

Advanced L^AT_EX

Tim Love

August 30, 2006

This document follows on from the *Word processing using L^AT_EX*¹ document. It describes the features of L^AT_EX that people at CUED are most likely to use. Further information is available from the LaTeX help page² and in the books available for loan from the operators in the DPO. Comments and bug reports to Tim Love (tpl@eng.cam.ac.uk).

Contents

1 L^AT_EX Concepts	2
1.1 Environments and commands	2
1.2 Classes and packages	3
1.3 Errors	4
1.4 Files created	5
1.5 How to use L ^A T _E X at CUED	5
2 Document structure	5
2.1 Counters and Length parameters	5
2.2 Document and page organisation	6
2.3 Pagebreaks, footnotes, etc	8
3 Color and Fonts	9
3.1 Colored text	9
3.2 Special characters	10
3.3 Font Sizes	11
3.4 Font Types	11
3.5 Postscript Fonts	11
3.6 Font attributes	12
3.7 Selection commands	13

¹http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/latex_basic/latex_basic.html

²http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/LaTeX_intro.html

Copyright ©1999-2006 by T.P. Love. This document may be copied freely for the purposes of education and non-commercial research. Cambridge University Engineering Department, Cambridge CB2 1PZ, England.

4	Environments	14
4.1	Alignments	14
4.2	Listing Items	14
4.3	Tabular	16
4.4	Array	17
4.5	Pictures	17
4.6	Maths	18
4.7	Figures	18
4.8	Tabbing	19
4.9	Verbatim	19
4.10	Quote, abstract	20
4.11	Letter	20
4.12	Curriculum Vitae	20
5	Customising	20
5.1	Macros	20
5.2	Modifications	20
5.3	New Commands	21
5.4	Packages	21
5.5	An Example	22
6	More Information	22

1 L^AT_EX Concepts

Writing a L^AT_EX document is rather like writing a program. This makes using L^AT_EX more difficult in some respects than using a word processor, but there are advantages too. For instance creating a table of contents is trivial. Beginners often use unnecessary ‘\’ sequences and write ‘{\large \textbf{2.1 Method}}\’ when ‘\subsection{Method}’ would be much better. Users who think they know more about typesetting than L^AT_EX (those who, for example, like underlining) will waste a lot of time too.

If you find you’re adding many L^AT_EX constructions you might save time in the long run by familiarising yourself with the packages (add-ons) described in the packages³ section of the online L^AT_EX page. You could even write your own macros and add-ons.

1.1 Environments and commands

L^AT_EX is a macro-package for T_EX which has many preset *environments* where much of the setting up that T_EX users have to do explicitly is done for you. An environment has the form

```
\begin{environment name}
```

³http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/LaTeX_intro.html#Packages

```
\end{environment name}
```

L^AT_EX also has commands which affect the formatting of the document. Their arguments are given in braces. For example,

```
\textit{This is much more important} than this.
```

produces as output *This is much more important* than this.

The related `\itshape` command doesn't take an argument. It affects all the subsequent text in the environment where it's used.

L^AT_EX tries to enforce the idea that the visual appearance of the document (use of fonts, indentation, etc) should derive from the logical structure of the document (*i.e.* rather than manually putting the section titles into bold, you should let the `\section` command do it). Resisting this philosophy can lead to extra (usually unnecessary) work.

L^AT_EX is expandable. Many macros can be loaded in to provide added features. You can also create your own commands and environments. Commands can take arguments that modify their action

- Some commands have a `*-form`, a variant on the standard command that you get by adding a `*` to the command name.
- Mandatory arguments are enclosed in `{ }` braces
- Optional arguments are enclosed in `[]` brackets.

1.2 Classes and packages

At the top of your file you will have a line something like

```
\documentclass[12pt]{article}
```

which determines the font size and document class – the type of document you're writing. Each class has an associated `*.cls` file in the system directories which is read in at start-up time. Other options accepted by `article` include `10pt`, `11pt` and `twocolumn`.

Then you'll probably need to load in extra macros with the `\usepackage` command. Each package has an associated `*.sty` file in the system directories. The packages inherit the options from the `\documentclass` line and can be given others of their own. *E.g.*

```
\usepackage[dvips]{graphicx}
\usepackage[dvips]{color}
```

tells L^AT_EX that you want to use the extra `graphicx` and `color` macros, and that you are going to use `dvips` to convert the resulting file to postscript. You can shorten this to

```
\usepackage[dvips]{graphicx,color}
```

1.3 Errors

Errors can be reported either from the L^AT_EX phase (in which case it is reported as such) or the lower level T_EX phase. Don't be too put off by the half-digested text displayed. A line number is reported which usually helps to detect the error. Typing 'h' sometimes produces helpful diagnostics. The most common errors reported are

- a command misspelt
- a mismatched brace
- improper use of special characters
- forgetting to have the appropriate `\usepackage{...}` lines.
- an error like this

```
Overfull \hbox (15.42563pt too wide) in paragraph at lines
285--288 \OT1/pp1/m/n/10 You can cre-ate ver-tical space
between lines or ho-ri-zontal space between
```

means that lines 285 to 288 are producing a text line about 15 pts (about 5mm) too wide. `\OT1/pp1/m/n/10` is the specification of the font used. When T_EX does right and left alignment, it works out how much space it needs to leave between words and where to hyphenate words if necessary. But the amount of space it's prepared to leave has to fall within a certain range and it will only split words in certain places (shown in the error message by a hyphen). If these restrictions mean that L^AT_EX can't produce a satisfactory line, it will produce as much of the line as it can. A re-phrasing of the offending sentence will usually solve the problem. Another thing you might try is to control the way a troublesome word is hyphenated using something like

```
\hyphenation{furthermore fur-ther-more}
```

at the top of your file. If all else fails, use `{\sloppy...}` to enclose the offending text.

- L^AT_EX issues a `*`.
This means L^AT_EX needs more input. It probably means you've missed out an `\end{document}`, but if not you may be able to get L^AT_EX to continue processing as best it can by typing `<Return>`.
- `!` Argument of `\label` has an extra `}`
If you're sure you haven't left out a left brace, then maybe you need to *protect* the inmost (fragile) command. For example, in some older versions of L^AT_EX `\label` is fragile which causes a problem in sequences like
`\caption{Picture\label{margin}},`
so the safer construction
`\caption{Picture\protect\label{margin}}`
has to be used.

1.4 Files created

More than just the *.dvi file may be produced. Don't worry about them – except perhaps for the log file you won't need to look at them.

- *.aux - cross references, etc
- *.toc - created by `\tableofcontents`
- *.lof - created by `\listoffigures`
- *.lot - created by `\listoftables`
- *.log - a copy of the diagnostic output that usually comes out on the screen.

Some common errors can be found using the `lacheck` program on the \LaTeX file.

1.5 How to use \LaTeX at CUED

From the command line – Edit the latex source code using any editor you want, then run `latex` at least twice, you should be able to preview your *.dvi document using `xdvi` and print it using `lp`, but if postscript is implicated at all (if you load in graphics, use postscript fonts, scale, use color, or rotate) then you should convert your *.dvi document to postscript. A typical sequence of commands to process `doc.tex` would be

```
latex doc
latex doc
dvips doc.dvi -o doc.ps
ghostview doc.ps
lp doc.ps
```

If you want a PDF document, then use `ps2pdf` to convert the postscript file, or use `pdflatex` instead of `latex`

Using a front-end – we have several latex-based programs ranging from `xlatex` (which just has buttons to replace command-line operations) to `Kile` (which includes an editor and many extra facilities).

2 Document structure

2.1 Counters and Length parameters

- **Counters** :- \LaTeX maintains many counter variables (e.g. page, part, equation, footnote, chapter, paragraph, section, subsection, subsubsection, enumi, etc). You can set these counters yourself. Some examples:-

```
\setcounter{page}{0}
\addtocounter{chapter}{2}
```

- **Length Parameters** :- \LaTeX accepts the following units of length: in, cm, mm, pt (there are 72.27 pts to an inch), em (width of an M), ex (height of an x). These units can be used to set the values of length variables using `\setlength`. For example,

```
\setlength{\parindent}{0in}
```

sets to zero the amount by which the first line of a paragraph is indented. Other useful length parameters are:-

parskip:- determines the gap between paragraphs.

baselineskip:- determines the usual distance from the bottom of one line to the bottom of the next. You can adjust this to produce double spacing, but a better way, which takes a stretch *factor* as argument is to use `\linespread`. For instance *before* `\begin{document}` you can do

```
\linespread{1.6}
```

to get doublespacing through your document. The `setspace` package offers more options.

2.2 Document and page organisation

- **Document classes**: The standard classes are `article`, `book`, `letter`, `report`, `slides`. The differences between these are often minor. A book can have chapters. In a report sections begin at 0 whereas in an article they begin at 1. Just about all our handouts are articles.
- **Big Documents** :- It's best to split your document into smaller files and have a master file looking like this

```
\documentclass[dvips,12pt]{book}
\usepackage{a4,color,graphics,palatino,fancyhdr}
\begin{document}
\pagestyle{empty}
\tableofcontents
\listoffigures
\pagestyle{fancy}
\input{chapter1}
\input{chapter2}
\input{fig2}
\input{chapter3}
\input{chapter4}
\appendix
\input{appendices}
\end{document}
```

on which you can run `latex` just as if the master file contained all the text of `chapter1.tex` etc. The advantage of this is that once you have a chapter correct, you can comment out the corresponding 'input' line and avoid unnecessary processing. Remember to take out the `\begin{document}` and `\end{document}` lines from the component files.

- **Page Size** :- You can choose the margin sizes yourself by changing the following dimensions *before* the `\begin{document}` line.

```
\setlength{\topmargin}{-0.4in}
\setlength{\topskip}{0.3in} % between header and text
\setlength{\textheight}{9.5in} % height of main text
\setlength{\textwidth}{6in} % width of text
\setlength{\oddsidemargin}{0.75in} % odd page left margin
\setlength{\evensidemargin}{0.75in} % even page left margin
```

Note that a margin width of 0cm gives you a margin 4cm wide. Rather than set absolute sizes you can modify the default sizes using commands like the following –

```
\addtolength{\evensidemargin}{-1cm}
\addtolength{\oddsidemargin}{-1cm}
\addtolength{\textwidth}{2cm}
```

Better support for control of page layout is provided by the `geometry`⁴ package. To see the current values of these dimensions, use the `layout` package, which defines a `\layout` command.

For A3 output, add `\usepackage{a3}` to your latex source file (`foo.tex` say), run latex on it, convert the resulting file to Postscript using `"dvips -t a3 foo.dvi -ofoo.ps"` then print using `"lp -oa3 foo.ps"`.

- **Title Pages** :- The title page of this document was created by the following \LaTeX commands.

```
\title{Advanced \LaTeX}
\author{Tim Love}
\date{\today}
\maketitle
```

- **Contents** :- Use `\tableofcontents` to create a contents list at that point in the document. \LaTeX will pick out the sections, subsections, etc for you. You'll have to run \LaTeX at least twice though.
- **Page numbers and Headings** :- These are determined by the argument given in `pagestyle{}`.

plain :- the page numbers are put at the bottom of the page. The top of the page is empty. This is the default mode.

empty :- this suppresses page numbering altogether, except on the title page if you're using `\maketitle`. The workaround in this case is to do

```
\maketitle
\thispagestyle{empty}
```

with no gap between the two lines.

headings :- this puts the page numbers at the top and adds a header whose contents depend on the document style.

`fancyhdr`⁵ is a popular package that adds useful page headers when the command `\pagestyle{fancy}` is used. This handout uses it.

Long section titles can cause trouble in headers. The section commands let you specify an extra, shorter title for use in the header and contents page. Section 2.3 was specified as follows

```
\subsection[Pagebreaks, footnotes, etc]
{Pagebreaks, space, footnotes, references, boxes, etc}
```

⁴<http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/geometry.dvi>

⁵<http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/fancyhdr.dvi>

- **Sectioning** :- To start a chapter called **Life** in a book, just use `\chapter{Life}`. Similar commands to start a part, section or subsection also exist in most document classes (articles don't have chapters or parts though).

If you use the *-form of the command then the sections will not be numbered, neither will it appear in the table of contents.

A title will only be numbered if its 'depth' isn't more than `secnumdepth` and will only appear in the contents page if the 'depth' isn't more than the value of `tocdepth`. So, for example, doing

```
\setcounter{tocdepth}{2}
\setcounter{secnumdepth}{3}
```

will cause section 1 . 4 . 3 to be numbered, but it won't appear in the contents.

2.3 Pagebreaks, space, footnotes, references, boxes, etc

- **Page Breaks** :- you can force a page break using `\newpage`.
- **Preventing line breaks** :- If there's a word that you don't want broken, put it in an `mbox`. *E.g.*,

```
One shouldn't try to break up \mbox\emph{relationships}
```

- **Space** :-

You can create vertical space `\vspace{.5cm}` between lines or horizontal space `\hspace{1.5cm}` between words. The arguments to these commands can be negative.

You can create vertical space between lines or horizontal space between words.

The arguments to these commands can be negative. `\vspace*` will create space even at the top of a page. It's sometimes useful to create stretchable space. The following creates space that pushes the letters to the edge of the page

```
\noindent A\hspace{\stretch{1}} B\
C\hspace{\stretch{1}} D
```

```
A B
C D
```

- **Footnotes** :-

```
Do them like this\footnote{I told you so.}
```

Do them like this⁶ The footnotes are numbered by default. If you want to use symbols (stars, daggers etc) then you need to redefine how the footnote counter is displayed.

```
\def\thefootnote{\fnsymbol{footnote}}
```

⁶I told you so.

- **Margin notes :-**

Do them like this `\marginpar{margin note}`

Do them like this

margin note

- **Cross References :-** At the end of this source file is the line `\label{THE_END}`.

The last page (page `\pageref{THE_END}`) is in section `\ref{THE_END}` produces The last page (page 23) is in section 6. You'll have to run \LaTeX at least twice to pick up forward references like these.

- **Boxed Text :-** For short pieces of text, use `\fbox`

Help. I'm `\fbox{trapped}`

Help. I'm trapped

- **Comments :-** Anything to the right of a `%` character is ignored by \LaTeX .

- **Rules :-** The `\rule` takes 2 arguments; a width and a height, so `\rule{\textwidth}{1pt}` produces

3 Color and Fonts

3.1 Colored text

Commands that control foreground and background colors need

```
\usepackage[dvips]{color}
```

after the `\documentclass` line but before `\begin{document}`.

- `\textcolor{colorname}{text}` writes text in a color which can be specified by name (black, white, red, green, blue or a color name you've defined), RGB components, or grayscale.
- `\colorbox{colorname}{text}` writes text in a box with a colored background.
- `\fcolorbox{framecolor}{boxcolor}{text}` writes text in a colored frame.
- `\pagecolor{colorname}` sets the color of the page's background.
- `\definecolor{colorname}{color specification}` lets you define new color names.

```
\definecolor{gold}{rgb}{0.85,.66,0}
```

This is in `\textcolor{red}{red}` and this box is `\colorbox{gold}{gold}`.

Text color can be set using RGB values

```
(\textcolor[rgb]{0,1,0}{like so}), or \textcolor[gray]{0.2}{shades}
\textcolor[gray]{0.5}{of} \textcolor[gray]{0.8}{grey}.
```

produces

This is in **red** and this box is gold. Text color can be set using RGB values (**like so**), or shades of grey.

3.2 Special characters

† is created by `\dag`, ‡ by `\ddag`, § by `\S`, ¶ by `\P`, £ by `\pounds`, ö by `\" {ö}`, © by `\copyright`. Many others are available in the `math` environment, including all the lower case greek letters and most of the upper case ones. If you only want to use a few characters you can bracket the symbols using `$` and `$` rather than `\begin{math}` and `\end{math}`. You can put a slash through any of these characters by prefacing them with `\not`

\sqrt{i}	<code>\sqrt{i}</code>	$\sqrt[5]{x+iy}$	<code>\sqrt[5]{x+iy}</code>	...	<code>\ldots</code>
...	<code>\cdots</code>	:	<code>\vdots</code>	⋯	<code>\ddots</code>
α	<code>\alpha</code>	β	<code>\beta</code>	γ	<code>\gamma</code>
δ	<code>\delta</code>	ω	<code>\omega</code>	Γ	<code>\Gamma</code>
Θ	<code>\Theta</code>	Ω	<code>\Omega</code>	\pm	<code>\pm</code>
\mp	<code>\mp</code>	\times	<code>\times</code>	\div	<code>\div</code>
$*$	<code>\ast</code>	$*$	<code>\star</code>	\circ	<code>\circ</code>
•	<code>\bullet</code>	•	<code>\cdot</code>	\cap	<code>\cap</code>
\bigcap	<code>\bigcap</code>	\cup	<code>\cup</code>	\bigcup	<code>\bigcup</code>
\uplus	<code>\uplus</code>	\biguplus	<code>\biguplus</code>	\sqcup	<code>\sqcup</code>
\sqcup	<code>\sqcup</code>	\bigsqcup	<code>\bigsqcup</code>	\vee	<code>\vee</code>
\bigvee	<code>\bigvee</code>	\wedge	<code>\wedge</code>	\bigwedge	<code>\bigwedge</code>
\setminus	<code>\setminus</code>	\wr	<code>\wr</code>	\diamond	<code>\diamond</code>
\bigtriangleup	<code>\bigtriangleup</code>	\bigtriangledown	<code>\bigtriangledown</code>	\triangleleft	<code>\triangleleft</code>
\triangleright	<code>\triangleright</code>	\oplus	<code>\oplus</code>	\otimes	<code>\otimes</code>
\bigoplus	<code>\bigoplus</code>	\ominus	<code>\ominus</code>	\oslash	<code>\oslash</code>
\bigotimes	<code>\bigotimes</code>	\oslash	<code>\oslash</code>	\odot	<code>\odot</code>
\bigodot	<code>\bigodot</code>	\bigcirc	<code>\bigcirc</code>	\amalg	<code>\amalg</code>
\leq	<code>\leq</code>	\prec	<code>\prec</code>	\preceq	<code>\preceq</code>
\ll	<code>\ll</code>	\subset	<code>\subset</code>	\subseteq	<code>\subseteq</code>
\in	<code>\in</code>	\vdash	<code>\vdash</code>	\succ	<code>\succ</code>
\geq	<code>\geq</code>	\succ	<code>\succ</code>	\succeq	<code>\succeq</code>
\gg	<code>\gg</code>	\supset	<code>\supset</code>	\supseteq	<code>\supseteq</code>
\sqsupseteq	<code>\sqsupseteq</code>	\ni	<code>\ni</code>	\dashv	<code>\dashv</code>
\equiv	<code>\equiv</code>	\sim	<code>\sim</code>	\simeq	<code>\simeq</code>
\asymp	<code>\asymp</code>	\approx	<code>\approx</code>	\cong	<code>\cong</code>
\neq	<code>\neq</code>	\doteq	<code>\doteq</code>	\propto	<code>\propto</code>
\models	<code>\models</code>	\perp	<code>\perp</code>	\mid	<code>\mid</code>
\parallel	<code>\parallel</code>	\bowtie	<code>\bowtie</code>	\smile	<code>\smile</code>
\frown	<code>\frown</code>	\leftarrow	<code>\leftarrow</code>	\Leftarrow	<code>\Leftarrow</code>
\rightarrow	<code>\rightarrow</code>	\Rightarrow	<code>\Rightarrow</code>	\leftrightarrow	<code>\leftrightarrow</code>
\Leftrightarrow	<code>\Leftrightarrow</code>	\mapsto	<code>\mapsto</code>	\hookrightarrow	<code>\hookrightarrow</code>
\leftharpoonup	<code>\leftharpoonup</code>	\leftharpoondown	<code>\leftharpoondown</code>	\rightharpoons	<code>\rightharpoons</code>
\longleftarrow	<code>\longleftarrow</code>	\Leftrightarrow	<code>\Leftrightarrow</code>	\longrightarrow	<code>\longrightarrow</code>
\Longrightarrow	<code>\Longrightarrow</code>	\longleftrightarrow	<code>\longleftrightarrow</code>	\Leftrightarrow	<code>\Leftrightarrow</code>
\longmapsto	<code>\longmapsto</code>	\hookrightarrow	<code>\hookrightarrow</code>	\rightarrow	<code>\rightarrow</code>
\rightharpoondown	<code>\rightharpoondown</code>	\uparrow	<code>\uparrow</code>	\Uparrow	<code>\Uparrow</code>
\downarrow	<code>\downarrow</code>	\Downarrow	<code>\Downarrow</code>	\Downarrow	<code>\Downarrow</code>
\nearrow	<code>\nearrow</code>	\searrow	<code>\searrow</code>	\swarrow	<code>\swarrow</code>
\nwarrow	<code>\nwarrow</code>	\aleph	<code>\aleph</code>	\hbar	<code>\hbar</code>
\imath	<code>\imath</code>	\jmath	<code>\jmath</code>	ℓ	<code>\ell</code>
\wp	<code>\wp</code>	\Re	<code>\Re</code>	\Im	<code>\Im</code>
\prime	<code>\prime</code>	\emptyset	<code>\emptyset</code>	∇	<code>\nabla</code>
\surd	<code>\surd</code>	\top	<code>\top</code>	\perp	<code>\perp</code>
\parallel	<code>\parallel</code>	\angle	<code>\angle</code>	\forall	<code>\forall</code>
\exists	<code>\exists</code>	\neg	<code>\neg</code>	\flat	<code>\flat</code>
\natural	<code>\natural</code>	\sharp	<code>\sharp</code>	\backslash	<code>\backslash</code>
∂	<code>\partial</code>	∞	<code>\infty</code>	\triangle	<code>\triangle</code>

♣	<code>\clubsuit</code>	◇	<code>\diamondsuit</code>	♥	<code>\heartsuit</code>
♠	<code>\spadesuit</code>	∑	<code>\sum</code>	∏	<code>\prod</code>
∏	<code>\coprod</code>	∫	<code>\int</code>	∫	<code>\oint</code>

3.3 Font Sizes

These are the available sizes.

tiny scriptsize footnotesize small normalsize
 large Large LARGE huge Huge

If, for example, you want to use the smallest size, do

```
{\tiny ... }
```

If Huge isn't big enough for you, you can scale a *postscript* font up using the commands in the `graphicx` package. `\resizebox{!}{5cm}{BIG}` produces

BIG

3.4 Font Types

Independent of size, these font types are at your disposal :- `\textrm` (roman), `\textit` (*italic*), `\textsc` (SMALL CAPS), `\emph` (*emphasis, but note that if you use *emph* within emphasized text, you will get roman text*), `\textsl` (*slanting*), `\texttt` (teletype), `\textbf` (**boldface**), `\textsf` (sans serif). As long as there's no conflict, these commands can be combined so that, for instance, **this is bold sans serif** can be produced by `\textsf{\textbf{this is bold sans serif}}`.

3.5 Postscript Fonts

It is easy to write a document that has postscript fonts. We have package support for helvetica (`helvetic`), utopia, times, optima, newcentury (`newcentu`), palatino and courier. To use palatino, for instance, all you need to do is add

```
\usepackage{palatino}
```

to your file. The `pifont` package has special commands for using the Zapf Dingbats font. `\dingfill{40}` completes the line with the specified symbol ☞ ☞

and `\dingline{36}` draws a whole line of symbols.

☞ ☞

It's a good idea to use a font that's installed in the printer you intend to use. See the table on page 12.

3.6 Font attributes

The commands above should give you sufficient control over fonts. If you don't want to know more at the moment then turn to section 4

Every text font in L^AT_EX has five *attributes*:

encoding This specifies the order that characters appear in the font. The most common values for the font encoding is OT1.

family The name for a collection of fonts, usually grouped under a common name by the font foundry. For example, 'Adobe Times' and Knuth's 'Computer Modern Roman' are font families. There are far too many font families to list them all, but some common ones are:

<i>Internal fontname</i>	<i>Fontname</i>	<i>In ljmr1?</i>
cmr	Computer Modern Roman	No
cmss	Computer Modern Sans	No
cmtt	Computer Modern Typewriter	No
cmm	Computer Modern Math Italic	No
cmsy	Computer Modern Math Symbols	No
cmex	Computer Modern Math Extensions	No
ptm	Adobe Times	Yes
phv	Adobe Helvetica	Yes
pcr	Adobe Courier	Yes
pun	Univers	No
ppl	Palatino	Yes
pagk	AvantGarde-Book	Yes
pagd	AvantGarde-Demi	Yes
pbk	Bookman	Yes
put	Utopia	No
pop	Optima	No
pnc	New Century Schoolbook	Yes
pzd	ZapfDingbats	Yes
rpad	Garamond	No

series How heavy or expanded a font is. For example, 'medium weight', 'narrow' and 'bold extended' are all series. The most common values for the font series are:

m	Medium
b	Bold
bx	Bold extended
sb	Semi-bold
c	Condensed

shape The form of the letters within a font family. For example, 'italic', 'oblique' and 'upright' are all font shapes. The most common values for the font shape are:

n Normal (that is ‘upright’ or ‘roman’)
 it Italic
 sl Slanted (or ‘oblique’)
 sc Caps and small caps

size The design size of the font, for example ‘10pt’.

These five parameters specify every L^AT_EX font, for example:

	L ^A T _E X specification			Font	
OT1	cmr	m	n	10pt	Computer Modern Roman 10pt
OT1	cmss	m	sl	12pt	Computer Modern Sans Oblique 12pt
OML	cmm	m	it	10pt	Computer Modern Math Italic 10pt
T1	ptm	b	it	18pt	Adobe Times Bold Italic 18pt

3.7 Selection commands

There are commands to set attributes one at a time:

Command	Attribute	Value in article class, 10pt
<code>\textrm{..}</code> or <code>\rmfamily</code>	family	cmr
<code>\textsf{..}</code> or <code>\sffamily</code>	family	cmss
<code>\texttt{..}</code> or <code>\ttfamily</code>	family	cmtt
<code>\textmd{..}</code> or <code>\mdseries</code>	series	m
<code>\textbf{..}</code> or <code>\bfseries</code>	series	bx
<code>\textup{..}</code> or <code>\upshape</code>	shape	n
<code>\textit{..}</code> or <code>\itshape</code>	shape	it
<code>\textsl{..}</code> or <code>\slshape</code>	shape	sl
<code>\textsc{..}</code> or <code>\scshape</code>	shape	sc
<code>\tiny</code>	size	5pt
<code>\scriptsize</code>	size	7pt
<code>\footnotesize</code>	size	8pt
<code>\small</code>	size	9pt
<code>\normalsize</code>	size	10pt
<code>\large</code>	size	12pt
<code>\Large</code>	size	14.4pt
<code>\LARGE</code>	size	17.28pt
<code>\huge</code>	size	20.74pt
<code>\Huge</code>	size	24.88pt

The low-level commands used to change font attributes are as follows.

```

\fontencoding{encoding}
\fontfamily{family}
\fontseries{series}
\fontshape{shape}
\fontsize{size}{baselineskip}

```

Each of these commands sets one of the font attributes; `\fontsize` also sets `\baselineskip`. The actual font in use is not altered by these commands, but the current attributes are used to determine which font to use after the next `\selectfont` command.

`\selectfont` selects a text font, based on the current values of the font attributes. There *must* be a `\selectfont` command immediately after any settings of the font parameters by (some of) the five `\font<parameter>` commands, before any following text. For example, it is legal to say:

```
\fontfamily{ptm}\fontseries{b}\selectfont Some text.
```

to select bold Times Roman, but it is *not* legal to say:

```
\fontfamily{ptm} Some \fontseries{b}\selectfont text.
```

```
\usefont{encoding}{family}{series}{shape}
```

is short hand for the equivalent `\font...` commands followed by `\selectfont`.

4 Environments

Some examples of how to use environments are given here.

4.1 Alignments

In these environments `\\` starts a new line.

```
\begin{flushleft}
Some people like to stay firmly\\ on the left whereas others
\end{flushleft}
\begin{flushright}
feel more at home on the right.\\
\end{flushright}
\begin{center}
but most of us prefer to stay dead in the center.
\end{center}
```

Some people like to stay firmly
on the left whereas others

feel much more at home
on the right.

but most of us prefer to stay dead in the center.

4.2 Listing Items

The items can be marked in one of three way:

```
\begin{itemize}
\item just by a bullet, using \texttt{itemize}

\item numbered, using \texttt{enumerate}

\begin{enumerate}
\item one
\item two
\item three
\end{enumerate}
```

```

\item or with a label, using \texttt{description}

\begin{description}
\item[itemize] bullets
\item[enumerate] automatic numbering
\item[description] labelling
\end{description}

\end{itemize}

```

- just by a bullet, using `itemize`
- numbered, using `enumerate`
 1. one
 2. two
 3. three
- or with a label, using `description`
 - itemize** bullets
 - enumerate** automatic numbering
 - description** labelling

The `pifont` package includes a variant of the `itemize` command that will replace the usual 'bullet' by a Zapf Dingbat symbol of your choice

```

\begin{dinglist}{43}
\item First
\item Second
\end{dinglist}

```

- ☞ First
- ☞ Second

and a variant of the `enumerate` command that given an initial Zapf Dingbat symbol will increment the symbol for each item.

```

\begin{dingautolist}{172}
\item First
\item Second
\end{dingautolist}

```

- ① First
- ② Second

4.3 Tabular

Tabular output is supported. When you create the environment you specify how many columns to have and how the contents are to be aligned (use `l`, `c` or `r` to represent each column with either left, center or right alignment) and where you want vertical lines (use `|`). The contents of the columns are separated by a `'&'` and rows by `\\`. Here's a simple example

```
\begin{tabular}{l|c|r}
left & centre & right\\
more left & more centre & more right\\
\end{tabular}
```

left	centre	right
more left	more centre	more right

To draw a full horizontal line, use `\hline` otherwise draw a line across selected columns using `\cline`. The `\multicolumn` command allows items to span columns. It takes as its first argument the number of columns to span. The following, more complicated example shows how to use these facilities.

```
\begin{tabular}{||l|lr||} \hline
\textbf{Veg} & \multicolumn{2}{|c|}{\textbf{Detail}}\\
carrots & per pound & \pounds 0.75 \\
& each & 20p \\
mushrooms & dozen & 86p \\
toadstools & pick your own & free \\
\end{tabular}
```

Veg	Detail	
carrots	per pound	£0.75
	each	20p
mushrooms	dozen	86p
toadstools	pick your own	free

Tables won't continue on the next page if they're too long. The `longtable` or `supertabular` commands are needed to do this. See the `Supertabular`⁷ document for details and examples.

If the text in a column is too wide for the page, \LaTeX won't automatically text-wrap. Using `p{5cm}` instead of `c`, `l` or `r` in the `tabular` line will wrap-around the text in a 5 cm wide column.

There are various packages to assist with table creation. The `array` package adds some helpful features, including the ability to add formatting commands that control a whole column at a time, like so

```
\begin{tabular}{>{\ttfamily}l>{\scshape}c>{\Large}r}
Text & More Text & Large Text\\
Left & Centred & Right
\end{tabular}
```

Text	MORE TEXT	Large Text
Left	CENTRED	Right

⁷<http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/supertab.dvi>

The `rotating`⁸ package is useful if you have a wide table that you want to display in landscape mode. You need to put your table inside `\begin{sidewaystable}` and `\end{sidewaystable}`.

If you want the table to have a caption and *float* (float up the page if it's started right near the foot of a page, for example), use

```
\begin{table}[htbp]
\begin{tabular}...
...
\end{tabular}
\caption{...}
\end{table}
```

See section 4.7 for details.

4.4 Array

The `array` environment (not to be confused with the `array` package) is similar to the `tabular` but must be within a *math* environment. This

```
\begin{math}
\left(
\begin{array}{cccc}
a+b+c & uv & x-y & 27 \\
x+y & w & +z & 363
\end{array}
\right)
\end{math}
```

produces $\left(\begin{array}{cccc} a+b+c & uv & x-y & 27 \\ x+y & w & +z & 363 \end{array} \right)$

4.5 Pictures

\LaTeX has some graphics capabilities. It's much better to import an encapsulated postscript file. See the *\LaTeX Maths and Graphics*⁹ document for more details.

```
\newcounter{cms}
\setlength{\unitlength}{1mm}
\begin{picture}(50,39)
\put(0,7){\makebox(0,0)[bl]{cm}}
\multiput(10,7)(10,0){5}{\addtocounter
{cms}{1}\makebox(0,0)[b]{\arabic{cms}}}
\put(15,20){\circle{6}}
\put(30,20){\circle{6}}
\put(15,20){\circle*{2}}
\put(30,20){\circle*{2}}
\put(10,24){\framebox(25,8){a box}}
\put(10,32){\vector(-2,1){10}}
\multiput(1,0)(1,0){49}{\line(0,1){2,5}}
```

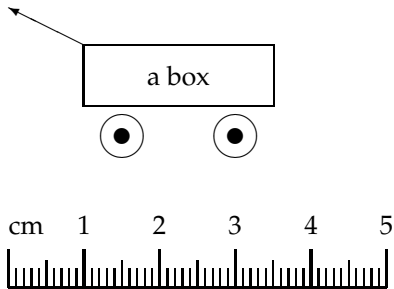
⁸<http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/rotating.ps>

⁹http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/latex_maths+pix/latex_maths+pix.html

```

\multiput(5,0)(10,0){5}{\line(0,1){3,5}}
\thicklines
\put(0,0){\line(1,0){50}}
\multiput(0,0)(10,0){6}{\line(0,1){5}}
\end{picture}

```



4.6 Maths

Maths is dealt with in the *L^AT_EX Maths and Graphics*¹⁰ document. Here are some examples

```

\begin{math}
\lim_{n \rightarrow \infty} x = 0 \ \backslash\
x^{2y} \ \backslash\
x_{2y} \ \backslash\
x^{2y}_{1} \ \backslash\
\frac{x+y}{1 + \frac{1}{n+1}} \ \backslash\
\end{math}

```

produces $\lim_{n \rightarrow \infty} x = 0$

x^{2y}

x_{2y}

x^{2y}_1

$\frac{x+y}{1 + \frac{1}{n+1}}$

4.7 Figures

To include graphics, use the `\includegraphics` command (provided by the `graphicx` package) inside `figure` environment. The arguments to `figure` specify where the space will be made, preferentially

- h here
- t top of page
- b bottom of page
- p on a page with no text

```

\begin{figure}[htbp]
\vspace{1cm}
\caption{1 cm of space}
\end{figure}

```

¹⁰<http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/latex.maths+pix/latex.maths+pix.html>

Figure 1: 1 cm of space

Putting ! as the first argument in the square brackets will encourage L^AT_EX to do what you say, even if the result's sub-optimal.

If you have a label defined in the caption, L^AT_EX may give an error message. \label is a *fragile* command (see the L^AT_EX book for details) so you'll need to do something like

```
...
\caption{1 cm of space\protect\label{EMPTY}}
```

or simply put the \label command after the caption. Note that if you put the \label *before* the caption, the resulting reference will be the section number and not the figure number. See the *L^AT_EX Maths and Graphics*¹¹ document for more details.

4.8 Tabbing

Within this environment tabs can be set by \= and the next tab moved to by using \>.

```
\begin{tabbing}
if \= it's raining\\      % set tab here, after the 'if'
  \> get an umbrella \=\\ % go to the defined tab and set a new one
else\\
  \> get wet \> * next tab is here\\
endif
\end{tabbing}
```

```
if it's raining
  get an umbrella
else
  get wet      * next tab is here
endif
```

4.9 Verbatim

Within this environment things come out unformatted. It's useful for showing examples of typed input and provides a way of printing characters that have a special meaning for L^AT_EX.

```
\begin{verbatim} caret is ^, tilde is ~ and backslash is \
\end{verbatim}
```

produces

```
caret is ^, tilde is ~ and backslash is \
```

If you just want to quote a few characters, use \verb|*quoted text*|. The characters delimiting the quote can be anything as long as they are the same.

¹¹http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/latex_maths+pix/latex_maths+pix.html

4.10 Quote, abstract

These widen the margins and change the font. The `abstract` environment also adds a title.

4.11 Letter

See `letter.tex` in `/export/Examples/LaTeX`

4.12 Curriculum Vitae

See `cv.tex` in `/export/Examples/LaTeX`

5 Customising

5.1 Macros

At the top of the source of this file is

```
\def\xdt{\cal X\!\!\$\texttt{.desktop}}
```

which defines `\xdt` to be `\cal X.desktop`. Using such constructions can make your document much tidier, and saves on typing.

5.2 Modifications

Many of the features of a \LaTeX document are easily customised, but you'll often have to look at the class (`.cls`) files to find out what to do. For example, suppose you wanted to have **References** in a book rather than **Bibliography**. If you look in `book.cls` you'll see

```
\newcommand\bibname{Bibliography}
```

so adding

```
\renewcommand{\bibname}{References}
```

to your file should achieve what you want.

Counters (*e.g.* `figure`) have related commands (*e.g.* `thefigure`) to control their appearance, so they're easy to customise

```
\renewcommand\thefigure{\roman{figure}}
```

produces figure numbers in lower case roman numerals. Longer commands can be adapted too. Remember however, that if the command involves a `@` you have to enclose your changes in `\makeatletter ... \makeatother`. Here's an example that changes the appearance of section headings -

```
\makeatletter
\renewcommand{\section}{\@startsection{section}{1}{0mm}}
```

```

{\baselineskip}%
{\baselineskip}{\normalfont\normalsize\scshape\centering}}%
\makeatother
\begin{document}

```

5.3 New Commands

A completely new command can be created using

```

\newcommand{\commandname}[number of arguments]{command text, using
#1, #2 etc to denote arguments}

```

For example,

```

\newcommand{\ve}[1]{\(#1_1 ... #1_n\)}
\ve{x}

```

produces as output $x_1\dots x_n$

A problem with this example is that it shouldn't change to math mode if L^AT_EX is already in that mode. A better try would be

```

\newcommand{\ve}[1]{\ensuremath{#1_1 ... #1_n}}

```

which will only change to math mode if it's necessary.

A new environment is just as easily created - give the name of the environment, and what you want to happen on entering and leaving the environment. The following provides a variant of the `itemize` command.

```

\newenvironment{emlist}{\begin{itemize} \em}{\end{itemize}}
\begin{emlist}
\item first comment
\item second comment
\end{emlist}

```

The end of the environment ends the scope of the emphasis.

- *first comment*
- *second comment*

The end of the environment ends the scope of the emphasis.

5.4 Packages

There are many features and options not mentioned in this handout. See the `packages`¹² section of the online L^AT_EX page for more details. You can often find out what you want by looking at the files in the system directories. If your file begins

```

\documentclass[12pt]{article}

```

then various macro files are read when you run `latex`, namely (`article.cls` and `size12.sty`). Some of these files are well-enough commented to be useful documentation.

¹²http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/LaTeX_intro.html#Packages

5.5 An Example

`minipage` creates a miniature page complete with its own footnotes etc. It can be created to any set width. Footnotes within a `minipage` use a different counter to other footnotes¹³. This example ensures that all the numbers follow on in sequence. The footnotes in `minipages` are marked by letters rather than numbers, so here the type is changed to arabic.

```
\begin{minipage}{\textwidth}
% Set the minipage footnote counter
\setcounter{mpfootnote}{\value{footnote}}
% Redefine the command that produces the footnote number
\renewcommand{\thempfootnote}{\arabic{mpfootnote}}
\begin{tabular}{|l|}\hline
one & two\footnote{A minipage footnote}\\\hline
\end{tabular}
\setcounter{footnote}{\value{mpfootnote}}
\end{minipage}
```

one	two ¹⁴
-----	-------------------

¹⁴A `minipage` footnote

You might find this trick useful if you want footnotes in tables. They don't come out otherwise.¹⁵

Here's an example of using a package. The `multicol` file lets one change the number of columns easily. To switch into 3-column text use

```
\begin{multicols}{3}{
Put the text here. Maths, tables, pictures etc are all ok, but not
figures. But you have to remember to load in the \texttt{multicol}
package at the top of your document.
}
\end{multicols}
```

Put the text here. ures. But you have to top of your document.
Maths, tables, pictures remember to load in the
etc are all ok, but not fig- `multicol` package at the

L^AT_EX uses the value of the environmental variable `TEXINPUTS` to decide where to look for `sty` files. If you create a directory called (say) `inputs`, copy a system `*.sty` file into it and do

```
export TEXINPUTS=inputs//:$ {TEXINPUTS:-:}
```

then your copy will be read in preference to the system one and you can customise easily.

6 More Information

- See the LaTeX help page¹⁶. Much of the material available from there isn't available in printed form. The guides to producing Posters and booklets Re-

¹³like this one

¹⁵Not for me anyway

¹⁶http://www-h.eng.cam.ac.uk/help/tpl/textprocessing/LaTeX_intro.html

ports and OHP slides are useful.

- See the `comp.text.tex` newsgroup.
- Use the CTAN Archive <http://www.tex.ac.uk/tex-archive/>
- Engineering Department users can see the examples in `/export/Examples/LaTeX`. The `Tutorial` subdirectory contains exercises.
- Engineering Department users can borrow a \LaTeX manual from the machine room.
- Look at the \LaTeX files in your system. When you run \LaTeX you'll be shown where the files are.