there will be extra space (“secondary leading”).

\largeitlepart \largeitlepart{\{*title part to be set large*\}}
\mediumtitlepart \mediumtitlepart{\{*title part to be set in medium size*\}}
\smalltitlepart \smalltitlepart{\{*title part to be set small*\}}

\publisher **Change Your Publisher** \publisher{*publisher name*}
We already set this to `admb-project.org` as a default, but you are free to change it.

2.4.5 Fill Your Index

\x \X{\langle index item\rangle}
You can use this as a shorthand for the indexing command \index{\langle index item\rangle}.

\xx \X{\langle index item\rangle}{\langle index sub-item\rangle}
You can use this as a shorthand for the indexing command \index{\langle index item\rangle!\langle index sub-item\rangle}

\fontindexentry \fontindexentry{\langle font abbreviation\rangle}{\langle index item\rangle}
Deprecated, as it doesn't work with \index, just with \X and \XX. We'll describe it in case you run into it in other manuals' code.
\langle font abbreviation\rangle is a (mostly) two-letter abbreviation for the \textXX font commands, e.g., \textit, where XX can be one of `bf`, `it`, `md`, `normal`, `rm`, `sc`, `sf`, `sl`, `tt`, and `up`. When used with \X and \XX, it does what, e.g., `foo@\textit{foo}` does. So, if you want *foo* (in italic) to be in the index, type \index{foo@\textit{foo}}. Don't use this construct in the middle of an index entry, though, e.g., \index{foo bar@\texttt{bar}}, as it will make the entry disappear!

2.4.6 Code the Code

code You probably will want some environment for typesetting things, such as computer code, whose linebreaks and mono-spaced letter widths you wish to preserve. This is the environment hopefully you'll use the most, as it is the largest font of the three code environments. Put your code between \begin{code}... \end{code}. If your material's lines are too long to fit in the width of the page (and we suggest you don't let things go out into the margin—it wrecks the document design), then use the environments below, which set things in smaller type. This environment sets things in the largest type of them all.

smallcode As with the `code` environment above, but using smaller type.

tinycode As with the `smallcode` environment above, but using yet smaller type. In fact, it's so small that you should only use it in emergencies—or if it's only going to be read by young eyes only.

\aftercodething \begin{code}\langle some code\rangle\end{code}\aftercodething{\langle text/image block\rangle}

The above code environments leave extra space top and bottom, so if you need to set something that fits snug up under a line of code—as you might if you have a command followed by its description—then you use this. You should not put anything between the code and the `\aftercodething` command, not even “invisible” indexing commands. Comments are OK, though. Actually, you can use this command to get rid of extra space after almost anything you don't want.

We've not tested everything, but go ahead and try it. This command adds extra space after its argument.

`lstlisting` In addition to the above, you can also typeset the C code using the `lstlisting` environment, which is set up to recognize the code's syntax and thus typeset it in a more sophisticated way.

`\afterlistingthing` `\begin{lstlisting}{some code}\end{lstlisting}`
`\afterlistingthing{\text/image block}`

This is analogous to `\aftercodething`, except you use it with the `lstlisting` environment.

2.4.7 Put Images and Tabulars in Figures and Tables

You can use the `\includegraphics` command to include image files, which you can do either inside or outside `figure` environments. If you do it outside, then know that you've therefore not allowed it to "float" elsewhere (as you would have had you included it in a figure). So, if the image is too big to fit on the (rest of the) page, be prepared for some unsightly vertical spacing. The same is true for PicTeXpictures. As `figure` environments are to images, so `table` environments are to, e.g., `tabular` environments.

Here are some things that we've added to help you with things figure and table-like:

`\emptycaption` `\emptycaption`

You can use this with a figure or table instead of `\caption{\text}`. It will print out, e.g., "Figure 1" without a colon.

`wrapfigure` You can use this environment from the `wrapfig` package to wrap text around your figures or tables. See their documentation for more details.

`\fiverm`

`\eightrm`

`\ninemr`

`\twelverm`

`\eighteenrm`

You can use these commands within PicTeX code to change font sizes. If you need another size, you can look at the code for `\@loadrms` below, or ask a friendly TeXnician what to do.

2.4.8 Breaking Lines by Hand

`\br` You can use this command to force a line break in, e.g., heads. We turn this into a space when the head gets put in the table of contents, so you won't have absurd-looking short lines there. Don't add the usual backslash after this command, or you'll get two spaces there in the table-of-contents lines.

`\BR` OK, if you really want a line break in both the section *and* the table of contents, then use this command.

2.5 Take Some Shortcuts

You can use some commands as shortcuts to getting good typography (or just not have to type as much).

`\cplus` You can use this to get a prettier version of "C++" with the "++" much smaller and raised up.

<code>\e</code>	$\langle real\ number\ factor \rangle \backslash e \{ \langle integer\ exponent \rangle \}$
	You can use this when you want your scientific notation to be in the form of “times 10 exponential” (e.g., 1.34×10^n).
<code>\pluseq</code>	E.g., $\{ \langle number\ variable \rangle \} \backslash pluseq \{ \langle number\ variable\ or\ number \rangle \}$
<code>\minuseq</code>	You can use these when you want “assignment by” operators (e.g., $+=$, $-=$, etc.) to be spaced more closely than they are when L ^A T _E X sets them by default as separate binary operators.
<code>\multiplyeq</code>	
<code>\divideeq</code>	
<code>\scAB</code>	You can use this to get ADMB.
<code>\scAR</code>	You can use this to get ADMB-RE.
<code>\scAD</code>	You can use this to get AUTODIF.
<code>\ADM</code>	You can use this to get AD Model Builder with no space after it.
<code>\ADMS</code>	You can use this to get AD Model Builder followed by a space.

3 Implementation

In this section, “we” refers to the implementors.

We want to be able to define things with @ in their name, so the user can’t access them.

```
1 \makeatletter
```

3.1 Initialization

We derive everything from the standard `book` class. We want 12-point body type.

```
2 \LoadClass[12pt]{book}
```

We load all the packages at the top, to be able to rearrange if there’s a conflict. (For instance, `chappg` has to go after `hyperref`, but in the sections below, it appears in a different order). We also put it here before our code, so the packages don’t redefine anything we’ve (re)defined.

```
3 \RequirePackage{ifthen}
4 \RequirePackage[T1]{fontenc}
5 \RequirePackage{smallcap}
6 \RequirePackage{amsmath}
7 \RequirePackage{amssymb}
8 \RequirePackage{bm}
9 %\RequirePackage{anyfontsize}
10 \RequirePackage{relsize}
11 \RequirePackage{fancyhdr}
12 % \RequirePackage{chappg}
13 \RequirePackage{tocbibind}
14 \RequirePackage{makeidx}
15 \RequirePackage{multicol}
16 \RequirePackage{fancyvrb}
17 \RequirePackage{listings}
18 \RequirePackage{graphicx}
19 \RequirePackage{caption}
20 \RequirePackage{wrapfig}
```

```

21 \RequirePackage{rawfonts}
22 \RequirePackage{pictxwd}
23 \RequirePackage{etex}
24 \RequirePackage{xy}
25 \RequirePackage{graphics}
26 \RequirePackage[usenames]{color}
27 \RequirePackage[pagebackref=true,linktocpage=true,colorlinks=true]{hyperref}
28 \RequirePackage{all}{hypcap}
29 \RequirePackage{chappg}

```

3.1.1 Some Programming Stuff

We want to be able to use the `\ifthenelse` construct in our macro writing.

```
30 % \RequirePackage{ifthen}
```

3.2 The Overall Document

3.2.1 Fonts

We want to be able to get bold small caps. The T1 optional argument extends the font set.

```

31 % \RequirePackage[T1]{fontenc}
32 % \RequirePackage{smallcap}

```

We want the user to be able to use all the nice AMS math symbols (and constructs).

```

33 % \RequirePackage{amsmath}
34 % \RequirePackage{amssymb}

```

We want to be able to typeset bold math symbols with `\bm`.

```
35 % \RequirePackage{bm}
```

We need some arbitrary font scaling for the title. (The package `fix-cm` would work, too, but this is more general.)

```
36 % \RequirePackage{anyfontsize}
```

We also need some relative font scaling.

```
37 % \RequirePackage{relsize}
```

We want to be able to set math in section and paragraph heads, so we add `\boldmath` to the fonts of all chapter, section, and paragraph commands.

`\@makechapterhead` We add `\boldmath` to the font.

```

38 \def\@makechapterhead#1{%
39   \vspace*{50\p@}%
40   {\parindent \z@ \raggedright \normalfont
41     \ifnum \c@secnumdepth > \m@ne
42       \huge\bfseries \chapapp\space \thechapter
43       \par\nobreak
44       \vskip 20\p@
45     \fi
46     \interlinepenalty\OM

```

```

47      \Huge \bfseries \boldmath #1\par\nobreak
48      \vskip 40\p@
49  }
50 }

\@makeschapterhead We add \boldmath to the font.
51 \def\@makeschapterhead#1{%
52   \vspace*{50\p@}%
53   {\parindent \z@ \raggedright
54     \normalfont
55     \interlinepenalty\@M
56     \Huge \bfseries \boldmath #1\par\nobreak
57     \vskip 40\p@
58   }
59 }

\section We add \boldmath to the font.
60 \renewcommand\section{\@startsection {section}{1}{\z@}%
61                               {-3.5ex \@plus -1ex \@minus -.2ex}%
62                               {2.3ex \@plus .2ex}%
63                               {\normalfont\Large\bfseries\boldmath}{}}

\subsection We add \boldmath to the font.
64 \renewcommand\subsection{\@startsection{subsection}{2}{\z@}%
65                               {-3.25ex \@plus -1ex \@minus -.2ex}%
66                               {1.5ex \@plus .2ex}%
67                               {\normalfont\large\bfseries\boldmath}{}}

\subsubsection We add \boldmath to the font.
68 \renewcommand\subsubsection{\@startsection{subsubsection}{3}{\z@}%
69                               {-3.25ex \@plus -1ex \@minus -.2ex}%
70                               {1.5ex \@plus .2ex}%
71                               {\normalfont\normalsize\bfseries\boldmath}{}}

\paragraph We add \boldmath to the font.
72 \renewcommand\paragraph{\@startsection{paragraph}{4}{\z@}%
73                               {3.25ex \@plus 1ex \@minus .2ex}%
74                               {-1em}%
75                               {\normalfont\normalsize\bfseries\boldmath}{}}

\ subparagraph We add \boldmath to the font.
76 \renewcommand\subparagraph{\@startsection{subparagraph}{5}{\parindent}%
77                               {3.25ex \@plus 1ex \@minus .2ex}%
78                               {-1em}%
79                               {\normalfont\normalsize\bfseries\boldmath}{}}

```

3.2.2 Page Layout

We need to reset the page margins, text body size, and footer placement. We use the `geometry` package to do so.

```
80 \RequirePackage[%  
81     top=1.125truein,  
82     outer=.875truein,  
83     inner=1.125truein,  
84     % textheight=8.5truein, % Allow LaTeX to infer this, both to get  
85     % textwidth=6.5truein, % rid of warnings and to allow any page size  
86     footskip=.875truein  
87 ]{geometry}
```

3.2.3 Page Numbering

```
88 % We want to define our own footers (and get rid of headers).  
89 % \RequirePackage{fancyhdr}
```

We need page numbers to be prepended with chapter numbers. This has to be loaded after the `hyperref` package.

```
90 % \RequirePackage{chappg}
```

`\tableofcontents` We need to end the front matter after the table of contents.

```
91 \let\savetableofcontents=\tableofcontents  
92 \renewcommand{\tableofcontents}{%  
93     \savetableofcontents  
94     \mainmatter  
95 }
```

`\manualname` We need to define the manual name so the footers can use it.

```
96 \newcommand*{\manualname}[1]{\gdef\@manualname{\#1}}
```

`\@dopagenumbering` `\@dopagenumbering`

We want our own type of headers and footers.

```
97 \newcommand{\@dopagenumbering}{%  
98     \pagestyle{fancy}  
99     \fancyhead{}  
100    \fancyfoot{}  
101    \renewcommand{\headrulewidth}{0pt}  
102    \renewcommand{\footrulewidth}{0pt}  
103    \fancyfoot[R]{\thepage} % basic one-sided footer, or alternatively:
```

Define odd pages, left and right.

```
104    % \fancyfoot[LO]{\small admbs-project.org}  
105    % \fancyfoot[RO]{\thepage}
```

Define even pages, left and right.

```
106    % \fancyfoot[LE]{\thepage}  
107    % \fancyfoot[RE]{\small\emph{\@manualname} User's Manual}  
108 }
```

We now call the above, to get the heads and footers the way we want.

109 \odopagenumbering

\pagenumbering We call this (from `chappg`) to get the page numbering to be the chapter number followed by the local (to the chapter) page number, separated by a hyphen.

110 \pagenumbering{bychapter}

\chapter We want our fancy footers to be in force, so we have to comment out the `\thispagestyle{plain}`.

```
111 \renewcommand\chapter{\if@openright\cleardoublepage\else\clearpage\fi%
112 % \thispagestyle{plain}%
113 \global\@topnum\z@%
114 \@afterindentfalse
115 \secdef\@chapter\@schapter
116 } % end \chapter
```

\appendix We need to add `\clear*pages`, so the `chappg` package doesn't get confused.

```
117 \renewcommand\appendix{
118   \if@openright
119     \cleardoublepage
120   \else
121     \clearpage
122   \fi
123   \par
124   \setcounter{chapter}{0}%
125   \setcounter{section}{0}%
126   \gdef\@chapapp{\appendixname}%
127   \gdef\thechapter{\@Alph\c@chapter}}
```

3.3 The Document Elements

3.3.1 Title Page

128 \definecolor{DarkBlue}{rgb}{0,0.08,0.45}

\maketitle We want to typeset a new design different from that of `book`, and to append a license page.

```
129 \if@titlepage
130 \renewcommand\maketitle{\begin{titlepage}%
131 \let\footnotesize\small
132 \let\footnoterule\relax
133 \let \footnote \thanks
134 \null\vfil
135 \vskip 60\p@
136 \begin{center}%
137   {\@title \par}%
138   \vskip 4em%
139   {%
140     \Large
```

```

141      \lineskip .75em%
142      \begin{tabular}[t]{c}%
143          \author%
144      \end{tabular}\par
145  }%
146      \vfill%
147      \hspace{9pt}%
148      \includegraphics[width=10pc]{ADMB-logo.png}\par
149      \vspace{-3.75pc}%
150      {\large \textcolor{DarkBlue}{\textbf{\textit{\textrm{\textit{@publisher}}}}}}\par
151      \vskip 1.5em%

```

We ignore the date.

```

152      \% {\large \date \par}%
153      \% \vspace*{1pc} % Add extra optical space at the bottom.
154      \end{center}\par
155      \thanks
156      \vfil\null
157      \end{titlepage}%
158      \setcounter{footnote}{0}%
159      \global\let\thanks\relax
160      \global\let\maketitle\relax
161      \global\let\@thanks\empty
162      \global\let\@author\empty
163      \global\let\@date\empty
164      \global\let\@title\empty
165      \global\let\title\relax
166      \global\let\author\relax
167      \global\let\date\relax
168      \global\let\and\relax

```

We go to the next page.

```
169 \clearpage
```

We omit the page number from the next page.

```
170 \thispagestyle{empty}%
```

If there is a license file, then set it. If not, leave a blank page.

```

171 \IfFileExists{license.tex}%
172 {%
173     \centerline{\Large License}%
174     \bigskip
175     \InputIfFileExists{license.tex}{}{%
176         \vfil\null
177     }
178     {
179     }
180     \frontmatter
181 } % End \maketitle
182 \fi

```

`\largeitlepart` We want a very large font for the main title.

```
183 \newcommand{\largeitlepart}[1]{%
184   \%{\sffamily\slshape\bfseries
185   {\sffamily\slshape
186   \fontsize{36}{36}\selectfont
187   #1\par
188   \vspace{.5\baselineskip}}%
189 }
```

`\mediumtitlepart` We also want a title font smaller than the large title one, but not that small.

```
190 \newcommand{\mediumtitlepart}[1]{%
191   \medskip
192   {%
193     \sffamily\slshape
194     \fontsize{24}{26}\selectfont
195     {#1}\par
196   }%
197   \bigskip
198   \medskip
199 }
```

`\mediumtitlepartnonslanted` We want a nonslanted version of `\mediumtitlepart`.

```
200 \newcommand{\mediumtitlepartnonslanted}[1]{%
201   \medskip
202   {%
203     \sffamily
204     \fontsize{24}{26}\selectfont
205     {#1}\par
206   }%
207   \bigskip
208   \medskip
209 }
```

`\smalltitlepart` We want a yet smaller title font, again still large enough for display (as opposed to text).

```
210 \newcommand{\smalltitlepart}[1]{%
211   \medskip
212   {%
213     \sffamily\slshape
214     \fontsize{18}{20}\selectfont
215     {#1}\par
216   }%
217   \bigskip
218   \medskip
219 }
```

`\smalltitlepartnonslanted` We want a nonslanted version of `\smalltitlepart`

```
220 \newcommand{\smalltitlepartnonslanted}[1]{%
221   \smallskip
```

```

222  {%
223   \sffamily
224   \fontsize{18}{20}\selectfont
225   {\#1}\par
226  }%
227  \bigskip
228  \medskip
229 }

\publisher We add a new entity to the title page.
230 \newcommand*{\publisher}[1]{\gdef\@publisher{\#1}}


We set the publisher to a default value.
231 \publisher{admb-project.org}

```

3.3.2 Table of Contents

We want the reference section and the index to show up in the table of contents.
 (We still had to do put the index in the TOC by hand, though, as we redefine the main index environment.)

```

232 % \RequirePackage{tocbibind}

\l@chapter We want to be able to set math in TOC chapter heads, so we add \boldmath to
the font for the chapter TOC entry.

```

```

233 \renewcommand*{\l@chapter}[2]{%
234   \ifnum \c@tocdepth >\m@ne
235     \addpenalty{-\@highpenalty}%
236     \vskip 1.0em \oplus \p@
237     \setlength{\tempdima}{1.5em}%
238     \begingroup
239       \parindent \z@ \rightskip \pnumwidth
240       \parfillskip -\pnumwidth

```

Add the \boldmath command here.

```

241       \leavevmode \bfseries \boldmath
242       \advance\leftskip\tempdima
243       \hskip -\leftskip
244       #1\nobreak\hfil \nobreak\hb@xt@\pnumwidth{\hss #2}\par
245       \penalty\@highpenalty
246     \endgroup
247   \fi
248 }

```

3.3.3 References

We want the reference chapter's name to be "References" rather than "Bibliography."

```

249 \renewcommand\tocbibname{References}
250 %
251 \let\savebibliography=\bibliography

```

```

252 \renewcommand{\bibliography}{%
253   \cleardoublepage
254   \pagenumbering[\tocbibname]{bychapter}
255   \savebibliography
256 }

```

3.3.4 The Index

We want an index.

```
257 % \RequirePackage{makeidx}
```

We use this to implement a two-column index where the columns don't crash into each other.

```
258 % \RequirePackage{multicol}
```

\l@section We need to increase the space for TOC number labels (the third argument), that is, for four—not three—numbers and a period.

```
259 \renewcommand*\l@section{\@dottedtocline{1}{1.5em}{3em}}
```

\X An abbreviation for `\index` with one item argument.

```
260 \newcommand\X[1]{\index{#1}}
```

\XX An abbreviation for `\index` with an item and a sub-item.

```
261 \newcommand\XX[2]{\index{#1!#2}}
```

\fontindexentry *Deprecated, as can only be used with \X and \XX.*

```
262 \newcommand{\fontindexentry}[2]{\#20\csname text#1\endcsname{\#2}}
```

theindex We want the index to have two columns that don't crash into each other.⁴

```

263 \renewenvironment{theindex}{%
264   \if@twocolumn
265     \restonecolfalse
266   \else
267     \restonecoltrue
268   \fi
269   \setlength{\columnseprule}{0pt}
270   \setlength{\columnsep}{45pt}
271   \begin{multicols}{2}[\section*{\indexname}]
272   \% \markboth{\MakeUppercase{\indexname}}%
273   \% {\MakeUppercase{\indexname}}%
274   \% \thispagestyle{plain}
275   \addcontentsline{toc}{chapter}{\indexname}
276   \setlength{\parindent}{0pt}
277   \setlength{\parskip}{0pt plus 0.3pt}
278   \relax
279   \let\item\@idxitem
280   \end{multicols}\if@restonecol\onecolumn\else\clearpage\fi}

```

⁴Modified from Juanjo on LC L^AT_EX Community.

```

281 %
282 % We want our special footers for the index pages.
283 % \begin{macrocode}
284 \let\saveprintindex=\printindex
285 \renewcommand{\printindex}{%
286   \cleardoublepage
287   \dopagenumbering
288   \pagenumbering[Index]{bychapter}
289   \saveprintindex
290 }

```

3.4 Some Document Elements

3.4.1 Typesetting Code

We want so-called “verbatim” environments for setting computer code.

```
291 % \RequirePackage{fancyvrb}
```

code We want the code set a bit smaller than normal size.

```
292 \DefineVerbatimEnvironment{code}{Verbatim}{fontsize=\small}
```

smallcode We want something yet smaller if the above size doesn’t fit in the page width.

```
293 \DefineVerbatimEnvironment{smallcode}{Verbatim}{fontsize=\scriptsize}
```

tinycode We want something even smaller yet, in case of dire straits. This is pretty unreadable.

```
294 \DefineVerbatimEnvironment{tinycode}{Verbatim}{fontsize=\tiny}
```

\aftercodething We want something that removes extra space above it, and adds extra space below.

```
295 \newcommand{\aftercodething}[1]{%
296   \unskip
297   #1\par
298   \medskip
299 }
```

We also can use something that formats code more smartly.

```
300 % \RequirePackage{listings}
```

```
301 \lstset{language=C++, numbers=none, basicstyle={\ttfamily}, columns=flexible, showstringspaces=false}
```

\afterlistingthing We want something that removes extra space above it, and adds extra space below.

```
302 \newcommand{\afterlistingthing}[1]{%
303   \unskip\vspace{-5\baselineskip}
304   #1\par
305   \medskip
306 }
```

3.4.2 Figure and Table Things

We group together things that might be found in figures and tables, or used with them.

Images We want to be able to include image files. They can also be included outside figures/tables.

```
307 % \RequirePackage{graphicx}
```

Captions We want to be able to have empty captions.

```
308 % \RequirePackage{caption}
```

\@setupemptycaption We want a caption that has no following colon. We use this to implement the command `\emptycaption`.

```
309 \newcommand{\@setupemptycaption}{%
310   \captionsetup{labelsep=none, justification=centering}
311 }
```

\@setupnonemptycaption We also want a caption that does have a following colon. We use this to implement `\emptycaption`.

```
312 \newcommand{\@setupnonemptycaption}{\captionsetup{labelsep=colon}}
```

\emptycaption We want a command to get an empty caption without a following colon.

```
313 \newcommand{\emptycaption}{%
314   \@setupemptycaption
315   \caption{}
316   \@setupnonemptycaption
317 }
```

Wrapping around Things We want to be able to wrap text around figures or tables. This is the package (out of this, `floatfiy`, and `picinpar`) that works the best. `floatflt` doesn't work if two figures come right after another, and `picinpar` doesn't work if the text is shorter than figure.

```
318 % \RequirePackage{wrapfig}
```

Math Pictures We want to be able to use PicTeX.

We need `rawfonts` to define, e.g., `\fiverm` for PicTeX, which was written before LaTeX2e. We have to put it before loading `pictexwd`.

```
319 % \RequirePackage{rawfonts}
```

\@loadrms We want some more sizes that the package `rawfonts` doesn't define.

```
320 \newcommand{\@loadrms}{%
321 \ifx\undefined\eightrm
322   \font\eightrm=cmr8
323 \fi
324 \ifx\undefined\ninerm
325   \font\ninerm=cmr9
326 \fi
327 \ifx\undefined\twelverm
328   \font\twelverm=cmr12
329 \fi
```

```

330 \ifx\undefined\eighteenrm
331   \font\eighteenrm=cmr17
332 \fi
333 }
334 \@loadrms

335 % \RequirePackage{pictexwd}

```

To use XY, we need more dimension registers than come with standard LaTeX, so we load `etex`.

```
336 % \RequirePackage{etex}
```

We want to typeset graphs and diagrams using TeX.

```
337 % \RequirePackage{xy}
```

```
338 % \RequirePackage{graphics}
```

We need color in some of the PicTeX files.

```
339 % \RequirePackage[usenames]{color}
```

3.5 The Finer Tuning

3.5.1 Breaking Lines by Hand

- \br We want to be able to force line breaks in, e.g., heads. This is disabled in contents lines. Omit the trailing bashslash.

```
340 \newcommand{\br}{\hfill\break}
```

- \BR We want to be able to force line breaks in, e.g., heads. This stays in force in contents lines.

```
341 \newcommand{\BR}{\hfill\break}
```

3.6 Beyond the Traditional Document

3.6.1 Hyperlinking

We want things hyperlinked in the PDF. This has to be loaded before the `chappg` package!

```
342 % \RequirePackage[letterpaper=true,pagebackref=true,linktocpage=true,colorlinks=true]{hyperref}
343 % \RequirePackage{all}{hypcap}
```

- \addcontentsline We modify the `hyperref` definition, not that of `classes`.

Usage:

```
\addcontentsline{file}{sec_unit}{entry}, e.g.,
\addcontentsline{toc}{chapter}{My Chapter Title}.
```

```
344 \renewcommand\addcontentsline[3]{% toc extension, type, tag
345   \begingroup
```

We want to ignore the by-hand breaks `\br` in heads, by turning them back into spaces.

```

346   \renewcommand{\br}{\ }
347   \let\label\@gobble
348   \ifx\@currentHref\@empty
349     \Hy@Warning{%
350       No destination for bookmark of \string\addcontentsline,%
351       \MessageBreak destination is added%
352     }%
353     \phantomsection
354   \fi
355   \expandafter\ifx\csname toclevel@#2\endcsname\relax
356     \begin{group}
357       \def\Hy@tempa{#1}%
358       \ifx\Hy@tempa\Hy@bookmarkstype
359         \Hy@WarningNoLine{%
360           bookmark level for unknown #2 defaults to 0%
361         }%
362       \else
363         \Hy@Info{bookmark level for unknown #2 defaults to 0}%
364       \fi
365     \end{group}
366     \expandafter\gdef\csname toclevel@#2\endcsname{0}%
367   \fi
368   \edef\Hy@toclevel{\csname toclevel@#2\endcsname}%
369   \Hy@writebookmark{\csname the#2\endcsname}%
370   {#3}%
371   {\@currentHref}%
372   {\Hy@toclevel}%
373   {#1}%
374   \ifHy@verbose
375     \def\Hy@tempa{#3}%
376     \onelevel@sanitize\Hy@tempa
377     \typeout{pdftex: bookmark at \the\inputlineno: %
378             {\csname the#2\endcsname}%
379             {\Hy@tempa}%
380             {\@currentHref}%
381             {\Hy@toclevel}%
382             {#1}%
383           }%
384   \fi
385   \addtocontents{#1}{%
386     \protect\contentsline{#2}{#3}{\thepage}{\@currentHref}%
387   }%
388 \end{group}
389 }
```

3.7 Macros to Improve the Look of Things

\cplus We want the plus signs in “C++” to not dominate because of their size.

```
390 \newcommand{\cplus}{\raise.5ex\hbox{\relsize{-3}++}}
```

\e We want scientific notation to be in the form of “times 10 exponential”.

```
391 \newcommand{\e}[1]{\ensuremath{\times 10^{\#1}}}
```

3.7.1 “Assignment By” Operators

We want “assignment by” to be spaced well. Since L^AT_EX has no notion of these operators, we have to space and kern them by hand.

\pluseq {<number variable>} \pluseq {<number variable or number>}
392 \newcommand{\pluseq}{\ensuremath{:+\kern-.25em=}}

\ttpluseq Like \pluseq, but for code mode, not math mode.

```
393 \newcommand{\ttpluseq}{\tt \protect\raisebox{.107ex}{+}=}
```

\minuseq {<number variable>} \minuseq {<number variable or number>}
394 \newcommand{\minuseq}{\ensuremath{:-\kern-.25em=}}

\ttminuseq Like \minuseq, but for code mode, not math mode.

```
395 \newcommand{\ttminuseq}{\tt \protect\raisebox{.273ex}{-}=}
```

\multiplyeq {<number variable>} \multiplyeq {<number variable or number>}
396 \newcommand{\multiplyeq}{\ensuremath{:\ast\kern-.22em=}}

\ttmultiplyeq Like \multiplyeq, but for code mode, not math mode.

```
397 \newcommand{\ttmultiplyeq}{\tt \protect\raisebox{-.066ex}{*}=}
```

\divideeq {<number variable>} \divideeq {<number variable or number>}
398 \newcommand{\divideeq}{\ensuremath{:/\kern-.33em=}}

\ttdivideeq Like \divideeq, but for code mode, not math mode.

```
399 \newcommand{\ttdivideeq}{\texttt{/=}}
```

3.7.2 Small-Cap Words

\scAB To get ADMB.

```
400  
401 \newcommand{\scAB}{\textsc{admb}}
```

\scAR To get ADMB-RE.

```
402 \newcommand{\scAR}{\mbox{\textsc{admb-re}}}
```

\scAD To get AUTODIF.

```
403 \newcommand{\scAD}{\textsc{autodif}}
```

3.7.3 Some Abbreviations

\ADM We don't want everything tied together, as we could get bad line breaks.

404 \newcommand\ADM{AD~Model Builder}

\ADMS Same as \ADM, but followed by a space.

405 \newcommand\ADMS{AD~Model Builder }

406 \makeatother