

2.1 Getting Started

L^AT_EX¹ is a highly versatile and robust typesetting program derived from T_EX. T_EX was created by Donald E. Knuth and it is quite simply, “[t]he most powerful formatting program for producing book quality text of scientific and technical works.” [1] Because T_EX’s main emphasis is on text formatting it relies on primitive command structures and is difficult to use. L^AT_EX is a macro language developed by Leslie Lamport to simplify the task of utilizing T_EX by packaging it with many higher level commands structures, often referred to as “packages.” L^AT_EX 2_ε is the current official standard distribution of L^AT_EX, and it incorporates many of the packages, that have been created since its’ initial release. Whereas T_EX is still primarily used by “hard-core” programmers it is possible for a complete novice to learn to create publication quality documents using L^AT_EX 2_ε in only a few sessions.

Before proceeding there are two subtleties about the T_EX compiler that are well worth remembering: (1) **spaces do not matter to T_EX**, (2) **every empty line begins a new paragraph.** e.g.,

When this sentence was written many extraneous spaces were included and a new paragraph was began here

by simply adding an empty line in the text editor.

When this sentence was written many extraneous spaces were included and a new paragraph was began here

by simply adding an empty line in the text editor.

2.1.1 Basic Program Structure

In general all T_EX programs are made up of two parts: the **preamble**, and the **document**. The preamble specifies the type of document as well as any special packages or commands that the compiler should use. The anatomy of a basic L^AT_EX 2_ε program is shown in figure 2.1. Note that all L^AT_EX 2_ε programs begin as plain-text files and must be saved with a “.tex” extension.

¹The pronunciation is “lay tech”, not “lay tex”.

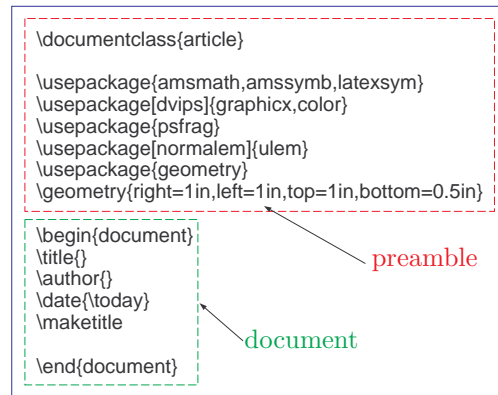










Figure 2.1: A complete L^AT_EX 2_ε program contains both a **preamble** and a **document** environment. Here the standard packages recommended by the author are shown along with the commands needed to produce a title with author and date.


2.1.2 Compiling a “.tex” File

As with all programming languages T_EX is very particular about its’ grammar. Because of this almost all users are going to make mistakes from time to time. Luckily the standard compilation process yields plenty of information to fix all but the most stubborn of problems.

As mentioned above every L^AT_EX 2_ε program begins as a plain-text file, with a “.tex” extension. As an example assume that an author begins with the file, “fname.tex”. Compilation of this file begins by pressing the  button on the WinEdt toolbar. This action automatically saves the file and opens an MS-DOS window. Text messages are then simultaneously echoed to the screen and written to the “fname.log” file. If the compilation is successful, then the MS-DOS window closes and the file “fname.dvi” is generated². The “.dvi” extension stands for “device-independent.” This independence is one of the main contributions that T_EX made to the typesetting community, and is one of the reasons that T_EX may be used to create high-quality documents in many different formats.

As illustrated in Figure 2.2 the author’s customization of WinEdt’s utilities allows a user to begin to compile a “.tex” file, by simply pressing the  button. This generates a device-independent(DVI) format³ which can then be converted to a Postscript(PS) format by pressing the  button. Once this is accomplished the PS file may be viewed directly by pressing the  button, or converted to a Portable Document Format (PDF) by pressing the  button and then viewed by pressing the  button. Alternatively a Hypertext Markup Language (HTML) version of the file may be generated by pressing the  button and viewed by pressing the  button. Note however that while the conversion to both PS and PDF is complete, the successful conversion of advanced L^AT_EX 2_ε documents to HTML has not yet been accomplished.

Errors

The first clue that something has gone wrong in the compilation process is that the MS-DOS window opened when the  button was pressed remains open and looks similar to the one shown in Figure 2.3.

²Actually a large number of files may be generated, depending on the packages and commands used. For a complete list of the files generated by L^AT_EX 2_ε, see [1], section 8.9.

³While the DVI may be viewed, any incorporated graphics will not be shown.

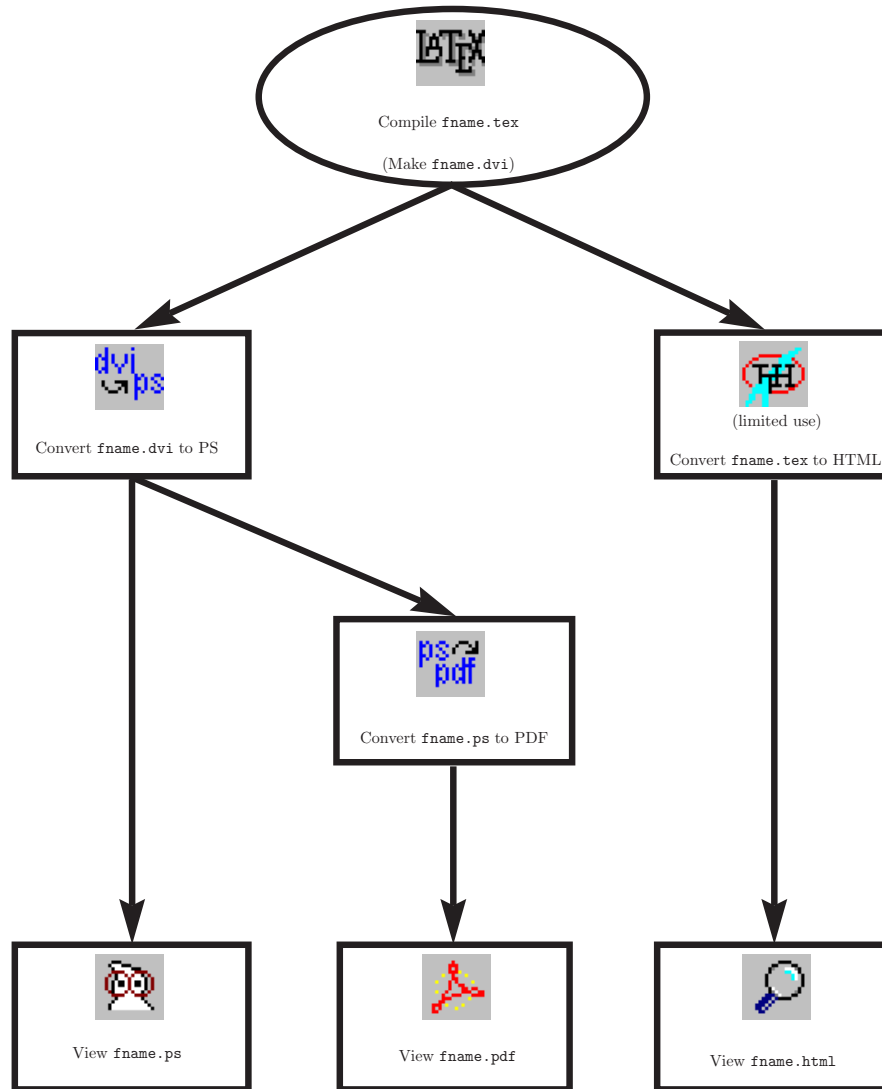


Figure 2.2: With the author's customization of WinEdt's utilities the task of compiling and converting a $\text{\LaTeX}2_{\epsilon}$ document to either PS, PDF, or HTML is accomplished by simply pressing the appropriate toolbar buttons in the indicated order.

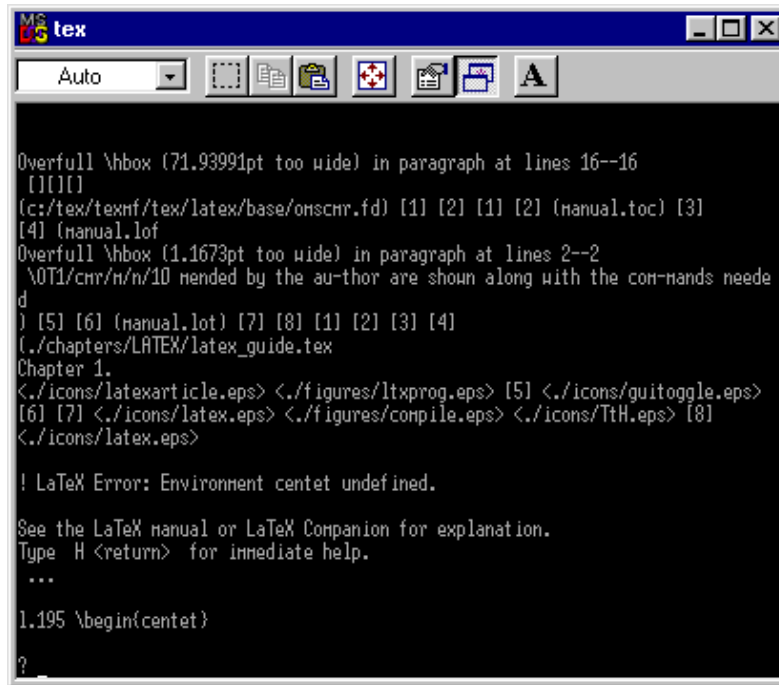


Figure 2.3: A typical error encountered during the compilation of a L^AT_EX 2_ε program.

The relevant error text is,

```
! LaTeX Error: Environment centet undefined.

See the LaTeX Manual or LaTeX Companion for explanation.
Type H <return> for immediate help.
...

1.195 \begin{centet}

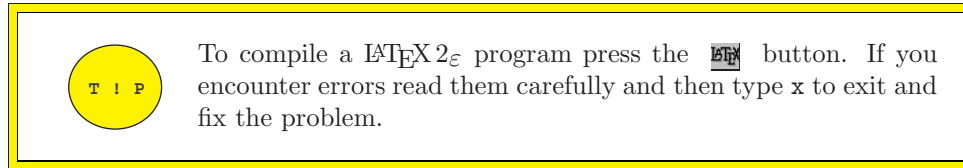
?
```

and indicates that the environment `center` is undefined. The `1.195` indicates that the error can be found on line 195 of the document. In the likely event that errors are of this kind are encountered during the compilation process the following commands can be typed at the `?` prompt⁴:

- `<return>` *return mode*:
Tells T_EX to continue processing after attempting to fix the problem. In the case of an unknown command name, the error treatment is to ignore it.
- `'S'` *scroll mode*:
T_EX continues the processing, writing further error messages to the screen as they are encountered, but without stopping for a user response.
- `'I line of text'` *insert mode*:
T_EX puts the line of text entered at the prompt in the program where the error is indicated. The `.tex` file is unchanged, requiring the error to be fixed within an editor.
- `'X'` *exit mode*:
The T_EX processing is halted and the current and subsequent pages do not appear in the DVI file.

⁴Adapted from [1], pages 256-7.

- ‘H’ *help mode*:
An extended account of the error is echoed to the screen, which may help the user more readily ascertain the solution.



In most cases errors can be fixed with relative ease. For example, the error cited above could be fixed by either typing ‘I \begin{center}’ at the prompt or by typing ‘X’ and correcting the error manually within WinEdt. More generally, if an error can not be fixed, (or even found!) then *scroll mode* made be used. If the compilation is successful the document may then be viewed to help determine the nature (or location!) of the error.

2.1.3 Document Classes and Options

While L^AT_EX 2_ε offers several different document classes such as, `article`, `report`, and `book`, **for the purposes of this book only the article style will be needed**. It is possible however to add options to this basic style. The general syntax for the `documentclass` command is:

```
\documentclass[options]{article}
```

The most common options available for the article class include:[1]

- `onecolumn`: (default) Sets the article in a single column format.
- `twocolumn`: Sets the article in a two-column format.
- `letterpaper`: (default) Uses the American standard letter sized paper (11 × 8.5 in) to set margins.
- `legalpaper`: Uses legal sized paper (14 × 8.5 in) to set margins.
- `portrait`: (default) Document format is such that the longer dimension of the paper is the vertical one.
- `landscape`: Document format is such that the shorter dimension of the paper is the vertical one.
- `fleqn`: Displayed formulas will be set with a flush left justification instead of centered.
- `leqno`: Equation numbers in displayed formulas will be set with a flush left justification instead of on the right.

If more than one of these options is desired they are simply separated with a comma. For example, to make a two-column article with 11 pt text⁵, an author would use the following `documentclass` command and options:

```
\documentclass[11pt,twocolumn]{article}
```

In addition, to the above options which effect the entire document, an author may wish to “fine-tune” the appearance of the text by issuing the following commands either in the preamble or at points in the text:

⁵Note that 1 in = 72.27 pt. (See table 2.1 for a list of valid T_EX length units.)

Unit Name	Abbreviation	Conversion
Centimeter	cm	(2.54 cm = 1 in)
Millimeter	mm	(10 mm = 1 cm)
Inch	in	
Point	pt	(1 in = 72.27 pt)
Big Point	bp	(1 in = 72 bp)
Pica	pc	(1 pc = 12 pt)

Table 2.1: The standard T_EX length units. (Adapted from [1].)

- `\parindent`: Specifies the indentation from the left margin for the first line in a new paragraph.
- `\mathindent`: Specifies the indentation from the left margin for equation numbers when the `fleqn` option is selected.
- `\columnsep`: Specifies the space between the two columns of text when the `twocolumn` option is selected.
- `\textwidth`: Specifies the width of the standard line of text.

When using each of these commands the desired length would be set (or reset) with the `setlength` command:

```
\setlength{command name}{length}
```

The length may be any of the allowed T_EX lengths shown in Table 2.1. If an author preferred to have paragraphs in an entire document indented by one inch, the following command, `\setlength{\parindent}{1in}`, would be included in the preamble of the document. If the command is instead issued within the text of the document, the change is valid until the end of the current environment is reached or until another `setlength` command is issued.

Ex: 2.1 Using the basic program format shown in figure 2.1 generate your first L^AT_EX 2_ε program. Save the program as “`first.tex`” and be sure to include the following:

- A Title
- Your Name (as author)
- One Complete Page of Text (at least)

Print out both the “.tex” and “.ps” versions of your document. If you encountered errors during compilation, also print out the “.log” file. Modify the document `first.tex` so that it is in two-column mode, with a font of 12 pt. Use the `setlength` and `columnsep` commands to vary the distance between the columns. Print out a PDF version of the resulting document.

2.1.4 Special and Command Characters

Part of the grammar of T_EX includes command characters. These include the characters `#$~_~%{}`. If these are to appear in text they must be preceded by a `\`. i. e., `\# =` produces `#`.

To create opening quotes, the ‘ symbol should be used and the ’ or " symbols should be used to produce single or double closing quotes, respectively. In addition, there are certain special characters that do not appear on the typewriter, but that may be produced with simple commands. These are:

†	<code>\dag</code>	§	<code>\S</code>	©	<code>\copyright</code>
‡	<code>\ddag</code>	¶	<code>\P</code>	£	<code>\pounds</code>

As with all programming languages T_EX allows authors to make comments which are hidden from the compiler using the % command character, as illustrated in the following code:

```
The price should read, "$11.75/#. "           The price should read, ‘‘\$11.75/\#. ’’
                                           % Check this price!
```

Notice that the comment does not show up in the output.

2.1.5 Common Commands and Environments

A L^AT_EX 2_ε program will generally consist of **plain-text**, **commands**, and **environments**. Whereas plain-text is any character or string of characters (excluding the special and command characters) commands and environments are predefined or derived constructs. In general commands are preceded by the \ symbol and have the following syntax:

```
\name[option]{argument}
```

Examples of low-level commands with no options or arguments include `\textwidth` and `\LaTeXe`, which when used yield the width of a line of text in a document and type out the L^AT_EX 2_ε logo, respectively. Commands which require arguments include the various sectioning commands. i. e.,

```
\section{}
\subsection{}
\subsubsection{}
```

and the `\footnote{}` command. As the names indicate, these commands allow an author to divide a document into sections, sub-sections, and sub-sub-sections as needed and to add footnotes to the text. In the case of the sectioning commands, the necessary arguments are the desired title, and numbering of the various levels is be done automatically by L^AT_EX 2_ε. In the case of footnotes, the entire text is the argument, and is set in a slightly smaller font.

There are a multitude of predefined environments in L^AT_EX 2_ε. Like commands, environments can take options and arguments or may be quite simple and require no options or arguments. One such environment is the `center` environment, which is instantiated with the following code:

```
\begin{center}
  document elements
\end{center}
```

Any code placed within this environment is automatically centered on the page. i. e.,

<pre>This is an example of how text is automatically centered by the "center" environment.</pre>	<pre>\begin{center} This is an example of how text is automatically centered by the "center" environment. \end{center}</pre>
--	--

Other examples of environments include the `flushleft`, `flushright`, `verse`, `quote` and `itemize` environments. In each of these environments the names give a clear indication of

what is accomplished. In the case of the `itemize` environment however, an author needs to also use the `item` command. The syntax for the `itemize` environment is illustrated in the following example:

i) First Item	
ii) Second Item	<code>\begin{itemize}</code>
iii) Third Item	<code>\item[i]] First Item \\ \item[ii]] Second Item \\ \item[iii]] Third Item \\ \end{itemize}</code>

Here the “i), ii), and iii)” within the square brackets were options chosen by the author. In principle any construct could be used. If no option is given, nothing will delineate the list, and if the square brackets are left off altogether a bullet (•) appears.

Note that within most environments a `\\` command will cause a line to be ended. If an additional vertical spacing of 20 pt is desired the command,

```
\\[20pt]
```

may be used. More generally vertical or horizontal spacing may be inserted into a document with the following commands:

```
\vspace{value}
\hspace{value}
```

Many of the more advanced environments may be inserted using the drop-down menus in WinEdt. As shown in Figure 2.4 this is accomplished by first clicking on the **Insert** tab, dragging down to the **Environments** tab and then choosing among the options.

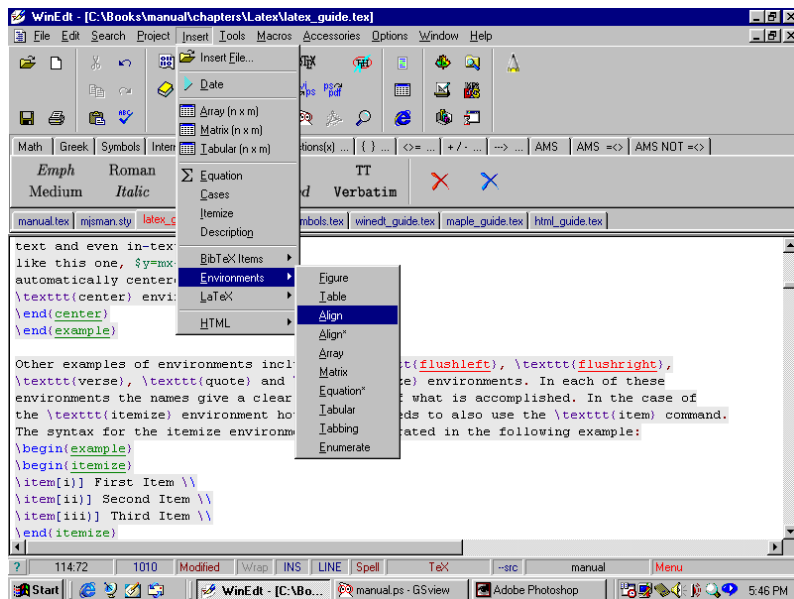


Figure 2.4: Many of the advanced environments available with L^AT_EX 2_ε are accessible with the drop-down menus in WinEdt.

2.1.6 Font Sizes and Faces

In addition to specifying the size of the font to be used for the entire document it is possible to use special commands to change the fonts and sizes locally. For example, the command `\tiny{text}\normalsize` produces, `text` and then returns the font size to the default(10pt Roman) setting. The font size commands available are