

# Compatibility mode for L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> emulating L<sup>A</sup>T<sub>E</sub>X 2.09

Alan Jeffrey and Frank Mittelbach

1995/12/27

## 1 Introduction

The file `latex209.def` is read in by L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> whenever it finds a `\documentstyle` rather than `\documentclass` command at the beginning of the file. This indicates a L<sup>A</sup>T<sub>E</sub>X 2.09 document, which should be processed in *compatibility mode*.

Any document which compiled under L<sup>A</sup>T<sub>E</sub>X 2.09 should compile under compatibility mode, unless it uses low-level commands such as `\tenrm`.

## 2 The docstrip modules

The following modules are used in the implementation to direct docstrip in generating the external files:

<code>driver</code>	produce a documentation driver file
<code>head</code>	produce the beginning of <code>latex209.def</code>
<code>tail</code>	produce the end of <code>latex209.def</code>
<code>article</code>	produce <code>article.sty</code>
<code>book</code>	produce <code>book.sty</code>
<code>report</code>	produce <code>report.sty</code>
<code>slides</code>	produce <code>slides.sty</code>
<code>letter</code>	produce <code>letter.sty</code>
<code>bezier</code>	produce <code>bezier.sty</code>
<code>fleqn</code>	produce <code>fleqn.sty</code>
<code>leqno</code>	produce <code>leqno.sty</code>
<code>openbib</code>	produce <code>openbib.sty</code>

Between the `head` and `tail` of `latex209.def`, the code for `oldlfont.sty` is included, so L<sup>A</sup>T<sub>E</sub>X 2.09 documents will automatically be run simulating the OFSS.

## 3 Driver

This section contains the driver for this documentation.

```
1 (*driver)
2 \documentclass{ltxdoc}
3 \DisableCrossrefs
4 % \OnlyDescription
5 \begin{document}
6   \DocInput{latex209.dtx}
7 \end{document}
8 </driver>
```

## 4 Beginning of latex209.def

This section describes the beginning of the file `latex209.def`.

```
9 (*head)
```

## 4.1 Identification

This file needs to be run with  $\text{\LaTeX} 2_{\epsilon}$ .

```
10 \NeedsTeXFormat{LaTeX2e}
```

Describe the file.

```
11 \ProvidesFile{latex209.def}[1998/05/13 v0.52 Standard LaTeX file]
```

Announce compatibility mode to the user.

```
12 \if@compatibility
13   \expandafter\endinput
14 \else
15   \typeout{^^J\space
16 \@spaces\@spaces\space Entering LaTeX 2.09 COMPATIBILITY MODE^^J\space
17 *****^^J\space
18 \space\space\space!!WARNING!!\space
19 \space\space\space!!WARNING!!\space
20 \space\space\space!!WARNING!!\space
21 \space\space\space!!WARNING!!\space\space\space ^^J\space
22 ^^J\space
23 This mode attempts to provide an emulation of the LaTeX 2.09^^J\space
24 author environment so that OLD documents can be successfully^^J\space
25 processed. It should NOT be used for NEW documents!^^J\space
26 ^^J\space
27 New documents should use Standard LaTeX conventions and start^^J\space
28 with the \string\documentclass\space command.^^J\space
29 ^^J\space
30 Compatibility mode is UNLIKELY TO WORK with LaTeX 2.09 style^^J\space
31 files that change any internal macros, especially not with^^J\space
32 those that change the FONT SELECTION or OUTPUT ROUTINES.^^J\space
33 ^^J\space
34 Therefore such style files MUST BE UPDATED to use^^J\space
35 \@spaces\@spaces\space Current Standard LaTeX: LaTeX2e.^^J\space
36 If you suspect that you may be using such a style file, which^^J\space
37 is probably very, very old by now, then you should attempt to^^J\space
38 get it updated by sending a copy of this error message to the^^J\space
39 author of that file.^^J\space
40 *****^^J}
41 \fi
```

## 4.2 Compatibility flag

$\text{\@compatibilitytrue}$   $\text{\LaTeX} 2_{\epsilon}$  has a flag  $\text{\if@compatibility}$  which can be used by document classes or packages to determine whether they are running in compatibility mode or not. This flag is set true by this file.

```
42 \@compatibilitytrue
```

## 4.3 Removing features

$\text{\usepackage}$  These  $\text{\LaTeX} 2_{\epsilon}$  commands are switched off in compatibility mode. This is done by saving the old definition, and redefining the command to call  $\text{\@latex@e@error}$  before executing the old version.

```
\listfiles
\ensuremath
\lrbbox
\newcommand
43 \def\@tempa#1#2{%
44   \expandafter\let\csname @@\string#1\endcsname#1%
45   \edef#1{%
46     \noexpand\@latex@e@error{\noexpand#2}%
47     \expandafter\noexpand\csname @@\string#1\endcsname
48   }%
49 }
50 \@tempa\usepackage\usepackage
51 \@tempa\listfiles\listfiles
52 \@tempa\ensuremath\ensuremath
```

```

53 \@tempa\lrbox{\begin{lrbox}}}%
54 \@tempa\xargdef{\newcommand{cmd}[args][def]}%

```

`\@latex@e@error` This error is produced if a user uses a  $\LaTeX 2_\epsilon$  command in compatibility mode. This is to encourage users to move over to using `\documentclass` as quickly as possible. During the preamble the error does nothing (so that packages can use  $\LaTeX 2_\epsilon$  commands) but it is redefined to be an error message at `\begin{document}`.

```

55 \let\@latex@e@error\gobble
56 \def\@latex@e@error#1{%
57   \@latexerr{%
58     LaTeX2e command \string#1\space in LaTeX 2.09 document%
59   }{%
60     This is a LaTeX 2.09 document, but it contains
61     \string#1.^J%
62     If you want to use the new features of LaTeX2e,
63     your document.^J%
64     should begin with \string\documentclass\space
65     rather than \string\documentstyle
66   }%
67 }

```

`\@ifdefinable` We trap the `\@notdefinable` error message to check to see if the command is a  $\LaTeX 2_\epsilon$  command, in which case we allow the definition to happen. We keep a list `\@old@ifdefinable` of commands which are allowed to be redefined this way in `\@latex@e@commands`, `\@@ifdefinable` and remove an entry each time it is defined. `\@latex@e@commands`

```

68 \let\@old@ifdefinable\@ifdefinable
69 \long\def\@ifdefinable#1{%
70   \def\@tempa##1##2##3##4\@tempa{%
71     \def\@latex@e@commands{##1##2}%
72     ##3% ##3 will either be \iftrue or \iffalse
73     \expandafter\@firstofone
74   \else
75     \expandafter\@old@ifdefinable\expandafter#1%
76   \fi
77   }%
78   \expandafter\@tempa\@latex@e@commands#1\iftrue#1\iffalse#1\@tempa%
79 }
80 \let\@@ifdefinable\@ifdefinable
81 \def\@latex@e@commands{%
82   \usepackage\listfiles\ensuremath\LaTeXe\lrbox
83   \th\dh\ng\dj\TH\DH\NG\DJ\k\r\SS
84   \guillemotleft\guillemotright\guilsinglleft
85   \guilsinglright\quotedblbase\quotesinglbase
86 }

```

`\@begin@tempboxa` If we were to switch off the new `\width`, `\height` and `\depth` commands, this is how to do it. This isn't done, since these commands may be used in packages.

```

\long\def\@begin@tempboxa#1#2{%
  \begingroup
  \setbox\@tempboxa#1{#2}}

```

#### 4.4 Document class hook

`\@documentclasshook` This macro is called by each use of `\documentclass`. We define it to define `\@normalsize` and `\normalsize` if necessary, to input each unused option as a package, and to switch off the new  $\LaTeX 2_\epsilon$  commands. However, we leave on the commands `\settoheight`, `\settowidth` and the new options to `\parbox` and `\minipage`, since these are likely to be used in packages.

The intention of the strange `\normalsize` tests below is that after the `\documentstyle` command has completed, if neither `\normalsize` nor `\@normalsize` was defined by the main style or one of its ‘substyles’ or ‘options’, then `\@normalsize` will be undefined and `\normalsize` will generate an error saying it hasn’t been defined.

If the style defined either `\normalsize` or `normalsize` then these two commands will be `\let` equal to each other, with the definition given by the style file.

If the style defines both `\normalsize` and `\@normalsize` then those two definitions are kept.

```

87 \def\@documentclasshook{%
88   \RequirePackage\@unusedoptionlist
89   \let\@unusedoptionlist\@empty
90   \def\@tempa{\@normalsize}%
91   \ifx\normalsize\@tempa
92     \let\normalsize\@normalsize
93   \fi
94   \ifx\@normalsize\@undefined
95     \let\@normalsize\normalsize
96   \fi
97   \ifx\normalsize\@undefined
98     \let\normalsize\original@normalsize
99   \fi
100  \let\@latex@e@error\@latex@e@error@}

```

`\original@normalsize` Save the original definition of `\normalsize` (which generates an error)

```
101 \let\original@normalsize\normalsize
```

`\normalsize` Some styles don’t define `\normalsize`, just `\@normalsize`.

```
102 \def\normalsize{\@normalsize}
```

## 4.5 Compatibility with L<sup>A</sup>T<sub>E</sub>X 2.09 document styles

`\@missingfileerror` If a `.cls` file is missing, we look to see if there is a file of the same name with a `.sty` extension.

```

103 \@ifundefined{saved@missingfileerror}{
104   \let\saved@missingfileerror=\@missingfileerror
105 }{}
106 \def\@missingfileerror#1#2{%
107   \ifx#2\@clsextension
108     \InputIfFileExists{#1.\@pkgextension}{%
109       \wlog{Compatibility mode: loading #1.\@pkgextension
110         \space rather than #1.#2.}%
111     }{%
112       \saved@missingfileerror{#1}{#2}%
113     }%
114   \else
115     \saved@missingfileerror{#1}{#2}%
116   \fi
117 }

```

`\@obsoletefile` For compatibility with the document styles which `\input` the standard L<sup>A</sup>T<sub>E</sub>X 2.09 document styles, we distribute files called `article.sty`, `book.sty`, `report.sty`, `slides.sty` and `letter.sty`. These use the command `\@obsoletefile`, which the L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> kernel defines to produce a warning message. We redefine it to just produce a message in the log file, and to pass any options from the old filename to the new filename.

```

118 \def\@obsoletefile#1#2{%
119   \expandafter\let\csname opt@#1\expandafter\endcsname
120   \csname opt@\@currname.\@current\endcsname

```

```
121 \wlog{Compatibility mode: inputting '#1'
122      instead of obsolete '#2'.}}
```

```
\footheight LATEX 2.09 supported these parameters, so for compatibility with old document
\@maxsep styles we allocate them.
\@dblmaxsep 123 \newdimen\footheight
124 \newdimen\@maxsep
125 \newdimen\@dblmaxsep
```

```
\mark LATEX 2.09 initialized an empty mark. Who knows, someone may have relied on
it:
126 \mark{{}{}}
```

## 4.6 Layout

```
\sloppy There is a new version of \sloppy in LATEX 2ε, so we restore the old one.
127 \def\sloppy{\tolerance \@M \hfuzz .5\p@ \vfuzz .5\p@}
```

```
\@finalstrut The strut which is used in a footnote has changed. This restores the old definition.
128 \def\@finalstrut#1{\unskip\strut}
```

```
\@marginparreset Restore the old spacing around floats.
```

```
\@floatboxreset 129 \let \@marginparreset \@empty
130 \let \@floatboxreset \@empty
```

```
\proclaim From plain TEX.
```

```
131 \outer\def\proclaim #1. #2\par{%
132 \medbreak
133 \noindent{\bfseries#1.\enspace}{\slshape#2\par}%
134 \ifdim\lastskip<\medskipamount
135 \remove\lastskip\penalty55\medskip
136 \fi}
```

```
\hang From plain TEX.
```

```
\textindent 137 \def\hang{\hangindent\parindent}
138 \def\textindent#1{\indent\llap{#1\enspace}\ignorespaces}
```

```
\ttraggedright
```

```
139 \def\ttraggedright{\reset@font\ttfamily\rightskip\z@ plus2em\relax}
```

```
\@footnotemark LATEX 2ε version has \nobreak to allow hyphenation.
```

```
140 \def\@footnotemark{%
141 \leavevmode
142 \ifhmode\edef\@x@sf{\the\spacefactor}\fi
143 \@makefnmark
144 \ifhmode\spacefactor\@x@sf\fi
145 \relax}
```

```
\@textsuperscript Fudge this command to remove the text font command which is always the first
thing in the argument. This is needed as in compatibility mode footnotes are
processed in math mode, but the standard classes call \@textsuperscript in the
definition of \thanks.
```

```
146 \def\@textsuperscript#1{${\m@th^{\@gobble#1}}$}
```

```
\@makefnmark LATEX 2ε version uses \@textsuperscript rather than math mode.
```

```
147 \def\@makefnmark{\hbox{${\@thefnmark}\m@th$}}
```

```
\thempfootnote LATEX 2ε version has an additional \itshape which would not work (and would
not make sense) in math mode.
```

```
148 \def\thempfootnote{\@alph{c@mpfootnote}}
```

`\@fnsymbol` L<sup>A</sup>T<sub>E</sub>X version uses `\ensuremath` which does not work in compatibility mode.

```
149 \def\@fnsymbol#1{\ifcase#1\or *\or \dagger\or \ddagger\or
150   \mathchar "278\or \mathchar "27B\or \|\or **\or \dagger\dagger
151   \or \ddagger\ddagger \else\@ctrerr\fi}
```

`\@inmathwarn` L<sup>A</sup>T<sub>E</sub>X (1995/12/01) checks for text commands being used in math mode. We switch this off in compatibility mode.

```
152 \let\@inmathwarn\@gobble
```

## 4.7 Verbatim

`\verb` We restore the old definition of `\verb`, but using `\verbatim@font` rather than `\tt`. The use of `\bgroup` and `\egroup` allows us to prefix it with `\hbox` in math mode.

```
153 \def\verb{%
154   \relax\ifmmode\hbox\fi\bgroup
155   \@noligs
156   \verbatim@font
157   \let\do\@makeother \dospecials
158   \@ifstar{\@sverb}{\@verb}%
159 }
160 \def\@sverb#1{%
161   \def\@tempa ##1#1{\leavevmode\null##1\egroup}%
162   \@tempa
163 }
```

`\verbatim@nolig@list` The only ligatures which should be switched off in 2.09 mode are the Spanish punctuation.

```
164 \def\verbatim@nolig@list{\do\‘}
```

`\@lquote` We restore the old definition of `\@lquote` in case any packages use it.

```
165 \def\@lquote{\leavevmode{\kern\z@}\‘}
```

## 4.8 Character codes

By default, L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> makes the input characters 0–8, 11, 14–31 and 128–255 illegal. In compatibility mode, we restore their old meanings.

```
166 \catcode0=9
167 \@tempcnta=1
168 \loop\ifnum\@tempcnta<32
169   \catcode\@tempcnta=12
170   \advance\@tempcnta by 1
171 \repeat%
172 \catcode\^^I=10\relax%
173 \catcode\^^L=13\relax%
174 \catcode\^^M=5\relax%
175 \catcode127=15
176 \@tempcnta=128
177 \loop\ifnum\@tempcnta<256
178   \catcode\@tempcnta=12
179   \advance\@tempcnta by 1
180 \repeat
```

## 4.9 Miscellaneous commands

`\SLiTeX` The S<sub>L</sub>I<sub>T</sub>E<sub>X</sub> logo.

```
181 \DeclareRobustCommand{\SLiTeX}{%
182   \normalfont S\kern -.06em
183   {\scshape l\kern -.035emi}\kern -.06em
184   \TeX}}
```

```

\+ The \+ command should be defined, so that it can be used in \renewcommand.
185 \let\+\@empty

\@cla LATEX 2.09 (and early versions of LATEX 2ε) used these count registers in the defi-
\@clb nition of \cline and \multispan. Declare them here in case they were used for
\mscount any other purposes.
186 \newcount\@cla
187 \newcount\@clb
188 \newcount\mscount

```

```

\@imakepicbox picture mode version
189 \long\def\@imakepicbox(#1,#2)[#3]#4{%
190   \vbox to#2\unitlength
191   {\let\mb@b\vss \let\mb@l\hss\let\mb@r\hss
192    \let\mb@t\vss
193    \@tfor\reserved@a :=#3\do{%
194     \if s\reserved@a
195     \let\mb@l\relax\let\mb@r\relax
196     \else
197     \expandafter\let\csname mb@\reserved@a\endcsname\relax
198     \fi}%
199   \mb@t
200   \hb@xt@ #1\unitlength{\mb@l #4\mb@r}%
201   \mb@b

```

This kern ensures that a b option aligns on the bottom of the text rather than the baseline. this is the documented behaviour in the L<sup>A</sup>T<sub>E</sub>X Book. The kern is removed in compatibility mode.

Remove kern for bug compatibility with 2.09.

```

202 %   \kern\z@
203   }}

```

\supereject

```

204 \def\supereject{\par\penalty-\@MM}

```

\nofiles This old version might change the vertical spacing when it is used. Some old document might depend on that changed spacing so...

```

205 \def\nofiles{%
206   \@fileswfalse
207   \typeout{No auxiliary output files.^^J}%
208   \long\def\protected@write##1##2##3{%
209     \let\makeindex\relax
210     \let\makeglossary\relax}

```

## 4.10 Packages and classes

\ProvidesPackage We redefine \ProvidesPackage and \ProvidesClass to produce a log message rather than a warning if they find an unexpected file.

```

\ProvidesClass
211 \def\ProvidesPackage#1{%
212   \xdef\@gtempa{#1}%
213   \ifx\@gtempa\@currname\else
214     \wlog{Compatibility mode: \@cls@pkg\space'\@currname' requested,
215       but '#1' provided.}%
216   \fi
217   \@ifnextchar[\@pr@videpackage{\@pr@videpackage[]}]%
218   \let\ProvidesClass=\ProvidesPackage

```

That ends the head of latex209.def.

```

219 </head>

```

## 5 Middle of latex209.def

At this point, the code for `oldfont.sty` is read in by the installation script.

## 6 End of latex209.def

This section describes the end of `latex209.def`.

```
220 <*tail>
```

### 6.1 Font commands

```
\ds@oldfont We declare oldfont, newfont, margid and nomargid options to mimic the
\ds@newfont LATEX 2.09 NFSS1 options.
\ds@margid 221 \def\ds@oldfont{%
\ds@nomargid 222 \no@font@optfalse
223 \let\math@bgroup\@empty
224 \let\math@egroup\@empty
225 \let\@math@bgroup\math@bgroup
226 \let\@math@egroup\math@egroup
227 }
228 \def\ds@newfont{%
229 \no@font@optfalse
230 \OptionNotUsed
231 }
232 \def\ds@margid{%
233 \no@font@optfalse
234 \let\math@bgroup\bgroup
235 \def\math@egroup##1{##1\egroup}%
236 \let \@math@bgroup \math@bgroup
237 \let \@math@egroup \math@egroup
238 }
239 \let\ds@nomargid\ds@oldfont
240 \onlypreamble\ds@oldfont
241 \onlypreamble\ds@newfont
242 \onlypreamble\ds@margid
243 \onlypreamble\ds@nomargid

\encodingdefault The default encoding for old documents is OT1 rather than T1.
244 \renewcommand{\encodingdefault}{OT1}

\cmex/m/n/10 Just in case a document style relies on \cmex/m/n/10 to exist (which may have
been hard-wired to \fam3) we load the font.
245 \expandafter\font\csname cmex/m/n/10\endcsname=cmex10

\normalshape These commands were used in older versions of NFSS.
\mediumseries 246 \def\normalshape{\fontshape\shapedefault\selectfont}
247 \def\mediumseries{\fontseries\seriesdefault\selectfont}

\DeclareOldFontCommand We redefine \DeclareOldFontCommand to do nothing. This means that any new
document classes will have their redefinitions of \rm, \bf etc. ignored.
248 \def \DeclareOldFontCommand #1#2#3{%
249 \wlog{Compatibility mode: definition
250 of \string#1\space ignored.}%
251 }

\@halfmag Some font-specifying commands from LATEX 2.09.
\@magscale 252 \def\@halfmag{ scaled \magstephalf}
\@ptscale 253 \def\@magscale#1{ scaled \magstep#1 }
254 \def\@ptscale#1{ scaled #100 }
```

`\font` The current font is set to be CMR 10pt, to match L<sup>A</sup>T<sub>E</sub>X 2.09.

```
255 \fontencoding{OT1} \fontfamily{cmr}
256 \fontsize{10}{12} \fontseries{m} \fontshape{n}
257 \selectfont
```

`\load` The `\load` command is no longer needed, it is therefore defined to do nothing.

```
258 \let\load@gobbletwo
```

Here are three delimiters which have been partly disabled by NFSS2 (the small variants) since the corresponding fonts are normally not preloaded as math symbol fonts.

```
259 \DeclareMathDelimiter{\lgroup} % extensible ( with sharper tips
260   {\mathopen}{bold}{"28}{largesymbols}{"3A}
261 \DeclareMathDelimiter{\rgroup} % extensible ) with sharper tips
262   {\mathclose}{bold}{"29}{largesymbols}{"3B}
263 \DeclareMathDelimiter{\bracevert} % the vertical bar that extends braces
264   {\mathord}{typewriter}{"7C}{largesymbols}{"3E}
```

In old documents we might find some usages of `\bffam` etc. Thus we add the following code:

```
265 \let\bffam\symbold
266 \let\sfam\symsans
267 \let\itfam\symitalic
268 \let\ttfam\symtypewriter
269 \let\scfam\symsmallcaps
270 \let\slfam\symslanted
271 \let\rmfam\symoperators
```

Below are the `\. .pt` commands with hopefully the same functionality as in the old `lfonts.tex`. Notice that the `\baselineskip` parameter wasn't set by these commands so that using them now shouldn't set this either. Thus we go low-level. This means that the commands are now fragile but I think they have been fragile before.

```
272 \newcommand\vpt   {\edef\f@size{\@vpt}\rm}
273 \newcommand\vipt  {\edef\f@size{\@vipt}\rm}
274 \newcommand\vipt  {\edef\f@size{\@vipt}\rm}
275 \newcommand\viipt {\edef\f@size{\@viipt}\rm}
276 \newcommand\ixpt  {\edef\f@size{\@ixpt}\rm}
277 \newcommand\xpt   {\edef\f@size{\@xpt}\rm}
278 \newcommand\xipt  {\edef\f@size{\@xipt}\rm}
279 \newcommand\xiip  {\edef\f@size{\@xiip}\rm}
280 \newcommand\xivpt {\edef\f@size{\@xivpt}\rm}
281 \newcommand\xvipt {\edef\f@size{\@xvipt}\rm}
282 \newcommand\xxpt  {\edef\f@size{\@xxpt}\rm}
283 \newcommand\xxvpt {\edef\f@size{\@xxvpt}\rm}
```

## 6.2 User customization

For sites which customized their version of L<sup>A</sup>T<sub>E</sub>X 2.09, we provide a file `latex209.cfg`, which is loaded every time we enter compatibility mode. If the file doesn't exist, we don't do anything.

```
284 \InputIfFileExists{latex209.cfg}{}{}
```

That ends the file `latex209.def`.

```
285 </tail>
```

## 7 Obsolete style files

For each of the standard L<sup>A</sup>T<sub>E</sub>X 2.09 document styles, we produce a file which points to the appropriate L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> document class file. This means that any styles which say `\input article.sty` should still work.

```

286 (*article | book | report | letter | slides)
287 \NeedsTeXFormat{LaTeX2e}
288 </article | book | report | letter | slides>
289 (*article)
290 \@obsoletedefile{article.cls}{article.sty}
291 \LoadClass{article}
292 </article>
293 (*book)
294 \@obsoletedefile{book.cls}{book.sty}
295 \LoadClass{book}
296 </book>
297 (*report)
298 \@obsoletedefile{report.cls}{report.sty}
299 \LoadClass{report}
300 </report>
301 (*letter)
302 \@obsoletedefile{letter.cls}{letter.sty}
303 \LoadClass{letter}
304 </letter>
305 (*slides)
306 \@obsoletedefile{slides.cls}{slides.sty}
307 \LoadClass{slides}
308 </slides>

```

We also produce empty `fleqn.sty` and `leqno.sty` files in case anyone has `\input` one of them.

```

309 (*fleqn)
310 \@obsoletedefile{fleqn.clo}{fleqn.sty}
311 \input{fleqn.clo}
312 </fleqn>
313 (*leqno)
314 \@obsoletedefile{leqno.clo}{leqno.sty}
315 \input{leqno.clo}
316 </leqno>

```

We also produce an empty `openbib.sty` in case anyone has `\input openbib.sty`. The `openbib` class option is now part of the kernel.

```

317 (*openbib)
318 \iffalse
319
320 The openbib option is now part of LaTeX thus this package is no
321 longer necessary. It is only retained for upward compatibility.
322 See the 2nd edition of the LaTeX book, or the file usrguide.tex
323 which comes with the LaTeX distribution, for more details.
324
325 \fi
326 </openbib>

```

We also produce an empty `bezier.sty` in case anyone has `\input bezier.sty`. The `\bezier` command is now part of the kernel.

```

327 (*bezier)
328 \iffalse
329
330 The \bezier command is now part of LaTeX thus this package is no
331 longer necessary. It is only retained for upward compatibility.
332 Also, please note that LaTeX now offers an extended bezier command
333 which automatically calculates the number of points needed for the
334 plot. See the 2nd edition of the LaTeX book, or the file
335 usrguide.tex which comes with the LaTeX distribution, for more
336 details.
337
338 \fi
339 </bezier>

```

We also produce a `t1enc` package, for compatibility with the Companion. This has been replaced by the `fontenc` package.

```
340 (*t1enc)
341 \NeedsTeXFormat{LaTeX2e}
342 \ProvidesPackage{t1enc}[1994/06/01 Standard LaTeX package]
343 \renewcommand{\encodingdefault}{T1}
344 \fontencoding{T1}\selectfont
345 </t1enc>
```