Chapter 4

Specialities

When putting together a large document, LaTeX will help with some special features like index generation, bibliography management, and other things. A much more complete description of specialities and enhancements possible with LaTeX can be found in the *LaTeX Manual* [1] and *The LaTeX Companion* [3].

4.1 Bibliography

Produce a bibliography with the `thebibliography` environment. Each entry starts with

`\bibitem[`*label*`]{`*marker*`}`

The *marker* is then used to cite the book, article or paper within the document.

`\cite{`*marker*`}`

If you do not use the *label* option, the entries will get enumerated automatically. The parameter after the `\begin{thebibliography}` command defines how much space to reserve for the number of labels. In the example below, `{99}` tells LaTeX to expect that none of the bibliography item numbers will be wider than the number 99.

```
Partl~\cite{pa} has
proposed that \ldots
\begin{thebibliography}{99}
\bibitem{pa} H.~Partl:
\emph{German \TeX},
TUGboat Volume~9, Issue~1 (1988)
\end{thebibliography}
```

Partl [1] has proposed that ...

Bibliography

[1] H. Partl: *German TEX*, TUGboat Volume 9, Issue 1 (1988)

For larger projects, you might want to check out the BibTEX program. BibTEX is included with most TEX distributions. It allows you to maintain a bibliographic database and then extract the references relevant to things you cited in your paper. The visual presentation of BibTEX-generated bibliographies is based on a style-sheets concept that allows you to create bibliographies following a wide range of established designs.

4.2 Indexing

A very useful feature of many books is their index. With LATEX and the support program `makeindex`,[1] an index can be generated quite easily. This introduction will only explain the basic index generation commands. For a more in-depth view, please refer to *The LATEX Companion* [3].

To enable their indexing feature of LATEX, the makeidx package must be loaded in the preamble with

```
\usepackage{makeidx}
```

and the special indexing commands must be enabled by putting the

```
\makeindex
```

command in the preamble.

[1] On systems not necessarily supporting filenames longer than 8 characters, the program may be called `makeidx`.

Table 4.1: Index Key Syntax Examples.

Example	Index Entry	Comment
`\index{hello}`	hello, 1	Plain entry
`\index{hello!Peter}`	Peter, 3	Subentry under 'hello'
`\index{Sam@\textsl{Sam}}`	*Sam*, 2	Formatted entry
`\index{Lin@\textbf{Lin}}`	**Lin**, 7	Formatted entry
`\index{Kaese@\textbf{K\"ase}}`	**Käse**, 33	Formatted entry
`\index{ecole@\'ecole}`	école, 4	Formatted entry
`\index{Jenny|textbf}`	Jenny, **3**	Formatted page number
`\index{Joe|textit}`	Joe, *5*	Formatted page number

The content of the index is specified with

`\index{`*key*`@`*formatted_ entry*`}`

commands, where *formatted_ entry* will appear in the index and *key* will be used for sorting. The *formatted_ entry* is optional. If it is missing the *key* will be used. You enter the index commands at the points in the text that you want the final index entries to point to. Table 4.1 explains the syntax with several examples.

When the input file is processed with LaTeX, each `\index` command writes an appropriate index entry, together with the current page number, to a special file. The file has the same name as the LaTeX input file, but a different extension (`.idx`). This `.idx` file can then be processed with the `makeindex` program:

`makeindex` *filename*

The `makeindex` program generates a sorted index with the same base file name, but this time with the extension `.ind`. If now the LaTeX input file is processed again, this sorted index gets included into the document at the point where LaTeX finds

`\printindex`

The showidx package that comes with LaTeX 2$_\varepsilon$ prints out all index entries in the left margin of the text. This is quite useful for proofreading a document and verifying the index.

Note that the `\index` command can affect your layout if not used carefully.

```
My Word \index{Word}. As opposed
to Word\index{Word}. Note the
position of the full stop.
```

My Word . As opposed to Word. Note the position of the full stop.

makeindex has no clue about characters outside the ASCII range. To get the sorting correct, use the @ character as shown in the Käse and école examples above.

4.3 Fancy Headers

The fancyhdr package,[2] written by Piet van Oostrum, provides a few simple commands that allow you to customize the header and footer lines of your document. Look at the top of this page, for an application of this package.

```
\documentclass{book}
\usepackage{fancyhdr}
\pagestyle{fancy}
% with this we ensure that the chapter and section
% headings are in lowercase.
\renewcommand{\chaptermark}[1]{%
        \markboth{#1}{}}
\renewcommand{\sectionmark}[1]{%
        \markright{\thesection\ #1}}
\fancyhf{}  % delete current header and footer
\fancyhead[LE,RO]{\bfseries\thepage}
\fancyhead[LO]{\bfseries\rightmark}
\fancyhead[RE]{\bfseries\leftmark}
\renewcommand{\headrulewidth}{0.5pt}
\renewcommand{\footrulewidth}{0pt}
\addtolength{\headheight}{0.5pt} % space for the rule
\fancypagestyle{plain}{%
   \fancyhead{} % get rid of headers on plain pages
   \renewcommand{\headrulewidth}{0pt} % and the line
}
```

Figure 4.1: Example fancyhdr Setup.

The tricky problem when customising headers and footers is to get things like running section and chapter names in there. LaTeX accomplishes this with a two-stage approach. In the header and footer definition, you use the commands \rightmark and \leftmark to represent the current section

[2] Available from CTAN://macros/latex/contrib/supported/fancyhdr.

and chapter heading, respectively. The values of these two commands are overwritten whenever a chapter or section command is processed.

For ultimate flexibility, the `\chapter` command and its friends do not redefine `\rightmark` and `\leftmark` themselves. They call yet another command (`\chaptermark`, `\sectionmark`, or `\subsectionmark`) that is responsible for redefining `\rightmark` and `\leftmark`.

If you want to change the look of the chapter name in the header line, you need only "renew" the `\chaptermark` command.

Figure 4.1 shows a possible setup for the fancyhdr package that makes the headers look about the same as they look in this booklet. In any case, I suggest you fetch the documentation for the package at the address mentioned in the footnote.

4.4 The Verbatim Package

Earlier in this book, you got to know the `verbatim` *environment*. In this section, you are going to learn about the verbatim *package*. The verbatim package is basically a re-implementation of the `verbatim` environment that works around some of the limitations of the original `verbatim` environment. This by itself is not spectacular, but the implementation of the verbatim package added new functionality, which is why I am mentioning the package here. The verbatim package provides the

`\verbatiminput{`*filename*`}`

command, which allows you to include raw ASCII text into your document as if it were inside a `verbatim` environment.

As the verbatim package is part of the 'tools' bundle, you should find it pre-installed on most systems. If you want to know more about this package, make sure to read [10].

4.5 Installing Extra Packages

Most LaTeX installations come with a large set of pre-installed style packages, but many more are available on the net. The main place to look for style packages on the Internet is CTAN (`http://www.ctan.org/`).

Packages such as geometry, hyphenat, and many others are typically made up of two files: a file with the extension `.ins` and another with the extension `.dtx`. There will often be a `readme.txt` with a brief description of the package. You should of course read this file first.

In any event, once you have copied the package files onto your machine, you still have to process them in a way that (a) tells your TeX distribution

about the new style package and (b) gives you the documentation. Here's how you do the first part:

1. Run LaTeX on the `.ins` file. This will extract a `.sty` file.
2. Move the `.sty` file to a place where your distribution can find it. Usually this is in your *…/localtexmf*`/tex/latex` subdirectory (Windows or OS/2 users should feel free to change the direction of the slashes).
3. Refresh your distribution's file-name database. The command depends on the LaTeXdistribution you use: TeXlive – `texhash`; web2c – `maktexlsr`; MiKTeX – `initexmf --update-fndb` or use the GUI.

Now extract the documentation from the `.dtx` file:

1. Run XeLaTeX on the `.dtx` file. This will generate a `.pdf` file. Note that you may have to run XeLaTeX several times before it gets the cross-references right.
2. Check to see if LaTeX has produced a `.idx` file among the various files you now have. If you do not see this file, then the documentation has no index. Continue with step 5.
3. In order to generate the index, type the following:
   `makeindex -s gind.ist` *name*
   (where *name* stands for the main-file name without any extension).
4. Run LaTeX on the `.dtx` file once again.
5. Last but not least, make a `.ps` or `.pdf` file to increase your reading pleasure.

Sometimes you will see that a `.glo` (glossary) file has been produced. Run the following command between step 4 and 5:
`makeindex -s gglo.ist -o` *name*`.gls` *name*`.glo`
Be sure to run LaTeX on the `.dtx` one last time before moving on to step 5.

4.6 LaTeX and PDF

By Daniel Flipo <Daniel.Flipo@univ-lille1.fr>

PDF is a portable hypertext document format. Much as in a web page, some words in the document are marked as hyperlinks. They link to other places in the document or even to other documents. If you click on such a hyperlink you get transported to the destination of the link. In the context of LaTeX, this means that all occurrences of `\ref` and `\pageref` become hyperlinks. Additionally, the table of contents, the index and all the other similar structures become collections of hyperlinks.

Most web pages you find today are written in HTML *(HyperText Markup Language)*. This format has two significant disadvantages when writing scientific documents:

1. Including mathematical formulae into HTML documents is not generally supported. While there is a standard for it, most browsers used today do not support it, or lack the required fonts.

2. Printing HTML documents is possible, but the results vary widely between platforms and browsers. The results are miles removed from the quality we have come to expect in the LaTeX world.

There have been many attempts to create translators from LaTeX to HTML. Some were even quite successful in the sense that they are able to produce legible web pages from a standard LaTeX input file. But all of them cut corners left and right to get the job done. As soon as you start using more complex LaTeX features and external packages things tend to fall apart. Authors wishing to preserve the unique typographic quality of their documents even when publishing on the web turn to PDF *(Portable Document Format)*, which preserves the layout of the document and permits hypertext navigation. Most modern browsers come with plugins that allow the direct display of PDF documents.

All modern TeX engines can generate PDF files out of the box. If you worked through this introduction until here you will already be familiar with the process.

4.6.1 Hypertext Links

The hyperref adds two cool features to your LaTeX PDF files:

1. The paper size is set according to your specification in the document class call.

2. All references in your your document turn into hyperlinks.

Just add `\usepackage{hyperref}` as the *last* command into the preamble of your document.

Many options are available to customize the behaviour of the hyperref package:

- either as a comma separated list after the pdftex option `\usepackage{hyperref}`
- or on individual lines with the command `\hypersetup{`*options*`}`.

In the following list the default values are written in an upright font.

bookmarks (=true,*false*) show or hide the bookmarks bar when displaying the document

unicode (=false,*true*) allows the use of characters of non-Latin based languages in Acrobat's bookmarks

pdftoolbar (=true,*false*) show or hide Acrobat's toolbar

pdfmenubar (=true,*false*) show or hide Acrobat's menu

pdffitwindow (=false,*true*) adjust the initial magnification of the PDF when displayed

pdftitle (={text}) define the title that gets displayed in the `Document Info` window of Acrobat

pdfauthor (={text}) the name of the PDF's author

pdfnewwindow (=false,*true*) define whether a new window should be opened when a link leads out of the current document

colorlinks (=false,*true*) surround the links by colour frames (`false`) or colour the text of the links (`true`). The colour of these links can be configured using the following options (default colours are shown):

linkcolor (=red) colour of internal links (sections, pages, etc.)

citecolor (=green) colour of citation links (bibliography)

filecolor (=magenta) colour of file links

urlcolor (=cyan) colour of URL links (mail, web)

If you are happy with the defaults, use

```
\usepackage{hyperref}
```

To have the bookmark list open and links in colour (the `=true` values are optional):

```
\usepackage[bookmarks,colorlinks]{hyperref}
```

When creating PDFs destined for printing, coloured links are not a good thing as they end up in gray in the final output, making it difficult to read. Use colour frames, which are not printed:

```
\usepackage{hyperref}
\hypersetup{colorlinks=false}
```

or make links black:

```
\usepackage{hyperref}
\hypersetup{colorlinks,%
            citecolor=black,%
            filecolor=black,%
            linkcolor=black,%
            urlcolor=black,%
            pdftex}
```

When you just want to provide information for the `Document Info` section of the PDF file:

```
\usepackage[pdfauthor={Pierre Desproges},%
            pdftitle={Des femmes qui tombent},%
            pdftex]{hyperref}
```

In addition to the automatic hyperlinks for cross references, it is possible to embed explicit links using

`\href{`*url*`}{`*text*`}`

The code

```
The \href{http://www.ctan.org}{CTAN} website.
```

produces the output "CTAN"; a click on the word "CTAN" will take you to the CTAN website.

If the destination of the link is not a URL but a local file, use the `\href` command without the 'http://' bit:

```
The complete document is \href{manual.pdf}{here}
```

which produces the text "The complete document is here". A click on the word "here" will open the file `manual.pdf`. (The filename is relative to the location of the current document).

The author of an article might want her readers to easily send email messages by using the `\href` command inside the `\author` command on the title page of the document:

```
\author{Mary Oetiker $<$\href{mailto:mary@oetiker.ch}%
       {mary@oetiker.ch}$>$
```

Note that I have put the link so that my email address appears not only in the link but also on the page itself. I did this because the link
`\href{mailto:mary@oetiker.ch}{Mary Oetiker}`
would work well within Acrobat, but once the page is printed the email address would not be visible anymore.

4.6.2 Problems with Links

Messages like the following:

```
! pdfTeX warning (ext4): destination with the same
  identifier (name{page.1}) has been already used,
  duplicate ignored
```

appear when a counter gets reinitialized, for example by using the command `\mainmatter` provided by the `book` document class. It resets the page number counter to 1 prior to the first chapter of the book. But as the preface of the book also has a page number 1 all links to "page 1" would not be unique anymore, hence the notice that "`duplicate` has been `ignored`."

The counter measure consists of putting `plainpages=false` into the hyperref options. This unfortunately only helps with the page counter. An even more radical solution is to use the option `hypertexnames=false`, but this will cause the page links in the index to stop working.

4.6.3 Problems with Bookmarks

The text displayed by bookmarks does not always look like you expect it to look. Because bookmarks are "just text," fewer characters are available for bookmarks than for normal LaTeX text. Hyperref will normally notice such problems and put up a warning:

```
Package hyperref Warning:
Token not allowed in a PDFDocEncoded string:
```

Work around this problem by providing a text string for the bookmarks, which replaces the offending text:

`\texorpdfstring{`*TEX text*`}{`*Bookmark Text*`}`

Math expressions are a prime candidate for this kind of problem:

```
\section{\texorpdfstring{$E=mc^2$}%
        {E = mc ** 2}}
```

which turns `\section{$E=mc^2$}` to "E = mc ** 2" in the bookmark area.

If you write your document in Unicode and use the `unicode` option for the hyperref package to use Unicode characters in bookmarks, this will give you a much larger selection of characters to pick from when when using `\texorpdfstring`.

4.7 Working with XeLaTeX and PDF

By Axel Kielhorn <A.Kielhorn@web.de>

Most of the things said in the previous section are valid for XeLaTeX as well.

There is a Wiki at http://wiki.xelatex.org/doku.php that collects information relevant to XeTeX and XeLaTeX.

4.7.1 The Fonts

In addition to the normal `tfm` based fonts, XeLaTeX is able to use any font known to the operating system. If you have the `Linux Libertine` fonts installed, you can simply say

```
\usepackage{fontspec}
\setmainfont[Ligatures=TeX]{Linux Libertine}
```

in the preamble. This will normally detect the italic and bold versions as well, so `\textit` and `\textbf` will work as usual. When the font is using OpenType technology you have access to many features which required switching to a separate font or using virtual fonts in the past. The main feature is the extended character set; a font may contain Latin, Greek and Cyrillic characters and the corresponding ligatures.

Many fonts contain at least two kinds of numerals, the normal lining numerals and so called old style (or lower case) numerals, which partly extend below the baseline. They may contain proportional numerals (the "1" takes less space than the "0") or monospaced numerals which are suitable for tables.

```
\newfontfamily\LLln[Numbers=Lining]{(font)}
\newfontfamily\LLos[Numbers=OldStyle]{(font)}
\newfontfamily\LLlnm[Numbers=Lining,Numbers=Monospaced]{(font)}
\newfontfamily\LLosm[Numbers=OldStyle,Numbers=Monospaced]{(font)}
```

Almost all OpenType fonts contain the standard ligatures (fl fi ffi) but there are also some rare or historical ligatures like st, ct and tz. You may not want to use them in a technical report but they are fine for a novel. To enable these ligatures use either of the following lines:

```
\setmainfont[Ligatures=Rare]{(font)}
\setmainfont[Ligatures=Historic]{(font)}
\setmainfont[Ligatures=Historic,Ligatures=Rare]{(font)}
```

Not every font contains both sets of ligature, consult the font documentation or just try it out. Sometimes these ligatures are language dependent; for example a ligature used in Polish (fk) is not used in English. You have to add

```
\setmainfont[Language=Polish]{(font)}
```

to enable the Polish ligatures.

Some fonts (like the commercial Adobe Garamond Premier Pro) contain alternative glyphs that are activated by default in XeLaTeX distributed with TeXLive 2010[3]. The result is a stylish "Q" with a descender reaching below the following "u". To disable this feature you have to define the font with disabled contextuals:

```
\setmainfont[Contextuals=NoAlternate]{(font)}
```

To learn about fonts in XeLaTeX read the fontspec manual.

Where do I get OpenType fonts?

If you have TeXLive installed, you already have some at .../texmf-dist/fonts/opentype, just install them in your operating system. This collection does not include DejaVu, which is available at http://dejavu-fonts.org/.

Make sure that each font is only installed *once*, otherwise interesting results may happen.

You can use every font installed on your computer, but remember that other users may not have these fonts. The Zapfino font used in the fontspec manual is included in Mac OSX, but is not available on Windows computers.[4]

Entering Unicode Characters

The number of characters in a font has grown but the number of keys on a regular keyboard has not. So, how do I enter non-ASCII characters?

If you write a large amount of text in a foreign language, you can install a keyboard for that language and print out the character positions. (Most operatings system have some sort of virtual keyboard, just make a screenshot.)

If you rarely need an exotic character, you can simply pick it in the character palette.

Some environments (e. g. the X Window System) offer many methods to enter non-ASCII characters. Some editors (e. g. Vim and Emacs) offer ways to enter these characters. Read the manual for the tools you are using.

4.7.2 Compatibility Between XeLaTeX and pdfLaTeX

There are a few things that are different between XeLaTeX and pdfLaTeX.

[3] The behavior has changed with this version, it was off by default in earlier releases.

[4] A commercial version of the font called Zapfino Extra is available.

- A XeLaTeX document has to be written in Unicode (UTF-8) while pdfLaTeX may use different input encodings.
- The microtype packages does not work with XeLaTeX yet, support for character protrusion is already under development.
- Everything font related has to be reviewed. (Unless you want to stick to Latin Modern.)

4.8 Creating Presentations

By Daniel Flipo <Daniel.Flipo@univ-lille1.fr>

You can present the results of your scientific work on a blackboard, with transparencies, or directly from your laptop using some presentation software.

pdfLaTeX combined with the beamer class allows you to create presentations in PDF, looking much like something you might be able to generate with LibreOffice or PowerPoint if you had a very good day, but much more portable because PDF readers are available on many more systems.

The beamer class uses graphicx, color and hyperref with options adapted to screen presentations.

When you compile the code presented in figure 4.2 with pdfLaTeX you get a PDF file with a title page and a second page showing several items that will be revealed one at a time as you step though your presentation.

One of the advantages of the beamer class is that it produces a PDF file that is directly usable without first going through a PostScript stage like prosper or requiring additional post processing like presentations created with the ppower4 package.

With the beamer class you can produce several versions (modes) of your document from the same input file. The input file may contain special instructions for the different modes in angular brackets. The following modes are available:

beamer for the presentation PDF discussed above.

trans for transparencies.

handout for the printed version.

The default mode is `beamer`, change it by setting a different mode as a global option, like `\documentclass[10pt,handout]{beamer}` to print the handouts for example.

The look of the screen presentation depends on the theme you choose. Pick one of the themes shipped with the beamer class or create your own. See the beamer class documentation in `beameruserguide.pdf` for more information on this.

```
\documentclass[10pt]{beamer}
\mode<beamer>{%
  \usetheme[hideothersubsections,
            right,width=22mm]{Goettingen}
}

\title{Simple Presentation}
\author[D. Flipo]{Daniel Flipo}
\institute{U.S.T.L. \& GUTenberg}
\titlegraphic{\includegraphics[width=20mm]{USTL}}
\date{2005}

\begin{document}

\begin{frame}<handout:0>
  \titlepage
\end{frame}

\section{An Example}

\begin{frame}
  \frametitle{Things to do on a Sunday Afternoon}
  \begin{block}{One could \ldots}
    \begin{itemize}
      \item walk the dog\dots \pause
      \item read a book\pause
      \item confuse a cat\pause
    \end{itemize}
  \end{block}
  and many other things
\end{frame}
\end{document}
```

Figure 4.2: Sample code for the beamer class

Let's have a closer look at the code in figure 4.2.

For the screen version of the presentation `\mode<beamer>` we have chosen the *Goettingen* theme to show a navigation panel integrated into the table of contents. The options allow us to choose the size of the panel (22 mm in this case) and its position (on the right side of the body text). The option *hideothersubsections*, shows the chapter titles, but only the subsections of the present chapter. There are no special settings for `\mode<trans>` and `\mode<handout>`. They appear in their standard layout.

The commands `\title{}`, `\author{}`, `\institute{}`, and `\titlegraphic{}` set the content of the title page. The optional arguments of `\title[]{}` and `\author[]{}` let you specify a special version of the title and the author name to be displayed on the panel of the *Goettingen* theme.

The titles and subtitles in the panel are created with normal `\section{}` and `\subsection{}` commands that you place *outside* the `frame` environment.

The tiny navigation icons at the bottom of the screen also allow to navigate the document. Their presence is not dependent on the theme you choose.

The contents of each slide or screen has to be placed inside a `frame` environment. There is an optional argument in angular brackets (< and >), it allows us to suppress a particular frame in one of the versions of the presentation. In the example the first page would not be shown in the handout version due to the `<handout:0>` argument.

It is highly recommended to set a title for each slide apart from the title slide. This is done with the command `\frametitle{}`. If a subtitle is necessary use the `block` environment as shown in the example. Note that the sectioning commands `\section{}` and `\subsection{}` do not produce output on the slide proper.

The command `\pause` in the itemize environment lets you reveal the items one by one. For other presentation effects check out the commands `\only`, `\uncover`, `\alt` and `\temporal`. In many place it is possible to use angular brackets to further customize the presentation.

In any case make sure to read through the beamer class documentation `beameruserguide.pdf` to get a complete picture of what is in store for you. This package is being actively developed, check out their website to get the latest information. (http://latex-beamer.sourceforge.net/)