

Preparing Documents with L^AT_EX

CS 2ME3/SE 2AA4

Steven Palmer

Department of Computing and Software
McMaster University

January 16 - 20

Outline

- 1 Introduction to L^AT_EX
 - What is L^AT_EX?
 - Installing L^AT_EX
- 2 Using L^AT_EX
 - Writing a Document in L^AT_EX
 - The Preamble
 - The Document Body
 - Compiling L^AT_EX
- 3 Example
 - A1 Report
- 4 Other Notes

L^AT_EX

L^AT_EX is a document preparation system used for preparing professional quality documents. Assuming you have primarily used MS Word or equivalent in the past, L^AT_EX will be a very new experience:

- L^AT_EX uses markup language source files, which can be compiled into .pdf documents.
- L^AT_EX produces documents in a What You See Is What You Mean (WYSIWYM) fashion, whereas Word-type programs are What You See Is What You Get (WYSIWYG).
- In other words, L^AT_EX gives you complete control over every aspect of your document (surely everyone has suffered the restrictiveness of Word at some point!)

Motivations to Learn L^AT_EX

Compared to Word-like editors, L^AT_EX has a much steeper learning curve (we will just be using basic L^AT_EX in this course, however, so it shouldn't be so bad!). There are of course significant benefits:

- Professional quality typesetting.
- A large library of packages that allow you to create documents however you envision: many of these packages are very useful for sfwr eng/comp sci students (presenting code, drawing graphs, etc.)
- You will be required to use L^AT_EX in upper year courses – better to get familiar with the basics now!

Installing TeX Distribution

You hopefully have already done this last week to get Doxygen working, but...

- You will need a T_EX distribution.
 - TeX Live is recommended.
 - For Ubuntu-based Linux distributions you can install via `apt-get install texlive-full`.
 - For other operating systems see <https://www.tug.org/texlive/> for installation details.

Structure of a L^AT_EX Source

L^AT_EX document sources use the following structure:

```
\documentclass [options]{<document type>}  
  ... preamble ...  
\begin{document}  
  ... document body ...  
\end{document}
```

For this course, you can just use the article class with 12 pt font for your document class:

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

The Preamble

The L^AT_EX preamble follows the `\documentclass` command. Here we declare which packages we want to use, perform setups for certain packages, define (or redefine) commands, and set certain fields (like title, author, date).

For this course, the following preamble should suffice:

```
\usepackage{fullpage} % extends margins to full page
\usepackage{booktabs} % for pretty tables
\usepackage{graphicx} % for including images in figures
\usepackage{hyperref} % for internal links

\title{Title goes here}
\author{Author goes here}
```

The Document Body

The body of your L^AT_EX document is contained in the document environment. Recall:

```
\begin{document}
  ...document body...
\end{document}
```

You will generally begin your document with a title page (and possibly a table of contents if you want):

```
\maketitle
\tableofcontents

% note that page breaks do not follow these by default
% to include page breaks use the following:
\maketitle
\newpage
\tableofcontents
\newpage
```

Sectioning

The following commands are used to create sections in L^AT_EX documents:

```
% creates a top level section with automatic numbering
\section{section title}

% creates a subsection of the current section
\subsection{subsection title}

% creates a subsubsection of the current subsection
\subsubsection{subsubsection title}
```

Sections are automatically numbered.

Paragraph Text

Paragraph text is simply typed directly into the source. A double line break indicates a new paragraph should be started. For example:

```
This is an example paragraph.  To start a new paragraph  
use a double line break.
```

```
This is a new paragraph.
```

Whitespace besides a double line break and single spaces is ignored. This will produce exactly the same output as before:

```
This is an example paragraph.  
To start a new paragraph  
use a double line break.
```

```
This      is      a      new      paragraph.
```

Environments

Several pieces of the document (lists, tables, figures) require special environments in L^AT_EX. Environments are used in the following way:

```
\begin{environment name} % this starts an environment
...environment body...  % do things
\end{environment name} % end the environment
```

We have seen this already with the document environment, in which the body of the document is written.

Lists

To make a list in L^AT_EX , we must use an environment:

```
% use the itemize environment for bullet lists
\begin{itemize}
  % use \item to start each bullet
  \item text for first bullet
  \item text for second bullet
  \item etc.
\end{itemize}

% use the enumerate environment for numbered lists
\begin{enumerate}
  % use \item to start each numbered point
  \item text for 1.
  \item text for 2.
  \item etc.
\end{enumerate}
```

Lists

The previous code will produce:

- text for first bullet
- text for second bullet
- etc.

- 1 text for 1.
- 2 text for 2.
- 3 etc.

Tables

To make a table in L^AT_EX , we use the table environment:

```
% use the table environment for tables
\begin{table}[<positions>]
  % <positions> tell LaTeX where our table is allowed
  % to be placed

  % you should use a combination of h, t, and b
  % h = here (in line)
  % t = top of page
  % b = bottom of page

  % Ex: \begin{table}[ht] means our table can either be
  % placed in line or at the top of a page (but not
  % at the bottom of a page) -- LaTeX will decide what
  % fits using a placement algorithm
\end{table}
```

Tables

We define the table using a tabular environment inside the table environment:

```
% use the table environment for tables
\begin{table}[h]    % h = in line
  \centering      % \centering centers the table
  % use the tabular environment to define your table
  \begin{tabular}{<columns>}
    % <columns> defines how many columns we will have
    % in our table, and their justification

    % use a combination of l, r, and c
    % l = left    r = right    c = center

    % Ex: \begin{tabular}{lcr} will create a table
    % with 3 columns, with the first left justified,
    % the second centered, and the last right justified
  \end{tabular}
\end{table}
```

Tables

Now we can construct the rows of the table:

```
\begin{table}[h]
  \centering
  \begin{tabular}{lr} % 2 columns
    \toprule % this creates a horizontal top line
    x-coord & y-coord\\ % these are our column headings
    \midrule % this creates a horizontal mid line
    1.2354 & 4.5543\\ % table entries
    3.4328 & 9.0033\\
    \bottomrule % this creates a horizontal bottom line
  \end{tabular}
\end{table}
```

Tables

Finally, we add a caption (table title), and a label (used to reference our table in text):

```
\begin{table}[h]
  \centering
  \begin{tabular}{lr}
    \toprule
    x-coord & y-coord\\
    \midrule
    1.2354 & 4.5543\\
    3.4328 & 9.0033\\
    \bottomrule
  \end{tabular}
  \caption{Example table} % table title
  \label{Table:example} % table reference name
  % NOTE: \label MUST come after \caption
\end{table}
```

Tables

The following table is produced by the code in the previous slide:

x-coord	y-coord
1.2354	4.5543
3.4328	9.0033

Table 1: Example table

For more examples and information on how to make tables of different sizes and stylings you can visit:

<https://en.wikibooks.org/wiki/LaTeX/Tables>

Figures

Figures in L^AT_EX are defined using the figure environment. It is similar to using the table environment:

```
% we use the figure environment for figures
% as with table, we must include permissible figure
% positions (combination of h, t, b)
\begin{figure}[h]
  \centering % to center figure
  \includegraphics{relative path to image file}
  \caption{Example figure} % figure title
  \label{Figure:example} % figure reference name
  % NOTE: \label MUST come after \caption
\end{figure}
```

Figures

Often, we want to scale our image to a certain fraction of the width of the document text (left margin to right margin). We can do this by adding a width option to the `\includegraphics` command:

```
\begin{figure}[h]
  \centering

  % assuming we have an image file called img.png the
  % same directory as our .tex source, this will create
  % a figure with the image img.png, and scale it to
  % 0.3 times the width of the document text
  \includegraphics[width=0.3\textwidth]{img.png}
  \caption{Example figure}
  \label{Figure:example}
\end{figure}
```

Figures

The code on the previous slide produces the following figure:



Figure 1: Example figure

Internal Referencing

Certain parts of the L^AT_EX document can be given labels using the `\label{...}` command. We saw this in tables and figures (sections can be labelled as well).

We can use the `\ref{...}` command in text to make internal references to these labels as follows:

```
% recall that we labelled our table "Table:example"  
% and our figure "Figure:example"  
  
% we can now reference their table/figure numbers:  
Refer to table \ref{Table:example}.  
  
See figure \ref{Figure:example}.
```

Internal Referencing

Here is the previous slide's referencing code in action:

Refer to table 1.

See figure 1.

Compiling a L^AT_EX Source to PDF

To compile a .tex source into a .pdf file, simply run the following command (on the command prompt/terminal):

```
pdflatex source.tex
```

Of course replacing source.tex with the actual name of your .tex file. If you installed TeX Live properly, pdflatex should be installed and found in your PATH.

Note: you may need to run pdflatex a second (sometimes even third) time to get all internal references working properly.

Putting it All Together

For an example we will flesh out parts of the first assignment's report.

First thing we will do is add our name and student number to the report.

- Open A1 report.tex (located in the report folder) in your favourite text editor
- Navigate to line 18

```
15 \newcounter{stepnum}~
16 ~
17 \title{Assignment 1 Solution}~
18 \author{Your Name and macid}~
19 \date{\today}~
20 ~
21 \begin {document}~
22 ~
```

Putting it All Together

- On line 18 replace Your name and macid with your actual name and ID

```
15 \newcounter{stepnum}
16 |
17 \title{Assignment 1 Solution}
18 \author{Henry M. 001020203}
19 \date{\today}
20 |
21 \begin {document}
22 |
```

- After you have replaced the information go to your terminal and type "pdflatex report.tex"

Putting it All Together

- Now open your report.pdf and ensure that your name and id is on your report

Assignment 1 Solution

Henry M. 000000000

January 11, 2018

Putting it All Together

- We are now going to make a table using the booktabs packages
- Open your report.tex file at the top of the file you will see a few `\usepackage{package}` statements, these are similar to include/import statements you have seen in Python or C. They allow us to use functions other people have created for LaTeX.
- Specifically we are going to add `\usepackage{booktabs}` to help us answer a question with a table.

Putting it All Together

```
1 \documentclass[12pt]{ar
2 |
3 \usepackage{graphicx}~
4 \usepackage{paralist}~
5 \usepackage{listings}~
6 \usepackage{booktabs}~
7 |
```

- Now we are going to navigate to the answers section of the report which should be located at or below line 47

```
46 \newpage~
47 |
48 \section{Answers}~
49 |
50 \begin{enumerate}~
51 |
```

Putting it All Together

- Navigate to the first question and where it says "Answer" we are going to insert a table in its place.
- To make a table we use `\begin{table}[h]` this tells LaTeX we want a table `\begin{table}` that is `[h]` inline.
- `\begin{tabular}{lr}` this tells LaTeX we want the table to have a `{lr}` left and right column.
- Now we are going to add our first row "Method & Type `\\`", the `&` symbol separates columns and `\\` separates rows.

```
55 \begin{table}[h]-
56   \begin{tabular}{lr}
57     Method & Type\\
58   \end{tabular}-
59 \end{table}-
60
```

Putting it All Together

- Run "pdflatex report.tex" again
- We now have a barebones table with two columns and one row that looks like this:

Method	Type
--------	------

- Lets now add some more rows, add two methods and what type they are.

Putting it All Together

- The table is still pretty bare, lets add some borders like this:

Method	Type
rm	???
size	???

- We will use `\toprule` `\midrule` `\bottomrule` to add borders to our table

Putting it All Together

```
55 \begin{table}[h]-
56   \begin{tabular}{lr}-
57     \toprule-
58     Method & Type\\-
59     \midrule-
60     rm & ???\\-
61     size & ???\\-
62     \bottomrule-
63   \end{tabular}-
64 \end{table}-
```

- Now we just have to run `pdflatex` on our report again and we will have:

Method	Type
rm	???
size	???

L^AT_EX Editors

You will probably want to use an editor with syntax highlighting (and maybe some additional features) when writing your L^AT_EX source. Some recommendations:

- TeXworks (multiplatform)
- WinEdt (Windows)
- NotePad++ with L^AT_EX syntax highlighting (Windows)
- gedit with L^AT_EX syntax highlighting (Linux)

Additional References

- If you want to do something in particular and don't know how, your best resource is Google: any question you have will likely have already been asked and answered on Stack Exchange or similar.
- A complete listing of all L^AT_EX packages, along with documentation for each package, can be found at <https://www.ctan.org/?lang=en> (Comprehensive T_EX Archive Network)

Fun Fact

This presentation was written as a L^AT_EX document. The source is available along with the pdf if you are interested – although do be advised it is much more advanced than the example tex file so don't worry if some parts look confusing!