

LATX1001

L^AT_EX TEMPLATES FOR UNIVERSITY

L^AT_EX Template Usage

Author

ES ALEXANDER

Release

2.2.0

April 27, 2019

Contents

1	Introduction	1
2	Basics	1
2.1	Commands	1
2.2	Environments	2
2.3	Comments	2
2.4	Text Mode	2
2.4.1	Paragraphs	2
2.4.2	Quotations	2
2.4.3	Unordered Lists	2
2.4.4	Ordered Lists	3
2.4.5	Columns and Page Breaks	3
2.4.6	Text Formatting	3
2.5	Maths Mode	4
2.5.1	Inline	4
2.5.2	Display	4
2.5.3	Equation	4
2.5.4	Aligned	4
2.5.5	Common Character/Font Commands	5
2.5.6	Spacing	5
3	Template Structure	5
4	Template Functionality	6
4.1	Comments	6
4.2	Header	6
4.3	Page Size/Orientation	6
4.4	Figures	6
4.5	Tables	7
4.5.1	Report-Exclusive Table Features	8
4.6	Code	9
4.7	Labels	9
4.8	Referencing	10
4.8.1	Report-Exclusive Referencing	10
4.9	Maths Notations	10
4.10	Template Specifics	11
4.10.1	Assignment Template	11
4.10.2	Report Template	11
4.10.3	Meeting Minutes Template	11

1 Introduction

L^AT_EX is a document preparation system often used for high-quality typesetting of formal documents. Its usage is prevalent in scientific and engineering journals, and is encouraged for university reports in both fields.

Often beginners have difficulty with the learning curve required when transitioning from a word processor such as Microsoft Word or Libre Office, so the templates in this repository (github.com/ES-Alexander/LaTeX-templates) have been provided to help ease the transition and provide a good point to start from.

The templates have been developed throughout my undergraduate study, and the setup and commands therein have been developed specifically for the tasks that I required in my undergraduate engineering degree. This document aims to outline some L^AT_EX basics, as well as the structure of the provided templates and how to use them.

If you are yet to decide on a L^AT_EX editor, my personal preferences are TeXMaker for offline use, and Overleaf for shared live online editing.

2 Basics

A basic understanding of L^AT_EX is essential to using it effectively, particularly when you require custom functionality. Fundamentally L^AT_EX supports text, maths, commands, and environments, which together encompass all the functionality required for professional quality presentation. Various ‘packages’ are available to make use of commands and environments that others have developed.

If you have used L^AT_EX before, feel free to skip this section.

2.1 Commands

Commands (or macros) are one of the components which truly set L^AT_EX apart from word processors. They have the potential to be incredibly powerful, and can significantly reduce repetition of work within and across documents.

The most basic of commands take no inputs, and simply perform an action or output some predefined text. Commands are denoted with a backslash, as:

```
\commandName
```

Some commands take in one or more required inputs, in curly braces:

```
\commandName{required1}{...}
```

Other commands have an optional input, in square brackets:

```
\commandName[optional]
```

And some take both an optional parameter and one or more required inputs:

```
\commandName[optional]{required1}{...}
```

Custom commands can be defined using

```
\newcommand{\commandName}{behaviour}
```

```
\newcommand{\commandName}[numInputs]{behaviour #N}
```

```
\newcommand{\commandName}[numInputs][optionalDefault]{behaviour #N},
```

where the Nth input is accessed using #N, and, if included, the optional input is #1. `optionalDefault` can be left blank if an optional input is desired with an empty default value.

2.2 Environments

Environments change the styling and/or behaviour of a block of L^AT_EX/text, and have a beginning and an end, as in:

```
\begin{environmentName}[options]
  <relevant code and/or text>
\end{environmentName}
```

All L^AT_EX documents begin by specifying a document type (often ‘article’), followed by relevant setup code in what’s called the ‘preamble’, and all aspects intended for output inside the `document` environment. A simple L^AT_EX document could be

```
\documentclass[11pt, a4paper]{article}
\begin{document}
  Hello
\end{document}
```

2.3 Comments

Sometimes in your L^AT_EX code it is desirable to include comments. To comment out the remainder of a line, use a percentage symbol before your comment.

```
\LaTeX\ code here % comment here    →    LATEX code here
```

Other times it’s useful to comment out multiple lines at once, which is implemented in this set of templates as the `\comment{comment text here}` command. To instead have a multi-line comment which is displayed in the document,

```
\dcom{comment text here}    →    comment here
```

2.4 Text Mode

A variety of formatting options are available for text, as environments or commands.

2.4.1 Paragraphs

Plain text, for paragraphs and the like, is written anywhere within the `document` environment. Generally paragraphs are ended with `\\`, and have an empty line between them.

2.4.2 Quotations

Open quotation marks are inserted using a backtick (``` → `‘`), and quotations end with apostrophes (`’` → `’`). Backtick is generally found on the same key as tilde (`~`), beneath ESC on a QWERTY keyboard. Indented quote blocks can be inserted using the `quote` environment.

2.4.3 Unordered Lists

Unordered lists can be created using the `itemize` environment. Each item begins with a `\item`, which takes an optional argument to specify the bullet type (e.g. `\item[-]` for a dashed bullet). Sub-lists are created with a nested `itemize`, and are automatically indented with a change of bullet, as in the following example.

<pre> \begin{itemize} \item item1 \item item2 \begin{itemize} \item subitem2.1 \end{itemize} \item[o] item3 \item[-] item4 \item ... \end{itemize} </pre>	\Longrightarrow	<pre> • item1 • item2 - subitem2.1 o item3 - item4 • ... </pre>
---	-------------------	---

2.4.4 Ordered Lists

Similarly to `itemize` for unordered lists, the `enumerate` environment can be used to create ordered lists, as below. Ordered lists can also be created as sub-lists to unordered lists, and vice versa.

<pre> \begin{enumerate} \item item1 \begin{enumerate} \item subitem1.1 \end{enumerate} \item ... \end{enumerate} </pre>	\Longrightarrow	<pre> 1. item1 (a) subitem1.1 2. ... </pre>
---	-------------------	---

2.4.5 Columns and Page Breaks

Columns are created using the `multicols` environment, which takes as a required parameter the number of columns (e.g. `\begin{multicols}{3}`). Text is automatically distributed amongst columns, but can be forced to the next column with a `\columnbreak`. Similar commands exist for forcing text to the next line (`\newline`) and to the next page (`\newpage`).

The `minipage` environment can be used to encase text, and can be useful where text in multiple columns should be vertically aligned. A combination of `multicols`, `minipages`, and vertical fills (`\vfill`) were used to create the list examples in Sections 2.4.3 and 2.4.4.

2.4.6 Text Formatting

Additional text formatting can be accomplished on an environment-wide or specified range of text. `\textit{text}`, `\textbf{text}`, `\underline{text}` and `\texttt{text}` make the specified text italicised, bold, underlined, or type-writer font. `\bfseries` and `\itshape` apply bold/italicised formatting to the entire surrounding section/environment (e.g. `not bold {\bfseries bold}`). `\centering` is the equivalent for centering, and the `center` environment can be used as an explicitly centred text environment.

2.5 Maths Mode

L^AT_EX has extensive maths display functionality, and is used extensively as the maths display engine across various websites and applications. Several display modes are provided to allow for maths presented as desired within your documents.

2.5.1 Inline

Inline maths mode is for maths within text, and is written as maths between single dollar signs (e.g. $\$a^2 + b^2 = c^2\$ \rightarrow a^2 + b^2 = c^2$).

2.5.2 Display

Display maths mode allows for displaying maths on its own line, and is written between double dollar signs, for example $\frac{\pi R^2}{4}$ becomes

$$\frac{\pi R^2}{4}.$$

2.5.3 Equation

The `equation` environment is like display maths mode, except with an equation number to the right of the line, which allows for easier referral in the rest of the document. As an example, `\begin{equation} \sum\limits_{n=0}^{\infty} \frac{\mathrm{lr}{-1}^n x^{2n}}{\mathrm{lr}{2n}!} = \mathrm{cos}{x} \eqlabel{just cos}\end{equation}` becomes

$$\sum_{n=0}^{\infty} \frac{(-1)^n x^{2n}}{(2n)!} = \cos(x) \tag{1}$$

Note that `\lr` and `\eqlabel` are defined as part of these templates, and `\cos` is overwritten to behave as displayed.

2.5.4 Aligned

The `align` maths environment allows for several lines of maths aligned with ampersands (&), as in

```
\begin{align}
\text{let } x \&= \pi \text{ in } \eqref{just cos} \nonumber \\
\rightarrow \cos(\pi) \&= \sum\limits_{n=0}^{\infty} \frac{\mathrm{lr}{-1}^n \pi^{2n}}{\mathrm{lr}{2n}!} \\
\therefore -1 \&= \sum\limits_{n=0}^{\infty} \frac{\mathrm{lr}{-1}^n \pi^{2n}}{\mathrm{lr}{2n}!}
\end{align}
```

becomes

$$\begin{aligned} &\text{let } x = \pi \text{ in (1)} \\ \rightarrow \cos(\pi) &= \sum_{n=0}^{\infty} \frac{(-1)^n \pi^{2n}}{(2n)!} \end{aligned} \tag{2}$$

$$\therefore -1 = \sum_{n=0}^{\infty} \frac{(-1)^n \pi^{2n}}{(2n)!} \tag{3}$$

The `align*` environment is identical to `align` but without line numbering.

2.5.5 Common Character/Font Commands

At times particular characters are required that don't match the standard italicised variables that are default in maths mode. `\mathscr{L}` gives a fancy script version of the specified letter. `\text{ text }` can be used to have text within a maths mode section, and `\mathrm{d}` converts the specified letter(s) to Roman characters (generally the standard font) to differentiate them from variable names.

2.5.6 Spacing

Maths mode ignores general white-space, and often requires more precisely defined spacing than text mode, which is facilitated with the following commands:

- `\,` (small space)
- `\` (normal space)
- `\!` (negative space)
- `\quad` (4 spaces)
- `\qquad` (8 spaces)

3 Template Structure

The provided templates all share the functionality provided in `CommonSetup2.tex`. Separate to this are `AssignmentSetup2.tex` and `ReportSetup2.tex` for functionality exclusive to assignments and reports respectively. `MinutesSetup2.tex` provides functionality exclusive to meeting minutes, and extends upon `ReportSetup2.tex`. As the setup files are not intended to change for different documents, they are linked to with absolute paths (in `\input{/Absolute/Path/From/Root/File.tex}` commands) that must be specified when you download the templates, and re-specified appropriately when using the template(s) online (e.g. on Overleaf) or after an update of the repository.

Each template also comes with a `<Name>Template2.tex` file, which should be copied each time you wish to create a new document with that template. The absolute paths allow the template file to be copied to anywhere in your computer and still maintain the links to the relevant setup files. Note that images being linked to in the documents are expected to be in an `images/` folder, within the same folder as the document itself. These should be linked to relative to the document, which can be done using just the folder name followed by a forward-slash, then the appropriate filename (where necessary).

The `Titlepage.tex` file used for generating report title-pages is something of a grey area, as titlepages are generally the same format, but at times a unique titlepage will be required for specific reports. The suggestion is to modify `Titlepage.tex` to match your most common titlepage format and provide an absolute link to it in your `ReportTemplate2.tex` file stored in the repository folder. In cases where a unique titlepage is desired, copy `Titlepage.tex` to the folder where your working document template is and change that copy instead of the shared one. Ensure you change the link for the working document to the relative titlepage, or the shared titlepage format will be used instead of the new one.

Inside the preamble of each `<Name>Template2.tex` file is all the relevant setup configuration code for that template. Any changes outside of that should be made in the `<Name>Setup2.tex` file or `CommonSetup2.tex` file for template-wide changes, or in a local copy of one or both of those documents for document-specific changes. Alternatively, document-specific changes can be incorporated into the preamble of the document, particularly new definitions or simple redefinitions.

4 Template Functionality

The set of templates comes with several definitions that are intended to reduce setup and formatting time by quickly and easily achieving desirable results.

4.1 Comments

Commenting functionality is added to enable multi-line comments with optional display in the document, using `\comment{text}` and `\dcom{text}` for hidden and displayed comments respectively. Commenting is discussed in greater depth in Section 2.3.

4.2 Header

Many formal documents include a header, with information such as course codes and document IDs displayed at the top of every page. These templates include the commands `\lefthead{val}` and `\righthead{val}` to specify the header text. Generally these are defined more meaningfully in the individual template files.

4.3 Page Size/Orientation

Page sizing and orientation are important aspects of document formatting that can cause significant frustration. The following commands are defined to simplify this, and change all subsequent pages of the document to the specified size until the next re-size is called.

- `\setafport` (set A4 portrait)
- `\setafland` (set A4 landscape)
- `\setatport` (set A3 portrait)
- `\setatland` (set A3 landscape)
- `\setpagesize{width}{height}` (set custom size, e.g. `\setpagesize{21cm}{27.9cm}`)

4.4 Figures

L^AT_EX re-imports figures on every compile, which allows for images to be updated and replaced which are then updated automatically in the document at the next compile.

To specify where L^AT_EX should search for your images, the `\graphicspath{{path1}/{path2}}` command can be used. Generally this will be in the preamble of the template files. Paths can be relative to the location of the main document (generally the template file), beginning with their folder name, or paths can be absolute from the top of the drive, beginning with a `/`. Each path is specified in a set of braces, and additional paths can be specified in additional braces.

A set of figure commands have been developed for ease of use, and function as follows:

- `\fig[alignment]{scale}{filename}{caption}`
 Inserts a centred figure with parameters:
 - `*alignment`: H by default for 'exactly in place' - can also be:
 - * h (approximately in place)
 - * b (at the bottom)
 - * t (at the top)
 - * p (on a page set aside for figures/tables)
 - * (some others possible if necessary)
 - `scale`: the scale of the inputted image (1 = real size)

- filename: the filename of the image to display
Don't include the file extension (e.g. use photo instead of photo.png) unless multiple files exist with the same name but different extensions.
- caption: the figure caption

Figures are auto-labelled by their filename, for cross-references in the document. Reference with `\figref{filename}` or `\reffig{filename}` for just the number.

- `\framefig[alignment]{scale}{filename}{caption}`
Inserts a centred figure with a frame. Parameters are the same as for `\fig`.
- `\multifig[alignment]{\subfigs}{caption}`
Inserts a figure with multiple subfigures, as specified by the `\subfigs` argument (see next command). Auto-labelling for multi-figs is based off the caption as there is no single filename.
- `\subfig[img-scale/height/width]{alignment}{fig-width}{filename}{caption}`
Inserts a sub-figure (generally within a multi-fig), with parameters:
 - *img-scale/height/width: defaults to `width=\textwidth` as this makes the specified image the same width as the sub-figure it resides in. Can also be `scale=N` (similar to normal figures but with the word specified) or `height=L`. Height and width require either relative measures, or numbers with units (e.g. `width=3cm`, `height=5in`, etc).
 - alignment: same options as normal figure alignment, but specifies the alignment of this subfig within the multifig. Generally set to `t` (top) or `b` (bottom).
 - fig-width: the width of this subfig within the multifig. Often easiest to set this as a decimal between 0 and 1 multiplied by the width of the multifig (e.g. `0.4\textwidth`). The decimal multiplier can be selected based on how many columns of subfigs you want, noting that sometimes the sum of widths must be slightly below `\textwidth` for a set of subfigs to sit in a single row of a multifig. Smaller numbers allow for more spacing between figures in a row, and `\` can be used to force the subsequent figures onto the next row.
 - filename/caption: same as for normal figures

Subfigs are auto-labelled by their filename, and can be referenced with `\sfigref{filename}` or `\reffig[s]{filename}` for just the number.

4.5 Tables

Tables are often known to be frustrating to deal with in standard L^AT_EX, but the following commands significantly simplify their use.

- `\easytable[alignment]{caption}{columns}{contents}`
Creates a table with parameters:
 - *alignment: same as for figures
 - caption: the table caption
 - columns: column distribution, types, and separation for the table, as a string of characters. To allow for full row styling, the first column specifier should be preceded by a `^` and subsequent column specifiers should each be preceded by `;` (e.g. `l^c|;c|;p{0.5\linewidth}|`). The allowable specifiers are:
 - + `p{width}` (justified, with width as specified (requires units (2cm, 5em, 12pt), or relative measure (`n\linewidth`)))
 - + `c` (centered, with column-width auto-scaled)
 - + `l` (left aligned)
 - + `r` (right aligned)
 - + `|` (vertical border line (can be repeated, e.g. `l^c||;l;r`))

Spaces in the column specification are ignored.

To style an entire column, precede its alignment specifier with:

- + `b` (bold)
- + `i` (italicised)
- + `>{style}` (styled as specified)
- contents: contents of the table (e.g.
`\headingrow Heading 1 & Heading 2 & Heading 3 \row`
`Side Heading & Text & Text \row`)

To style a single cell, use

- + `\textbf{text}` (bold)
- + `\textit{text}` (italicised)
- + `\normalfont{text}` (ignore column/row styling)

To style a row, precede it with

- + `\bfrow` (make this row bold)
- + `\itrow` (italicise this row)
- + `\rowstyle{style}` (styled as specified)

Table items are separated with ampersands (`&`).

- `\row` (end this row with a horizontal line beneath it)
- `\headingrow` (this row is a heading row)

`\hline` can be used to manually add a horizontal line to the table. `\` can be used to end the current row without a horizontal line beneath it.

As an example,

Check out the table styling in `\tabref{Table Styling Example}`.

```
\easytable{Table Styling Example}{|~bc|;ic;c|}{
  \headingrow Test1 & \normalfont{Test1} & Test1 \row
  Test2 & Test2 & Test2 \\
  \itrow Test3 & Test3 & Test3 \\
  \bfrow Test4 & Test 4 & Test 4 \row
}
```

becomes

Check out the table styling in Table 1.

Table 1: Table Styling Example

Test1	Test1	Test1
Test2	<i>Test2</i>	Test2
<i>Test3</i>	<i>Test3</i>	<i>Test3</i>
Test4	<i>Test 4</i>	Test 4

4.5.1 Report-Exclusive Table Features

Certain table features are exclusive to the report template as they were not deemed necessary for the assignment template. These features are as follows:

- coloured cells (`\rowcolor{colour}` and `\cellcolor{colour}` commands)
- tables spanning multiple pages using `\easylongtable`, which uses the same parameters as `\easytable`

- merged cells using
 - `\multicolumn{num_cols}{alignment}{text}`
span multiple columns with one cell, specifying how many columns to span, the alignment of the merged cell (e.g. c), and the cell's contents
 - `\multirow{num_rows}{*/width}{text}`
span multiple rows with one cell, specifying how many rows to span, the width of the new cell (generally * to match the existing column width), and the cell's contents

4.6 Code

Sometimes it is useful to include code snippets or entire files within a document. The `listings` package allows this with several styling and display options and support for over 50 different programming languages.

To treat written text as code, use the `\verb!text!` function (code between exclamation marks) for inline, or the `lstlisting` environment for display-mode/multi-line code. The `\intextCode[options]{caption}{code}` function can be used to include a caption with in-text code (auto-labelled by the caption).

When inputting code from a file, use

- `\python[options]{filename}{caption}` (Python code file)
- `\matlab[options]{filename}{caption}` (Matlab code file)
- `\inputCode[options]{filename}{caption}` (general code file)

The `\python` and `\matlab` commands are for coloured formatting and relevant styling of Python and Matlab code. Note that unlike with images, file extensions should be included in filenames when inputting code files. Auto-labelling for inputted files is by filename.

While many options (specified as key-word arguments) are possible (search online for 'LaTeX lstlistings options'), some which may be of particular interest are

- `language` (select the language of the code (e.g. `language=Python`))
- `firstline` (specify the first line of the file to include)
- `lastline` (specify the last line of the file to include)
- `firstnumber` (specify the number to start numbering at, if numbers are included - generally the same as 'firstline')

4.7 Labels

While considerable effort has been put towards auto-labelling, at times it may be appropriate to label something directly, or to create your own labelling and referencing functions. In these cases, the following functions can be used:

- `\plabel[prefix]{label}` (a label with a prefix (often object type))
For example, auto-labelling of figures internally uses `\plabel[fig]{label}`.
- `\eqlabel{label}` (a label for an equation, use in the `align` or `equation` environments)
- `\label{label}` (base label function, if necessary - doesn't distinguish between object types as easily as `\plabel`)

4.8 Referencing

Labelling is primarily useful because of the ability to cross-reference to objects in the document. Various referencing functions have been defined to simplify this process, allowing for either the number associated with a `\plabel` label, or the number with a meaningful descriptor. These are as follows:

- `\figref{figFilename/figCaption}` → Figure N
- `\sfigref{subfigFilename}` → Figure Na)
- `\tabref{tableCaption}` → Table N
- `\coderef{codeFilename/codeCaption}` → Listing N
- `\eqref{eqnLabel}` → (N)
- `\reffig[s]{figFilename/figCaption}` → N/Na
- `\reftab{tableCaption}` → N
- `\refcode{codeFilename/codeCaption}` → N

References with optional 's' values represent labels for objects that can be sub-objects (e.g. `\reffig{figFilename}` for normal figures becomes `\reffig[s]{figFilename}` for subfigures, and is `\reffig{figCaption}` for multfigs.

4.8.1 Report-Exclusive Referencing

Some additional referencing methods are exclusive to reports, allowing for referenced sections and easy input of hyperlinked urls.

- `\refapp[s/ss]{appendixName}` → A/A.B/A.B.C
- `\refsec[s/ss]{sectionName}` → N/N.M/N.M.L
- `\appref[s/ss]{appendixName}` → Appendix A/A.B/A.B.C
- `\secref[s/ss]{sectionName}` → Section N/N.M/N.M.L
- `\url{url}` (hyperlinked URLs without needing to deal with special characters)

4.9 Maths Notations

While many maths notations come with L^AT_EX by default, the following set are definitions or redefinitions that cater to my preferences and help speed up my workflow.

- Units in math mode:
 - `\unit[power]{unit}` (unit with an optional power)
 - `\nsunit[power]{unit}` (`\unit` but with no preceding space)
 - `\mathrm{units+powers}` (useful for several units with multiple powers)
- Powers of ten:
 - `\E{N}` → $\times 10^N$ (used in maths mode)
 - `\tE{N}` (used in text mode)
- Vertically auto-scaled parentheses/brackets to fit contents:
 - `\lr{parenthesesed}` → *(parenthesesed)*
 - `\lrs{squared}` → *[squared]*
 - `\lra{absoluted}` → *|absoluted|*
- Integration:
 - `\eint{f(x)}{x}` → $\int f(x) dx$
 - `\eiint{f(x,y)}{x}{y}` → $\iint f(x,y) dx dy$
 - `\dint{f(x)}{a}{b}{x}` → $\int_a^b f(x) dx$
 - `\diint{f(x,y)}{a}{b}{c}{d}{x}{y}` → $\int_a^b \int_c^d f(x,y) dx dy$

- Differentiation:

$$\begin{aligned}
 - \text{\del[order]\{f(x,y)\}\{x\}} &\rightarrow \frac{\partial f(x,y)}{\partial x} \\
 - \text{\der[order]\{f(x)\}\{d\}} &\rightarrow \frac{d^2 f(x)}{dx^2}
 \end{aligned}$$

- Redefined with brackets around argument:

$$\begin{aligned}
 - \text{\cos[order]\{x\}} &\rightarrow \cos^2(x) \\
 - \text{\sin[order]\{x\}} &\rightarrow \sin(x) \\
 - \text{\tan[order]\{x\}} &\rightarrow \tan^{-1}(x) \\
 - \text{\ln\{x\}} &\rightarrow \ln(x) \\
 - \text{\log[base]\{x\}} &\rightarrow \log_3(x) \\
 - \text{\exp\{x^2\}} &\rightarrow \exp(x^2)
 \end{aligned}$$

- $\text{\limit\{n\}\{\infty\}} \rightarrow \lim_{n \rightarrow \infty}$
- $\text{\D} \rightarrow d$ (maths differential)
- $\text{\deg[unit]} \rightarrow ^\circ\text{C}$ (degrees symbol with optional unit)
- $\text{\ve\{v\}} \rightarrow \mathbf{v}$ (bolded vector notation)
- $\text{\qf\{a\}\{b\}\{c\}} \rightarrow \frac{-(b) \pm \sqrt{(b)^2 - 4 \cdot a \cdot c}}{2 \cdot a}$ (quadratic formula)

4.10 Template Specifics

While most of the functionality of the templates is shared, some is specific to one template type, as has been seen with some report-exclusive features in previous sections.

4.10.1 Assignment Template

Assignments include the `\que{Question}` command, which creates numbered, bolded questions.

4.10.2 Report Template

Reports include a titlepage, along with auto-labelled sections and appendices. This auto-labelling requires modified commands, and is accessed by the existing section commands with an 'l' appended, or an 'al' for appendices (e.g. `\section{Name}` \rightarrow `\sectionl{Name}`, and `\subsubsection{AppendixName}` \rightarrow `\subsubsectional{AppendixName}`). These auto-labelled sections can then be accessed with the report-exclusive referencing commands in Section 4.8.1.

4.10.3 Meeting Minutes Template

Meeting minutes include the relevant timing, location, and people present, but also include the commands `\topic[owner]{topic}` to create a topic with an optional owner, and `\alloc{Name}` to allocate someone to a task in the final 'to-do' list.