How to make your own document style in \LaTeX{}2\epsilon

Dag F. Langmyhr

Department of Informatics
University of Oslo
dag@ifi.uio.no

1 Background

Both the old \LaTeX{}2.09 and the new \LaTeX{}2\epsilon is distributed with five standard document styles or classes:\footnote{\LaTeX{}2.09 uses the term `document style' while \LaTeX{}2 uses `document class' for the same thing, namely a definition of the appearance of a document; this choice of words is also used in \LaTeX{}2\epsilon to distinguish between the two versions.} \texttt{article}, \texttt{report}, \texttt{book}, \texttt{slides} and \texttt{letter}. Nearly all \LaTeX{} users base their documents on one of these standard classes, perhaps slightly modified by one or more of the myriad of options found on the net. Very few users design new document classes, even though there are several reasons why you might want to do just that:

- The standard classes may not fit your needs. Typical other useful classes would be for lecture notes, technical manuals, theater plays, exam questions, CVs, memos, official letters, etc.
- You are more likely to achieve a good result if you design a complete document class rather than apply several options from various sources.
- You get bored of always seeing the same \LaTeX{} classes.
- It is very easy to design a new document class, particularly since \LaTeX{}2\epsilon came along.

For my own personal use, I needed a class for lecture notes:

- I wanted more space in the margins and between paragraphs, as the students like to make notes.
- I needed an environment for showing short examples of program code, and a `list of examples' just like the `list of figures'.
- I wanted a quotation at the start of every chapter.
- I wanted something that did not look quite so `\LaTeX{}y'.

This class I called \texttt{lecnotes}. It was originally designed to work under the old \LaTeX{}2.09, and has since been updated to \LaTeX{}2\epsilon.

2 How to find information

Even though designing a new document class is not difficult, you need some information on how classes work. This information can be found in:


- The files \texttt{clsguide.tex}, which describes how to write a class file, \texttt{usrguide.tex}, which introduces \LaTeX{}2.09 users to \LaTeX{}2\epsilon, and \texttt{fntguide.tex}, which contains an introduction to the font handling mechanisms of \LaTeX{}2\epsilon. These three files are found in the \LaTeX{}2\epsilon distribution.
- The \LaTeX{}2\epsilon source code itself, in particular the two files \texttt{source2e.tex} and \texttt{classes.dtx}. These files can be processed by \LaTeX{}2\epsilon to give you an indexed and commented version of the source code.

You also need some proficiency in plain \TeX, as the class files are usually written in a mixture of plain \TeX{} and \LaTeX{}.

3 What is a document class?

A document class such as \texttt{lecnotes} is really a file \texttt{lecnotes.cls} containing plain \TeX{} and \LaTeX{} code. This file is read and processed when its name is found as a parameter to \texttt{\documentclass}. A class file in \LaTeX{}2\epsilon should contain five parts: identification, declaration of options, execution of options, package loading and the main code.

3.1 Writing a class file

When writing a class file for a new document style, you can do this in one of three different ways:

- You can write it from scratch. This is quite difficult and not recommended for beginners.
- You can start with a copy of one of the standard classes and modify it. This is a better approach, but will give you problems if essential parts of \LaTeX{} are changed in a later version.
- You can write a class file which loads a standard class and makes the necessary modifications. This will increase the processing time slightly, but you are much safer against problems with future revisions.

This is no doubt the preferred way to do it in \LaTeX{}2\epsilon, as a special command \texttt{\LoadClass} is available for just this purpose.

\texttt{Lecnotes} was written using \texttt{\LoadClass}, and it is based on the standard \texttt{report} class.\footnote{The examples shown here contain just the actual code file \texttt{lecnotes.cls}. This code is derived from the documented file \texttt{lecnotes.dtx}.}
3.2 The identification part

This part tells the name of the class and also specifies that it only works under \LaTeX{}:

\NeedsTeXFormat{LaTeX2e}
\ProvidesClass{lecnotes}[1994/08/26 v1.0
  \PrivateLaTeXe{} document class]

3.3 The declaration of options part

Some document classes have options and they are declared in this part of the class file. \textit{Lecnotes} has two options, or rather, one option \texttt{norsk} for specifying that the text is in Norwegian, and an invalidation of \texttt{twocolumn}:

\newcommand{\lnotes@exa@id}{Example}
\newcommand{\lnotes@exa@tit}{List of Examples}
\DeclareOption{norsk}{%
  \renewcommand{\lnotes@exa@id}{Eksempel}
  \renewcommand{\lnotes@exa@tit}{Eksempler}
}
\DeclareOption{twocolumn}{%
  \ClassWarningNoLine{lecnotes}{option `twocolumn' unsupported in this class}}

In addition, it has the usual size options \texttt{10pt}, \texttt{11pt} and \texttt{12pt}. All other options should just be passed on to the \texttt{report} class:

\DeclareOption{*
  \PassOptionsToClass{\CurrentOption}{report}}

3.4 The execution of options part

The previous part just declared the options; now is the time to have them executed. In \texttt{lecnotes} I always execute \texttt{10pt} in case the user does not specify a size:

\ExecuteOptions{10pt}

Then all the user-supplied options are executed:

\ProcessOptions

3.5 The package loading part

As mentioned previously, \texttt{lecnotes} is based on the \texttt{report} class, so that must be loaded:

\LoadClass[twoside]{report}

I supply the \texttt{twoside} option, as I always use that.

\texttt{Lecnotes} needs two packages (see \textit{Incorporating packages} below) and these must be loaded at this stage:

\RequirePackage{float}
\RequirePackage{moreverb}

(\texttt{Moreverb} also loads the \texttt{verbatim} package.)

3.6 The main code part

The rest of this document describes the main code part.

4 Adjusting parameters

There are several style parameters in the standard classes, and part of the work of designing a new class is simply to adjust some of these parameters.

4.1 Adjusting paragraph spacing

The two \TeX{} values \texttt{parindent} and \texttt{parskip} control paragraph indentation and the vertical space added between paragraphs, respectively. Since I wanted no indentation but instead some space between paragraphs, I defined:

\setlength{\parskip}{12\p@ + 4\p@ - 2\p@}
\setlength{\parindent}{0\p@}

I added some stretch and shrink to the paragraph spacing to avoid vertical spacing problems. The effect of these two definitions is shown in figure 1.

4.2 Adjusting figure placement

Several parameters control how floating material such as figures, tables and examples (see below) is placed. In \texttt{lecnotes} I wanted a different layout from the standard one:

- Because of all the program examples I have, I wanted floats to be allowed to fill up to 80\% of a text page.
- I wanted up to three floats on a text page, but no bottom floats.
- I wanted separate float pages when necessary, but only when they fill at least 75\% of the page.

The specification of this is

\setcounter{topnumber}{3}
\renewcommand{\topfraction}{0.80}
\setcounter{bottomnumber}{0}
\setcounter{totalnumber}{3}
\renewcommand{\textfraction}{0.20}
\renewcommand{\floatpagefraction}{0.75}

To avoid name conflicts, \LaTeX{} alters the \texttt{\catcode} of the `@' character so that it behaves as a letter when reading class and package files.

When writing a class file, it is a good idea to prefix each local name with the name of the class to avoid possible conflict with local names in other class or package files.

3 To save both storage space and processing time, \LaTeX{} uses commands for words that occur often, such as

\@depth for depth
\@height for height
\@minus for minus
\@plus for plus
\@width for width

It also uses a dimen register named \texttt{\z@} to contain the width 0pt.
4.3 Roman and Arabic page numbering

In lecnotes I wanted the page numbers for the initial part of the document (containing the title page, the abstract, the table of contents, the list of examples, etc.) to be in lower case Roman numbers:

\pagenumbering{roman}

The first \chapter command will change this to Arabic numbers and reset the page counter to 1:

\pagenumbering{arabic}
\setcounter{page}{1}

4.4 Other adjustments

I also made a few other adjustments. I wanted a ‘bolder’ look, so I made the rules in arrays, tables and framed boxes thicker:

\setlength{\fboxsep}{3\p@}
\setlength{\fboxrule}{0.6\p@}
\setlength{\arrayrulewidth}{0.8\p@}

5 Choosing a font

A document contains two kinds of text: the body text and the additional text used in headings, headers and footers.

\@evenfoot
\@oddfoot
\@evenhead
\@oddhead
\@evenhead
\@oddhead
\@evenfoot
\@oddfoot

5.1 Designing page styles

\LaTeX uses the four macros \texttt{\@evenhead}, \texttt{\@oddhead}, \texttt{\@evenfoot} and \texttt{\@oddfoot} to obtain the page header and footer for a left-hand and right-hand page.\footnote{\texttt{\@evenhead} and \texttt{\@oddhead} are used for the header on even and odd pages, respectively. \texttt{\@evenfoot} and \texttt{\@oddfoot} are used for the footer on even and odd pages, respectively.}

A call on a page style such as \texttt{\pagestyle{plain}} is just a different notation for calling the macro \texttt{\ps@plain} which defines figures, captions, table of contents, etc. Usually fonts from the same family are used for both kinds of text, but one might achieve a more lively result with two contrasting font families. In lecnotes I leave the choice of body text to the writer\footnote{In case the user does not have \texttt{Gill Sans} available, it is possible to choose another alternative.} and use \texttt{Gill Sans} in various weights for the additional text.\footnote{If the document does not use the \texttt{twoside} option, all pages are regarded as odd-numbered.}

Since the document class should work for 10, 11 and 12 point body text, I made a macro for each kind of font needed. The fonts defined for the \texttt{10pt} option are shown in figure 2.

6 Utilizing hooks

The standard document classes contain many hooks which allow a designer to easily modify the class by redefining simple macros.

6.1 Designing page styles
The main hook for designing section headings is \section which has six parameters:

- **name** is the name for the heading, like subsection.
- **level** is the level number for the heading, for instance 2 for a subsection.
- **indent** is the amount of horizontal indentation.
- **beforeskip** is the amount of vertical space to put above each heading.
- **afterskip** is the amount of vertical space to put below the heading.

This parameter serves an additional purpose. If its value is negative, the space above will be the absolute value of the parameter, but in addition the first text paragraph following the heading will not be indented.

Another thing to remember is that some commands like \chapter contain a call on \pagestyle{plain}. This implies that if one wants to modify the appearance of the first page of a chapter, one must either rewrite \chapter or redefine the \ps@plain macro. I chose the latter approach, and made plain a copy of headings but with empty headers:

```latex
\renewcommand{\ps@plain}{\ps@headings
   \renewcommand{\@oddevenhead}{}\%
   \renewcommand{\@oddevenfoot}{}\%
}
```

### 6.2 Modifying the section headings

The main hook for designing section headings is \section which has six parameters:

- **name** is the name for the heading, like subsection.
- **level** is the level number for the heading, for instance 2 for a subsection.
- **indent** is the amount of horizontal indentation.
- **beforeskip** is the amount of vertical space to put above each heading.
- **afterskip** is the amount of vertical space to put below the heading.

This parameter serves an additional purpose. If its value is negative, the space above will be the absolute value of the parameter, but in addition the first text paragraph following the heading will not be indented.
Chapter 5

Introduction

This is a handbook about \TeX, a new typesetting system intended for the creation of beautiful books—indeed, for books that contain a lot of mathematics. By preparing a manuscript in \TeX format, you will be telling the computer exactly how the manuscript is to be transformed into pages whose typographic quality is comparable to that of the world's finest printers, yet you won't need to do much more work than would be involved if you were simply typing the manuscript on an ordinary typewriter.

5.1 Background

This is a handbook about \TeX, a new typesetting system intended for the creation of beautiful books—indeed, for books that contain a lot of mathematics. By preparing a manuscript in \TeX format, you will be telling the computer exactly how the manuscript is to be transformed into pages whose typographic quality is comparable to that of the world's finest printers, yet you won't need to do much more work than would be involved if you were simply typing the manuscript on an ordinary typewriter.

5.1.1 Previous editions

This is a handbook about \TeX, a new typesetting system intended for the creation of beautiful books—indeed, for books that contain a lot of mathematics. By preparing a manuscript in \TeX format, you will be telling the computer exactly how the manuscript is to be transformed into pages whose typographic quality is comparable to that of the world's finest printers, yet you won't need to do much more work than would be involved if you were simply typing the manuscript on an ordinary typewriter.

The standard report class

style specifies which type style to use when printing the heading, for example \Large\bf.

Using this hook it is easy to define the various section headings. My definitions differ from the standard ones in both spacing and font style and are shown in figure 4:\footnote{The sign of the fourth and fifth parameters may not be as you expected, but that is because I have redefined \@startsection; see the section \texttt{Modifying the code} below.}

\begin{figure}[ht]
\centering\begin{tabular}{|l|l|}
\hline
\textbf{Chapter 5} & \textbf{Introduction} \\
\hline
The standard report class & The new lecnotes class \\
\hline
\end{tabular}
\caption{Chapter and section headings}
\end{figure}
The appearance of the section number is defined in a hook called \seccntformat having the section level name as its parameter. I redefined that to put the numbers into the margin (using \llap) and in a font of their own:

```latex
\renewcommand{\seccntformat}{%\llap }%
\newcommand{\lnotes@section}{%\llap \lnotes@num@sec \thesection}
\newcommand{\lnotes@subsection}{%\llap \lnotes@num@subsec \thesubsection}
\newcommand{\lnotes@subsubsection}{%\llap \lnotes@num@subsubsec \thesubsubsection}
```

The chapter and part headings are more complex and usually not made with \startsection. Chapter headings are defined by the macro \@makechapterhead (or \maketitle if \chapter* was used) and part headings by \@part (or \@part), and these can be modified if needed.

I redefined \maketitle to use different spacing and font, and also to include a short quotation defined previously using a new command called \chaptercite; the result can be seen in figure 4.

### 6.3 Modifying the chapter and part headings

The chapter and part headings are more complex and usually not made with \startsection. Chapter headings are defined by the macro \@makechapterhead (or \maketitle if \chapter* was used) and part headings by \@part (or \@part), and these can be modified if needed.

I redefined \maketitle to use different spacing and font, and also to include a short quotation defined previously using a new command called \chaptercite; the result can be seen in figure 4.

### 6.4 Modifying the table of contents

There exist hooks for designing the appearance of each line in the table of contents, and these hooks are called \@ptocline for defining the entries in the lists of figures and tables.

Each line on the table of contents file file.toc is on the form

```
\contentsline{section}{\numberline{2.1}Hex numbers}{14}
```

indicating that section number 2.1 titled Hex numbers starts on page 14. This is just a different notation for calling the right hook:

```
\@startsection{section}{1}{% 0em}{-0.8em}{2em}{
  \lnotes@section{\numberline{\thesection}}%}%
```

Since most tables of contents look rather alike, with a dotted line leading from the section title to the page number, \LaTeX{} provides a macro to make such a line. It is called \dottedtocline and has five parameters:

- \texttt{level} is the section level number, for instance 1 for a section.
- \texttt{indent} is the amount of indentation to the left.
- \texttt{numwidth} is the width to be reserved for the section number.
- \texttt{text} is the section title. This usually consists of two parts: the section number, given as a parameter to \numberline, and the title text. The macro \numberline just sets its parameter left justified in a box of the width specified by the third parameter to \dottedtocline.
- \texttt{page} is the page number. This is set right justified in a box of width \@pnumwidth, which may be redefined by the user.

It is now easy to define the three hooks \@section, \@subsection and \@subsubsection using \dottedtocline:

```
\renewcommand{\section}{%\@dottedtocline{1}{2.5em}{3em}}
\renewcommand{\subsection}{%\@dottedtocline{2}{4.5em}{3em}}
\renewcommand{\subsubsection}{%\@dottedtocline{3}{7.5em}{3em}}
```

(Since the two parameters to \vsection etc. are to be transmitted unaltered to \dottedtocline, they have been omitted in these definitions.)

### 6.5 Modifying the footnotes

In lecnotes I wanted the footnote markers to be more striking than in the standard classes, so I redefined \makefnmark to place them in a box as shown in figure 1:

```
\renewcommand{\makefnmark}{%\hspace*{0.1em}}
```

I also redefined \makefntext to change the appearance of the actual footnote; in particular I wanted the footnote marker to ‘hang out’ in the left margin, and I wanted a wider paragraph indentation:

```
\setlength{\parindent}{3em} %
\setlength{\parskip}{1pt plus 0.5pt minus 0.25pt} %
```

I wanted more space between the footnotes:

```
\setlength{\footnotesep}{12pt} %
```

Finally, I wanted a longer and thicker rule separating the footnotes from body text:

```
\renewcommand{\footnoterule}{%\vspace*{-1pt} %}
```

In lecnotes I use \dotteddottedtocline instead. It is my own very simple modification of \dottedtocline placing the dots in groups of two, as shown in figure 5.
7 Incorporating packages

There is little sense in reinventing features implemented by others, so when writing a document class some of the work is to make those features available, perhaps with a different interface.

To create a suitable command for showing program examples, as shown on figure 6, I found the material I needed in three packages:

**Verbatim** by Rainer Schöpf makes it possible to include verbatim code from a file.

**Moreverb** by Angus Duggan extends *verbatim* with line numbers and correct handling of TAB characters.

**Float** by Anselm Lingnau makes it possible to design new kinds of floats.

The loading of these packages was specified in the \texttt{loading} part of the class file.

I could now create the new float environment I needed:

\begin{example}[tp]\texttt{[loew][chapter]} \texttt{\floatname{example}{\lnotes{exa}@id}} \texttt{\floatstyle{plain}}

Using this, I could make a command \texttt{\examplefile} with two parameters: the name of the example file, and a caption:

\begin{verbatim}
\newcommand{\examplefile}[2]{% \begin{example} \% First the top rules:
\hrule \@height 1.0\p@ \vspace{2\p@} \hrule \@height 0.4\p@ \vspace{4\p@} \%

\% Then the example text:
\begin{verbatim}
\% After that, the bottom rules:
\end{verbatim} \hrule \@height 0.4\p@ \vspace{2\p@} \hrule \@height 1.0\p@ \relax
\%

\% And finally, the caption:
\caption{#2}
\end{example}
\end{verbatim}

I added the two rules above and below the code to make it stand out more clearly from the body text.

The final touch is the macro \texttt{\listofexamples}. All the necessary code was in the \texttt{float} package; I only needed to add some font specifications:

\begin{verbatim}
\newcommand{\listofexamples}{% \begin{verbatim}
\lnotes{lis}@txt
\setlength{\parskip}{\z@ \@plus 1\p@} \lnotes{font}@\@id\% \rmfamily \llist{example}{\lnotes{exa}@tit}%
\end{verbatim}}
\end{verbatim}

8 Modifying the code

Even though you can get quite a long way just using the parameters and hooks supplied by the \LaTeX{} designers, you sometimes need to modify the actual \LaTeX{} code.
CHAPTER 5. A SHORT INTRODUCTION

The standard report class

One example is the \@startsection command which does not allow negative space after a heading. Since I use \parskip>0, I get too much vertical space between the heading and the following text. The only solution is to redefine \startsection to use the fourth parameter to specify display or run-in heading rather than the fifth. No lines are indented in the lecnotes class anyway.

8.1 Redefining the abstract environment

Another example is the abstract environment which I want to start on an odd-numbered page, use a special font, and produce an extra blank page if twoside is used. The only way to achieve this is to redefine it:

\begin{xxx}
\newenvironment{abstract}{% 
}{}% 
\let\lnotes@endabstract = \endabstract% 
\let\lnotes@theabstract = \theabstract% 
\begin{titlepage}
\mbox{} \vfill
\centerline{\lnotes@abs@hed} 
\mbox{} \vfill
\begin{titlepage}
\if@twoside 
\begin{titlepage}
\mbox{} 
\end{titlepage}
\end{titlepage}
\fi
\end{xxx}

8.2 Modifying the theindex environment

One does not always have to redefine the whole piece of code, since it is easy to keep the original under a different name using a \let, and just specify some additional action. For instance, to ensure that the theindex environment always starts on an odd-numbered page, the following will suffice.\footnote{Internally, \LaTeX \ implements a \texttt{\begin{xxx} as a \texttt{xxx}, and an \texttt{\end{xxx} as an \texttt{endxxx}.}

\let\lnotes@theindex = \theindex% 
\let\lnotes@endtheindex = \endtheindex% 
\begin{xxx}
\renewenvironment{index}% 
{}% 
{\cleardoublepage \lnotes@theindex}% 
\end{xxx}

9 Conclusion

My intention with this description of lecnotes is not to promote it as a great new document class, which it is not, but to show that it is quite easy to create a new document class. My hope is that others will make classes of their own, and make them widely available.

The lecnotes class can be found at \texttt{ftp.ifi.uio.no: pub\tex\lecnotes.cls.}

---

Reprint MAPS#13 (94.2); Nov 1994

Dutch \TeX \ Users Group (NTG), P.O. Box 394, 1740 AJ Schagen, The Netherlands