HOW TO TEX A THESIS: THE PURDUE THESIS STYLES

A Manual

Submitted to the Graduate Students

of

Purdue University

by

James Darrell McCauley and Scott Hucker

In Partial Fulfillment of the

Requirements for the Degree

of

Master TEXnician

March 1993

DISCARD THIS PAGE

TABLE OF CONTENTS

DISCARD THIS PAGE

LIST OF FIGURES

DISCARD THIS PAGE

NOMENCLATURE

TEX	a typesetting system by Donald Knuth [2], also refers to the "plain" format. The proper pronunciation rhymes with "heck" and "peck" and does not sound like "hex" or "Rex."
LATEX	a set of TEX macros comprising the "lplain" format, originally written by Leslie Lamport [3]. The proper pronunciation is `lā·tek'` and not `lā'·teks` (see above).
BIBTEX	a bibliography generation program by Oren Patashnik [3] that can be used with either plain TEX or LATEX.
`epic`	Enhancements to the LATEX`picture` environment; a LATEX style option.
`eepic`	Enhancements to `epic`; a LATEX style option.
PICTEX	A collection of picture drawing commands, written by Michael Spivak; works with plain TEX or LATEX.

ABSTRACT

Thesis, James Darrell McCauley and Scott Hucker, Purdue University, March 1993. How to TEX a Thesis: The Purdue Thesis Styles. Major Professor: Dr. Purdue Pete.

This is not a thesis and Master TEXnician is not a degree granted by Purdue University.

This manual describes the usage of LATEX styles to typeset theses at Purdue University. Chapter 1 gives basic information about thesis formatting and adds a few disclaimers about this software and manual. Chapter 2 gives lots of "essential" information about the LATEX typesetting system. Chapter 3 gives a good introduction to BIBTEX, a bibliography generation system, and Chapter 4 gives a very brief overview of math environments. Chapter 5 describes the concept of a *float* along with methods of figure inclusion in LATEX, including `gnuplot` and `xfig`. Chapter 6 gives specific information about using the Purdue thesis styles for LATEX—Chapters 2–5 are general LATEX information and can be viewed as a separate but necessary tutorial. Finally, Chapter 7 gives pointers for finding help with LATEX and thesis formatting.

The style discussed in this manual were originally written by Dave Kraynie and have been modified by McCauley. Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted. This software and its documentation is provided "as is" without express or implied warranty.

1. THESES AT PURDUE WITH LaTeX

This manual is no substitute for the Purdue thesis guide [1] or the LaTeX book [3], but it should be used in conjunction with both. However, this chapter points out some of the major points of the thesis preparation requirements of Purdue and later chapters introduce LaTeX and describe its use.

1.1 Formatting Requirements

While care has been taken by authors of "The Purdue Thesis Style" to relieve the burden of formatting for LaTeX users, there are some points of which you should be aware.

1.1.1 Creative Use of Fonts

First, there is the restriction of typefaces. Bold face cannot be used in the text. Italic (or emphasized) text may not be used in the body of the thesis but it is acceptable "for books titles and foreign terms [1]." This manual, for the sake of readability, violates this point.

1.1.2 Spacing

The following is taken directly out of the guide [1]:

> The text of the thesis is double-spaced or one and a half spaced. Multi-lined headings, subdivision, figure and table captions, long quotations, footnotes and notes are single–spaced. The bibliography is double-spaced or one and a half spaced between entries and single-spaced within entries.

Format approval must occur at least two weeks before the last day of each semester. "The examining committee needs a period of at least two weeks from the time of format approval to the defense date to study the thesis [1]." For format approval, you need an appointment (494–2905). At the alloted time, bring a copy of the completed thesis to Thesis Format Checking in Special Collections room 279 Stewart Center. Degree candidates are completely responsible for the formatting of their thesis. See the Purdue thesis guide [1] for more information.

1.3 Disclaimer of Liability

Please note that this software and its documentation is provided "as is" without express or implied warranty. Furthermore, if you are using LaTeX on Purdue's Engineering Computer Network (ECN), also note that TeX, LaTeX, and all associated software (e.g., `xdvi`, `dvips`, and `xfig`) are classified as "unsupported software." This means that ECN staff assume no responsibility whatsoever. It is only out of a benovolent heart that they provide the disk space for such software and give certain ECN users the privilege to install and maintain it.

Those who support "unsupported software" do so on a voluntary basis. If you are unsatisfied with the level of support, you can volunteer to take over or share maintenance duties. This is all outlined in the file `/usr/unsup/README`.

Those who maintain TeX and LaTeX on ECN are no exception. While usually willing to answer questions and to provide "fixes" to problems as their schedules permit, they cannot assume responsibility for any deadlines missed or time "wasted" by using LaTeX or The Purdue Thesis Styles.

THE ENGINEERING COMPUTER NETWORK (ECN) AT PURDUE, ECN TEX SUPPORT VOLUNTEERS, AND JAMES DARRELL MCCAULEY (THE

AUTHOR OF THIS DOCUMENT) MAKE NO EXPRESS OR IMPLIED WARRANTIES (INCLUDING BY WAY OF EXAMPLE, MERCHANTABILITY) WITH RESPECT TO ANY ITEM, AND SHALL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL OR CONSEQUENTAL DAMAGES ARISING OUT OF THE POSSESSION OR USE OF ANY SUCH ITEM. LICENSEE AND/OR USER AGREES TO INDEMNIFY AND HOLD ECN, TEX SUPPORT, AND JAMES DARRELL MCCAULEY HARMLESS FROM ANY CLAIMS ARISING OUT OF THE USE OR POSSESSION OF SUCH ITEMS.

The message in this section should be clear. If there are any questions about it, contact any of the parties mentioned. If it makes you uncomfortable, do not use The Purdue Thesis Styles.

Otherwise, if you are new to TEX or LATEX, prepare yourself for a journey! After dabbling with LATEX, seeing how nice your articles, reports, or thesis looks, and realizing the time–saving advantages of such a system, your outlook on convential text processing will never be the same.

2. ESSENTIAL LaTeX

The purpose of this chapter is to familarize you with LaTeX in general, giving you "essential" information. It borrows liberally from a paper called "Essential LaTeX" by Jon Warbrick at Plymouth Polytechnic. The original paper may be obtained from the directory `/home/harbor2/texsuprt/TeX/guides/essential.ps`. While not necessarily focused on producing a thesis with LaTeX, it is considered prerequisite reading material.

2.1 Introduction

This chapter is an attempt to give you all the essential information that you will need in order to use the LaTeX Document Preparation System. Only very basic features are covered, and a vast amount of detail has been omitted. It is not possible to include everything that you might need to know, and if you intend to make extensive use of the program you should refer to a more complete reference.

The main reference for LaTeX is *The* LaTeX *User's guide and Reference Manual* by Leslie Lamport [3]. This contains all the information that you will ever need to know about the program, and you will need access to a copy if you are to use LaTeX seriously. You may purchase a copy at local bookstores.

2.2 How does LaTeX work?

In order to use LaTeX you generate a file containing both the text that you wish to print and instructions to tell LaTeX how you want it to appear. You will normally create this file using your system's text editor (e.g., `vi`, `emacs`, `textedit`). You can give the file any name you like, but it should end "`.tex`" to identify the file's

contents. You then get LaTeX to process the file, and it creates a new file of typesetting commands; this has the same name as your file but the "`.tex`" ending is replaced by "`.dvi`". This stands for '*D*evice *I*ndependent' and, as the name implies, this file can be used to create output on a range of printing devices. Your *local guide* will go into more detail (see Chapter 7).

Rather than encourage you to dictate exactly how your document should be laid out, LaTeX instructions allow you describe its *logical structure*. For example, you can think of a quotation embedded within your text as an element of this logical structure: you would normally expect a quotation to be displayed in a recognizable style to set it off from the rest of the text. A human typesetter would recognize the quotation and handle it accordingly, but since LaTeX is only a computer program it requires your help. There are therefore LaTeX commands that allow you to identify quotations and as a result allow LaTeX to typeset them correctly.

Fundamental to LaTeX is the idea of a *document style* that determines exactly how a document will be formatted. LaTeX provides standard document styles that describe how standard logical structures (such as quotations) should be formatted. You may have to supplement these styles by specifying the formatting of logical structures peculiar to your document, such as mathematical formulae. You can also modify the standard document styles or even create an entirely new one, though you should know the basic principles of typographical design before creating a radically new style.

There are a number of good reasons for concentrating on the logical structure rather than on the appearance of a document. It prevents you from making elementary typographical errors in the mistaken idea that they improve the aesthetics of a document—you should remember that the primary function of document design is to make documents easier to read, not prettier. It is more flexible, since you only need to alter the definition of the quotation style to change the appearance of all the quotations in a document. Most important of all, logical design encourages better writing. A visual system makes it easier to create visual effects rather than a coherent

structure; logical design encourages you to concentrate on your writing and makes it harder to use formatting as a substitute for good writing.

2.3 A Sample LaTeX file

Have a look at the example LaTeX file in Figure 2.1. It is a slightly modified copy of the standard LaTeX example file `small.tex`. It is available in the directory `/usr/unsup/lib/tex/inputs`. The line numbers down the left–hand side are not part of the file, but have been added to make it easier to identify various portions. Try copying this file, running LaTeX on it, and viewing the output:

```
% latex small
% xdvi small       # if your are running OpenWindows or X11
% dvips small      # to create a PostScript file
% lpr -P<printer> small.ps   # to print
```

2.3.1 Running Text

Most documents consist almost entirely of running text—words formed into sentences, which are in turn formed into paragraphs—and the example file is no exception. Describing running text poses no problems, you just type it in naturally. In the output that it produces, LaTeX will fill lines and adjust the spacing between words to give tidy left and right margins. The spacing and distribution of the words in your input file will have no effect at all on the eventual output. Any number of spaces in your input file are treated as a single space by LaTeX, it also regards the end of each line as a space between words (see lines 15–17). A new paragraph is indicated by a blank line in your input file, so don't leave any blank lines unless you really wish to start a paragraph.

LaTeX reserves a number of the less common keyboard characters for its own use. The ten characters

```
# $ % & ~ _ ^ \ { }
```

should not appear as part of your text, because if they do LaTeX will get confused.

```
% SMALL.TEX -- Released 5 July 1985
% USE THIS FILE AS A MODEL FOR MAKING YOUR OWN LaTeX INPUT FILE.
% EVERYTHING TO THE RIGHT OF A  %  IS A REMARK TO YOU AND IS IGNORED
% BY LaTeX.
%
% WARNING!  DO NOT TYPE ANY OF THE FOLLOWING 10 CHARACTERS EXCEPT AS
% DIRECTED:        &   $   #   %   _   {   }   ^   ~   \

\documentstyle[11pt]{article}     % YOUR INPUT FILE MUST CONTAIN THESE
\begin{document}                  % TWO LINES PLUS THE \end COMMAND AT
                                  % THE END

\section{Simple Text}          % THIS COMMAND MAKES A SECTION TITLE.

Words are separated by one or    more      spaces.  Paragraphs are
    separated by one or more blank lines.  The output is not affected
by adding extra spaces or extra blank lines to the input file.


Double quotes are typed like this: ``quoted text''.
Single quotes are typed like this: `single--quoted text'.

Long dashes are typed as three dash characters---like this.

Italic text is typed like this: {\em this is italic text}.
Bold   text is typed like this: {\bf this is  bold  text}.

\subsection{A Warning or Two}        % THIS MAKES A SUBSECTION TITLE.

If you get too much space after a mid-sentence period---abbreviations
like etc.\ are the common culprits)---then type a backslash followed by
a space after the period, as in this sentence.

Remember, don't type the 10 special characters (such as dollar sign and
backslash) except as directed!  The following seven are printed by
typing a backslash in front of them:  \$  \&  \#  \%  \_  \{  and  \}.
The manual tells how to make other symbols.

\end{document}                   % THE INPUT FILE ENDS LIKE THIS
```

Figure 2.1 A Sample LaTeX File

There are a number of words in the file that start '\' (see lines 9, 10 and 13). These are LaTeX *commands* and they describe the structure of your document. There are a number of things that you should realize about these commands:

- All LaTeX commands consist of a '\' followed by one or more characters.
- LaTeX commands should be typed using the correct mixture of upper- and lower–case letters. `\BEGIN` is *not* the same as `\begin`.
- Some commands are placed within your text. These are used to switch things, like different typestyles, on and off. The `\em` command is used like this to emphasise text, normally by changing to an *italic* typestyle (see line 25). The command and the text are always enclosed between '`{`' and '`}`'—the '`{\em`' turns the effect on and and the '`}`' turns it off.
- There are other commands that look like

  ```
  \command{text}
  ```

  In this case the text is called the "argument" of the command. The `\section` command is like this (see line 13). Sometimes you have to use curly brackets '`{}`' to enclose the argument, sometimes square brackets '`[]`', and sometimes both at once. There is method behind this apparent madness, but for the time being you should be sure to copy the commands exactly as given.
- When a command's name is made up entirely of letters, you must make sure that the end of the command is marked by something that isn't a letter. This is usually either the opening bracket around the command's argument, or it's a space. When it's a space, that space is always ignored by LaTeX. We will see later that this can sometimes be a problem.

2.3.3 Overall structure

There are some LaTeX commands that must appear in every document. The actual text of the document always starts with a `\begin{document}` command and ends with an `\end{document}` command (see lines 10 and 39). Anything that comes after the `\end{document}` command is ignored. Everything that comes before the `\begin{document}` command is called the *preamble*. The preamble can only contain LaTeX commands to describe the document's style.

One command that must appear in the preamble is the `\documentstyle` command (see line 9). This command specifies the overall style for the document. There are quite a number of styles that you can use, as you will find out later on in this chapter.

2.3.4 Other Things to Look At

LaTeX can print both opening and closing quote characters, and can manage either of these either single or double. To do this it uses the two quote characters from your keyboard: ‘ and ’. You will probably think of ’ as the ordinary single quote character which probably looks like ´ or ' on your keyboard, and ‘ as a “funny” character that probably appears as `. You type these characters once for single quote (see line 21), and twice for double quotes (see line 20). The double quote character " itself is almost never used and should not be used unless you want your text to look "funny" (compare quote in previous sentence).

LaTeX can produce three different kinds of dashes. A long dash, for use as a punctuation symbol, as is typed as three dash characters in a row, like this ‘---’ (see line 23). A shorter dash, used between numbers as in ‘10–20’, is typed as two dash characters in a row, while a single dash character is used as a hyphen.

From time to time you will need to include one or more of the LaTeX special symbols in your text. Seven of them can be printed by making them into commands by proceeding them by backslash (see line 36). The remaining three symbols can be produced by more advanced commands, as can symbols that do not appear on your keyboard such as †, ‡, §, £, ©, ♯ and ♣.

It is sometimes useful to include comments in a LaTeX file, to remind you of what you have done or why you did it. Everything to the right of a `%` sign is ignored by LaTeX, and so it can be used to introduce a comment.

2.4 Document Styles and Style Options

There are four standard document styles available in LaTeX:

`article` intended for short documents and articles for publication. Articles do not have chapters, and when `\maketitle` is used to generate a title (see Section 2.9) it appears at the top of the first page rather than on a page of its own.

`report` intended for longer technical documents. It is similar to `article`, except that it contains chapters and the title appears on a page of its own.

`book` intended as a basis for book publication. Page layout is adjusted assuming that the output will eventually be used to print on both sides of the paper.

`letter` intended for producing personal letters. This style will allow you to produce all the elements of a well laid out letter: addresses, date, signature, etc.

These standard styles can be modified by a number of *style options*. They appear in square brackets after the `\documentstyle` command. Only one style can ever be used but you can have more than one style option, in which case their names should be separated by commas. The standard style options are:

`11pt` prints the document using eleven–point type for the running text rather that the ten-point type normally used. Eleven–point type is about ten percent larger than ten–point.

`12pt` prints the document using twelve–point type for the running text rather than the ten-point type normally used. Twelve–point type is about twenty percent larger than ten–point.

`twoside` causes documents in the article or report styles to be formatted for printing on both sides of the paper. This is the default for the book style.

`twocolumn` produces two columns on each page.

`titlepage` causes the `\maketitle` command to generate a title on a separate page for documents in the `article` style. A separate page is always used in both the `report` and `book` styles.

An additional document style, the one used for this document and for Purdue theses, is `puthesis`. Style options for eleven–point type and twelve–point type are available.

2.5 Environments

We mentioned earlier the idea of identifying a quotation to LaTeX so that it could arrange to typeset it correctly. To do this you enclose the quotation between the commands `\begin{quotation}` and `\end{quotation}`. This is an example of a LaTeX construction called an *environment*. A number of special effects are obtained by putting text into particular environments.

2.5.1 Quotations

There are two environments for quotations: `quote` and `quotation`. `quote` is used either for a short quotation or for a sequence of short quotations separated by blank lines:

```
US presidents ... pithy remarks:
\begin{quote}
The buck stops here.

I am not a crook.
\end{quote}
```

US presidents have been known for their pithy remarks:

> The buck stops here.
>
> I am not a crook.

Use the `quotation` environment for quotations that consist of more than one paragraph. Paragraphs in the input are separated by blank lines as usual:

```
Here is some advice to remember:
\begin{quotation}
Environments for making
...other things as well.

Many problems
...environments.
\end{quotation}
```

Here is some advice to remember:

> Environments for making quotations can be used for other things as well.
>
> Many problems can be solved by novel applications of existing environments.

2.5.2 Centering and Flushing

Text can be centered on the page by putting it within the `center` environment, and it will appear flush against the left or right margins if it is placed within the `flushleft` or `flushright` environments.

Text within these environments will be formatted in the normal way, in particular the ends of the lines that you type are just regarded as spaces. To indicate a "newline" you need to type the `\\` command. For example:

```
\begin{center}
one
two
three \\
four \\
five
\end{center}
```

one two three
four
five

2.5.3 Lists

There are three environments for constructing lists. In each one each new item is begun with an `\item` command. In the `itemize` environment the start of each item is given a marker, in the `enumerate` environment each item is marked by a number. These environments can be nested within each other in which case the amount of indentation used is adjusted accordingly:

```
\begin{itemize}
\item Itemized lists are handy.
\item However, don't forget
  \begin{enumerate}
  \item The 'item' command.
  \item The 'end' command.
  \end{enumerate}
\end{itemize}
```

- Itemized lists are handy.
- However, don't forget
  1. The 'item' command.
  2. The 'end' command.

The third list making environment is `description`. In a description you specify the item labels inside square brackets after the `\item` command. For example:

```
Three animals that you should
know about are:
\begin{description}
  \item[gnat] A small animal...
  \item[gnu] A large animal...
  \item[armadillo] A ...
\end{description}
```

Three animals that you should know about are:

gnat A small animal that causes no end of trouble.

gnu A large animal that causes no end of trouble.

armadillo A medium-sized animal.

2.5.4 Tabbing

LaTeX will almost always convert a sequence of spaces, including tabs, into a single space. See what happens in this example:

```
\begin{flushleft}
Income  Expenditure Result   \\
20s 0d  19s 11d     happiness \\
20s 0d  20s 1d      misery  \\
\end{flushleft}
```

Income Expenditure Result
20s 0d 19s 11d happiness
20s 0d 20s 1d misery

The `tabbing` environment overcomes this problem. Within it you set tabstops and tab to them much like you do on a typewriter. Tabstops are set with the `\=` command, and the `\>` command moves to the next stop. The `\\` command is used to separate each line. A line that ends `\kill` produces no output, and can be used to set tabstops:

```
\begin{tabbing}
Income \=Expenditure \=    \kill
Income \>Expenditure \>Result \\
20s 0d \>19s 11d \>Happiness   \\
20s 0d \>20s 1d  \>Misery    \\
\end{tabbing}
```

Income	Expenditure	Result
20s 0d	19s 11d	Happiness
20s 0d	20s 1d	Misery

Unlike a typewriter's tab key, the `\>` command always moves to the next tabstop in sequence, even if this means moving to the left. This can cause text to be overwritten if the gap between two tabstops is too small.

2.5.5 Verbatim Output

Sometimes you will want to include text exactly as it appears on a terminal screen. For example, you might want to include part of a computer program. Not only do

you want LaTeX to stop playing around with the layout of your text, you also want to be able to type all the characters on your keyboard without confusing LaTeX. The `verbatim` environment has this effect:

```
The section of program in
question is:
\begin{verbatim}
{ this finds %a & %b }

for i := 1 to 27 do
  begin
  table[i] := fn(i);
  process(i)
  end;
\end{verbatim}
```

The section of program in question is:

```
{ this finds %a & %b }

for i := 1 to 27 do
   begin
   table[i] := fn(i);
   process(i)
   end;
```

2.6 Type Styles

We have already come across the `\em` command for changing typeface. Here is a full list of the available typefaces:

`\rm`	Roman	`\it`	*Italic*	`\sc`	Small Caps
`\em`	*Emphatic*	`\sl`	*Slanted*	`\tt`	`Typewriter`
`\bf`	**Boldface**	`\sf`	Sans Serif		

Remember that these commands are used *inside* a pair of braces to limit the amount of text that they effect. In addition to the eight typeface commands, there are a set of commands that alter the size of the type. These commands are:

`\tiny`	`\small`	`\large`	`\huge`
`\scriptsize`	`\normalsize`	`\Large`	`\Huge`
`\footnotesize`		`\LARGE`	

2.7 Sectioning Commands and Tables of Contents

Technical documents, like this one, are often divided into sections. Each section has a heading containing a title and a number for easy reference. LaTeX has a series of commands that will allow you to identify different sorts of sections. Once you have

done this LaTeX takes on the responsibility of laying out the title and of providing the numbers.

The commands that you can use are:

```
\chapter        \subsection          \paragraph
\section        \subsubsection       \subparagraph
```

The naming of these last two is unfortunate, since they do not really have anything to do with 'paragraphs' in the normal sense of the word; they are just lower levels of section. In most document styles, headings made with `\paragraph` and `\subparagraph` are not numbered. `\chapter` is not available in document style `article`. The commands should be used in the order given, since sections are numbered within chapters, subsections within sections, etc.

A seventh sectioning command, `\part`, is also available. Its use is always optional, and it is used to divide a large document into series of parts. It does not alter the numbering used for any of the other commands.

Including the command `\tableofcontents` in your document will cause a contents list to be included, containing information collected from the various sectioning commands. You will notice that each time your document is run through LaTeX the table of contents is always made up of the headings from the previous version of the document. This is because LaTeX collects information for the table as it processes the document, and then includes it the next time it is run. This can sometimes mean that the document has to be processed through LaTeX twice to get a correct table of contents.

2.8 Producing Special Symbols

You can include in you LaTeX document a wide range of symbols that do not appear on you your keyboard. For a start, you can add an accent to any letter:

```
ò \‘{o}    õ \~{o}    ǒ \v{o}    ǫ \c{o}    ó \’{o}
```

ō	`\={o}`	ő	`\H{o}`	o̩	`\d{o}`
ô	`\^{o}`	ȯ	`\.{o}`	o͡o	`\t{oo}`
o̲	`\b{o}`	ö	`\"{o}`	ŏ	`\u{o}`

A number of other symbols are available, and can be used by including the following commands:

†	`\dag`	§	`\S`	©	`\copyright`
‡	`\ddag`	¶	`\P`	£	`\pounds`
œ	`\oe`	Œ	`\OE`	æ	`\AE`
Æ	`\AE`	å	`\aa`	Å	`\AA`
ø	`\o`	Ø	`\O`	ł	`\l`
Ł	`\E`	ß	`\ss`	¿	`?‘`
¡	`!‘`	...	`\ldots`	LaTeX	`\LaTeX`

There is also a `\today` command that prints the current date. When you use these commands remember that LaTeX will ignore any spaces that follow them, so that you can type '`\pounds 20`' to get '£20'. However, if you type '`\LaTeX is wonderful`' you will get 'LaTeXis wonderful'—notice the lack of space after LaTeX. To overcome this problem you can follow any of these commands by a pair of empty brackets and then any spaces that you wish to include, and you will see that `\LaTeX{} really is wonderful!` (LaTeX really is wonderful!).

2.9 Titles

Most documents have a title. To title a LaTeX document, you include the following commands in your document, usually just after `begin{document}`.

```
\title{required title}
\author{required author}
\date{required date}
\maketitle
```

If there are several authors, then their names should be separated by `\and`; they can also be separated by `\\` if you want them to be centered on different lines. If the `\date` command is left out, then the current date will be printed.

```
\title{Essential \LaTeX}
\author{J Warbrick\and A N Other}
\date{14th February 1988}
\maketitle
```

Essential LaTeX

J Warbrick A N Other

14th February 1988

The exact appearance of the title varies depending on the document style. In styles `report` and `book` the title appears on a page of its own. In the `article` style it normally appears at the top of the first page, the style option `titlepage` will alter this (see Section 2.4).

2.10 Errors

When you create a new input file for LaTeX you will probably make mistakes. Everybody does, and it's nothing to be worried about. As with most computer programs, there are two sorts of mistake that you can make: those that LaTeX notices and those that it doesn't. To take a rather silly example, since LaTeX doesn't understand what you are saying, it isn't going to be worried if you mis-spell some of the words in your text. You will just have to accurately proof-read your printed output. On the other hand, if you mis-spell one of the environment names in your file then LaTeX won't know what you want it to do.

When this sort of thing happens, LaTeX prints an error message on your terminal screen and then stops and waits for you to take some action. Unfortunately, the error messages that it produces are rather user-unfriendly. Nevertheless, if you know where to look they will probably tell you where the error is and went wrong.

Consider what would happen if you mistyped `\begin{itemize}` so that it became `\begin{itemie}`. When LaTeX processes this instruction, it displays the following on your terminal:

```
LaTeX error.  See LaTeX manual for explanation.
              Type  H <return>  for immediate help.
```

```
! Environment itemie undefined.
\@latexerr ...for immediate help.}\errmessage {#1}
                                                \endgroup
l.140 \begin{itemie}

?
```

After typing the '?' LaTeX stops and waits for you to tell it what to do.

The first two lines of the message just tell you that the error was detected by LaTeX. The third line, the one that starts '!' is the *error indicator*. It tells you what the problem is, though until you have had some experience of LaTeX this may not mean a lot to you. In this case it is just telling you that it doesn't recognize an environment called `itemie`. The next two lines tell you what LaTeX was doing when it found the error, they are irrelevant at the moment and can be ignored. The final line is called the *error locator*, and is a copy of the line from your file that caused the problem. It starts with a line number to help you to find it in your file, and if the error was in the middle of a line it will be shown broken at the point where LaTeX realized that there was an error. LaTeX can sometimes pass the point where the real error is before discovering that something is wrong, but it doesn't usually get very far.

At this point you could do several things. If you knew enough about LaTeX you might be able to fix the problem, or you could type 'X' and press the return key to stop LaTeX while you go and correct the error. The best thing to do, however, is just to press the return key. This will allow LaTeX to go on running as if nothing had happened. If you have made one mistake, then you have probably made several and you may as well try to find them all in one go. It's much more efficient to do it this way than to run LaTeX over and over again fixing one error at a time. Don't worry about remembering what the errors were—a copy of all the error messages is being saved in a *log* file so that you can look at them afterwards.

If you look at the line that caused the error it's normally obvious what the problem was. If you can't work out what you problem is look at the hints below, and if they don't help consult Chapter 6 of the manual [3]. It contains a list of all of the error

messages that you are likely to encounter together with some hints as to what may have caused them.

Some of the most common mistakes that cause errors are

- A mispelled command or environment name.
- Improperly matched '`{`' and '`}`'—remember that they should always come in pairs.
- Trying to use one of the ten special characters `# $ % & _ { } ~ ^` and `\` as an ordinary printing symbol.
- A missing `\end` command.
- A missing command argument (that's the bit enclosed in '`{`' and '`}`').

One error can get LaTeX so confused that it reports a series of spurious errors as a result. If you have an error that you understand, followed by a series that you don't, then try correcting the first error—the rest may vanish as if by magic.

Sometimes LaTeX may write a `*` and stop without an error message. This is normally caused by a missing `\end{document}` command, but other errors can cause it. If this happens type `\stop` and press the return key.

Finally, LaTeX will sometimes print *warning* messages. They report problems that were not bad enough to cause LaTeX to stop processing, but nevertheless may require investigation. The most common problems are 'overfull' and 'underfull' lines of text. A message like:

```
Overfull \hbox (10.58649pt too wide) in paragraph at lines 172--175
[]\tenrm Mathematical for-mu-las may be dis-played. A dis-played
```

indicates that LaTeX could not find a good place to break a line when laying out a paragraph. As a result, it was forced to let the line stick out into the right-hand margin, in this case by 10.6 points. Since a point is about 1/72nd of an inch this may be rather hard to see, but it will be there nonetheless.

This particular problem happens because LaTeX is rather fussy about line breaking, and it would rather generate a line that is too long than generate a paragraph that doesn't meet its high standards. The simplest way around the problem is to enclose the entire offending paragraph between `\begin{sloppypar}` and `\end{sloppypar}` commands. This tells LaTeX that you are happy for it to break its own rules while it is working on that particular bit of text.

Alternatively, messages about "Underfull `\hbox`'es" may appear. These are lines that had to have more space inserted between words than LaTeX would have liked. In general there is not much that you can do about these. Your output will look fine, even if the line looks a bit stretched. About the only thing you could do is to re-write the offending paragraph!

2.11 A Final Reminder

You now know enough LaTeX to produce a wide range of documents. But this chapter has only scratched the surface of the things that LaTeX can do. This entire document was itself produced with LaTeX (with no sticking things in or clever use of a photocopier) and even it hasn't used all the features that it could. From this you may get some feeling for the power that LaTeX puts at your disposal.

Please remember what was said in the introduction: if you **do** have a complex document to produce then **go and read the manual**. You will be wasting your time if you rely only on what you have read here and you will be wasting other busy peoples' time if your questions could be easily answered by the manual (see also Chapter 7).

3. CITATIONS AND BIBLIOGRAPHIES

The task of compiling and formatting the sources cited in papers can be quite tedious, especially for large documents like theses. A program separate from LaTeX, called "BibTeX," can be used to automate this task [3].

3.1 The Citation Command

When referring to the work of someone else, the `\cite` command is used. This generates the citation in the text for you. In the above paragraph, the command `\cite{latex}` was used after the word "task." The formatting of your citation is handled by either the document style or a style option. The default citation style uses the number system (a number enclosed in square brackets). Other citations styles may use the author-date system, e.g. (Lamport, 1986), or the superscript[3] system.

3.2 Bibliography Styles

The way that a reference is formatted in your bibliography depends on the bibliography style, which is specified near the beginning of your document with the `\bibliographstyle{file}` command. `file.bst` is the name of the bibliography style file. Standard bibliography styles include `plain`, `unsrt`, `alpha`, and `abbrev`. The bibliography style governs whether or not references are sorted, whether first names or initials are used for authors, whether or not last names are listed first, the location of the year in the reference (after the author or at the end of the reference), whether or not the year is enclosed within parenthesis, etc. You may be required by your department or major professor to follow a style for a particular journal. If so,

then you will need to find a BibTeX style file to suits your needs (see Chapter 7 for information on how to do this). If you cannot locate an appropriate BibTeX style, then choose one that comes close to the required formatting style and modify your `bbl` by hand. (See §3.4 for a description of the `bbl` file.) Results of a few common (but non–standard) styles are shown below.

`acm.bst` (Assoc. for Computing Machinery)

[1] Jansen, N. B., Flickinger, M. C., and Tsao, G. T. Production of 2,3–butanediol from D–xylose by klebsiella oxytoca ATCC 8724. *Biotechnology and Bioengineering 26* (1984), 362–369.

`ieeetr.bst` (*IEEE Transactions*)

[1] N. B. Jansen, M. C. Flickinger, and G. T. Tsao, "Production of 2,3–butanediol from D–xylose by klebsiella oxytoca ATCC 8724," *Biotechnology and Bioengineering*, vol. 26, pp. 362–369, 1984.

`jacs-new.bst` (*J. of the Am. Chem. Soc.*)

[1] Jansen, N. B.; Flickinger, M. C.; Tsao, G. T. *Biotechnology and Bioengineering*, **1984**, *26*, 362–369.

3.3 The Database

The `\bibliography{file}` command is placed in your input file at the location where the "List of References" section should be. It specifies the name of your bibliographic data base, `file.bib`. An example entry in a BibTeX database follows:

```
@book {  lamport86 ,
author =     "Leslie Lamport" ,
title =      "\LaTeX: A Document Preparation System" ,
publisher =  "Addison--Wesley Pub.\ Co." ,
year =       "1986" ,
address =    "Reading, MA" ,
}
```

The citation key is the first field in this entry—citing this book in a LaTeX file would look like

```
According to Lamport~\cite{lamport86}...
```

The tilde (~) is used to tie the word "Lamport" to the citation generated. The space between these words is then unbreakable—the word "Lamport" and the citation [3] will not be split across two lines if they happen to occur near the end of a line.

A listing of all entry types with their required and optional fields is given in the Appendix. One tool to aid in building BibTeX databases is the BibTeX–mode of an editor called `emacs`. Other tools, some with graphical interfaces, are available on the net. An explanation of their usage is beyond the scope of this chapter. If you cannot figure out how to use these tools, then `vi` or any other text editor will suffice for creation of your database.

There are several rules that you must following when creating your database. Authors are always listed by full names, first name first, and multiple authors are separated by `and`. For example,

```
author = "John Jay Park and Frederick Gene Watson and
          Matthew Scott Livingston",
```

If you were using `abbrv` as your `bibliographystyle`, a reference for these authors may look like:

J. J. Park, F. G. Watson, and M. S. Livingston ...

Some styles only capitalize the first word of the title. If you use any acronyms or other words that should always be capitalized in titles, then they should be enclosed in `{}`'s (e.g., `{FORTRAN}`, `{N}ewton`). This protects the case of these characters.

There are several other rules for building BibTeX databases [3] that cannot be covered here.

3.4 Putting It All Together

To aid the reader in understanding how all this works together, the following excerpt was taken from Lamport [3]:

> When you ran LaTeX with the input file `sample.tex`, you may have noticed that LaTeX created a file named `sample.aux`. This file, called an *auxiliary* file, contains cross–referencing information. Since `sample.tex`

contains no cross–referencing commands, the auxiliary file it produces has no information. However, suppose that LaTeX is run with an imput file named `myfile.tex` that has citations and bibliography–making commands. The auxiliary file `myfile.aux` that it produces will contain all of the citation keys and the arguments of the `\bibliography` and `\bibliographystyle` commands. When BibTeX is run, it reads this information from the auxiliary file and produces a file named `myfile.bbl` containing LaTeX commands to produce the source list ... The next time LaTeX is run on `myfile.tex`, the `\bibliography` command reads the `bbl` file (`myfile.bbl`), which generates the source list.

4. MATHEMATICAL FORMULAE

One of the most attractive features of LaTeX and one of the compelling reasons to use LaTeX for your thesis is the aesthetic quality of mathematics produced. Rather than providing detailed instruction on typesetting math, this chapter provides only some basic information about math environments and provides a few examples to give your a peek at what is possible.

The primary environment for typesetting equations is the `equation` environment (see §2.5 for an introduction to LaTeX environments). The example below

```
\begin{equation}\label{eqn:example}
C\left(d, e\right) = 2d +e - \sqrt{d^2 +    e^2},\: d,e\in\cal{G}
\end{equation}
```

produces

$$C\left(d, e\right) = 2d + e - \sqrt{d^2 + e^2},\ d, e \in \mathcal{Z} \qquad [4.1]$$

The `\label{eqn:example}` part of the command is necessary if you want to automatically reference equations such as eqn. 4.1. (The LaTeX source for the previous sentence ends "`such as eqn.~\ref{eqn:example}.`"—for more imformation about labeling and referencing, see the LaTeX book [3]).

When equations are presented sequentially, the `eqnarray` environment is useful:

```
\begin{eqnarray*}
m_4 & = & \frac{1}{n}\sum_{i=1}^n\left(x_i-\bar{x}\right)^4\\
s^4 & = & \left(\frac{1}{n}\sum_{i=1}^n\left(x_i-\bar{x}
              \right)^2\right)^2
\end{eqnarray*}
```

$$m_4 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^4$$

$$s^4 = \left(\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2\right)^2$$

In the above example, the starred variation of the `eqnarray` environment was used. This differs from the un–starred version in that the former does not number the equations.

Finally, if you are not interested in equation numbering, you can use the `displaymath` environment:

```
\begin{displaymath}
A\left(\alpha,\beta\right) = \left[\begin{array}{ccc}
        1           & 0          & 0 \\
        0           & \beta      & 1 \\
        -3\alpha^2 & -2\beta^2 & -\alpha^2
        \end{array}\right]
\end{displaymath}
```

$$A\left(\alpha,\beta\right) = \left[\begin{array}{ccc} 1 & 0 & 0 \\ 0 & \beta & 1 \\ -3\alpha^2 & -2\beta^2 & -\alpha^2 \end{array}\right]$$

In these last three examples were three environments: `equation`, `eqnarray`, and `displaymath`. Also shown were use of delimters (paren and bracket), subscripts, superscripts, special symbols ($\in$, $\surd$), summations, arrays, calligraphy, and greek letters. This only scratches the surface of what is available. For more information, consult the LaTeX book [3]. This chapter ends with a few more (perhaps non–obvious) examples.

$$\frac{n!}{\prod_{j=1}^{n}(n_j!)} \qquad [4.2]$$

$$P_\delta = \begin{array}{c} \\ g_1 \\ g_2 \\ \\ N_g \end{array} \begin{array}{c} \begin{array}{cccc} g_1 & g_2 & & N_g \end{array} \\ \begin{pmatrix} c_{11} & c_{12} & \dots & c_{1N_g} \\ c_{21} & c_{22} & \dots & c_{2N_g} \\ \vdots & \vdots & \ddots & \vdots \\ c_{N_g1} & c_{N_g2} & \dots & c_{N_gN_g} \end{pmatrix} \end{array}$$

$$R = \begin{cases} 2N_y(N_x - 1), & \text{if } \delta = \pm(1,0) \\ 2N_x(N_y - 1), & \text{if } \delta = \pm(0,1) \\ 2(N_x - 1)(N_y - 1), & \text{if } \delta = \pm(1,-1) \\ & \text{or } \delta = \pm(1,-1) \end{cases}$$

$$M = \left[\begin{array}{c|cc} A & B & C \\ \hline D & E & F \\ G & H & I \end{array}\right]$$

$$\sum_{x \in A}{}' f(x) \stackrel{\text{def}}{=} \sum_{\substack{x \in A \\ x \neq 0}} f(x)$$

$$\underbrace{\{\overbrace{a, \dots, a}^{k\ a\text{'s}}, \overbrace{b, \dots, b}^{l\ b\text{'s}}\}}_{k+l \text{ elements}}$$

5. TABLES AND FIGURES

In §2.5, the concept of a LaTeX environment was introduced. Special environments exist for tables and figures. These environments are special because they are allowed to *float*—that is, LaTeX doesn't always put them at the exact place that they occur in your input file. An algorithm is used to place the floating environments, or floats, at locations which are typographically correct. This may cause endless frustration if you want to have a figure or table occur at a specific location. There are a few methods for solving this.

You can exert some influence on LaTeX's float placement algorithm by using float *position specifiers*. These specifiers, listed below, tell LaTeX what you prefer.

`h`	"here"	do not move this object
`p`	"page"	put this object on a following page
`b`	"bottom"	bottom of a page
`t`	"top"	top of a page

Any combination of these can be used:

```
\begin{figure}[hptb]
 ...
\caption{The graph of x versus y.}
\end{figure}
```

In this example we asked LaTeX to "put the figure 'here,' if possible. If it is not possible (according to the rules encoded in your algorithm), put in on the next 'page.' If this isn't possible, try to put it at the 'top' of a page. The last thing that I want you to do is to put the figure at the 'bottom' of a page."

The remainder of this chapter deals with the ellipsis (...) in the above example, i.e., what to put in a figure. For the formatting of tables, refer to the LaTeX manual [3].

Most laser printers on ECN are PostScript printers. These include Apple LaserWriters, Sun SPARCprinters, and others. Thus, one method of putting a figure into a LaTeX document is instruct the printer driver (`dvips` in our case) to include a PostScript file when creating the output file. Methods of PostScript figure inclusion are discussed in the *Local Guides*—see Chapter 7.

However, because so many vendors' software do not write conforming PostScript code, you will undoubtably have problems using this method at some point. While it may be the only route for some situations, it is certainly not the best method of figure inclusion. For example, figures can look very ugly when fonts used in the figure do not match fonts used in the text. The next two sections describe ways of producing much nicer figures where the problem of non-matching fonts can be avoided.

5.2 GNUPLOT

This section gives instruction for using `gnuplot`, a scientific plotting program available on ECN Suns. With it, you can view output with MIT X Windows (or `openwin`) and save output as LaTeX `pictures`.

5.2.1 The X Window System

If you are fortunate enough to have access to a Sun console or an X terminal, you can instruct `gnuplot` to plot graphs on your screen. A script of a session with `gnuplot` is given below.

```
% gnuplot
gnuplot> set ylabel 'Concentration'
gnuplot> set xlabel 'Time'
gnuplot> plot 'data.dat', 'data.dat' using 1:3
```

After typing the last line, a window pops up on your screen. The plot command tells `gnuplot` to plot two sets of data. The first implicitly uses the first column as the

abscissa and the second column as the ordinate. The second set of data in the file is plotted by explicitly specifying columns.

To exit, type `quit` at the `gnuplot>` prompt. When repeating a graph several times, it may be useful to use the scripting capabilities of `gnuplot`. To do this, put the gnuplot commands in a text file (using your favorite editor, such as `vi` or `emacs`) and add "`pause -1`" as the last line.

```
set ylabel 'Concentration'
set xlabel 'Time'
plot 'data.dat', 'data.dat' using 1:3
pause -1
```

If you call this file `mygraph.ins`, then use this filename as an argument to the command which begins the program:

```
% gnuplot mygraph.ins
```

This instructs `gnuplot` to plot the data and wait for keyboard input before exiting. If you press the "Return" key, the pop–up window disappears and `gnuplot` exits, leaving you back at the UNIX prompt.

You must be running MIT X Windows or OpenWindows for this example to work. For information about running these windowing systems, see your ECN site specialist. These windowing systems must be run on a Sun console or special graphics terminal. If you do not have access to these, then hardcopy output is available.

5.2.2 EEPIC

EEPIC is an extension of LaTeX's `picture` environment. `gnuplot` has a terminal driver which allows you to write EEPIC code. The above example is easily modified to do this:

```
set ylabel 'Concentration'
set xlabel 'Time'
set term eeppic
set out 'mygraph.tex'
plot 'data.dat', 'data.dat' using 1:3
```

Saving this as `mygraph.ins`, the command

```
% gnuplot mygraph.ins
```

now saves the graph in the file `mygraph.tex`. This file may be included in your LaTeX document by adding the the `epic` and `eepic` style options:

```
\documentstyle[epic,eepic]{article}
```

Then, you can simply `\input` the file into a figure environment:

```
\begin{figure}[hbtp]
\begin{center}
\input mygraph
\end{center}
\caption{Example output from {\tt gnuplot}.}
\end{figure}
```

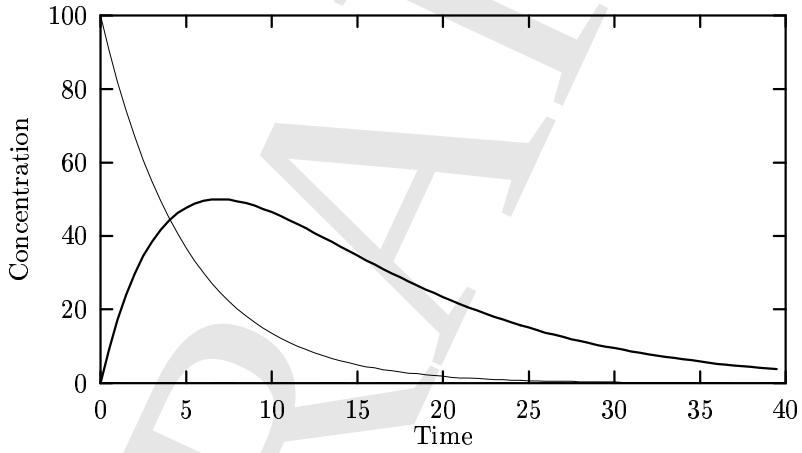

Figure 5.1 Example output from `gnuplot`.

When using the `eepic` (or `latex`, `pictex`, etc) terminal driver, not only will your fonts be the same, but you can take advantage of LaTeX's powerful math capabilities. For example, if you wanted to use the following equation as your `xlabel`:

$$N_{Re,n} = \frac{N^{2-n} D_a^2 \rho}{K \left(du/dy\right)_{av}^{n-1}}$$

you would tell `gnuplot` to

```
set xlabel '$N_{Re,n} = \frac{N^{2-n}D_a^2\rho} ... $'
```

A manual for `gnuplot` is available via anonymous ftp from `dartmouth.edu` (ask your site specialist about anonymous ftp).

5.3 Xfig

`xfig` is an interactive drawing tool that runs under MIT X Windows or `openwin`. It allows you to draw many different kinds of objects, including arcs, circles, splines, ellipses, polygons, and boxes. Drawings are saved in a format called "Fig." It has the same font advantages as `gnuplot` (i.e., can do LaTeX math) and can export files in the following formats: `eepic`, LaTeX `picture`, PostScript, PiCTeX, and others. Each of these can be used in LaTeX. The program is menu driven and its usage is fairly straightforward.

It should be noted that `gnuplot` has a terminal driver called `fig`. Using this terminal driver, a `fig` file of your graph can be produced, which can later be edited and annotated using `xfig`.

5.4 Rotation

To be quite honest, a minor detail was left out in the instructions producing figure 5.1. The y-axis label in this figure is rotated, which is not something that `gnuplot`'s `eepic` terminal driver does by default. This was accompished by using a dirty trick that relies on `dvips`. Another style option was used:

```
\documentstyle[...,rotate,...]{...}
```

and the output of `gnuplot` (saved in `mygraph.tex`) was modified. The line that read

```
\put(45,472){\makebox(0,0)[l]{\shortstack{Concentration}}}
```

was changed to

```
\put(125,472){\makebox(0,0)[c]{\rotate{Concentration}}}
```

This style option may be useful for other purposes, like rotating large tables so that they appear in landscape mode.

6. USING THE PUTHESIS STYLE

You can get a copy of the LaTeX style for creating a Purdue thesis by executing the following command on ECN:

```
tar xvf /usr/unsup/lib/tex/inputs/PUthesis.tar
```

This will create a directory with the style files (`puthesis.sty`, `puthe11.sty`, and `puthe12.sty`) and a few template files to get you started.

Because the size of theses, it usually prudent to keep each chapter (or section) in a separate file. These files are then `\included` in a *root* file. It is the *root* file that you will run LaTeX on.

The `\documentstyle` command is used to tell LaTeX that you will be using the `puthesis` document style and is the first command in your root file. Style options, such as `11pt` or `12pt` are specified here:

```
\documentstyle[12pt]{puthesis}
```

The area immediately proceding this command is called the "preamble" and is used for things like defining macros and declaring the page style. A special page style is available with the `puthesis` style:

```
\pagestyle{thesisdraft}
```

Using this page style makes the words "Draft: Do Not Distribute" appear at the bottom of every page along with the date and time that the file was LaTeXed and the name of the input file.

For draft copies, you may also want to used boldface type for chapter and section headings. This is easily accomplished by putting the following command in the preamble:

```
\renewcommand{\bface}{\bf}
```

Another draft feature in a box outlining each page (showing the margins) and the word "DRAFT" written lightly across the page. The can be accomplished by putting the following command in the preamble:

```
\draftmargins
```

This margins drawn by this command are very useful for useful for pointing out "Overfull `\hbox`'es." Finally, another useful feature for draft copies is offered by `psfig` (if you are including PostScript figures). The command:

```
\psdraft
```

shows the bounding box that the figure would occupy (instead of actually including the figure). For slower PostScript printers (e.g., Apple LaserWriters), this will decreased the time required to get a printout (since many PostScript figures require a lot of processing time). This also decreases the amount of toner used, which makes site specialists happy.

The next usual command is `\begin{document}`. The following example is part of the root file used for this manual.

```
\begin{document}
\bibliographystyle{plain}
\include{prelude}
\include{intro}
\include{essential}
\include{bibtex}
\include{math}
\include{usage}
\include{help}
\include{conslusion}
\bibliography{example}
\include{app}
\end{document}
```

6.1 Prelude

After the `\begin{document}` command comes the preliminary information found in theses. In this manual, this information is kept in `prelude.tex` (see above). You will want to tell LaTeX to number these pages with lowercase roman numerals.

```
\clearpage\pagenumbering{roman}
```

Next comes your thesis title, your name, date of graduation, and degree.

```
\title{How to \TeX\ a Thesis}
\author{James Darrell McCauley}
\date{March 1993}
\degree{Master \TeX{nician}}
%\bypass
%\prelim
```

If the degree is not specified, "Doctor of Philosophy" is assumed. For a masters bypass report, specify `\bypass`. For a preliminary report, specify `\prelim`. Once the above are defined, use `\maketitle` to generate the titlepage.

```
\maketitle
```

Next comes the dedication and acknowledgements:

```
\begin{dedication}
To all those who ...
\end{dedication}

\begin{acknowledgments}
I'd like to thank ...
\end{acknowledgments}
```

You must tell LaTeX to generate a table of contents, a list of tables, and a list of figures and to begin numbering pages with arabic numerals:

```
\tableofcontents
\listoftables
\listoffigures
\clearpage\pagenumbering{arabic}
```

This concludes the preliminary section of your thesis.

At the beginning of `intro.tex`, there is the following command:

```
\chapter{Theses at Purdue}
```

Following that is the text of the chapter (note that the argument to `\chapter` is automagically changed to uppercase in the output). The body of your thesis is separated by sectioning commands like `\chapter`. The heiarchy of sectioning commands is:

```
\chapter{}  \section{}   \subsection{}    \subsubsection{}
            \paragraph{} \subparagraph{}
```

Remember the basic rules of outlining that you picked up in grammar school: "you cannot have an 'A' if you do not have a 'B.'" Take care to have at least two `\sections` if you use this comand; have at least two `\subsections` if you use `\subsection`, etc.

6.3 Bibliography or References

As a final note, the default title for the references chapter is "LIST OF REFERENCES." Since some people prefer "BIBLIOGRAPHY", the command `\altbibtitle` has been added to change the chapter title.

6.4 Appendices

Two commands are available which suppress the writing of auxiliary information (to the `lot` and `lof` files). They are:

```
\noappendixtables   % Turn off Appendix header in LIST OF TABLES
\noappendixfigures  % Turn off Appendix header in LIST OF FIGURES
```

These commands are used before you begin the appendix chapter.

There are two environments for doing the appendix chapter: `appendix` and `appendices`. If you have only one chapter in the appendix, as does this manual, use the `appendix` environment. Otherwise, use the `appendices` environment.

```
\begin{appendix}
\chapter{\BibTeX\ Entries}
\input{entries}
\end{appendix}
```

The difference between these two environments is the way that the chapter header is created and how this is listed in the table of contents.

6.5 Fonts

Many find it much more convenient to use `\bf` and `\tt` fonts for initial version. For final versions, you need not edit them out. Simply redefine the macros to do nothing:

```
\let\bf=\relax
\let\tt=\relax
\let\sf=\relax
\let\sc=\relax
```

in the preamble of the thesis. You may want to conditionally redefine `\em` and `\it`, though careful usage of these would be much easier.

7. WHERE TO GET HELP

As stated in Chapter 1, TEX is unsupported software on ECN, so it's best if you are a self–starter. ECN TEX Support consists of individuals who have volunteered their time and are not paid by ECN. However, there are places to get help:

- have a friend help you
- ask the USENET newsgroup `purdue.ecn.unsup`
- send mail to `texsuprt@ecn.purdue.edu` (ECN TEX Support)
- read the USENET newsgroup `comp.text.tex`

There is another way to gain some insight, but unless you have a propensity for programming, it may be more confusing than helpful (NBA—No Babies Allowed). Simply, read the LATEX source. LATEX, or `lplain`, is built upon `plain` TEX. The files `plain.tex` and `latex.tex` may be found in the directory

`/usr/unsup/lib/tex/inputs.`

You should be able to locate the Purdue Thesis Style (`puthesis.sty`) and various style options. Realistically, these files are not very difficult to comprehend—they just take a little study (typically, I do not use the manuals [2, 3] but rather go straight to the TEX source when answering user's questions).

General information can be found in `/home/harbor2/texsuprt/TeX/guides` on Suns. The Postscript files can be printed or previewed with `ghostview` (if you are using X11 or openwin on Suns). The files `local_guide*.ps` describe use of TEX and LATEX on ECN, including how to include PostScript figures in your documents.

If you cannot access this information, send electronic mail to ECN TEX Support (`texsuprt@ecn.purdue.edu`).

Finally, there are the references listed at the end of this manual. The LaTeX manual [3] is sufficient for beginners. Unless you either have some extra money lying around or think that you will be doing some serious TeXing (which I encourage), *The TeXbook* [2] may be a reference that you can safely avoid.

8. CONCLUSION

The style files described in Chapter 6 should take you a long way in formatting your thesis. However, at the present, there are a few details left uncovered:

1. multi-volume theses,
2. inclusion of a paper for publication (required by some departments?),
3. facing page captions for tables and figures,
4. generation of the thesis acceptance form, and
5. theses in foreign languages (for which TEX is very adept).

Any other deficiencies should be reported to ECN TEX Support.

LIST OF REFERENCES

LIST OF REFERENCES

[1] Oliver Dunn, Jane Ganfield, Page Karling, Ellen DeNelsky, Constance Aveni, Helen Q. Schroyer, Keith Dowden, and Martha K. Becker. *A Manual for the Preparation of Graduate Theses.* Graduate School's Committe on Thesis and Publications, The Graduate School, Purdue University, West Lafayette, IN, 4th revised edition, 1992.

[2] Donald E. Knuth. *The TEXbook.* Addison–Wesley Pub. Co., Reading, MA, 1984.

[3] Leslie Lamport. LATEX*: A Document Preparation System.* Addison–Wesley Pub. Co., Reading, MA, 1986.

APPENDIX

APPENDIX
BibTeX ENTRIES

The following shows the fields required in all types of BibTeX entries. Fields with OPT prefixed are optional (the three letters OPT should be not be used). If an optional field is not used, the entire field can be deleted.

```
@Unpublished{,
  author =       "",
  title =        "",
  note =         "",
  OPTyear =      "",
  OPTmonth =     ""
}

@TechReport{,
  author =       "",
  title =        "",
  institution =          "",
  year =         "",
  OPTtype =      "",
  OPTnumber =    "",
  OPTaddress =   "",
  OPTmonth =     "",
  OPTnote =      ""
}

@Proceedings{,
  title =        "",
  year =         "",
  OPTeditor =    "",
  OPTpublisher =         "",
  OPTorganization =      "",
  OPTaddress =   "",
  OPTmonth =     "",
  OPTnote =      ""
}
```

```
@Manual{,
  title =        "",
  OPTauthor =    "",
  OPTorganization =      "",
  OPTaddress =   "",
  OPTedition =   "",
  OPTyear =      "",
  OPTmonth =     "",
  OPTnote =      ""
}

@InProceedings{,
  author =       "",
  title =        "",
  booktitle =    "",
  year =         "",
  OPTeditor =    "",
  OPTpages =     "",
  OPTorganization =      "",
  OPTpublisher =         "",
  OPTaddress =   "",
  OPTmonth =     "",
  OPTnote =      ""
}
```

```
@PhDThesis{,
  author =       "",
  title =        "",
  school =       "",
  year =         "",
  OPTaddress =   "",
  OPTmonth =     "",
  OPTnote =      ""
}
```

```
@Misc{,
  OPTauthor =    "",
  OPTtitle =     "",
  OPThowpublished =      "",
  OPTyear =      "",
  OPTmonth =     "",
  OPTnote =      ""
}
```

```
@MastersThesis{,
  author =       "",
  title =        "",
  school =       "",
  year =         "",
  OPTaddress =   "",
  OPTmonth =     "",
  OPTnote =      ""
}
```

```
@InCollection{,
  author =       "",
  title =        "",
  booktitle =    "",
  publisher =    "",
  year =         "",
  OPTeditor =    "",
  OPTchapter =   "",
  OPTpages =     "",
  OPTaddress =   "",
  OPTmonth =     "",
  OPTnote =      "" }
```

```
 @InBook{,
  author =       "",
  title =        "",
  chapter =      "",
  publisher =    "",
  year =         "",
  OPTeditor =    "",
  OPTpages =     "",
  OPTvolume =    "",
  OPTseries =    "",
  OPTaddress =   "",
  OPTedition =   "",
  OPTmonth =     "",
  OPTnote =      ""
}
```

```
@Article{,
  author =       "",
  title =        "",
  journal =      "",
  year =         "",
  OPTvolume =    "",
  OPTnumber =    "",
  OPTpages =     "",
  OPTmonth =     "",
  OPTnote =      ""
}
```