Using LaTeX to Write a PhD Thesis

Nicola L.C. Talbot

School of Computing Sciences
University of East Anglia
Norwich. NR4 7TJ. U.K.
http://theoval.cmp.uea.ac.uk/~nlct/

Tuesday 6th November, 2007 (version 1.2)

The base URL for this document is: `http://theoval.cmp.uea.ac.uk/~nlct/latex/thesis/`

Any opinions made herein are my own, and do not necessarily reflect the opinions of the University of East Anglia.

Abstract

This document is aimed at PhD students who want to use LaTeX to typeset their PhD thesis. If you are unfamiliar with LaTeX I recommend that you first read LaTeX for Complete Novices [5].

Contents

Chapter 1

Introduction

Many PhD students in the sciences are being encouraged to produce their PhD thesis in LaTeX, particularly if their work involves a lot of mathematics. This document is intended as a brief guide on how to structure your document, and how to define new page styles, chapter headings and various other components that are usually required for a thesis. If you have never used LaTeX before, I recommend that you first read LaTeX for Complete Novices [5], as this document assumes you have a basic knowledge of LaTeX.

Throughout this document, source code is illustrated using a typewriter font like this:

```
This is an \textbf{example}.
```

Where both sample source code and the corresponding output is illustrated, then the source code is illustrated in the form:

↑Input

```
This is an \textbf{example}.
```

↓Input

and the corresponding output is illustrated like this:

↑Output

This is an **example**.

↓Output

Command definitions are shown in a typewriter font in the form:

`\documentclass[`*<options>*`]{`*<class file>*`}` Definition

where words *<like this>* (such as *<options>* and *<class file>*) indicate the type of thing you need to substitute. Note that the angle brackets are merely a visual aid to indicate that the text is a metasyntactic variable, so the angle brackets should not appear in your code. For example if you want to use the **report** class file, you would need to substitute *<class file>* with `report`, and if you want the `a4paper` option, you would substitute *<options>* with `a4paper`, like this:

```
\documentclass[a4paper]{report}
```

Links to related topics in the UKTUG FAQ are displayed in the margin like this: [What is TEX?]

Chapter 2

Getting Started

If you have been told to use a particular class file, use that one, otherwise I recommend that you use the report or scrreprt class file. Before you start your document, consider first what kind of structure it should have. Unless you have been told otherwise, I recommend that you start out with a skeletal document that looks something like the following:

[Replacing the standard classes]

```
\documentclass[a4paper]{report}

\begin{document}

\title{A Sample PhD Thesis}
\author{A. N. Other}
\date{July 2004}

\maketitle

\pagenumbering{roman}
```

```
\tableofcontents
\listoffigures
\listoftables

\chapter*{Acknowledgements}

\begin{abstract}
\end{abstract}

\pagenumbering{arabic}

\chapter{Introduction}
\label{ch:intro}

\chapter{Technical Introduction}
\label{ch:techintro}

\chapter{Method}
\label{ch:method}

\chapter{Results}
\label{ch:results}
```

```
\chapter{Conclusions}
\label{ch:conc}

\bibliographystyle{plain}
\bibliography{thesis}

\end{document}
```

If you do this, it will help ensure that your document has the correct structure before you begin with the actual contents of the document. (Note that the chapter titles will naturally vary depending on your subject or institution, and you may need a different paper size if you are not in Europe. I have based the above on my own PhD thesis which I wrote in the early to mid 1990s in the Department of Electronic Systems Engineering at the University of Essex, and it may well not fit your own requirements.)

Note that I have included the lines

```
\bibliographystyle{plain}
\bibliography{thesis}
```

however I haven't yet created the bibliography database `thesis.bib`. I will cover this in Chapter 5, but you will still be able to run the document through LaTeX. If you haven't started yet, go ahead and try this. Creating a skeletal document can have an amazing psychological effect on some people: for very little effort it can produce a document several pages long, which can give you a sense of achievement which can help give you sufficient momentum to get started[1].

If you are using the scrreprt class you can use the commands `\frontmatter`, `\mainmatter` and `\backmatter` to delineate the various logical divisions of your document. These commands are also defined in some other classes, such as book and memoir.

[1] but of course, it's not guaranteed to work with everyone.

Chapter 3

Splitting a Large Document into Several Files

Some people prefer to place each chapter of a large document in a separate file. You can do this by using the command

\include{<*filename*>**}** Definition

If you only want to work on one or two chapters, you can tell LaTeX to only include those files using the command

\includeonly{<*file list*>**}** Definition

in the preamble, where <*file list*> is a comma separated list of files you want included. LaTeX will still read in all the cross-referencing information for the missing chapters, but won't include them in the DVI file. There is a definite advantage to this if you have, say, a large number of images in your results chapter, which you don't need when you're working on, say, the technical introduction. You can still reference all the figures in the omitted chapter, as long as you have previously LaTeXed the document without the `\includeonly` command.

The example given in Chapter 2 can now be split into various files:
File thesis.tex:

```
\documentclass[a4paper]{report}

\begin{document}

\title{A Sample PhD Thesis}
\author{A. N. Other}
\date{July 2004}

\maketitle

\pagenumbering{roman}
\tableofcontents
\listoffigures
\listoftables

\chapter*{Acknowledgements}
```

```
\begin{abstract}
\end{abstract}

\pagenumbering{arabic}

\include{intro}

\include{techintro}

\include{method}

\include{results}

\include{conc}

\bibliographystyle{plain}
\bibliography{thesis}

\end{document}
```

File intro.tex:

```
\chapter{Introduction}
```

```
\label{ch:intro}
```

File techintro.tex:

```
\chapter{Technical Introduction}
\label{ch:techintro}
```

File method.tex:

```
\chapter{Method}
\label{ch:method}
```

File results.tex:

```
\chapter{Results}
\label{ch:results}
```

File conc.tex:

```
\chapter{Conclusions}
\label{ch:conc}
```

If you only want to work on, say, the Method and the Results chapters, you can place the following command in the preamble:

```
\includeonly{method,results}
```

3.1 Excluding Files

There is also a command called `\excludeonly` defined in the excludeonly package which performs the reverse of `\includeonly`.

3.2 `\input` and `\include`

Some people become confused over the difference between `\include` and `\input`.

[What's going on in my `\include` commands?]

Definition

`\input{`*<filename>*`}`

acts as though the contents of the file called *<filename>* were inserted into the document at the point where the `\input` command occurs. For example, if you have a file called `myfile.tex` which contained the following lines:

```
Hello World!
Goodbye World!
```

and you had another file called `mydoc.tex` which contained the following:

```
\documentclass{article}
```

```
\begin{document}
\input{myfile}
\end{document}
```

then `mydoc.tex` is equivalent to

```
\documentclass{article}
\begin{document}
Hello World!
Goodbye World!
\end{document}
```

Whereas

`\include{`<*filename*>`}` Definition

does more than simply read the contents of the file called <*filename*>`.tex`. Firstly, an associated auxiliary file is created called <*filename*>`.aux`. This file contains all the cross-referencing information (produced by `\label` and `\cite`) that occurs in <*filename*>`.tex`. This means that any labels in the files that have been excluded (either by not being listed in `\includeonly` or by being listed in `\excludeonly`) can still be referenced in other parts of the document.

Secondly, `\clearpage` is issued, then the file name is checked to determine if it is in the included list. If it is, the file contents will then be read and the cross-referencing information will be written to <*filename*>`.tex`, otherwise the file contents will be ignored. At the end of the file, another `\clearpage` is issued. This is why it makes sense to only use `\include` where the included file contains an entire chapter (including `\chapter` and corresponding `\label` commands.)

Chapter 4
Changing the Document Style

It is possible to redefine `\chapter`, `\section` etc in order to change the heading style for your document. I recommend that you first write your thesis, and then worry about changing the document style; the ability to do this is one of the advantages of using LaTeX over a word processor. Remember that writing your thesis is more important than the layout. Whilst it may be that your school or department may insist on a certain style, it should not take precedence over the actual task of writing.

Some class files, such as the KOMA script classes (which include scrreprt mentioned in Chapter 2) and the memoir class provide commands to help you modify the document style. There are also packages available to help you modify the appearance of chapter and section headings. Alternatively, you may prefer to write your own class or package which will produce a document that conforms to your school's guidelines (perhaps you have friends who may also benefit from this.)

[The style of section headings]

If you want to know how a particular class or package will enable you to modify the document style, then you should read the user guide for that class or package. In this tutorial I shall illustrate how you can create your own style which will be based on the report class. Note that if you want to redefine commands such as `\chapter` and `\section`, using the methods described below, it is better to create a class or package rather than putting the commands directly in your document. There are two main reasons for this: firstly, some of the commands involved use an `@` character which behaves differently depending on whether or not it occurs in a class/package or in a normal `.tex` file, and secondly, if you place all these commands in your main document, you may confuse the spell checker or word count application[1].

[Learning to write LaTeX classes and packages]

[How many words have you written?]

So, should you create a package or a class file? Packages should be designed to be independent of the class file. For example, the graphicx package works irrespective of whether you are using the report, article, slide etc class file. If the commands or environments that you want to define are somehow dependent on a particular class file, then you should create a new class file that is based on the one you want. If you are redefining chapter or section styles, then this is dependent on the overall document style, that is, it's dependent on the class file. So, you should create a new class file that modifies the existing one, rather than creating a package.

[1] for information on counting the number of words in your document, see the documentation for the cmpreprt class file

Let's have an example. If you want to create a new class called, say, mythesis, you will need to create a file called `mythesis.cls`, and the start of your file should look something like:

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesClass{mythesis}
```

Next you need to specify what to do with any options passed to this class file. Since we don't need to define any new options for this example, we can simply pass all options on to the report class file:

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

Once all options have been declared, they need to be processed:

```
\ProcessOptions
```

Now the report class needs to be loaded:

```
\LoadClass{report}
```

and the very last line of your file should have the command

```
\endinput
```

The contents of this new class file should be inserted between the `\LoadClass{report}` and `\endinput` commands. You will then need to modify your source code, `thesis.tex`, so that it uses this new class file:

```
\documentclass[a4paper]{mythesis}
```

If the class file is only intended to conform to a specific university or school's guidelines, it makes more sense for your new class file to set the paper size to a fixed size. For example, a European university may insist on A4 paper, in which case the paper size is no longer an option. In this case you can either specify the paper size as an option to `\LoadClass`:

```
\LoadClass[a4paper]{report}
```

or you can load the **geometry** package, in which case the following line should go after `\LoadClass{report}`[2]:

```
\RequirePackage[a4paper]{geometry}
```

Either way, you no longer need the `a4paper` option in your document:

```
\documentclass{mythesis}
```

If you need to set additional page layout parameters such as the margin widths, then it is better to use the **geometry** package. For example, to set the paper size to A4 with 1 inch margins do:

```
\RequirePackage[a4paper,margin=1in]{geometry}
```

See the **geometry** package documentation for further details.

[Changing the margins in LaTeX]

[2] Note that in a class or package you should use `\RequirePackage` instead of `\usepackage`

4.1 Modifying Object Specific Text

The report class file defines various commands that produce words such as “Contents”, “Chapter”, “Bibliography”. These commands, and their default values are listed in Table 4.1.

[How to change LATEX’s “fixed names”]

Table 4.1: Default Names

`\contentsname`	Contents
`\listfigurename`	List of Figures
`\listtablename`	List of Tables
`\bibname`	Bibliography
`\indexname`	Index
`\figurename`	Figure
`\tablename`	Table
`\partname`	Part
`\chaptername`	Chapter
`\appendixname`	Appendix
`\abstractname`	Abstract

So, suppose you want your figures and tables to be labelled Fig. and

Tab. instead of Figure and Table, then you could add the following lines to `mythesis.cls`:

```
\renewcommand{\figurename}{Fig.}
\renewcommand{\tablename}{Tab.}
```

4.2 Changing the Section Headings

It is possible to customise the way your section, subsection etc headings appear by redefining the corresponding commands `\section`, `\subsection` etc using the command:

`\@startsection{`*<type>*`}{`*<level>*`}{`*<indent>*`}{`*<beforeskip>*`}{`*<afterskip>*`}{`*<style>*`}`

Definition

The six arguments are as follows:

<type> The sectioning type. This should be one of: `section`, `subsection`, `subsubsection`, `paragraph` or `subparagraph`. (Note no backslash.)

<level> This is the sectioning level as indicated in Table 4.2.

<indent> This should be a length, specifying the indentation from the left margin.

<beforeskip> The absolute value of the *<beforeskip>* specifies how much vertical distance to leave before the heading. If *<beforeskip>* is negative, the first paragraph following the section heading will not be indented.

<afterskip> The absolute value of the *<afterskip>* specifies how much vertical distance to leave after the heading. If *<afterskip>* is negative, the text following the sectioning command will appear on the same level as the section heading.

<style> The *<style>* are the declarations required to set the style of the heading (e.g. `\itshape` for an italic heading.) Note that the last command in *<style>* may be a command which takes a single argument, but all the others must be declarations.

(Remember that `\paragraph` and `\subparagraph` produce sub-sub-sub-sections and sub-sub-sub-sub-sections, respectively, though most class files suppress their numbering and give them a negative *<afterskip>*.)

[How to create a `\subsubsubsection`

As an example, suppose you want to change the section headings so that they appear in a large italic font, you could do something like:

```
\renewcommand{\section}{\@startsection
{section}%                          % the name
{1}%                                % the level
```

Table 4.2: Section Levels

part	-1
chapter	0
section	1
subsection	2
subsubsection	3
paragraph	4
subparagraph	5

```
{0mm}%                          % the indent
{-\baselineskip}%               % the before skip
{0.5\baselineskip}%             % the after skip
{\normalfont\large\itshape}} % the style
```

As mentioned above, the final command within the <*style*> argument may be a command which takes an argument, so you could also do something like:

```
\renewcommand{\section}{\@startsection
{section}%                      % the name
{1}%                            % the level
{0mm}%                          % the indent
{-\baselineskip}%               % the before skip
{0.5\baselineskip}%             % the after skip
{\normalfont\large\MakeUppercase}} % the style
```

which would convert the section heading to uppercase. See *A Guide to LaTeX* [2] for further information.

There is a counter called `secnumdepth` that controls what level the sections have numbers. The levels correspond to those shown in Table 4.2. By default this value is 2, so only parts, chapters, sections and subsections have associated numbers. You can use `\setcounter` to change the value of `secnumdepth`. So, for example, if you want the `\paragraph` command to produce a number, do

```
\settocounter{secnumdepth}{4}
```

4.3 Changing the Chapter Headings

If you want to change the chapter or part heading style, you can't use `\@startsection`. Instead you should use the `\secdef` command. If you load `report.cls` into a text editor, you will see that both the `\part` and `\chapter` commands use `\secdef`. The definition of `\chapter` has the line

```
\secdef\@chapter\@schapter
```

and `\part` has the line

```
\secdef\@part\@spart
```

The first argument to `\secdef` tells LaTeX what to do if the unstarred version is used, and the second argument tells LaTeX what to do if the starred version is used. So the command

```
\chapter{Introduction}
```

will use the command `\@chapter`, whereas the command

```
\chapter*{Acknowledgements}
```

will use the command `\@schapter`. The commands `\@chapter` and `\@schapter` use the commands `\@makechapterhead` and `\@makeschapterhead`, respectively, to format the chapter heading, so if you want to change the chapter format, you will need to redefine the commands `\@makechapterhead` and `\@makeschapterhead`. The easiest way to do this is to look for the code for these commands in `report.cls` and copy them over to your new class file, `mythesis`, described earlier, and edit the appropriate formatting commands.

For example, suppose you want a line to appear above and below the chapter heading, and have the chapter heading appear in small capitals, you could do:

```
\renewcommand{\@makechapterhead}[1]{%
  \vspace*{50\p@}%
  {\parindent \z@ \raggedright \normalfont
    \hrule                                       % horizontal line
    \vspace{5pt}%                                % add vertical space
    \ifnum \c@secnumdepth >\m@ne
        \huge\scshape \@chapapp\space \thechapter % Chapter number
        \par\nobreak
        \vskip 20\p@
    \fi
    \interlinepenalty\@M
    \Huge \scshape #1\par                        % chapter title
    \vspace{5pt}%                                % add vertical space
    \hrule                                       % horizontal rule
    \nobreak
    \vskip 40\p@
  }}

\renewcommand{\@makeschapterhead}[1]{%
  \vspace*{50\p@}%
  {\parindent \z@ \raggedright
    \normalfont
    \hrule                                            % horizontal line
    \vspace{5pt}%                                     % add vertical space
    \interlinepenalty\@M
    \Huge \scshape #1\par                             % chapter title
    \vspace{5pt}%                                     % add vertical space
    \hrule                                            % horizontal line
    \nobreak
    \vskip 40\p@
  }}
```

You can download the file mythesis.cls which includes all the examples covered so far in this chapter.

4.4 Adding to the Table of Contents

Starred versions of the sectioning commands are not added to the table of contents by default, but they can be added using:

\addcontentsline{<*ext*>}{<*type*>}{<*text*>} Definition

<*ext*> This should be the extension of the file where the contents are written. So this will be **toc** for the table of contents, **lof** for the list of figures and **lot** for the list of tables.

<*type*> This is the type of object you are adding to the contents. e.g. chapter, section, figure.

<*text*> This is the text that should go in the contents.

For example, the bibliography is generated using a starred version of the **\chapter** command, so it doesn't get added to the table of contents. To add it to the table of contents, you can do

```
\addcontentsline{toc}{chapter}{\bibname}
```

The counter **tocdepth** controls the section level depth in the table of contents. The levels corresponding to the sections are shown Table 4.2.

The **report** class file sets **tocdepth** to 2, which means that only the parts, chapters, sections and subsections will be entered into the table of contents. You can use **\setcounter** to change the value of **tocdepth**. For example, to also include the subsubsections, paragraphs and subparagraphs, do:

[The format of the Table of Contents, etc.]

```
\setcounter{tocdepth}{5}
```

4.5 Defining a New Page Style

There are two page styles pre-defined by LaTeX[3]: `empty` and `plain`. These page styles can be selected either using:

[Alternative head- and footlines in LaTeX]

Definition

```
\pagestyle{<style>}
```

to change the style "from this point onwards", or

Definition

```
\thispagestyle{<style>}
```

to change the style for a specific page.

Both these commands call the command `\ps@<style>`, and it is this command which redefines the header and footer. So, `\pagestyle{plain}` calls the command `\ps@plain` which in turn calls the commands that redefine the header and footer, and `\pagestyle{empty}` calls the command `\ps@empty` and so on.

So, to define a new page style called, say, `thesis`, you first need to define a command called `\ps@thesis`. Since the command name contains an `@` character, this definition needs to go in a style file or class file.

[3] most of the standard class files, including report, also define the page styles `headings` and `myheadings`

The headers and footers for odd and even numbered pages can be specified by redefining the commands: `\@oddhead`, `\@evenhead`, `\@oddfoot` and `\@evenfoot`.

For example, suppose you want the new page style to have empty headers, and the footers to contain the page number with a dash on either side (e.g. -28-) centred, then you could do:

```
\newcommand{\ps@thesis}{
   \renewcommand{\@oddhead}{}%     header blank
   \renewcommand{\@evenhead}{}%    header blank
   \renewcommand{\@oddfoot}{\hfill-\thepage-\hfill}%
   \renewcommand{\@evenfoot}{\hfill-\thepage-\hfill}%
}
```

Note that if you are using the default `oneside` option to the report class file, only the `\@oddhead` and `\@oddfoot` commands will have any effect. If you want the odd and even numbered pages to be different, you must remember to use the `twoside` option[4]. It is also possible to customise page styles using the fancyhdr package. See *A Guide to LaTeX* [2] for an example.

Unless you are told otherwise, I recommend that you use the `headings` page style for your thesis.

[4] this generally isn't appropriate for a thesis

Chapter 5

Generating a Bibliography

When you are writing a large document such as a PhD thesis, I strongly recommend that you use BibTeX rather than typing up the bibliography in a thebibliography environment. If you use BibTeX:

1. Only the references that you cite are included in the bibliography. (Examiners tend to fault uncited references[1].)

2. References are displayed in a consistent manner.

3. Entries can be sorted in order of citation or alphabetically.

4. The style can easily be changed by simply using a different bibliography style file.

[1] When your examiners read through your thesis, they can check off each citation they encounter against your bibliography. When they reached the end of the thesis, they can then look through the bibliography for unchecked entries. One or two will appear the result of carelessness, whereas a large quantity will look like padding and may lead the examiners to suspect a certain amount of duplicity on your part.

Recall that the example file listed in Chapter 2 had the lines:

```
\bibliographystyle{plain}
\bibliography{thesis}
```

The command

`\bibliographystyle{<style>}` Definition

indicates which BibTeX style file (`.bst`) to use without the extension. The above example uses `plain.bst`. The command

`\bibliography{<database>}` Definition

indicates which database (`.bib`) to use. The above example uses the database `thesis.bib`, which we will need to create. Since the document currently doesn't have any `\cite` commands, and `thesis.bib` does not yet exist, the DVI file does not yet have a bibliography.

There are many bibliography styles, but the basic ones are:

[Choosing a bibliography style]

abbrv Entries sorted alphabetically with abbreviated first names, months and journal names.

`alpha` Entries sorted alphabetically with the citation represented by abbreviated author surname and year instead of a number.

`plain` Entries sorted alphabetically, with the citation represented by a number.

`unsrt` Entries sorted according to citation with the citation represented by a number.

See *A Guide to LaTeX* [2] or *The LaTeX Companion* [1] for information about other bibliography styles, and check with your supervisor to see if there is a particular style you should be using.

Entries in the bibliography database should have the following form:

[Creating a BibTeX bibliography]

```
@<entry type>{<keyword>,
   <field name> = "<text>",
                  ⋮
   <field name> = "<text>"
}
```

where <*entry type*> indicates the type of entry (e.g. book or article). Standard entry types are listed in Table 5.1.

Within an entry, <*keyword*> is a short label that is used to cite this work with the `\cite` command. If you have written bibliographies with the thebibliography environment, it's the same as the argument to `\bibitem`. There then follows a comma-separated list of fields of the form <*field name*> `=` <*value*>. The <*field name*> indicates what kind of field it is, e.g. `title`, `author`. Table 5.2 lists the standard fields. Note that some bibliography styles may define additional non-standard fields, such as `email` or `url`. See *A Guide to LaTeX* [2] or *The LaTeX Companion* [1] for information about other fields not listed in Table 5.2.

[URLS in BibTeX bibliographies]

The required and optional fields for the standard entry types are listed in Table 5.3. If an entry has a field that is neither required nor optional, BibTeX will ignore it. This means that you can have a field called, say, `abstract`, which will be ignored by the standard bibliography styles, but will be included if you use a bibliography style that has an `abstract` field. So you can store additional information in the database which won't appear in the bibliography.

Table 5.1: Standard BiBTeX entry types

`article`	Article from a journal
`book`	Published book
`booklet`	Printed work without a publisher
`conference`	Identical to **`inproceedings`**
`inbook`	Part, chapter, section etc of a book
`incollection`	A chapter of a book with its own author and title
`inproceedings`	An article in a conference proceedings
`manual`	Technical documentation
`mastersthesis`	A master's thesis
`misc`	Non-standard work
`phdthesis`	PhD thesis
`proceedings`	Conference proceedings
`techreport`	Report published by an institution
`unpublished`	Unpublished work with an author and title

Table 5.2: Standard BiBTeX fields

`address`	Publisher/Institution's address
`author`	Author names
`booktitle`	Title of book where only a part of the book is being cited
`chapter`	Chapter or section number
`edition`	The edition of the book
`howpublished`	How a non-standard work was published
`institution`	The institute sponsoring the work
`journal`	The name of the journal
`month`	The month the work was published
`note`	Any additional information
`number`	The number of the journal, technical report etc
`organization`	Organization sponsoring conference or manual
`pages`	Page number or page range
`publisher`	Publisher's name
`school`	Academic institution where thesis was written
`series`	Name of a series
`title`	The title of the work
`type`	The type of technical report
`volume`	The volume number.

Table 5.3: Required and Optional Fields

Entry Type	**Required Fields**	**Optional Fields**
article	author, title, journal, year	volume, month, note, number, pages
book	author or editor, title, publisher, year	address, edition, volume or number, month, note, pages, series
booklet	title	author, address, howpublished, month, note, year
inbook	author or editor, chapter or pages, title, publisher, year	address, edition, volume or number, month, note, series, type
incollection	author, title, booktitle, publisher, year	address, chapter, editor, edition, volume or number, month, note, pages, series, type
inproceedings	author, title, booktitle, year	address, editor, volume or number, month, note, organization, pages, publisher, series, type
manual	title	author, address, edition, month, note, organization, year

Required and Optional Fields Cont.

Entry Type	**Required Fields**	**Optional Fields**
mastersthesis	author, title, school, year	address, month, note, type
misc	—	author, howpublished, month, note, title, year
phdthesis	author, title, school, year	address, month, note, type
proceedings	title, year	editor, organization, address, volume or number, series, month, publisher, note
techreport	author, title, institution, year	type, number, address, month, note
unpublished	author, title, note	month, year

BibTeX uses the European assumption that names are composed of forenames, an optional "von" part which starts with a lower case letter, a surname and an optional "jr" part. In order to enable BibTeX to correctly identify these components, names must be entered in one of the following formats:

[BibTeX sorting and name prefixes]

- *<forenames> <von> <surname>*
- *<von> <surname>, <forenames>*

- *<von> <surname>, <jr>, <forenames>*

Examples (using a style that converts forenames to initials):

[Transcribed initials in BibTeX]

Entry	**Output (“abbrv” style)**
`"Alex Thomas von Neumann"`	A.T. von Neumann
`"John Chris {Smith Jones}"`	J.C. Smith Jones
`"van de Klee, Mary-Jane"`	M.-J. van de Klee
`"Smith, Jr, Fred John"`	F.J. Smith, Jr
`"Maria {\uppercase{d}e La} Cruz"`	M. De La Cruz

Compare the last example with: `"Maria De La Cruz"` which would produce: M. D. L. Cruz, which is incorrect. Let's analyse this last example in more detail: BibTeX always expects the “von” part to start with a lower case letter, but De and La both start with an upper case letter, so BibTeX will assume that these form part of the forenames. However, BibTeX will ignore any LaTeX commands such as `\uppercase` in `\uppercase{d}e` since it assumes that the command is an accent command. So when it parses `\uppercase{d}e` it will skip `\uppercase` and look at the following letter. In this case it is d which is lower case, so from BibTeX's point of view the word `\uppercase{d}e` starts with a lower case letter, so it is therefore the “von” part. You can either do the same with the “La” part, or, as in the above example, you can place it in the same group as `\uppercase{d}e`. If the names in your bibliography don't look correct, then it is likely that

[Accents in bibliographies]

you haven't followed the correct name format in your .bib file. (Note that this also applies to the names in the editor field.)

Multiple authors should be separated by the keyword **and**. Here is an example using the book entry:

[BibTeX doesn't understand lists of names]

```
@book{goossens97,
   author = "Goossens, Michel and Rahtz, Sebastian and
             Mittelbach, Frank",
   title = "The \LaTeX\ graphics companion: illustrating
            documents with \TeX\ and {PostScript}",
   publisher = "Addison Wesley Longman, Inc",
   year = 1997
}
```

In this example, the <*keyword*> is goossens97, so you cite the entry using the command \cite{goossens97}. The standard bibliography styles usually convert titles to lower case, so the name PostScript is enclosed in curly braces to prevent this from happening.

Note that curly braces {} can be used instead of double quotes. The above example can just as easily be written:

```
@book{goossens97,
   author = {Goossens, Michel and Rahtz, Sebastian and
             Mittelbach, Frank},
   title = {The \LaTeX\ graphics companion: illustrating
            documents with \TeX\ and {PostScript}},
   publisher = {Addison Wesley Longman, Inc},
   year = 1997
}
```

Numbers (such as the year 1997) don't need to be delimited with quotes or braces. So you can have

```
pages = 10
```

but a page range would need to be written:

```
pages = "10--45"
```

Bibliography styles always have three-letter abbreviations for months: `jan`, `feb`, `mar`, etc. These should be used instead of typing them in explicitly, as their format depends on the bibliography style. These abbreviations should be entered without quotes. e.g.:

```
@inproceedings{talbot97,
   author    = "Talbot, Nicola and Cawley, Gavin",
   title     = "A fast index assignment algorithm for
                robust vector quantisation of image data",
   booktitle = "Proceedings of the I.E.E.E. International
```

```
                 Conference on Image Processing",
   address    = "Santa Barbara, California, USA",
   month      = oct,
   year       = 1997
}
```

The following is an example of a bibliography database (you can download it if you want):

```
@book{goossens97,
   author = "Goossens, Michel and Rahtz, Sebastian and
             Mittelbach, Frank",
   title = "The \LaTeX\ graphics companion: illustrating
            documents with \TeX\ and {PostScript}",
   publisher = "Addison Wesley Longman, Inc",
   year = 1997
}

@inproceedings{talbot97,
   author    = "Talbot, Nicola L. C. and Cawley, Gavin C.",
   title     = "A fast index assignment algorithm for
                robust vector quantisation of image data",
   booktitle = "Proceedings of the I.E.E.E. International
```

```
                Conference on Image Processing",
   address   = "Santa Barbara, California, USA",
   month     = oct,
   year      = 1997
}

@article{cawley96,
   author    = "Cawley, Gavin C. and Talbot, Nicola L. C.",
   title     = "A fast index assignment algorithm for vector
                quantization over noisy transmission channels",
   journal   = "I.E.E. Electronic Letters",
   number    = 15,
   volume    = 32,
   pages     = "1343--1344",
   month     = jul,
   year      = 1996
}

@incollection{wainwright93,
   author    = "Wainwright, Robert B.",
   title     = "Hazards from {Northern} Native Foods",
   booktitle = "\emph{Clostridium botulinum}: Ecology and
```

```
                Control in Foods",
   chapter   = 12,
   pages     = "305--322",
   editor    = "Hauschild, Andreas H. W. and Dodds,
                Karen L.",
   publisher = "Marcel Dekker, Inc",
   year      = 1993
}
```

Once you have set up your bibliography database, you will need to first LaTeX your document, then call BibTeX and then LaTeX your document twice to get all the cross references up to date. If you are using TeXnicCenter, when you create a new project, click on the 'Uses BiBTeX' option, and it will automatically call BibTeX when you click on the build icon. If you are using a command prompt, then if your file is called, say, **thesis.tex**, you will need to type the following commands:

["Normal" use of BibTeX from LaTeX]

```
latex thesis
bibtex thesis
latex thesis
latex thesis
```

Note that you are specifying the *auxiliary file* when calling BibTeX, *without* the extension. You can have a bibliography database that has a different name from your LaTeX file, but you use the name of the auxiliary file[2] when calling BibTeX. For example, if your thesis is saved in the file **thesis.tex** and your bibliography database is saved in the file **ref.bib**, then you still need to do:

```
latex thesis
bibtex thesis
latex thesis
latex thesis
```

In fact, you can use multiple bibliography databases (which isn't the same as having multiple bibliographies in your document.) Suppose your references are defined in the files **ref1.bib** and **ref2.bib**, then you need to specify both databases in **thesis.tex**:

[Multiple bibliographies?]

```
\bibliography{ref1}
\bibliography{ref2}
```

Alternatively you can specify the databases in a list:

```
\bibliography{ref1,ref2}
```

[2] This will typically have the same base name as your main document file, but may be different if you are using a bibliography managing package such as bibunits.

If you have references which you find yourself frequently using, such as your own publications, you may prefer to keep a `.bib` file containing these references in a central location, such as in your local texmf tree. If you are using a UNIX-like operating system, this will typically be in `~/texmf/bibtex/bib/`. If you are using Windows, this may be in the folder `c:\localtexmf\bibtex\bib\` but check your TEX installation documentation. If you do this, remember to refresh the TEX database.

[Installing things on a (La)TeX system]

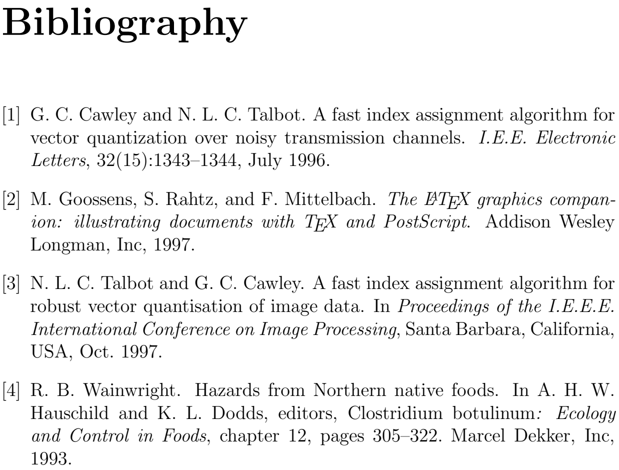
Illustrations of some of the common bibliography styles are shown in Figures 5.1, 5.2, 5.3, 5.4, 5.5, 5.6 and 5.7. Note that the **apalike** bibliography style requires the **apalike** package.

5.1 Back-References

The **backref** package supplied with the **hyperref** bundle will place a comma-separated list of section or page numbers on which the work was cited at the end of each item in the bibliography. Each bibliography item in the **thebibliography** environment must be separated by a blank line, but as BIBTEX does this automatically, you only need to worry about it if you are creating your **thebibliography** environment without the aid of BIBTEX.

[References from the bibliography to the citation]

The list of numbers will by default refer to the page numbers in which the corresponding `\cite` commands are located, but this can be changed to the section numbers by passing the options `ref` to the **backref** package. If



Bibliography

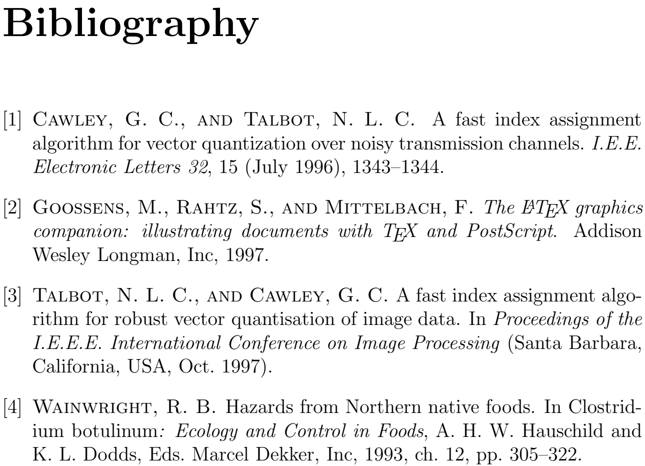
[1] G. C. Cawley and N. L. C. Talbot. A fast index assignment algorithm for vector quantization over noisy transmission channels. *I.E.E. Electronic Letters*, 32(15):1343–1344, July 1996.

[2] M. Goossens, S. Rahtz, and F. Mittelbach. *The LaTeX graphics companion: illustrating documents with TeX and PostScript*. Addison Wesley Longman, Inc, 1997.

[3] N. L. C. Talbot and G. C. Cawley. A fast index assignment algorithm for robust vector quantisation of image data. In *Proceedings of the I.E.E.E. International Conference on Image Processing*, Santa Barbara, California, USA, Oct. 1997.

[4] R. B. Wainwright. Hazards from Northern native foods. In A. H. W. Hauschild and K. L. Dodds, editors, Clostridium botulinum*: Ecology and Control in Foods*, chapter 12, pages 305–322. Marcel Dekker, Inc, 1993.

Figure 5.1: `abbrv` bibliography style



Bibliography

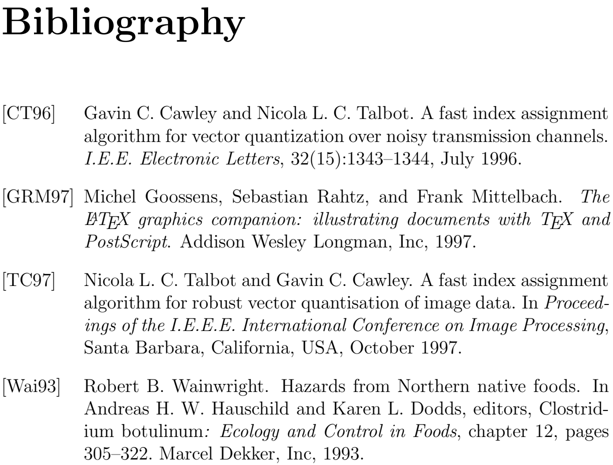
[1] Cawley, G. C., and Talbot, N. L. C. A fast index assignment algorithm for vector quantization over noisy transmission channels. *I.E.E. Electronic Letters 32*, 15 (July 1996), 1343–1344.

[2] Goossens, M., Rahtz, S., and Mittelbach, F. *The LaTeX graphics companion: illustrating documents with TeX and PostScript*. Addison Wesley Longman, Inc, 1997.

[3] Talbot, N. L. C., and Cawley, G. C. A fast index assignment algorithm for robust vector quantisation of image data. In *Proceedings of the I.E.E.E. International Conference on Image Processing* (Santa Barbara, California, USA, Oct. 1997).

[4] Wainwright, R. B. Hazards from Northern native foods. In Clostridium botulinum*: Ecology and Control in Foods*, A. H. W. Hauschild and K. L. Dodds, Eds. Marcel Dekker, Inc, 1993, ch. 12, pp. 305–322.

Figure 5.2: `acm` bibliography style



Bibliography

[CT96] Gavin C. Cawley and Nicola L. C. Talbot. A fast index assignment algorithm for vector quantization over noisy transmission channels. *I.E.E. Electronic Letters*, 32(15):1343–1344, July 1996.

[GRM97] Michel Goossens, Sebastian Rahtz, and Frank Mittelbach. *The LaTeX graphics companion: illustrating documents with TeX and PostScript*. Addison Wesley Longman, Inc, 1997.

[TC97] Nicola L. C. Talbot and Gavin C. Cawley. A fast index assignment algorithm for robust vector quantisation of image data. In *Proceedings of the I.E.E.E. International Conference on Image Processing*, Santa Barbara, California, USA, October 1997.

[Wai93] Robert B. Wainwright. Hazards from Northern native foods. In Andreas H. W. Hauschild and Karen L. Dodds, editors, Clostridium botulinum*: Ecology and Control in Foods*, chapter 12, pages 305–322. Marcel Dekker, Inc, 1993.

Figure 5.3: `alpha` bibliography style

Bibliography

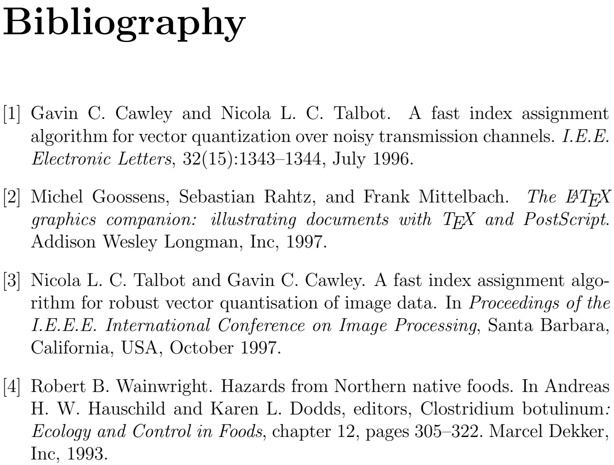
[1] M. Goossens, S. Rahtz, and F. Mittelbach, *The LaTeX graphics companion: illustrating documents with TeX and PostScript*. Addison Wesley Longman, Inc, 1997.

[2] N. L. C. Talbot and G. C. Cawley, "A fast index assignment algorithm for robust vector quantisation of image data," in *Proceedings of the I.E.E.E. International Conference on Image Processing*, (Santa Barbara, California, USA), Oct. 1997.

[3] G. C. Cawley and N. L. C. Talbot, "A fast index assignment algorithm for vector quantization over noisy transmission channels," *I.E.E. Electronic Letters*, vol. 32, pp. 1343–1344, July 1996.

[4] R. B. Wainwright, "Hazards from Northern native foods," in Clostridium botulinum*: Ecology and Control in Foods* (A. H. W. Hauschild and K. L. Dodds, eds.), ch. 12, pp. 305–322, Marcel Dekker, Inc, 1993.

Figure 5.4: `ieeetr` bibliography style



Bibliography

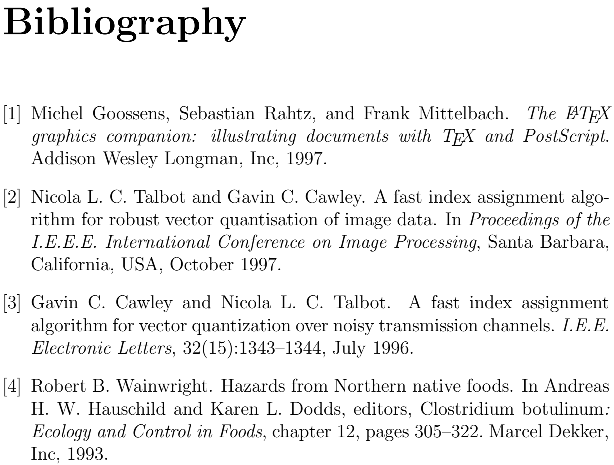
[1] Gavin C. Cawley and Nicola L. C. Talbot. A fast index assignment algorithm for vector quantization over noisy transmission channels. *I.E.E. Electronic Letters*, 32(15):1343–1344, July 1996.

[2] Michel Goossens, Sebastian Rahtz, and Frank Mittelbach. *The LaTeX graphics companion: illustrating documents with TeX and PostScript*. Addison Wesley Longman, Inc, 1997.

[3] Nicola L. C. Talbot and Gavin C. Cawley. A fast index assignment algorithm for robust vector quantisation of image data. In *Proceedings of the I.E.E.E. International Conference on Image Processing*, Santa Barbara, California, USA, October 1997.

[4] Robert B. Wainwright. Hazards from Northern native foods. In Andreas H. W. Hauschild and Karen L. Dodds, editors, Clostridium botulinum*: Ecology and Control in Foods*, chapter 12, pages 305–322. Marcel Dekker, Inc, 1993.

Figure 5.5: `plain` bibliography style



Bibliography

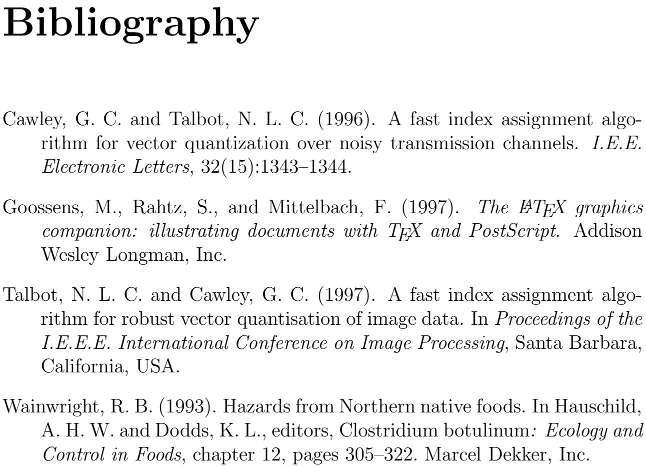
[1] Michel Goossens, Sebastian Rahtz, and Frank Mittelbach. *The LaTeX graphics companion: illustrating documents with TeX and PostScript*. Addison Wesley Longman, Inc, 1997.

[2] Nicola L. C. Talbot and Gavin C. Cawley. A fast index assignment algorithm for robust vector quantisation of image data. In *Proceedings of the I.E.E.E. International Conference on Image Processing*, Santa Barbara, California, USA, October 1997.

[3] Gavin C. Cawley and Nicola L. C. Talbot. A fast index assignment algorithm for vector quantization over noisy transmission channels. *I.E.E. Electronic Letters*, 32(15):1343–1344, July 1996.

[4] Robert B. Wainwright. Hazards from Northern native foods. In Andreas H. W. Hauschild and Karen L. Dodds, editors, Clostridium botulinum*: Ecology and Control in Foods*, chapter 12, pages 305–322. Marcel Dekker, Inc, 1993.

Figure 5.6: `unsrt` bibliography style



Bibliography

Cawley, G. C. and Talbot, N. L. C. (1996). A fast index assignment algorithm for vector quantization over noisy transmission channels. *I.E.E. Electronic Letters*, 32(15):1343–1344.

Goossens, M., Rahtz, S., and Mittelbach, F. (1997). *The LaTeX graphics companion: illustrating documents with TeX and PostScript*. Addison Wesley Longman, Inc.

Talbot, N. L. C. and Cawley, G. C. (1997). A fast index assignment algorithm for robust vector quantisation of image data. In *Proceedings of the I.E.E.E. International Conference on Image Processing*, Santa Barbara, California, USA.

Wainwright, R. B. (1993). Hazards from Northern native foods. In Hauschild, A. H. W. and Dodds, K. L., editors, Clostridium botulinum*: Ecology and Control in Foods*, chapter 12, pages 305–322. Marcel Dekker, Inc.

Figure 5.7: `apalike` bibliography style (requires `apalike` package)

you are using the hyperref package, then the backref package will be loaded if you use the hyperref package options `backref` or `backref=section`.

The backref package uses the command `\backref` to control the format of the list of back-references. Without the hyperref package, the list of back-references has an introductory text supplied by `\backrefpagesname` or `\backrefsectionsname`. See the backref package documentation for further detail.

This document uses the backref package. The effect can be seen in the bibliography on page 124.

5.2 Troubleshooting

- BibTeX writes the thebibliography environment to a `.bbl` file, which is then input into the document by `\bibliography`. If you have made a LaTeX error in the `.bib` file, this error will be copied to the `.bbl` file. If you have corrected the error in the `.bib` file, but you are still getting an error when you LaTeX your document, try deleting the `.bbl` file.

- Remember to use double quotes or braces to delimit the field names in your `.bib` file.

- Remember to put a comma at the end of each field (except the last).

- It is better to only use alphanumerical characters in the keywords. Some punctuation characters such as `-` should be fine, but spaces are not recommended, and commas should definitely be avoided.

- The LaTeX comment symbol (`%`) is not a comment character in a `.bib` file.

- If you have entered a field in the `.bib` file, but it doesn't appear in the bibliography, check to make sure that the field is required or optional for that type of entry, and check the spelling.

- Check the log file (`.blg`) generated by BibTeX for messages.

Chapter 6

Formatting

6.1 Double Spacing

Double spacing is usually frowned upon in the world of modern typesetting, however it is usually a requirement when you are writing a PhD thesis as it gives the examiners extra space to write comments.

Double spacing can either be achieved using the `setspace` package, or by redefining the value of `\baselinestretch`. The value depends on the font size (see Table 6.1). To switch back to single spacing set `\baselinestretch` back to 1.

[Double-spaced documents in LaTeX]

Table 6.1: Double spacing values for `\baselinestretch`

Font Size	10pt	11pt	12pt
`\baselinestretch`	1.67	1.62	1.66

For example, if you are using 12pt font, you will need the following line:

```
\renewcommand{\baselinestretch}{1.66}
```

It is however better to use the setspace package which provides the declarations `\singlespacing`, `\onehalfspacing` and `\doublespacing`.

6.2 Changing the Title Page

The title page style generated by `\maketitle` may not be appropriate for the school/university's specifications. If this is the case, you can use the titlepage environment instead. For example:

↑Input

```
\begin{titlepage}
\begin{center}
\vspace*{1in}
{\LARGE A Sample PhD Thesis}
\par
\vspace{1.5in}
{\large A. N. Other}
```

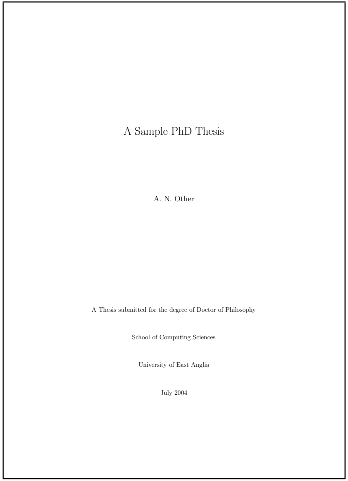
```
\par
\vfill
A Thesis submitted for the degree of Doctor of Philosophy
\par
\vspace{0.5in}
School of Computing Sciences
\par
\vspace{0.5in}
University of East Anglia
\par
\vspace{0.5in}
July 2004
\end{center}
\end{titlepage}
```

↓Input

The resulting output is shown in Figure 6.1.

Check with your supervisor to see if there is a particular layout required for the title page. Note that some classes, such as memoir and scrreprt provide commands to modify the title layout. The titling package also provides such facilities.

[The style of document titles]



A Sample PhD Thesis

A. N. Other

A Thesis submitted for the degree of Doctor of Philosophy

School of Computing Sciences

University of East Anglia

July 2004

Figure 6.1: Example Title Page

6.3 Verbatim Text

There may be times when you want to include text exactly as you have typed it into your source code. For example, you may want to include a short segment of computer code. This can be done using the verbatim environment. For example:

[Code listings in LaTeX]

↑Input

```
\begin{verbatim}

#include <stdio.h>

int main()
{
   printf{"Hello World\n"};

   return 1;
}

\end{verbatim}
```

↓Input

would produce the following output:

↑Output

```
#include <stdio.h>

int main()
{
   printf{"Hello World\n"};

   return 1;
}
```

↓Output

[Including a file verbatim in LaTeX]

Definition

The contents of a file can be included verbatim using the command:

\verbatiminput{<*filename*>}

defined in the **verbatim** package. For example:

```
\verbatiminput{helloW.c}
```

where `helloW.c` is the filename (remember to use a forward slash `/` as a directory divider, even if you are using Windows).

Note: it is not usually appropriate to have reams of listings in your thesis. It can annoy an examiner if you have included every single piece of code you have written during your PhD, as it comes across as padding to make it look as though your thesis is a lot larger than it really is. (Examiners are not easily fooled, and it's best not to irritate them as it is likely to make them less sympathetic towards you.) If you want to include listings in your thesis, check with your supervisor first to find out whether or not it is appropriate.

6.4 Tabbing

The tabbing environment lets you create tab stops so that you can tab to a particular distance from the left margin. Within the tabbing environment, you can use the command `\=` to set a tab stop, `\>` to jump to the next tab stop, `\<` to go back a tab stop, `\+` to shift the left border by one tab stop to the right, `\-` to shift the left border by one tab stop to the left. In addition, `\\` will start a new line and `\kill` will set any tabs stops defined in that line, but will not typeset the line itself.

[Accents misbehave in tabbing]

Examples:

1. This first example sets up three tab stops:

↑Input

```
\begin{tabbing}
Zero \=One \=Two \=Three\\
\>First tab stop\\
\>A\>\>B\\
\>\>Second tab stop
\end{tabbing}
```

↓Input

This produces the following output:

↑Output

```
Zero One Two Three
     First tab stop
     A       B
         Second tab stop
```

↓Output

2. This second example sets up four tab stops, but ignores the first line:

↑Input

```
\begin{tabbing}
AAA \=BBBB \=XX \=YYYYYY \=Z \kill
\>\>\>Third tab stop\\
\>a \>\>b \>c
\end{tabbing}
```

↓Input

This produces the following output:

↑Output

```
               Third tab stop
     a         b         c
```

↓Output

6.5 Theorems and Algorithms

A PhD thesis can often contain theorems, lemmas, definitions etc. These structures can be created using the command

\newtheorem{<*type*>**}{**<*title*>**}[**<*outer counter*>**]** Definition

where <*type*> is the type of your structure (e.g. theorem), <*title*> is the word that is printed in bold at the start of the structure (e.g. Theorem) and if the optional argument <*outer counter*> is present, then the structure's counter should depend on <*outer counter*> (as in the optional argument to `\newcounter`.)

You should typically define your new theorem either in the preamble or in a package or class file. Once you have defined your new theorem, a new environment is created whose name is given by <*type*>. This environment has an optional argument that you can use to specify a caption for the structure.

Examples:

1. Define a theorem structure. The counter belonging to this structure is not dependent on any other counter:

⊤Input

```
\newtheorem{theorem}{Theorem}

\begin{theorem}
If $\lambda$ is an eigenvalue of $\mathbf{B}$ with
eigenvector $\vec{\xi}$, then $\lambda^n$ is an
eigenvalue of $\mathbf{B}^n$ with eigenvector $\vec{\xi}$.
\end{theorem}
```

↓Input

This gives the following output:

↑Output

Theorem 1 *If λ is an eigenvalue of $\mathbf{B}$ with eigenvector $\vec{\xi}$, then λ^n is an eigenvalue of $\mathbf{B}^n$ with eigenvector $\vec{\xi}$.*

↓Output

(See LaTeX for Complete Novices [5] if you want to know how to redefine the `\vec` command so that the vector appears in bold.)

2. In this example, the theorem is defined to be dependent on the chapter counter. The theorem counter will be reset each time a new chapter is started:

↑Input

```
\newtheorem{theorem}{Theorem}[chapter]

\begin{theorem}
If $\lambda$ is an eigenvalue of $\mathbf{B}$ with
eigenvector $\vec{\xi}$, then $\lambda^n$ is an
eigenvalue of $\mathbf{B}^n$ with eigenvector $\vec{\xi}$.
\end{theorem}
```

↓Input

This gives the following output:

↑Output

Theorem 6.1 *If* λ *is an eigenvalue of* $\mathbf{B}$ *with eigenvector* $\vec{\xi}$*, then* λ^n *is an eigenvalue of* $\mathbf{B}^n$ *with eigenvector* $\vec{\xi}$*.*

↓Output

3. In this example, the theorem is given a caption:

↑Input

```
\newtheorem{theorem}{Theorem}[chapter]

\begin{theorem}[Eigenvector Powers]
If $\lambda$ is an eigenvalue of $\mathbf{B}$ with
eigenvector $\vec{\xi}$, then $\lambda^n$ is an
eigenvalue of $\mathbf{B}^n$ with eigenvector $\vec{\xi}$.
\end{theorem}
```

↓Input

This gives the following output:

↑Output

Theorem 6.1 (Eigenvector Powers) *If λ is an eigenvalue of $\mathbf{B}$ with eigenvector $\vec{\xi}$, then λ^n is an eigenvalue of $\mathbf{B}^n$ with eigenvector $\vec{\xi}$.*

↓Output

4. In this example an algorithm structure is created. The commands `\hfill\par` are used to prevent the tabbing environment from running into the algorithm title.

↑Input

```
\newtheorem{algorithm}{Algorithm}

\begin{algorithm}[Gauss-Seidel Algorithm]
\hfill\par
\begin{tabbing}
1. \=For $k=1$ to maximum number of iterations\\
\>2. For \=$i=1$ to $n$\\
\>\>Set
\begin{math}
x_i^{(k)} =
\frac{b_i-\sum_{j=1}^{i-1}a_{ij}x_j^{(k)}
         -\sum_{j=i+1}^{n}a_{ij}x_j^{(k-1)}}%
     {a_{ii}}
\end{math}
\\
\>3. If $\|\vec{x}^{(k)}-\vec{x}^{(k-1)}\| < \epsilon$,
where $\epsilon$ is a specified stopping criteria, stop.
\end{tabbing}
\end{algorithm}
```

↓Input

This will give the following output:

↑Output

Algorithm 1 (Gauss-Seidel Algorithm)

```
1. For $k = 1$ to maximum number of iterations
    2. For $i = 1$ to $n$
            Set $x_i^{(k)} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k-1)}}{a_{ii}}$
    3. If $\|\vec{x}^{(k)} - \vec{x}^{(k-1)}\| < \epsilon$, where $\epsilon$ is a specified stopping criteria, stop.
```

↓Output

The last example doesn't look right, as algorithms tend to be displayed in an upright font not an italic font. The package amsthm

[Typesetting pseudocode in LATEX]

extends the functionality of `\newtheorem` and provides three theorem styles:

plain Title and number in bold, body in italic (default).

definition Title and number in bold, body in normal font.

remark Title and number in italic, body in normal font.

The above example can now be changed to:

↑Input

```
\theoremstyle{definition}
\newtheorem{algorithm}{Algorithm}

\begin{algorithm}[Gauss-Seidel Algorithm]
\hfill\par
\begin{tabbing}
1. \=For $k=1$ to maximum number of iterations\\
\>2. For \=$i=1$ to $n$\\
\>\>Set
\begin{math}
x_i^{(k)} =
\frac{b_i-\sum_{j=1}^{i-1}a_{ij}x_j^{(k)}
         -\sum_{j=i+1}^{n}a_{ij}x_j^{(k-1)}}%
     {a_{ii}}
\end{math}
\\
\>3. If $\|\vec{x}^{(k)}-\vec{x}^{(k-1)}\| < \epsilon$,
where $\epsilon$ is a specified stopping criteria, stop.
\end{tabbing}
\end{algorithm}
```

↓Input

This will give the following output:

↑Output

Algorithm 1 (Gauss-Seidel Algorithm)

```
1. For $k = 1$ to maximum number of iterations
  2. For $i = 1$ to $n$
        Set $x_i^{(k)} = \frac{b_i-\sum_{j=1}^{i-1}a_{ij}x_j^{(k)}-\sum_{j=i+1}^{n}a_{ij}x_j^{(k-1)}}{a_{ii}}$
  3. If $\|\vec{x}^{(k)}-\vec{x}^{(k-1)}\| < \epsilon$, where $\epsilon$ is a specified stopping criteria, stop.
```

↓Output

(You can download an example of this.)

Alternatively, if you want your algorithms to behave like figures and tables, you can use the `\newfloat` command defined in the float package:

`\newfloat{`*<type>*`}{`*<placement>*`}{`*<ext>*`}[`*<outer counter>*`]` Definition

where *<type>* is the name of your new float, *<placement>* is the default placement specifier (**t**, **b**, **p** and **h**), *<ext>* is the extension for the list of *<type>* and as before, the presence of *<outer counter>* indicates that the counter associated with this new float should depend on *<outer counter>*.

You can also specify the style of your new floats by issuing the command:

`\floatstyle`{*<style>*} Definition

before defining your new floats, where *<style>* can be one of:

plain Same as the standard figure and table floats, except that the caption is always placed at the end of the float.

boxed The body of the float is placed in a box, and the caption is printed below the box.

ruled The caption is printed at the top with a rule above and below it, and there is a rule at the end of the float.

The name associated with a float is defined using the command:

\floatname{<*type*>}{<*name*>} Definition

where <*type*> is the name of the float environment (as defined in \newfloat) and <*name*> is the name associated with that float.

The list of <*type*> can be produced using the command:

\listof{<*type*>}{<*title*>} Definition

So, instead of defining our algorithm environment using \newtheorem, we could instead define it using \newfloat as follows:

⊤Input

```
\floatstyle{ruled}
```

```
\newfloat{algorithm}{htbp}{loa}
\floatname{algorithm}{Algorithm}

\begin{algorithm}
\caption{Gauss-Seidel Algorithm}
\label{alg:GS}

\begin{tabbing}
1. \=For $k=1$ to maximum number of iterations\\
\>2. For \=$i=1$ to $n$\\
\>\>Set
\begin{math}
x_i^{(k)} =
\frac{b_i-\sum_{j=1}^{i-1}a_{ij}x_j^{(k)}
         -\sum_{j=i+1}^{n}a_{ij}x_j^{(k-1)}}{a_{ii}}
\end{math}
\\
\>3. If $\|\vec{x}^{(k)}-\vec{x}^{(k-1)}\| < \epsilon$,
where $\epsilon$ is a specified stopping criteria, stop.
\end{tabbing}
\end{algorithm}
```

↓Input

This would produce the following output:

Algorithm 1 Gauss-Seidel Algorithm

```
1. For $k = 1$ to maximum number of iterations
    2. For $i = 1$ to $n$
            Set $x_i^{(k)} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k-1)}}{a_{ii}}$
    3. If $\|\vec{x}^{(k)} - \vec{x}^{(k-1)}\| < \epsilon$, where $\epsilon$ is a specified stopping criteria, stop.
```

The following line can then go after the list of figures and list of tables:

```
\listof{algorithm}{List of Algorithms}
```

(You can download an example of this.)

Chapter 7

Generating an Index or a Glossary

It is fairly straight-forward to create an index or glossary using LaTeX, and using the `makeindex` application makes it even easier. It is a good idea to include a glossary in a thesis, particularly if there are any symbols or abbreviations in your document, as there are a number of different ways some symbols can be interpreted. For example, x' could mean the derivative of x or it could mean an updated value of x (or it could even mean the transpose of x, but in this case x should be formatted as a vector.) It is not wise to assume that your reader uses the same notation as you. It isn't quite so common to include an index in a PhD thesis, however, the LaTeX user's guide [3] states that any nonfiction work of more than twenty pages ought to have an index.

7.1 Generating an Index

If you want to generate an index, you will need the command

[Generating an index in (La)TeX]

```
\makeindex
```

Definition

in the preamble. The command

```
\index{<entry>}
```

Definition

is used to index *<entry>* at that point in the document. For example, the following code:

↑Input

```
Eigenvectors\index{eigenvector} are defined \ldots
```

↓Input

will produce the output

↑Output

> Eigenvectors are defined . . .

↓Output

and place the entry ‘eigenvector’ in the `.idx` file with the associated page number.

Note that if you don’t use `\makeindex` in the preamble, no `.idx` file will be created and `\index` will ignore its argument.

The package makeidx provides the command

```
\printindex
```

Definition

which should be placed at the point in the document where you want your index to appear.

Provided you have used `\makeindex` and `\index`, once you have LaTeXed your document, there will be a file with the extension `.idx` containing all the indexing information as a series of `\indexentry` commands. This command is not defined by LaTeX, so you should not input the `.idx` file into your document. The `.idx` file needs to be processed by an external application such as `makeindex` to create a file which contains all the LaTeX commands necessary to typeset the index. This new file has the extension `.ind`, and it is this file which is input by `\printindex` on the next LaTeX run.

If you are using `TeXnicCenter` you will need to select "uses makeindex" when you create a new project, if you are using a command prompt, you will need to do:

```
latex filename.tex
makeindex filename.idx
latex filename.tex
```

(where **filename** is the base name of your source file, e.g. **thesis**) If you are also using BibTeX, you will need to do:

```
latex filename.tex
bibtex filename
makeindex filename.idx
latex filename.tex
latex filename.tex
```

It's a good idea to have sub-entries within an index, to assist the reader. For example, you may want to index the term "matrix", but your document may mention many different types of matrices, such as diagonal, block or singular. In which case it would be better to index the term matrix for general occurrences, and have sub-entries indexing specific types of matrices, so that the matrix entry in the index would look something like:

matrix, 4, 10, 22–24

diagonal, 12

block, 20, 24

singular, 33

A sub-entry can be generated using the `!` character. So the above can be generated using the following commands:

Preamble:	`\makeindex`
Page 4:	`\index{matrix}`
Page 10:	`\index{matrix}`
Page 12:	`\index{matrix!diagonal}`
Page 20:	`\index{matrix!block}`
Page 22:	`\index{matrix}`
Page 23:	`\index{matrix}`
Page 24:	`\index{matrix}`
Page 24:	`\index{matrix!block}`
Page 33:	`\index{matrix!singular}`
End of Doc:	`\printindex`

Note that the same entries on pages 22, 23 and 24 have been turned into a page range 22–24. For larger page ranges, you can specify the start of the page range by appending `|(` to the end of the index entry and the end of the page range by appending `|)` to the end of the index entry. For example:

Preamble:	`\makeindex`
Page 4:	`\index{matrix}`
Page 10:	`\index{matrix}`
Page 12:	`\index{matrix!diagonal}`
Page 20:	`\index{matrix!block}`
Page 22:	`\index{matrix|(}`
Page 24:	`\index{matrix!block}`
Page 30:	`\index{matrix|)}`
Page 33:	`\index{matrix!singular}`
End of Doc:	`\printindex`

would produce the following output in the index:

matrix, 4, 10, 22–30

- diagonal, 12
- block, 20, 24
- singular, 33

An index entry can refer to another entry using `|see{`<*reference*>`}`[1]. For example,

[1] This in fact tells `makeindex` to use the command `\see{`<*reference*>`}` which uses `\seename` to typeset the word "see". The `babel` package will redefine this so that it uses the relevant translation, or you can redefine `\seename` using `\renewcommand`

```
\index{singular matrix|see{matrix, singular}}
```

would produce the entry

singular matrix, *see* matrix, singular

The format of the page number can be changed using `|`<*style*> where <*style*> is the name of a formatting command *without* the backslash. Suppose in the above example, the term "matrix" is defined on page 10, then you may want the page number to appear in bold to indicate that this is a primary reference. The command `\textbf` produces bold text, so you would need to append `|textbf` to the index entry[2]. For example, the code:

[2] The argument to the formatting command will be the page number. In fact, `\see` takes two arguments, the first is the redirection text which you must supply within the argument of `\index` (as shown in the example) and the second argument is the page number which `\see` ignores

Preamble:	`\makeindex`
Page 4:	`\index{matrix}`
Page 10:	`\index{matrix|textbf}`
Page 12:	`\index{matrix!diagonal}`
Page 20:	`\index{matrix!block}`
Page 22:	`\index{matrix|(}`
Page 24:	`\index{matrix!block}`
Page 30:	`\index{matrix|)}`
Page 33:	`\index{matrix!singular}`
End of Doc:	`\printindex`

would produce the following output in the index:

matrix, 4, **10**, 22–30

    diagonal, 12

    block, 20, 24

    singular, 33

Note that if you want to apply more than one formatting command, say you want the number to be bold and italic, then you should define a new command with one argument which will set the argument in that font, for example:

```
\newcommand{\textbfit}[1]{\textit{\bfseries #1}}
```

and then use this command name (without the backslash) in the `\index` command. (It is possible to do, say `\index{matrix|itshape\textbf}`, but since `\itshape` is a declaration, it will set the rest of your index in that shape, until counteracted by another font changing command. You definitely *must not* do something along the lines of `\index{matrix|textit\textbf}` since this will be equivalent to `\textit{\textbf}{`<*page number*>`}` which will of course produce an error from LaTeX since it is syntactically incorrect.)

The application `makeindex` sorts the index according to the entries specified, so the word "matrix" would come before the word "modulus", but `$\mu$` will be sorted on the characters `$`, `\`, `m`, `u` and then `$`, so μ would come before "matrix". This may not be appropriate, so it is possible to specify how to sort the entry and how to format the entry separately using the `@` character:

```
\index{mu@$\mu$}
```

In this case the sorting is performed on the string `mu`, so it will appear after the word "modulus", but it will appear in the index as μ. For more information about generating an index see the LaTeX user's guide [3], *The LaTeX Companion* [1] or *A Guide to LaTeX* [2].

7.1.1 Troubleshooting

- My index hasn't appeared.

  1. Make sure you have the command `\printindex` at the place where you want the index to appear (this command is defined in the makeidx package).
  2. Make sure you have the command `\makeindex` in the preamble.
  3. Make sure you LaTeX the document, then run `makeindex`, then LaTeX the document again.
  4. Check `makeindex`'s log file (which has the extension `.ilg` by default) for error messages.

- I want to index the character `"`, `@`, `!` or `|` but it's not working.

  If you want any of these symbols in your index, you will need to prepend the character with the double quote symbol `"`. For example:

  ```
  \index{"@}
  ```

  will index the `@` character.

- I have multiple entries of the same item. For example:

identity matrix, 10, 22-30

identity matrix, 4

Check to make sure the sort argument to each of the corresponding `\index` commands is the same, pay particular attention to spaces as `makeindex` will treat the following entries differently:

```
\index{identity matrix}
\index{identity  matrix}
```

LaTeX however, treats multiple spaces the same as a single space, so the text will appear the same in the index.

- LaTeX says that the command `\printindex` is undefined.

  You have forgotten to load the makeidx package.

7.2 Generating a Glossary

There are a number of packages available to assist creating a glossary, these include makeglos (analogous to makeidx), nomencl, glossaries[3], glosstex and gloss. The first three use LaTeX in conjunction with `makeindex`,

[3] The glossaries package has replaced the now obsolete glossary package

glosstex uses LaTeX in conjunction with `makeindex` and `glosstex` whilst gloss uses LaTeX in conjunction with BibTeX. This document only describes glossaries. If you are interested in using the others, you should read their accompanying documentation.

The glossaries package has the advantage over makeglos and nomencl in that you don't have to worry about escaping `makeindex`'s special characters as they are dealt with internally. The glossary information is set using keys and you can override the default plural form for plurals that aren't formed by appending the letter "s" to the singular form. In addition, you can specify alternative text for the first time the term is used in the document, and you can also define an associated symbol. This guide gives a brief overview of the glossaries package. For further details you will need to read the package documentation.

7.2.1 Defining Glossary Entries

Firstly, in order to make the glossary (or glossaries, if you have more than one) appear, you must use the command

```
\makeglossaries
```

Definition

in the preamble. This is analogous to the **`\makeindex`** command described earlier.

Next you need to define the terms you want to appear in the glossary. Again, this must be done in the preamble. This is done using the command

```
\newglossaryentry{<label>}{<key-val list>}
```

Definition

The first argument <*label*> is a unique label to allow you to refer to this entry in your document text. The entry will only appear in the glossary if you have referred to it in the document using one of the commands listed later. The second argument is a comma separated <*key*>=<*value*> list. Available keys are as follows:

`name` The name of the entry (as it will appear in the glossary)

`description` A brief description of this term (to appear in the glossary)

`text` How this entry will appear in the document text where the singular form is required. If this key is omitted, the value of the **`name`** key is used.

`first` How this entry will appear in the document text the first time it is used, where the first use requires the singular form. If this field is omitted, the value of the **`text`** key is used.

`plural` How this entry will appear in the document text where the plural form is required. If this key is omitted, the value is obtained by appending the letter “s” to the value of the `text` key.

`firstplural` How this entry will appear in the document text the first time it is used, where the first use requires the plural form. If this field is omitted, the value is obtained by appending the letter “s” to the value of the `first` key.

`symbol` This key is provided to allow the user to specify an associated symbol, but most glossary styles ignore this value. If omitted, the value is set to `\relax`.

`sort` This value indicates how `makeindex` should sort this entry. If omitted, the value is given by the `name` key.

`type` This is the glossary type to which this entry belongs. If omitted, the main glossary is assumed.

For example, the following defines the term “set” and assigns a brief description. The term is given the label `set`. This is the minimum amount of information you must give:

```
\newglossaryentry{set}% the label
```

```
{name=set,             % the term
 description={a collection of objects} % a brief description
}
```

The following entry also has an associated symbol:

```
\newglossaryentry{U}   % the label
{name={universal set}, % the term
 description={the set of all things} % a brief description
 symbol={\ensuremath{\mathcal{U}}}  % the associated symbol
}
```

The following example uses the vertical bar symbol `|` which is one of `makeindex`'s special characters, but the glossaries package deals with it behind the scenes, so I don't need to do anything special:

```
\newglossaryentry{card}% the label
{name=cardinality, % the term
 description={the number of objects within a set}, % brief description
 symbol={\ensuremath{|\mathcal{S}|}} % the associated symbol
}
```

The plural of the word "matrix" is "matrices" not "matrixs", so the term needs the plural form set explicitly:

```
\newglossaryentry{matrix}% the label
{name=matrix, % the term
 description={a rectangular table of elements}, % brief description
 plural=matrices % the plural
}
```

The **glossaries** package also provides the shortcut command

```
\newacronym[<key-val list>]{<label>}{<abbrv>}{<long>}
```

Definition

This is equivalent to:

```
   \newglossaryentry{<label>}{type=\acronymtype,
   name={<abbrv>},
   description={<long>},
   text={<abbrv>},
   first={<long> (<abbrv>)},
   plural={<abbrv>s},
   firstplural={<long>s (<abbrv>s)},
   <key-val list>
   }
```

Note that the glossary type is given as `\acronymtype`. This will be the main glossary by default. If you specify the package option `acronym` then a new glossary type will be created called `acronym` and `\acronymtype` will be set to this value.

7.2.2 Displaying Terms in the Document

Any glossary term that has been defined in the preamble using `\newglossaryentry` or `\newacronym`, as described above, can be used in the document text using one of the commands described in this section. Note the term will only appear in the glossary if it has been used, in the same way that when you are using BibTeX, only those references you cite in the text will appear in the bibliography.

`\glslink[`<*options*>`]{`<*label*>`}{`<*text*>`}` Definition

This command adds the term given by <*label*> to the relevant glossary but instead of displaying the term in the document, it displays <*text*> at that point. If you have hyperlinks enabled (for example, you are using the hyperref or html package) the <*text*> will be a hyperlink to the relevant entry in the glossary. The optional argument <*options*> is a comma separated <*key*>=<*value*> list which may take any of the following keys:

`format` This specifies how to format the associated number for this entry. It is equivalent to the `\index` `|` special character, described in Section 7.1, where, as with `\index`, you should not include the initial backslash. As before, if you want to specify, say bold italic, you will need to define a new command to do this. Again you may also use `(` and `)` to denote a page range.

If you are using the hyperref or html package, you will need to use one of the `\hyper`<*xx*> commands that are defined by the glossaries package, such as `\hyperbf`, if you want to retain a hyperlink. If you instead use `\textbf` you will lose the hyperlink. See the glossaries documentation for further details.

`counter` This specifies which counter to use for the associated number in the glossary entry. This is usually the page number, but can be changed to, say, the section in which the term is used.

`hyper` This is a boolean key which can be used to enable/disable the hyperlink to the relevant entry in the glossary. Note that setting `hyper=true` will only have an effect if hyperlinks are supported (through loading the hyperref or html packages.)

`\gls[`<*options*>`]{`<*label*>`}[`<*insert*>`]` Definition

This is the same as `\glslink` except that the link text is determined from the value of the `text` or `first` keys supplied when the term was defined by `\newglossaryentry`. The first optional argument is the same as that for `\glslink`. The final optional argument <*insert*> allows you to insert some additional text into the link text. By default, this will append <*insert*> to the end of the link text. One of the examples above defined a new glossary entry labelled `matrix`. Suppose in my document I want to write, say, "the matrix's dimensions are given by n and m", then I can do:

```
the \gls{matrix}['s] dimensions are given by $n$ and $m$
```

The text "matrix's" will appear as a link. Of course, you can simply do:

```
the \gls{matrix}'s dimensions are given by $n$ and $m$
```

If there are no hyperlinks (you haven't loaded hyperref or html) then there will be no noticeable difference between the two lines of code above[4]. If you do have hyperlinks enabled then the second way will look a little odd if you use the `colorlinks` hyperref option, and will look ugly if you use the default boxed link style.

`\Gls[<options>]{<label>}[<insert>]` Definition

[4] assuming you haven't changed the way in which the inserted text is added.

This is like `\gls` except that the first letter of the link text is converted to upper case in the event that the term appears at the start of a sentence.

`\GLS[<`*options*`>]{<`*label*`>}[<`*insert*`>]` Definition

This is like `\gls` except that the entire link text is converted to upper case. This is less useful as you shouldn't use this command in titles or page headers in the same way that you shouldn't put commands such as `\index` in similar places. If you want a glossary term to appear in a heading or title you should use

`\glsdisplaytext{<`*label*`>}` Definition

This produces the value assigned with the `text` key when the entry was defined, but it does not add any information to the glossary nor does it generate a hyperlink. There are also analogous commands `\Glsentrytext` (make the first letter upper case), `\glsentryfirst` (the value assigned with `first` key) and `\Glsentryfirst` (as previous, but makes the first letter upper case.)

`\glspl[<`*options*`>]{<`*label*`>}[<`*insert*`>]` Definition

This is analogous to `\gls` but produces the plural form as specified by either the `plural` or `firstplural` keys. Again there are analogous commands

`\Glspl[<options>]{<label>}[<insert>]` Definition

`\GLSpl[<options>]{<label>}[<insert>]` Definition

The same caveats above apply here. If you want plural forms in headings then use `\glsentryplural{<label>}` or `\Glsentryplural{<label>}`, which are analogous to `\glsentrytext` and `\Glsentrytext`, or `\glsentryfirstplural{<label>}` or `\Glsentryfirstplural{<label>}`, which are analogous to `\glsentryfirst` and `\Glsentryfirst`.

You can add a line to the glossary without generating text using

`\glsadd[<options>]{<label>}` Definition

The optional argument is the same as that for `\glslink` except that the `hyper` key has no meaning since no text is generated by the command.

If you want to add all the entries you have defined for a given glossary, you can do so using

```
\glsaddall[<glossary list>]
```

Definition

If you have defined additional glossaries, you can specify to add only those entries which belong to the glossaries listed in <*glossary list*>.

7.2.3 Displaying the Glossary

To display the glossary, you can either use

```
\printglossary[<options>]
```

Definition

or

```
\printglossaries
```

Definition

at the point where you want the glossaries to appear. It is simpler to use just `\printglossaries` which will display the glossaries in the order in which they were defined, otherwise you will need to specify a separate `\printglossary` for each glossary.

The optional argument to `\printglossary` is a <*key*>=<*value*> list of options, where the following keys are defined:

`type` The value of this key specifies which glossary to display. If omitted the main glossary is assumed.

`title` The glossary's title (overriding the title specified when the glossary was defined.)

`toctitle` The title for the table of contents (if the `toc` package option is used.)

`style` The glossary style to use for this glossary. There are many predefined styles available, check the glossaries documentation for details.

Remember that if you use the `acronym` package option, your document will have at least two glossaries, so if you don't use `\printglossaries` you would need to do

```
\printglossary[type=acronym]
```

to ensure that the list of acronyms appears.

7.2.4 Generating the Glossary Files

As with creating bibliographies with BibTeX and creating an index with makeindex it is necessary to use an external application in order to generate the files containing the glossaries which are then loaded by `\printglossary` or `\printglossaries`. In this case we again use `makeindex` but more information is required than was necessary to create an index. If you miss this step, or if something goes wrong with this step, the glossary will not appear in the document.

Recall that to make the glossaries appear, it was necessary to use the command `\makeglossaries`. This performs two functions: firstly, it allows the glossary information to be written to external files, secondly, it generates a customized `makeindex .ist` style file. This file needs to be passed to `makeindex` via the `-s` switch. In addition, you need to specify the output file, and `makeindex` needs to be called *separately* for each glossary. If you have multiple glossaries, but you specify them all in a single call to `makeindex` then it will concatenate all the entries into a single glossary. If your main document file is called, say `thesis.tex`, then to create the main glossary you will need to do:

```
makeindex -s thesis.ist -t thesis.glg -o thesis.gls thesis.glo
```

If you have specified the `acronym` package option you will also need to do:

```
makeindex -s thesis.ist -t thesis.alg -o thesis.acr thesis.acn
```

If you have created any additional glossaries, you will need to do something similar for each additional glossary. This is fairly cumbersome, so the glossaries package comes with a Perl script which will automate this process for you. All you have to do is specify the name of the auxiliary file *without* the extension, and `makeindex` will be called the required number of times with the necessary settings.

If you want to add any extra information to the start or end of the glossary, you can redefine the commands `\glossarypreamble` and `\glossarypostamble`. The latest version of the glossaries package can be downloaded from CTAN or from `http://theoval.cmp.uea.ac.uk/~nlct/latex/packages/`. The glossaries package has a FAQ available at `http://theoval.cmp.uea.ac.uk/~nlct/latex/packages/faq/`.

You can download for an example of how to create a list of acronyms or you can download for an example of how to create a glossary containing symbols.

Chapter 8

Too Many Unprocessed Floats

A common problem PhD student's encounter when writing a thesis is the "too many unprocessed floats" error. This is usually caused by having too many figures and tables in the results chapter and not enough surrounding text. If this happens, there are a number of things you can try doing:

["Too many unprocessed floats]

1. Make sure you haven't been too restrictive in where you want your floats to go. If you use a placement specifier, give LaTeX as many options as possible. For example:

   ```
   \begin{figure}[htbp]
   ```

   which indicates that the figure can be placed "here" (`h`), at the top of a page (`t`), at the bottom of the page (`b`) or on a page solely consisting of floats (`p`). If you just use the `h` placement specifier then you are stating: "I want it *here* and *nowhere else*!" If TeX can't put it *exactly here*, then you have given no alternative place to put it, and it won't get placed anywhere, unless a `\clearpage` command is issued, at which point all remaining unprocessed floats will be dumped at that point. If you are determined that an image must be placed *exactly here* then it should not be placed in a floating environment.

2. Try increasing the amount of text in the chapter. Remember that you should never simply print all the figures and tables in a results chapter without discussing them to some extent.

3. If all else fails, try using the `\clearpage` command. This forces all unprocessed floats to be processed immediately, and start a new page. This may result in the page ending prematurely, if you wish to avoid this, you can use the afterpage package, and use the command:

   ```
   \afterpage{\clearpage}
   ```

For other problems, check the FAQ on the TEX Archive [4].

Chapter 9

General Thesis Writing Advice

This chapter is not specific to LaTeX. Some of the points have already been mentioned in asides or footnotes. Remember that each college or university or even school within a university may have different requirements, and requirements will also vary according to country, so some of this advice may not apply to you. I am writing from the point of view of an English scientist, and am basing it on my own experience and on the comments of English science-based PhD examiners and supervisors. I cannot guarantee that your own department or university will agree with them.

1. Find out the thesis style requirements from your supervisor or your department's website. Many universities still require double-spaced, single sided documents with wide margins. Double-spacing is by and large looked down on in the world of typesetting, but this requirement for a PhD thesis has nothing to do with æsthetics or readability. In England the purpose of the PhD viva is to defend your work[1].

[1] I gather this is not the case in some other countries, where the viva is more

Before your viva, paper copies of your thesis are sent to your examiners. The double spacing and wide margins provide the examiners room to write the comments and criticisms they wish to raise during the viva, as well as any typographical corrections. Whilst they could write these comments on a separate piece of paper, cross-referencing the relevant page in the thesis, it is more efficient for the comments to actually be on the relevant page of the thesis. That way, as they go through the manuscript during your viva, they can easily see the comments, questions or criticisms they wish to raise alongside the relevant text. If you present them with a single-spaced document with narrow margins, you are effectively telling them that you don't want them to criticise your work!

2. Don't try to pad your thesis with irrelevant information. This includes adding items in your bibliography that are not referenced in the text, adding figures or tables that are not explained in the text, and supplying all the source code you have written. The outcome of your viva will not depend on the physical size of your thesis, but on the clarity of your writing and on the quality of your work.

3. Clearly delineate your thesis through the use of chapters and sec-

informal, and the decision to pass or fail you has already been made before your viva.

tions, outlining your original aims and objectives, an overview of the subject matter including references to other people's work in the area, the methods you employed to extend or innovate the field, your results and conclusions.

4. Make sure your references include some recent journal or conference papers to illustrate that you are aware of new developments in your field. Remember that due to the nature of publishing, most books are dated by the time they reach the book shelves. Journal and conference papers are likely to be more up-to-date[2].

5. Always explain acronyms, technical terms and symbols. It is a good idea to include a glossary of terms, list of notation or list of acronyms to avoid confusion.

6. If you have equations, make sure you explain the variables used, and how you go from one equation to the next. Depending on your field, you might also consider clarifying the mathematics by providing

[2] Having said that, I know someone who submitted an article to a journal, and it took three and a half years before the reviewers came back with comments. In the end, the author withdrew the manuscript because by that time the topic was out of date.

graphical representations of the equations[3].

7. If you include any graphs, bar charts, pie charts or any other form of data plot, make sure it is clearly labelled and no distortion is introduced (such as using three-dimensional bar charts or pie charts[4].)

8. If you have used a computer application to generate numerical results, make sure you have some understanding of the underlying process and what the results mean. This doesn't necessarily mean that you need to understand complex computer code, or complex algorithms, but what you shouldn't do is say something along the lines of "well, I clicked on this button, and it said $m = 0.678$." What is the purpose of the button? What does m represent? What does the result $m = 0.678$ signify? What value were you expecting or hoping to get? Numbers on their own are meaningless. If I ran into a room shouting "I've got 42!" What does that mean? Forty-two what? Forty-two brilliant reviews? (Great!) Forty-two percent in an exam? (Not good.) Forty-two spots on my face? (Very bad!)

[3] When I was a PhD student, I was once rendered speechless when asked to provide a graphical illustration of an equation involving a quadruple summation that had no graphical meaning from my point of view. Perhaps this was a drawback of being a mathematician doing a PhD in an electronics department.

[4] The sole purpose of 3D pie charts or bar charts appears to be to look pretty and impress people who have no understanding of mathematics.

9. Don't waste time worrying about the best way to word your thesis. Write first, then edit it later or you will never get started.

10. If your supervisor offers to edit chapters of your thesis, take them up on their offer! Such offers are not made out of politeness, but a desire to ensure that you pass. Don't be embarrassed and worry that it's not good enough, that's the whole point in your supervisor helping you improve it[5].

11. Write in a clear concise manner. A thesis is a technical document, not a novel, so don't be tempted to write something along the lines of: "I awaited with bated breath, my whole body quivering with excitement at the eager anticipation that my algorithm would prove superior to all others, and, oh joy, my experiments proved me right."

12. Don't decorate your thesis with irrelevant clip art. It is unprofessional and highly inappropriate in the sciences.

13. Make regular back-ups of your work. Be prepared for any of the following: accidentally deleting your thesis, accidentally overwriting your thesis with another file, software failure, hardware failure, fire and theft.

[5] but don't expect your supervisor to actually write your thesis!

Items 9 and 10 above were supplied by Dr Gavin Cawley[6] who has been both a PhD supervisor and examiner.

[6] School of Computing Sciences, University of East Anglia

GNU Free Documentation License

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "**Document**", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "**you**". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "**Modified Version**" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “**Secondary Section**” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “**Invariant Sections**” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “**Cover Texts**” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “**Transparent**” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "**Opaque**".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "**Title Page**" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

A section “**Entitled XYZ**” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “**Acknowledgements**”, “**Dedications**”, “**Endorsements**”, or “**History**”.) To “**Preserve the Title**” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled “Acknowledgements” or “Dedications”, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties–for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as

Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with … Texts." line with this:

> with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Bibliography

[1] "The LaTeX Companion", Michel Goossens, Frank Mittelbach and Alexander Samarin, Addison-Wesley (1994). 31, 32, 83

[2] "A Guide to LaTeX2ε: document preparation for beginners and advanced users", Helmut Kopka and Patrick W. Daly, Addison-Wesley (1995). 22, 28, 31, 32, 83

[3] "LaTeX : a document preparation system", Leslie Lamport, 2nd ed. Addison-Wesley (1994). 75, 83

[4] The TeX Archive. http://www.tex.ac.uk/ 101

[5] "LaTeX for Complete Novices", Nicola Talbot. http://theoval.cmp.uea.ac.uk/~nlct/latex/novices/ (2007). b, 1, 64

Index

Symbols
A N
B O
C P
D Q
E R
F S
G T
H U
I V
J W
K X
L Y
M Z

Symbols
A N
B O
C P
D Q
E R
F S
G T
H U
I V
J W
K X
L Y
M Z

Symbols
A N
B O
C P
D Q
E R
F S
G T
H U
I V
J W
K X
L Y
M Z

Symbols
A N
B O
C P
D Q
E R
F S
G T
H U
I V
J W
K X
L Y
M Z

Symbols
A N
B O
C P
D Q
E R
F S
G T
H U
I V
J W
K X
L Y
M Z