

Rolling your own Document Class: Using \LaTeX to keep away from the Dark Side

Peter Flynn
Silmaril Consultants

Abstract Document classes in \LaTeX provide automation to improve consistency, productivity, and accuracy in creating and maintaining documents, thereby avoiding the inefficiencies of wordprocessors. However, users who want to package their macros or applications as a document class are often put off by the apparent complexity of the sample classes in the standard distribution. This paper describes what the code in the article document class file does and suggests solutions to some of the popular requirements for changes.

1 Know thine enemy

One of the key features of \TeX systems is the extensibility offered by re-usable pieces of programming called macros. Rudimentary macros exist in many text-handling packages (in fact they were at the heart of the first editors for markup applications), and some wordprocessors make use of general-purpose programming languages such as *Visual Basic* or *Java*; but only typesetters have dedicated languages to doing typesetting, and \TeX 's is by far the most accessible.

This has led to several large and well-known macro packages (\LaTeX , Con \TeX t, \TeX info, *eplain*, etc) which have all but taken the place of Knuth's original language as the end-user's primary interfaces. Most users now only have to use the macro commands of their chosen interface instead of having to write their own macros afresh or maintain a large private collection of personal macros.

This is not to say that there is no place for homebrew macros in plain \TeX : some people have perfectly valid reasons for avoiding the aforementioned packages and continuing to use \TeX in the raw. Using one of the above 'standards' does not always mean that you avoid raw \TeX in your own code, because you may

need some advanced operations which operate at a lower level than normal. It nevertheless remains true that the use of macros to perform groups of frequently-used functions provides a level of automation not found in most wordprocessing systems, and is a major factor in helping users become and remain more productive.

1.1 Standard document classes

The standard document classes installed with L^AT_EX (article, report, book, and letter) were written in a hybrid of L^AT_EX and plain T_EX code. Sometimes this was because the function Lamport wanted was not worth writing a single-use L^AT_EX macro for; sometimes it is because (as Knuth describes in another context) “T_EX is only ‘half obedient’ while these definitions are half finished” [4, p.352]; and sometimes because of the need mentioned above to perform lower-level functions. While the L^AT_EX_{2 ϵ} developers and maintainers have replaced much of the earlier plain T_EX code with updated L^AT_EX equivalents, the code remains fairly dense and is not immediately obvious to the beginner; and the mix of syntax variants can be confusing to the user accustomed to the fairly small set of commands used for common L^AT_EX documents. Plain T_EX itself has some 900 ‘control sequences’ (commands) of which about 350 are ‘primitives’ (indivisible low-level operations), whereas many regular L^AT_EX users get by with some 20–30 commands, if even that.

Users who have started to write their own macros, or who have encountered the need to modify L^AT_EX’s defaults for whatever reason, sometimes find the need to encapsulate their favourite format as a document class, governing the entire document, rather than just a package (style file) handling one or two specific features. In this paper we will dissect one of the common document classes and examine what the features and functions are.

1.2 Caveats

This paper uses the article class as the example. The book and report are structured very similarly and the user who has examined the following sections should have no difficulty in identifying the differences.

The letter class, however, is a very different animal. It implements a vertically-centered format once common in typewritten letters but rarely seen nowadays, and has no provision for many of the features users expect to be able to find in a letter template. For this reason I do not refer any further to this format.

The ConT_EXt system implements a different and extensible set of document classes—including letters—in a radically different manner to L^AT_EX and has been discussed and presented extensively in recent years. The *eplain* macros implement many of the features of the L^AT_EX internal mechanisms, but without imposing any document format at all, leaving the plain T_EX user free to write those herself.

1.3 More background

The essential documentation to read before you start writing your own classes is *L^AT_EX2_ε for class and package writers* [8] (available on all modern T_EX installations by typing `texdoc clsguide`, and *The L^AT_EX Companion* [6, App:A.4]. These describe in detail the additional commands available to class and package authors. There are also some special declarations explained in *Companion* [6, p.847]. The article by Hefferon [3] which I refer to later is a good example of how to build on an existing class. If you have to deal with an obsolete L^AT_EX 2.09 style file, there is an older paper in TUGboat [1].

2 Dissection of `article.cls`

In this example, we use the file from the T_EX Live 2005 distribution (so the line numbers refer to that version only). Lines 1–53 are comments and are omitted here for brevity: they explain where the file came from and how it can be used. This is auto-generated because the document class and package files in the standard distributions of L^AT_EX are derived from master copies maintained in docT_EX (`.dtx`) format [7], which combines documentation and L^AT_EX code in a single file, much in the same way that Knuth’s *WEB* system does for many programming languages [9].¹ A short explanation of the sources of the class files is in the *T_EX FAQ* [2, label:ltxcmds].

1. If you intend making your document class available to the rest of the L^AT_EX community (eg via CTAN), you should make it a docT_EX document so that you can combine documentation with your code. Actually, you should probably be doing this anyway. . .

2.1 Version and identification

The first thing a document class or package must do is identify itself by name, and specify which is the oldest version of L^AT_EX it will work with (it is assumed that it will therefore work with all later versions).

```

54 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
55 \ProvidesClass{article}
56     [2004/02/16 v1.4f
57     Standard LaTeX document class]

```

In your new document class file you should set the date and version to the earliest version you have tested your code with (probably your current version). The name of the document class being provided gets checked against the name requested in the `\documentclass` declaration, and L^AT_EX will give a warning if there is a discrepancy. You may provide a label for the class as well: this will appear in the log file. The linebreaks and indentation are for human readability only.

```

\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesClass{ladingbill} [2006/07/01 v0.01 Bill of Lading
    specialist LaTeX document class]

```

2.2 Initial code and compatibility

On a number of occasions, classes define values as null or a default for later use, so that subsequent code won't trip up as it would if they were undefined. In most cases you will probably need to keep the internal definitions (such as `\@ptsize` here) for use later on (see section 2.4.1 on p. 8).²

2. The use of the '@' sign may be unfamiliar to newcomers: in normal L^AT_EX it is classified as an 'other' character [4, p. 37]. This means it cannot be used as part of a control sequence (command) in your document. But in class and package files, L^AT_EX reclassifies it as a 'letter', and uses it in command definitions which are intended to be inaccessible to the normal user. Its use here indicates that the `\@ptsize` command is going to be given a value that the end-user should not be able to interfere with, or even know exists.

```

58 \newcommand\@optsizet{}
59 \newif\if@restonecol
60 \newif\if@titlepage
61 \@titlepagefalse

```

The conditionals `\if@restonecol` (which flags the restoration of one-column layout after using L^AT_EX’s built-in two-column usage, as distinct from using the `multicol` package) and `\if@titlepage` (which flags use of the separate title-page layout—set to `false` in the following line) are used in the default `\maketitle` command in section 2.4.4 on p. 16. If you’re planning to rewrite `\maketitle` to your own design you may need to take these conditionals into account.³

If you are going to invoke additional packages to provide facilities needed by your options, use the `\RequirePackage` command here, before the options section. If the additional packages are unconnected with your option definitions, use the `\RequirePackage` command after the options are executed (see section 2.3.4 on p. 8).

2.3 Options

In an ideal world we wouldn’t have to support obsolete versions of software, but the L^AT_EX defaults still allow v2.09-type `\documentstyle` declarations to be processed, with a warning. However, for a modern class file this is not necessary, so in your own class you can omit all the tests for `\@ifcompatibility` and their `\else` and terminating `\fi` commands, here and throughout, leaving just the code that was in the `\else` blocks.

2.3.1 Paper sizes

How many paper size options you want to support in your class is entirely up to you. You should allow at least A4 and Letter for normal office work.

3. How much to cater for and how much to ignore will depend on how much your class deviates from the default. Many L^AT_EX users will expect to be able to use options like `twocolumn` and `titlepage` simply because they are available in the default classes. But if you are writing a much more prescriptive format, you may want to remove these options entirely, which means removing all references to conditional flags which depend on them.

```

62 \if@compatibility\else
63 \DeclareOption{a4paper}
64   {\setlength\paperheight {297mm}%
65    \setlength\paperwidth  {210mm}}
66 \DeclareOption{a5paper}
67   {\setlength\paperheight {210mm}%
68    \setlength\paperwidth  {148mm}}
69 \DeclareOption{b5paper}
70   {\setlength\paperheight {250mm}%
71    \setlength\paperwidth  {176mm}}
72 \DeclareOption{letterpaper}
73   {\setlength\paperheight {11in}%
74    \setlength\paperwidth  {8.5in}}
75 \DeclareOption{legalpaper}
76   {\setlength\paperheight {14in}%
77    \setlength\paperwidth  {8.5in}}
78 \DeclareOption{executivepaper}
79   {\setlength\paperheight {10.5in}%
80    \setlength\paperwidth  {7.25in}}
81 \DeclareOption{landscape}
82   {\setlength\@tempdima  {\paperheight}%
83    \setlength\paperheight {\paperwidth}%
84    \setlength\paperwidth  {\@tempdima}}
85 \fi

```

In some cases you may be writing for a highly specific environment such as your own office or employer, where only one size is required. If so, just omit all the other declarations and add the one option to the `\ExecuteOptions` command (see section 2.3.4 on p. 8).

2.3.2 Type sizes and layout options

As mentioned above, the compatibility settings in this block can be removed in your own class, because modern class files use default option settings via the `\DeclareOption` command instead.

```

86 \if@compatibility
87   \renewcommand\@ptsize{0}
88 \else
89   \DeclareOption{10pt}{\renewcommand\@ptsize{0}}
90   \fi
91   \DeclareOption{11pt}{\renewcommand\@ptsize{1}}
92   \DeclareOption{12pt}{\renewcommand\@ptsize{2}}
93 \if@compatibility\else

```

```

94 \DeclareOption{oneside}{\@twosidefalse \@mparswitchfalse}
95 \fi
96 \DeclareOption{twoside}{\@twosidetrue \@mparswitchtrue}
97 \DeclareOption{draft}{\setlength\overfullrule{5pt}}
98 \ifcompatibility\else
99 \DeclareOption{final}{\setlength\overfullrule{0pt}}
100 \fi
101 \DeclareOption{titlepage}{\@titlepagetrue}
102 \ifcompatibility\else
103 \DeclareOption{notitlepage}{\@titlepagefalse}
104 \fi
105 \ifcompatibility\else
106 \DeclareOption{onecolumn}{\@twocolumnfalse}
107 \fi
108 \DeclareOption{twocolumn}{\@twocolumntrue}
109 \DeclareOption{leqno}{\input{leqno.clo}}
110 \DeclareOption{fleqn}{\input{fleqn.clo}}
111 \DeclareOption{openbib}{%
112 \AtEndOfPackage{%
113 \renewcommand\@openbib@code{%
114 \advance\leftmargin\bibindent
115 \itemindent -\bibindent
116 \listparindent \itemindent
117 \parsep \z@
118 }%
119 \renewcommand\newblock{\par}}%
120 }

```

The other options should probably be retained, as users may expect them to work, bearing in mind the comment about two-column and title-page settings above. Note that the openbib declaration is 10 lines long, and defers itself to end of the package as a `\renewcommand` so that it doesn't conflict with the command being declared later.

As with paper sizes above, if your class only needs one specific size setup, just invoke it in `\ExecuteOptions`.

2.3.3 Your own options

Now is the time to add your own option declarations, if any. Note that option names have no backslash; otherwise the `\DeclareOption` command works the same way as the `\newcommand` command (but with no parameters).

Details of how to preserve the options of an existing class you are 'borrowing' via the `\LoadClass` command are discussed in section 3.1 on p. 31.

2.3.4 Applying options

Two commands control when the options are applied:

```
121 \ExecuteOptions{letterpaper,10pt,oneside,onecolumn,final}  
122 \ProcessOptions
```

`\ExecuteOptions` applies all the options you specify in the argument, in order, as your selected defaults. The `\ProcessOptions` command then applies any options the user has selected in their `\documentclass` declaration.

2.4 Layout

A large number of size and shape settings depend on the selected point size (default 10pt, otherwise as selected in your options). The exact sizes of type chosen for all the different type-size commands are kept in three Class Option files, `size10.clo`, `size11.clo`, and `size12.clo`. There are some others available from CTAN, such as James Kilfiger’s `size14.clo` for readers needing larger type editions, but the three mentioned above cover the vast majority of normal text setting.

If you are going to invoke additional packages that are unconnected with your option definitions, put the `\RequirePackage` commands here (see section 3.2 on p. 31). Be aware that some packages expect certain variables or definitions already to be present, so their invocation may need to be deferred until after everything else. In this case, enclose the `\RequirePackage` command in a `\AtEndOfPackage` or `\AtBeginDocument` command. This will invoke the package[s] at the specified point in processing, and thus avoid error messages or interference with code in the class file that has not yet been executed.

2.4.1 Type size

To invoke the right settings, the `\@ptsize` command is embedded in the argument to an `\input` command:

```
123 \input{size1\@ptsize.clo}  
124 \setlength\lineskip{1\p@}
```

```

125 \setlength\normallineskip{1\p@}
126 \renewcommand\baselinestretch{}
127 \setlength\parskip{0\p@ \@plus \p@}

```

A number of basic settings are then made using the internal definition of a point (\p@). The class option files contain a lot of other size-specific settings as well as the font size specifications for the chosen body size.

2.4.1.1 Identity and basic sizes The class option files identify themselves in the same way as class files, but using the \ProvidesFile instead of \ProvidesClass.

```

----- size10.clo -----
54 %% Grave accent \‘ Left brace \{ Vertical bar \|
55 %% Right brace \} Tilde \~}
56 \ProvidesFile{size10.clo}
57 [2004/02/16 v1.4f
58 Standard LaTeX file (size option)]
59 \renewcommand\normalsize{%
60 \setfontsize\normalsize\@xpt\@xiipt
61 \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
62 \abovedisplayshortskip \z@ \@plus3\p@
63 \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
64 \belowdisplayskip \abovedisplayskip
65 \let\@listi\@listI}
66 \normalsize
67 \newcommand\small{%
68 \setfontsize\small\@ixpt{11}%
69 \abovedisplayskip 8.5\p@ \@plus3\p@ \@minus4\p@
70 \abovedisplayshortskip \z@ \@plus2\p@
71 \belowdisplayshortskip 4\p@ \@plus2\p@ \@minus2\p@
72 \def\@listi{\leftmargin\leftmargini
73 \topsep 4\p@ \@plus2\p@ \@minus2\p@
74 \parsep 2\p@ \@plus\p@ \@minus\p@
75 \itemsep \parsep}%
76 \belowdisplayskip \abovedisplayskip
77 }
78 \newcommand\footnotesize{%
79 \setfontsize\footnotesize\@viipt{9.5}%
80 \abovedisplayskip 6\p@ \@plus2\p@ \@minus4\p@
81 \abovedisplayshortskip \z@ \@plus\p@
82 \belowdisplayshortskip 3\p@ \@plus\p@ \@minus2\p@
83 \def\@listi{\leftmargin\leftmargini
84 \topsep 3\p@ \@plus\p@ \@minus\p@
85 \parsep 2\p@ \@plus\p@ \@minus\p@
86 \itemsep \parsep}%
87 \belowdisplayskip \abovedisplayskip
88 }
89 \newcommand\scriptsize{\setfontsize\scriptsize\@viipt\@viipt}

```

```

90 \newcommand\tiny{\@setfontsize\tiny\@vpt\@vpt}
91 \newcommand\large{\@setfontsize\large\@xiipt{14}}
92 \newcommand\Large{\@setfontsize\Large\@xivpt{18}}
93 \newcommand\LARGE{\@setfontsize\LARGE\@xviipt{22}}

```

The first block defines the standard L^AT_EX sizes. These are named using roman numerals (eg \@xiipt for 12pt) because digits are not allowed in control sequence names. The more frequently-used sizes also define the display math spacing and the spacing for top-level lists (list definition names also use roman numerals like \@listi).

2.4.1.2 Spacing This section controls paragraph indentation (differing between one-column and two-column setting); the dimensions of the three ‘shortcut’ spacing commands (small, med, and big) but not the actual commands themselves, which are defined in L^AT_EX itself; and some top-of-page and bottom-of-page spacing settings (normally reset using the geometry package).

```

----- size10.clo -----
94 \newcommand\huge{\@setfontsize\huge\@xxpt{25}}
95 \newcommand\Huge{\@setfontsize\Huge\@xxvpt{30}}
96 \if@twocolumn
97 \setlength\parindent{1em}
98 \else
99 \setlength\parindent{15\p@}
100 \fi
101 \setlength\smallskipamount{3\p@ \@plus 1\p@ \@minus 1\p@}
102 \setlength\medskipamount{6\p@ \@plus 2\p@ \@minus 2\p@}
103 \setlength\bigskipamount{12\p@ \@plus 4\p@ \@minus 4\p@}
104 \setlength\headheight{12\p@}
105 \setlength\headsep {25\p@}
106 \setlength\topskip {10\p@}
107 \setlength\footskip{30\p@}

```

2.4.1.3 Text area Text width and text height are set to depend on the columnar setting and a multiple of line-heights respectively.

```

----- size10.clo -----
108 \if@compatibility \setlength\maxdepth{4\p@} \else
109 \setlength\maxdepth{.5\topskip} \fi
110 \if@compatibility
111 \if@twocolumn

```

```

112     \setlength\textwidth{410\p@}
113     \else
114     \setlength\textwidth{345\p@}
115     \fi
116 \else
117     \setlength\@tempdima{\paperwidth}
118     \addtolength\@tempdima{-2in}
119     \setlength\@tempdimb{345\p@}
120     \if@twocolumn
121     \ifdim\@tempdima>2\@tempdimb\relax
122     \setlength\textwidth{2\@tempdimb}
123     \else
124     \setlength\textwidth{\@tempdima}
125     \fi
126     \else
127     \ifdim\@tempdima>\@tempdimb\relax
128     \setlength\textwidth{\@tempdimb}
129     \else
130     \setlength\textwidth{\@tempdima}
131     \fi
132     \fi
133     \fi
134 \ifcompatibility\else
135     \@settopoint\textwidth
136     \fi
137 \ifcompatibility
138     \setlength\textheight{43\baselineskip}
139 \else
140     \setlength\@tempdima{\paperheight}
141     \addtolength\@tempdima{-2in}
142     \addtolength\@tempdima{-1.5in}
143     \divide\@tempdima\baselineskip
144     \@tempcnta=\@tempdima
145     \setlength\textheight{\@tempcnta\baselineskip}

```

(The compatibility-mode settings were absolute values in points.) As with paper size and type size, you can preselect exact values for the text area and margins (see next section) using the geometry package.

2.4.1.4 Page margins Margins also depend on the column settings, and on the one-side/two-side setting.

```

----- size10.clo -----
146     \fi
147     \addtolength\textheight{\topskip}
148     \if@twocolumn
149     \setlength\marginparsep {10\p@}
150     \else

```

```

151 \setlength\marginparsep{11\p@}
152 \fi
153 \setlength\marginparpush{5\p@}
154 \ifcompatibility
155 \if@twoside
156 \setlength\oddsidemargin {44\p@}
157 \setlength\evensidemargin {82\p@}
158 \setlength\marginparwidth {107\p@}
159 \else
160 \setlength\oddsidemargin {63\p@}
161 \setlength\evensidemargin {63\p@}
162 \setlength\marginparwidth {90\p@}
163 \fi
164 \if@twocolumn
165 \setlength\oddsidemargin {30\p@}
166 \setlength\evensidemargin {30\p@}
167 \setlength\marginparwidth {48\p@}
168 \fi
169 \else
170 \if@twoside
171 \setlength\@tempdima {\paperwidth}
172 \addtolength\@tempdima {-\textwidth}
173 \setlength\oddsidemargin {.4\@tempdima}
174 \addtolength\oddsidemargin {-1in}
175 \setlength\marginparwidth {.6\@tempdima}
176 \addtolength\marginparwidth {-\marginparsep}
177 \addtolength\marginparwidth {-0.4in}
178 \else
179 \setlength\@tempdima {\paperwidth}
180 \addtolength\@tempdima {-\textwidth}
181 \setlength\oddsidemargin {.5\@tempdima}
182 \addtolength\oddsidemargin {-1in}
183 \setlength\marginparwidth {.5\@tempdima}
184 \addtolength\marginparwidth {-\marginparsep}
185 \addtolength\marginparwidth {-0.4in}
186 \addtolength\marginparwidth {-0.4in}
187 \fi
188 \ifdim \marginparwidth >2in
189 \setlength\marginparwidth{2in}
190 \fi
191 \@settopoint\oddsidemargin
192 \@settopoint\marginparwidth
193 \setlength\evensidemargin {\paperwidth}
194 \addtolength\evensidemargin{-2in}
195 \addtolength\evensidemargin{-\textwidth}
196 \addtolength\evensidemargin{-\oddsidemargin}
197 \@settopoint\evensidemargin
198 \fi
199 \ifcompatibility
200 \setlength\topmargin{27pt}
201 \else
202 \setlength\topmargin{\paperheight}
203 \addtolength\topmargin{-2in}

```

```

204 \addtolength\topmargin{-\headheight}
205 \addtolength\topmargin{-\headsep}
206 \addtolength\topmargin{-\textheight}
207 \addtolength\topmargin{-\footskip} % this might be wrong!
208 \addtolength\topmargin{-.5\topmargin}

```

Again, the compatibility-mode settings are absolute, whereas the modern defaults are computed.

2.4.1.5 Footnote space Spacing for footnotes and floats is flexible (plus and minus a certain amount) so that the page-breaking routine doesn't become too rigid.

```

----- size10.clo -----
209 \settopoint\topmargin
210 \fi
211 \setlength\footnoteseq{6.65\p@}
212 \setlength{\skip\footins}{9\p@ \@plus 4\p@ \@minus 2\p@}
213 \setlength\floatsep {12\p@ \@plus 2\p@ \@minus 2\p@}
214 \setlength\textfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
215 \setlength\intextsep {12\p@ \@plus 2\p@ \@minus 2\p@}
216 \setlength\dblfloatsep {12\p@ \@plus 2\p@ \@minus 2\p@}
217 \setlength\dbltextfloatsep{20\p@ \@plus 2\p@ \@minus 4\p@}
218 \setlength\@fptop{0\p@ \@plus 1fil}
219 \setlength\@fpsep{8\p@ \@plus 2fil}
220 \setlength\@fpbot{0\p@ \@plus 1fil}
221 \setlength\@dblftop{0\p@ \@plus 1fil}
222 \setlength\@dblfpsep{8\p@ \@plus 2fil}

```

2.4.1.6 Lists Finally, for the values dependent on type size, the dimensions of lists are set. As mentioned above, names are fabricated using roman numerals (i to vi).

```

----- size10.clo -----
223 \setlength\@dblfpbot{0\p@ \@plus 1fil}
224 \setlength\partopsep{2\p@ \@plus 1\p@ \@minus 1\p@}
225 \def\@listi{\leftmargin\leftmargini
226 \parsep 4\p@ \@plus2\p@ \@minus\p@
227 \topsep 8\p@ \@plus2\p@ \@minus4\p@
228 \itemsep4\p@ \@plus2\p@ \@minus\p@}
229 \let\@listI\@listi
230 \@listi
231 \def\@listii {\leftmargin\leftmarginii
232 \labelwidth\leftmarginii
233 \advance\labelwidth-\labelsep

```

```

234         \topsep    4\p@ \@plus2\p@ \@minus\p@
235         \parsep    2\p@ \@plus\p@ \@minus\p@
236         \itemsep    \parsep}
237 \def\@listiii{\leftmargin\leftmarginiii
238         \labelwidth\leftmarginiii
239         \advance\labelwidth-\labelsep
240         \topsep    2\p@ \@plus\p@\@minus\p@
241         \parsep    \z@
242         \partopsep \p@ \@plus\z@ \@minus\p@
243         \itemsep    \topsep}
244 \def\@listiv {\leftmargin\leftmarginiv
245         \labelwidth\leftmarginiv
246         \advance\labelwidth-\labelsep}
247 \def\@listv  {\leftmargin\leftmarginv
248         \labelwidth\leftmarginv
249         \advance\labelwidth-\labelsep}
250 \def\@listvi {\leftmargin\leftmarginvi
251         \labelwidth\leftmarginvi

```

2.4.2 Spacing penalties

Three penalties are set which get invoked in various decisions on paragraph-breaking. You probably don't want to change these unless you are doing deep surgery.

article.cls

```

128 \@lowpenalty 51
129 \@medpenalty 151
130 \@highpenalty 301
131 \setcounter{topnumber}{2}
132 \renewcommand\topfraction{.7}
133 \setcounter{bottomnumber}{1}
134 \renewcommand\bottomfraction{.3}
135 \setcounter{totalnumber}{3}
136 \renewcommand\textfraction{.2}
137 \renewcommand\floatpagefraction{.5}
138 \setcounter{dbltopnumber}{2}
139 \renewcommand\dbltopfraction{.7}
140 \renewcommand\dblfloatpagefraction{.5}

```

The fractions and numbers refer to the proportions of the page that can be taken up by figures and tables, and the number of floats allowed, when calculating the location of floats.

2.4.3 Running heads

Depending on the imposition (one-sided or two-sided), the default running heads are specified as in the *L^AT_EX book* [5].

```
article.cls
141 \if@twoside
142   \def\ps@headings{%
143     \let\@oddfoot\@empty\let\@evenfoot\@empty
144     \def\@evenhead{\thepage\hfil\slshape\leftmark}%
145     \def\@oddhead{\slshape\rightmark\hfil\thepage}%
146     \let\@mkboth\markboth
147     \def\sectionmark##1{%
148       \markboth {\MakeUppercase{%
149         \ifnum \c@secnumdepth >\z@
150           \thesection\quad
151         \fi
152         ##1}}{}}%
153     \def\subsectionmark##1{%
154       \markright {%
155         \ifnum \c@secnumdepth >\@ne
156           \thesubsection\quad
157         \fi
158         ##1}}}}
159 \else
160   \def\ps@headings{%
161     \let\@oddfoot\@empty
162     \def\@oddhead{\slshape\rightmark\hfil\thepage}%
163     \let\@mkboth\markboth
164     \def\sectionmark##1{%
165       \markright {\MakeUppercase{%
166         \ifnum \c@secnumdepth >\m@ne
167           \thesection\quad
168         \fi
169         ##1}}}}
170 \fi
171 \def\ps@myheadings{%
172   \let\@oddfoot\@empty\let\@evenfoot\@empty
173   \def\@evenhead{\thepage\hfil\slshape\leftmark}%
174   \def\@oddhead{\slshape\rightmark\hfil\thepage}%
175   \let\@mkboth\@gobbletwo
176   \let\sectionmark\@gobble
177   \let\subsectionmark\@gobble
178 }
```

In many cases it may be preferable to use the `fancyhdr` package instead. This lets you specify a very wide range of header and footer layouts, with left/right switching for double-sided work.

2.4.4 Titling

This is possibly the first big change you'll need to make. There are two `\maketitle` commands defined, one for use on a separate title page (without facilities for attribution), and one for normal use on the starting page (with attributions, and allowing for two columns, using the `\@maketitle` command as well). Both are controlled by the `\if@titlepage` switch.

```

----- article.cls -----
179 \if@titlepage
180 \newcommand\maketitle{\begin{titlepage}%
181 \let\footnotesize\small
182 \let\footnoterule\relax
183 \let \footnote \thanks
184 \null\vfil
185 \vskip 60\p@
186 \begin{center}%
187 {\LARGE \@title \par}%
188 \vskip 3em%
189 {\large
190 \lineskip .75em%
191 \begin{tabular}[t]{c}%
192 \@author
193 \end{tabular}\par}%
194 \vskip 1.5em%
195 {\large \@date \par}% % Set date in \large size.
196 \end{center}\par
197 \@thanks
198 \vfil\null
199 \end{titlepage}%
200 \setcounter{footnote}{0}%
201 \global\let\thanks\relax
202 \global\let\maketitle\relax
203 \global\let\@thanks\@empty
204 \global\let\@author\@empty
205 \global\let\@date\@empty
206 \global\let\@title\@empty
207 \global\let\title\relax
208 \global\let\author\relax
209 \global\let\date\relax
210 \global\let\and\relax
211 }
212 \else
213 \newcommand\maketitle{\par
214 \begingroup
215 \renewcommand\thefootnote{\@fnsymbol\c@footnote}%
216 \def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}}%
217 \long\def\@makefntext##1{\parindent 1em\noindent
218 \hbext@1.8em{%
219 \hss\@textsuperscript{\normalfont\@thefnmark}}##1}%
220 \if@twocolumn
```

```

221     \ifnum \col@number=\@ne
222         \@maketitle
223     \else
224         \twocolumn[\@maketitle]%
225     \fi
226 \else
227     \newpage
228     \global\@topnum\z@    % Prevents figures from going at top of page.
229     \@maketitle
230     \fi
231     \thispagestyle{plain}\@thanks
232 \endgroup
233 \setcounter{footnote}{0}%
234 \global\let\thanks\relax
235 \global\let\maketitle\relax
236 \global\let\@maketitle\relax
237 \global\let\@thanks\@empty
238 \global\let\@author\@empty
239 \global\let\@date\@empty
240 \global\let\@title\@empty
241 \global\let\title\relax
242 \global\let\author\relax
243 \global\let\date\relax
244 \global\let\and\relax
245 }
246 \def\@maketitle{%
247     \newpage
248     \null
249     \vskip 2em%
250     \begin{center}%
251     \let \footnote \thanks
252     {\LARGE \@title \par}%
253     \vskip 1.5em%
254     {\large
255         \lineskip .5em%
256         \begin{tabular}[t]{c}%
257             \@author
258         \end{tabular}\par}%
259     \vskip 1em%
260     {\large \@date}%
261     \end{center}%
262     \par
263     \vskip 1.5em}
264 \fi

```

In all of these you can redefine the size, location, and spacing of the three basic titling elements, \@title, \@author, and \@date. If you are not using two-column setting, or a title-page option, you could replace the whole lot with a single `\renewcommand{\maketitle}{...}` of your own design

You can also make up your own additional elements, for example an optional subtitle:

```
\def\@subtitle{\relax}
\newcommand{\subtitle}[1]{\gdef\@subtitle{#1}}
\renewcommand{\maketitle}{
  \begin{titlepage}
    \huge\@author\par
    \Large\@title\par
    \if\@subtitle\relax\else\large\@subtitle\par\fi
    \normalsize\@date\par
  \end{titlepage}
}
```

This lets the phantom `\@subtitle` exist unused, set to `\relax` unless an author explicitly uses the `\subtitle` command, because the titling routine can test whether it is still set to `\relax`, and if not, format it accordingly. This technique can be used to add many of the items of metadata used by publishers, such as author affiliations, email and web addresses, and dates of submission.

2.5 Structure

Unless you are doing a very rigid class for data-handling, you probably want to keep the basic sectional structures for normal continuous text as they are, and only change the formatting.

```
----- article.cls -----
265 \setcounter{secnumdepth}{3}
266 \newcounter {part}
267 \newcounter {section}
268 \newcounter {subsection}[section]
269 \newcounter {subsubsection}[subsection]
270 \newcounter {paragraph}[subsubsection]
271 \newcounter {subparagraph}[paragraph]
272 \renewcommand \thepart {\@Roman\c@part}
273 \renewcommand \thesection {\@arabic\c@section}
274 \renewcommand \thesubsection {\thesection.\@arabic\c@subsection}
275 \renewcommand \thesubsubsection{\thesubsection .\@arabic\c@subsubsection}
276 \renewcommand \theparagraph {\thesubsubsection.\@arabic\c@paragraph}
277 \renewcommand \thesubparagraph {\theparagraph.\@arabic\c@subparagraph}
```

```

278 \newcommand\part{%
279   \if@noskipsec \leavevmode \fi
280   \par
281   \addvspace{4ex}%
282   \@afterindentfalse
283   \secdef\@part\@spart}

```

The `\part` command is defined separately, as it operates like `\chapter` in other classes, with more space and a prefix (the book and report classes define a separate `\chapter` command).

```

----- article.cls -----
285 \def\@part[#1]#2{%
286   \ifnum \c@secnumdepth >\m@ne
287     \refstepcounter{part}%
288     \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
289   \else
290     \addcontentsline{toc}{part}{#1}%
291   \fi
292   {\parindent \z@ \raggedright
293     \interlinepenalty \@M
294     \normalfont
295     \ifnum \c@secnumdepth >\m@ne
296       \Large\bfseries \partname\nobreakspace\thepart
297       \par\nobreak
298     \fi
299     \huge \bfseries #2%
300     \markboth{}{\par}%
301     \nobreak
302     \vskip 3ex
303     \@afterheading}
304 \def\@spart#1{%
305   {\parindent \z@ \raggedright
306     \interlinepenalty \@M
307     \normalfont
308     \huge \bfseries #1\par}%
309   \nobreak
310   \vskip 3ex
311   \@afterheading}

```

The sectional formatting is one of the most common features of a document class that need to change. Details of the operation of the `\@startsection` command are in the *L^AT_EX book* [5] if you want to do a complete rewrite, but in many cases one of the packages like `sectsty` can be used to change fonts or spacing without you having to redo everything from scratch.

```

312 \newcommand\section{\@startsection {section}{1}{\z@}%
313         {-3.5ex \@plus -1ex \@minus -.2ex}%
314         {2.3ex \@plus .2ex}%
315         {\normalfont\Large\bfseries}}
316 \newcommand\subsection{\@startsection{subsection}{2}{\z@}%
317         {-3.25ex\@plus -1ex \@minus -.2ex}%
318         {1.5ex \@plus .2ex}%
319         {\normalfont\large\bfseries}}
320 \newcommand\subsubsection{\@startsection{subsubsection}{3}{\z@}%
321         {-3.25ex\@plus -1ex \@minus -.2ex}%
322         {1.5ex \@plus .2ex}%
323         {\normalfont\normalsize\bfseries}}
324 \newcommand\paragraph{\@startsection{paragraph}{4}{\z@}%
325         {3.25ex \@plus 1ex \@minus .2ex}%
326         {-1em}%
327         {\normalfont\normalsize\bfseries}}
328 \newcommand\subparagraph{\@startsection{subparagraph}{5}{\parindent}%
329         {3.25ex \@plus 1ex \@minus .2ex}%
330         {-1em}%
331         {\normalfont\normalsize\bfseries}}

```

2.6 Indents and margins

In this section the class file defines the internal margins set around block elements like lists. For controlling lists, L^AT_EX provides four levels of indentation. As explained earlier, because digits are not permitted in command names, all these parameters end in the Roman-numeral equivalents.

```

332 \if@twocolumn
333     \setlength\leftmargini {2em}
334 \else
335     \setlength\leftmargini {2.5em}
336 \fi
337 \leftmargin \leftmargini
338 \setlength\leftmarginii {2.2em}
339 \setlength\leftmarginiii {1.87em}
340 \setlength\leftmarginiv {1.7em}
341 \if@twocolumn
342     \setlength\leftmarginv {.5em}
343     \setlength\leftmarginvi {.5em}
344 \else
345     \setlength\leftmarginv {1em}
346     \setlength\leftmarginvi {1em}
347 \fi
348 \setlength \labelsep {.5em}

```

```

349 \setlength \labelwidth{\leftmargini}
350 \addtolength\labelwidth{-\labelsep}
351 \@beginparpenalty -\@lowpenalty
352 \@endparpenalty -\@lowpenalty
353 \@itempenalty -\@lowpenalty
354 \renewcommand\theenumi{\@arabic\c@enumi}
355 \renewcommand\theenumii{\@alph\c@enumii}
356 \renewcommand\theenumiii{\@roman\c@enumiii}
357 \renewcommand\theenumiv{\@Alph\c@enumiv}
358 \newcommand\labelenumi{\theenumi.}
359 \newcommand\labelenumii{(\theenumii)}
360 \newcommand\labelenumiii{\theenumiii.}
361 \newcommand\labelenumiv{\theenumiv.}
362 \renewcommand\p@enumii{\theenumi}
363 \renewcommand\p@enumiii{\theenumi(\theenumii)}
364 \renewcommand\p@enumiv{\p@enumiii\theenumiii}
365 \newcommand\labelitemi{\textbullet}
366 \newcommand\labelitemii{\normalfont\bfseries \textendash}
367 \newcommand\labelitemiii{\textasteriskcentered}
368 \newcommand\labelitemiv{\textperiodcentered}
369 \newenvironment{description}
370     {\list{}{\labelwidth\z@ \itemindent-\leftmargin
371             \let\makelabel\descriptionlabel}}
372     {\endlist}
373 \newcommand*\descriptionlabel[1]{\hspace\labelsep
374     \normalfont\bfseries #1}

```

The variables and their meaning are described in more detail in *L^AT_EX book* [5] and *Companion* [6], but essentially:

`\leftmarginrr` are the list level indentations from outer page margin to the start of the text;

`\labelsep` is the space between the number or bullet and the start of the text;

`\labelwidth` is how much space to allow for the numbering or bulleting;

`\theenumrr` controls the style of numbering;

`\labelenumrr` controls the style of bulleting.

In all these cases, you can remove the conditional code surrounding the variants for two-column work, and have just one setting, if you are not going to provide for two-column setting.

The description environment works slightly differently: the `\makelabel` command is equated to a `\descriptionlabel` command to indent and format the

item label. This is easily redefined, for example to make the labels use the sans-serif font instead of the default roman typeface, and add an automatic em-rule afterwards:

```
\renewcommand*\descriptionlabel[1]{\hspace\labelsep
\relax\sffamily{\bfseries #1}~---\space\ignorespaces}
```

2.7 Abstract

The default abstract is formatted differently according to where it appears: on the first page or on a page by itself after a separate title page.

```

----- article.cls -----
375 \if@titlepage
376 \newenvironment{abstract}{%
377 \titlepage
378 \null\vfil
379 \@beginparpenalty\@lowpenalty
380 \begin{center}%
381 \bfseries \abstractname
382 \@endparpenalty\@M
383 \end{center}}%
384 {\par\vfil\null\endtitlepage}
385 \else
386 \newenvironment{abstract}{%
387 \if@twocolumn
388 \section*{\abstractname}%
389 \else
390 \small
391 \begin{center}%
392 {\bfseries \abstractname\vspace{-.5em}\vspace{\z@}}%
393 \end{center}%
394 \quotation
395 \fi}
396 {\if@twocolumn\else\endquotation\fi}
397 \fi
```

One common requirement is for the Abstract formatting to follow the pattern of a subsection when it appears on a separate page, eg

```
\newenvironment{abstract}{%
\titlepage
```

```

\subsection*{\abstractname}}%
{\par\vfil\null\endtitlepage}

```

Some styles require turning off the initial indentation when the Abstract is on the first page, for consistency with the default Anglo-American style used in sections:

```

\newenvironment{abstract}{%
  \if@twocolumn
    \section*{\abstractname}%
  \else
    \small
    \begin{center}%
      {\bfseries \abstractname\vspace{-.5em}\vspace{\z@}}%
    \end{center}%
    \quotation\noindent\ignorespaces
  \fi}
{\if@twocolumn\else\endquotation\fi}

```

Note that if you will be adding to an existing class in the manner described in section 3.1 on p.31, these last two examples will use the `\renewenvironment` command instead.

2.8 Structural elements

The default classes contain some rudimentary environments for verse and quotations, and a compatibility setting for L^AT_EX 2.09 users, which can be omitted from new classes (make sure you keep one definition of the titlepage environment, though!

```

----- article.cls -----
398 \newenvironment{verse}
399     {\let\\\@centercr
400     \list{}{\itemsep \z@
401         \itemindent -1.5em%
402         \listparindent\itemindent
403         \rightmargin \leftmargin}

```

```

404             \advance\leftmargin 1.5em}%
405             \item\relax}
406         {\endlist}
407 \newenvironment{quotation}
408     {\list{}{\listparindent 1.5em%
409             \itemindent \listparindent
410             \rightmargin \leftmargin
411             \parsep \z@ \@plus\p@}%
412     \item\relax}
413     {\endlist}
414 \newenvironment{quote}
415     {\list{}{\rightmargin\leftmargin}%
416     \item\relax}
417     {\endlist}
418 \if@compatibility
419 \newenvironment{titlepage}
420     {%
421     \if@twocolumn
422     \@restonecoltrue\onecolumn
423     \else
424     \@restonecolfalse\newpage
425     \fi
426     \thispagestyle{empty}%
427     \setcounter{page}\z@
428     }%
429     {\if@restonecol\twocolumn \else \newpage \fi
430     }
431 \else
432 \newenvironment{titlepage}
433     {%
434     \if@twocolumn
435     \@restonecoltrue\onecolumn
436     \else
437     \@restonecolfalse\newpage
438     \fi
439     \thispagestyle{empty}%
440     \setcounter{page}\@ne
441     }%
442     {\if@restonecol\twocolumn \else \newpage \fi
443     \if@twoside\else
444     \setcounter{page}\@ne
445     \fi
446     }
447 \fi
448 \newcommand\appendix{\par
449     \setcounter{section}{0}%
450     \setcounter{subsection}{0}%
451     \gdef\thesection{\@Alph\c@section}}

```

The quotation environment is another which benefits from the removal of the initial indentation:

```

\newenvironment{quotation}
  {\list{}{\listparindent 1.5em%
  \itemindent \z@
  \rightmargin \leftmargin
  \parsep \z@ \@plus\p@}%
  \item\relax}
{\endlist}

```

For the reasons noted in section 2.7 on p. 22, this may need to be a `\renewcommand`.

This section ends with a the definition for `\appendix` which switches the `\section` settings to produce labels with A, B, C, etc instead of 1, 2, 3.

2.9 Figures and Tables

These are controlled by a number of dimensions which you may already be familiar with, such as `\tabcolsep` for the gap between table columns. The `\fboxsep` and `\fboxrule` dimensions control the gap and rule thickness around boxed text.

article.cls

```

452 \setlength\arraycolsep{5\p@}
453 \setlength\tabcolsep{6\p@}
454 \setlength\arrayrulewidth{.4\p@}
455 \setlength\doublerulesep{2\p@}
456 \setlength\tabbingsep{\labelsep}
457 \skip\@mpfootins = \skip\footins
458 \setlength\fboxsep{3\p@}
459 \setlength\fboxrule{.4\p@}
460 \renewcommand \theequation {\@arabic\c@equation}
461 \newcounter{figure}
462 \renewcommand \thefigure {\@arabic\c@figure}
463 \def\fps@figure{tbp}
464 \def\ftype@figure{1}
465 \def\ext@figure{lof}
466 \def\fnm@figure{\figurename\nobreakspace\thefigure}
467 \newenvironment{figure}
468     {\@float{figure}}
469     {\end@float}
470 \newenvironment{figure*}
471     {\@dblfloat{figure}}
472     {\end@dblfloat}
473 \newcounter{table}
474 \renewcommand\thetable{\@arabic\c@table}

```

```

475 \def\fps@table{tbp}
476 \def\ftype@table{2}
477 \def\ext@table{lot}
478 \def\fnm@table{\tablename\nobreakspace\thetable}
479 \newenvironment{table}
480     {\@float{table}}
481     {\end@float}
482 \newenvironment{table*}
483     {\@dblfloat{table}}
484     {\end@dblfloat}
485 \newlength\abovecaptionskip
486 \newlength\belowcaptionskip
487 \setlength\abovecaptionskip{10\p@}
488 \setlength\belowcaptionskip{0\p@}
489 \long\def\@makecaption#1#2{%
490     \vskip\abovecaptionskip
491     \sbox\@tempboxa{#1: #2}%
492     \ifdim \wd\@tempboxa >\hsize
493         #1: #2\par
494     \else
495         \global \@minipagefalse
496         \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
497     \fi
498     \vskip\belowcaptionskip}

```

At the end of this section is the `\@makecaption` command, another popular candidate for redesign, but consider also using the `ccaption` package.

2.10 Legacy support

The obsolescent commands `\rm`, `\it`, `\bf`, etc are declared here to function as their modern equivalents.

```

----- article.cls -----
499 \DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
500 \DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
501 \DeclareOldFontCommand{\tt}{\normalfont\ttfamily}{\mathtt}
502 \DeclareOldFontCommand{\bf}{\normalfont\bfseries}{\mathbf}
503 \DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
504 \DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}
505 \DeclareOldFontCommand{\sc}{\normalfont\scshape}{\@nomath\sc}
506 \DeclareRobustCommand*\cal{\@fontswitch\relax\mathcal}
507 \DeclareRobustCommand*\mit{\@fontswitch\relax\mathnormal}

```

2.11 Table of Contents

The Table of Contents section starts with some commands which evaluate to dimensions, plus the `\tableofcontents` command itself.

article.cls

```
508 \newcommand\@pnumwidth{1.55em}
509 \newcommand\@tocrmarg{2.55em}
510 \newcommand\@dotsep{4.5}
511 \setcounter{tocdepth}{3}
512 \newcommand\tableofcontents{%
513   \section*\@contentsname
514     \@mkboth{%
515       \MakeUppercase\@contentsname}{\MakeUppercase\@contentsname}}%
516   \@starttoc{toc}%
517 }
518 \newcommand*\l@part[2]{%
519   \ifnum \c@tocdepth >-2\relax
520     \addpenalty\@secpenalty
521     \addvspace{2.25em \@plus\p@}%
522     \setlength\@tempdima{3em}%
523     \begingroup
524       \parindent \z@ \rightskip \@pnumwidth
525       \parfillskip -\@pnumwidth
526       {\leavevmode
527         \large \bfseries #1\hfil \hb@xt@\@pnumwidth{\hss #2}}\par
528         \nobreak
529         \if@compatibility
530           \global\@nobreaktrue
531           \everypar{\global\@nobreakfalse\everypar{}}%
532         \fi
533       \endgroup
534     \fi}
535 \newcommand*\l@section[2]{%
536   \ifnum \c@tocdepth >\z@
537     \addpenalty\@secpenalty
538     \addvspace{1.0em \@plus\p@}%
539     \setlength\@tempdima{1.5em}%
540     \begingroup
541       \parindent \z@ \rightskip \@pnumwidth
542       \parfillskip -\@pnumwidth
543       \leavevmode \bfseries
544       \advance\leftskip\@tempdima
545       \hskip -\leftskip
546       #1\nobreak\hfil \nobreak\hb@xt@\@pnumwidth{\hss #2}\par
547     \endgroup
548     \fi}
549 \newcommand*\l@subsection{\@dottedtocline{2}{1.5em}{2.3em}}
550 \newcommand*\l@subsubsection{\@dottedtocline{3}{3.8em}{3.2em}}
551 \newcommand*\l@paragraph{\@dottedtocline{4}{7.0em}{4.1em}}
552 \newcommand*\l@subparagraph{\@dottedtocline{5}{10em}{5em}}
553 \newcommand\listoffigures{%
```

```

554     \section*{\listfigurename}%
555     \mkboth{\MakeUppercase\listfigurename}%
556           {\MakeUppercase\listfigurename}%
557     \@starttoc{lof}%
558   }
559 \newcommand*\l@figure{\@dottedtocline{1}{1.5em}{2.3em}}
560 \newcommand\listoftables{%
561   \section*{\listtablename}%
562   \mkboth{%
563     \MakeUppercase\listtablename}%
564     {\MakeUppercase\listtablename}%
565   \@starttoc{lot}%
566   }
567 \let\l@table\l@figure

```

There are `\l@ttt` commands (`\l@part`, `\l@section`, etc) which produce the ToC lines from the `.aux` file. The List of Tables and List of Figures is implemented in the same way as the ToC. As with other features, consider the `tocloft` package for common modifications

2.12 Bibliography and Index

Bibliography styles themselves are implemented in `.bst` files, but the style of the section can be changed here, including indentation and spacing.

```

----- article.cls -----
568 \newdimen\bibindent
569 \setlength\bibindent{1.5em}
570 \newenvironment{thebibliography}[1]
571   {\section*{\refname}%
572   \mkboth{\MakeUppercase\refname}{\MakeUppercase\refname}%
573   \list{\@biblabel{\@arabic\c@enumiv}}%
574     {\settowidth\labelwidth{\@biblabel{#1}}%
575     \leftmargin\labelwidth
576     \advance\leftmargin\labelsep
577     \@openbib@code
578     \usecounter{enumiv}%
579     \let\p@enumiv\empty
580     \renewcommand\theenumiv{\@arabic\c@enumiv}}%
581   \sloppy
582   \clubpenalty4000
583   \@clubpenalty \clubpenalty
584   \widowpenalty4000%
585   \sfcode'\.\@m}
586 {\def\@noitemerr
587   {\@latex@warning{Empty 'thebibliography' environment}}%
588   \endlist}

```

```

589 \newcommand\newblock{\hskip .11em\@plus.33em\@minus.07em}
590 \let\@openbib@code\@empty
591 \newenvironment{theindex}
592     {\if@twocolumn
593      \@restonecolfalse
594      \else
595      \@restonecoltrue
596      \fi
597      \twocolumn[\section*{\indexname}]}%
598      \@mkboth{\MakeUppercase\indexname}%
599      {\MakeUppercase\indexname}%
600      \thispagestyle{plain}\parindent\z@
601      \parskip\z@ \@plus .3\p@\relax
602      \columnseprule \z@
603      \columnsep 35\p@
604      \let\item\@idxitem}
605     {\if@restonecol\onecolumn\else\clearpage\fi}
606 \newcommand\@idxitem{\par\hangindent 40\p@}
607 \newcommand\subitem{\@idxitem \hspace*{20\p@}}
608 \newcommand\subsubitem{\@idxitem \hspace*{30\p@}}
609 \newcommand\indexspace{\par \vskip 10\p@ \@plus5\p@ \@minus3\p@\relax}

```

2.13 Odds ‘n’ ends

The final section starts with the footnote ‘fence’ and the footnote alignment. There is also a list of the section names, which are the ones which get customised for other languages when you use the babel multilingual/multicultural package).

```

----- article.cls -----
610 \renewcommand\footnoterule{%
611     \kern-3\p@
612     \hrule\@width.4\columnwidth
613     \kern2.6\p@}
614 \newcommand\@makefnmark[1]{%
615     \parindent 1em%
616     \noindent
617     \hb@xt@1.8em{\hss\@makefnmark}#1}
618 \newcommand\contentsname{Contents}
619 \newcommand\listfigurename{List of Figures}
620 \newcommand\listtablename{List of Tables}
621 \newcommand\refname{References}
622 \newcommand\indexname{Index}
623 \newcommand\figurename{Figure}
624 \newcommand\tablename{Table}
625 \newcommand\partname{Part}
626 \newcommand\appendixname{Appendix}
627 \newcommand\abstractname{Abstract}
628 \def\today{\ifcase\month\or
629     January\or February\or March\or April\or May\or June\or

```

```

630     July\or August\or September\or October\or November\or December\fi
631     \space\number\day, \number\year}
632 \setlength\columnsep{10\p@}
633 \setlength\columnseprule{0\p@}
634 \pagestyle{plain}
635 \pagenumbering{arabic}
636 \if@twoside
637 \else
638     \raggedbottom
639 \fi
640 \if@twocolumn
641     \twocolumn
642     \sloppy
643     \flushbottom
644 \else
645     \onecolumn
646 \fi
647 \endinput

```

To end with, there is the `\today` date, which non-Americans can recode as:

```

\def\today{\number\day\space\ifcase\month\or
  January\or February\or March\or April\or May\or June\or
  July\or August\or September\or October\or November\or
  December\fi\space\number\year}

```

The last few lines include the column spacing, page style, and page numbering setups. Single-sided work is allowed to have a slightly variable text height (the `\raggedbottom` command), and two-column setting has a strict height but slightly greater tolerance on justification.

3 Rolling your own

Having seen what the article class does and how it works, you have a choice: create your new class file from scratch, or build onto an existing class.

Writing a wholly new class requires a significant knowledge of L^AT_EX and T_EX internals, but will have the advantage of being dedicated to the specific task on hand, and may offer more scope for automation, particularly if the process of generating the output is to be embedded within a larger application.

3.1 Re-using an existing class

Building on the work of other classes is more common, and has been described for a specific application (Minutes of meetings) in [3]. This involves loading the existing class file, handling any existing or new options, and then adding or modifying the commands and environments it provides.

We have already seen the use of `\renewcommand` (section 2.4.4 on p.16) and its counterpart for environments, `\renewenvironment` (section 2.7 on p.22), but you need to ensure the command and environments you are replacing are correctly preloaded. Hefferon [3] describes in detail the use of the `\LoadClass` and `\DeclareOption*` commands to specify the class on which you want to base yours, and how to preserve existing options and add your own.

3.2 Packages

As well as rewriting or modifying the code of an existing class, you can also invoke extra packages. In most cases this is faster, more reliable, and easier to do than rewriting the code of the existing class.

We have mentioned several useful packages so far:

geometry for the text area and page margins;

multicol for multiple columns of text;

fancyhdr for running headers and footers;

sectsty for changes to section and title styles;

ccaption for changes to the layout of Table and Figure captions;

tocloft for changes to the layout of the Table of Contents and Lists of Figures and Tables;

babel for working in multiple languages.

In your new class file, use the `\RequirePackage` command after the options (see section 2.3.4 on p.8). If an option needs to refer to a specific package, put the `\RequirePackage` after the version and identification section but before your options (see section 2.2 on p.4).

3.3 Four Last Things

The *Companion* [6, p.888] specifies that ‘every class file *must* contain four things’:

1. a definition of `\normalsize`;
2. a value for `\textwidth`;
3. a value for `\textheight`;
4. a specification for `\pagenumbering`.

Beyond that, it’s up to you! If you have been documenting your class file in `docTeX` format as you go along, as explained in the first paragraph in section 2, you should now consider releasing it for general use by submitting it to the CTAN maintainers so that other can use it.

References

- [1] Johannes Braams. Document classes and packages in $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}2_{\epsilon}$. *TUGboat*, 15(3), Sep 1994.
- [2] Robin Fairbairns (Ed). Frequently Asked Questions about $\text{T}_{\text{E}}\text{X}$. Technical report, UK $\text{T}_{\text{E}}\text{X}$ Users Group, Cambridge, UK, Nov 2005.
- [3] Jim Hefferon. Minutes in less than hours: Using $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ resources. *The PracTeX Journal*, (4), Oct 2005.
- [4] Donald Knuth. *The TeXbook*. Addison-Wesley, Reading, MA, Jun 1986.
- [5] Leslie Lamport. *L^AT_EX: A document preparation system*. Addison-Wesley, Reading, MA, 2nd edition, 1994.
- [6] Frank Mittelbach, Michel Goossens, Johannes Braams, David Carlisle, Chris Rowley, Christine Detig, and Joachim Schrod. *The L^AT_EX Companion: Tools and Techniques for Computer Typesetting*. Addison-Wesley, Reading, , MA, 2nd edition, May 2004.
- [7] Scott Pakin. How to package your $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ package. *CTAN*, Nov 2005.

- [8] The L^AT_EX3 Project. L^AT_EX2_ε for class and package writers, Dec 2003.
- [9] Wayne Sewell. *Literate Programming in WEB*. Van Nostrand Reinhold, New York, NY, 1989.