

Using L^AT_EX at VTSU

Peter Chapin*
Vermont State University

Last Revised: June 11, 2024

Abstract

This article describes how to install and use L^AT_EX for the creation of lab reports and similar documents as required in certain courses at Vermont State University (VTSU). This article is not a general tutorial on L^AT_EX. Instead it focuses on quickly showing you how to handle the immediate formatting needs of VTSU's courses.

1 Introduction

In certain courses you will be required to write lab reports using the L^AT_EX typesetting system. L^AT_EX is a powerful document formatting tool that can produce professional quality output. It is used in the printing industry to typeset certain books, particularly those with a lot of mathematical content, and by academic researchers publishing scholarly papers. Learning how to use the system is worthwhile but it does require some effort. The intent of making L^AT_EX a requirement in this course is to force you to take the effort necessary to learn L^AT_EX basics. I think you will find that for simple documents L^AT_EX is not hard to use and that the high quality results it produces promotes a satisfying feeling of pride in one's work.

2 Installation

L^AT_EX may be installed on the lab machines you are using. However, there are L^AT_EX distributions for all commonly used environments: Windows, macOS, and Linux. You will likely want to install L^AT_EX on your own system.

*peter.chapin@vermontstate.edu

To prepare L^AT_EX documents you can use an ordinary text editor. Many advanced text editors support special editing modes or extensions for L^AT_EX. There are also L^AT_EX IDEs. In this document, I will describe how to set up Visual Studio Code for editing L^AT_EX documents.

If you are interested in installing L^AT_EX on your own machine, the best way to proceed depends on your host operating system. In the subsections below I describe my recommendations for several different hosts.

2.1 Windows

On Microsoft Windows I recommend MiKTeX [1]. There are two installers a “basic” version and a “complete” version. The basic version installs a minimal set of packages whereas the complete version installs a much larger set. Notice that the installer for the complete version actually downloads the packages during the installation.

By default the basic installer configures MiKTeX to download missing packages automatically when they are first used. This gives you access to a rich collection of packages without taking up space on your system for packages you will never need. The disadvantage of this configuration is that you need network access whenever a new package needs to be downloaded.

2.2 Linux

Most Linux distributions have L^AT_EX as one of their standard packages. In fact there are often a bewildering array of packages related to T_EX and L^AT_EX. Try typing `pdflatex` at a command prompt to see if the system is already installed. If not, use your distribution’s package manager to install it. If you can’t find “latex” listed as a package by itself try looking for “texlive.”

2.3 Mac OS

On Mac OS systems you might try the MacTeX distribution at <http://www.tug.org/mactex/>. This entails a large download followed by a fairly standard installation process.

2.4 Visual Studio Code

Although you can use any text editor to prepare a L^AT_EX document, you may want to use an editor with L^AT_EX-specific features. I recommend Visual Studio

Code (VSCode) as a good editor for writing L^AT_EX documents. It also had the advantage of being cross-platform to Windows, macOS, and Linux.

There are a bewildering number of VSCode extensions with “LaTeX” in their names, so it’s hard to know which ones are essential. I recommend the “LaTeX Workshop” extension. I also recommend the “L^AT_EX LanguageTool grammar/spell checking” extension installed. I believe LaTeX Workshop will automatically install it. When you first load a LaTeX file, the L^AT_EX extension downloads the grammar-checking engine, which is fairly large, that’s a one-time download.

LaTeX Workshop will try to compile your document every time you save it. If you prefer different behavior, you can change the setting “LaTeX *i* Auto Build: Run” to “never.” In that case, you have to compile the document manually when, and only when, you are ready.

Another extension I recommend is the Rewrap extension. It adds a command (Alt+Q on Windows and Linux, Option-Q on macOS) that rewraps the current paragraph to a specified width. Writing each paragraph as a single, very long line is problematic in general, even the long line is wrapped in the window of *your* editor. However, manually adjust line lengths after editing a paragraph isn’t fun. With the Rewrap extension, You can just type Alt+Q (Option+Q on macOS), and the paragraph is adjusted to look good with no fuss.

3 L^AT_EX Demonstration

This section demonstrates various L^AT_EX markup. You can use this as a reference. Be sure to compare the formatted version of this document against the L^AT_EX source used to create that output. This should help you to understand the basic L^AT_EX commands. Feel free to copy constructs out of this document into your own documents when appropriate.

You can use **bold text**, *italicized text*, and even ***bold italicized text***. However, it is normally better to avoid specifying font faces and instead use more abstract commands that describe the intended effect. For example you can *emphasize* a word, format **file names** in a special way, and format **commands** in a special way.

Notice that the handling of file names and commands are by way of custom macros defined at the top of the document. By changing the definitions of these macros you can easily change the way a certain entity is formatted throughout your document. You can define your own macros as well. In fact L^AT_EX’s underlying macro engine, T_EX, is a full Turing complete programming language. Many advanced formatting functions can be written as T_EX macros.

3.1 Lists

L^AT_EX supports various list making environments. These lists can be nested.

1. The first sort of list is an enumerated (numbered) list.
 - (a) This is a nested enumerated list.
 - (b) Two...
 - (c) Three...
2. The second sort of list is an itemized list with bullets.
 - This is a nested itemized list.
 - An item...
 - An item...

It is also possible to create custom lists environments using a format you define. This document does not demonstrate that technique, however.

3.2 Mathematics

L^AT_EX's support for formatting mathematics is second to none. In fact, the system was originally created in part to make typesetting complex equations easy while still producing very high quality output. You may not need to write elaborate equations in your documents, but if you do, L^AT_EX makes it easy. Knowing the commands to format equations is useful for another reason: those commands often appear in other contexts.

All mathematical formula should be enclosed in $\$$ characters: $c^2 = a^2 + b^2$. This includes formula that involve just numbers: $1 + 2 = 3$. This is because the rules for typesetting math are different than the rules for typesetting ordinary text. You can also use Greek letters in math mode: The asymptotic running time of QUICKSORT is $\Theta(n \log n)$. This is far superior to BUBBLESORT's $\Theta(n^2)$ running time. The difference is especially significant for large n .

There is no need to use a special "equation editor" as you might find in a word processor. Once you know the commands to format equations, you can type them naturally along with whatever other text you want. In fact, some word processors let you use L^AT_EX commands to format equations if you don't want to deal with all the menus and buttons you would otherwise have to navigate.

Large equations can be displayed by themselves centered on the page using a `displaymath` environment.

$$r_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

It is also possible to create an array of equations. Use double backslashes to end each line in the formatted output. The ‘*’ in the name of the environment suppresses the numbering of the equations. In an equation array the & symbols are used to separate the left, central, and right components of the equation. It is common for the central component to be an = symbol but any appropriate formula is also fine. If \mathcal{A} , \mathcal{B} , and \mathcal{C} are wfs of L , then the following are axioms of L

$$\begin{aligned} & (\mathcal{A} \rightarrow (\mathcal{B} \rightarrow \mathcal{A})) \\ ((\mathcal{A} \rightarrow (\mathcal{B} \rightarrow \mathcal{C})) \rightarrow ((\mathcal{A} \rightarrow \mathcal{B}) \rightarrow (\mathcal{A} \rightarrow \mathcal{C}))) \\ (((\neg \mathcal{B}) \rightarrow (\neg \mathcal{A})) \rightarrow (((\neg \mathcal{B}) \rightarrow \mathcal{A}) \rightarrow \mathcal{B})) \end{aligned}$$

To truly appreciate L^AT_EX’s powers for handling mathematics, consider Maxwell’s equations governing the behavior of electromagnetic fields as follows.

$$\oint \mathbf{E} \cdot d\mathbf{L} = -\frac{d}{dt} \int_S \mathbf{B} \cdot d\mathbf{S} \quad (1)$$

$$\oint \mathbf{H} \cdot d\mathbf{L} = \int_S \left(\mathbf{J} + \frac{\partial \mathbf{D}}{\partial t} \right) \cdot d\mathbf{S} \quad (2)$$

$$\oint_S \mathbf{D} \cdot d\mathbf{S} = \int_V \rho dV \quad (3)$$

$$\oint_S \mathbf{B} \cdot d\mathbf{S} = 0 \quad (4)$$

$$\oint_S \mathbf{J} \cdot d\mathbf{S} = -\frac{d}{dt} \int_V \rho dV \quad (5)$$

Notice that the equations above are numbered. This allows you to refer to them nicely in the text. For example, Equation 2 is Ampere’s Law. The term $\partial \mathbf{D} / \partial t$ was added by Maxwell to fix a mathematical inconsistency that otherwise existed between the equations. This term is called *displacement current*. Equation 5 is the Continuity Equation. It says that the total current leaving a volume of space equals the rate at which the charge contained in that volume decreases.

3.3 Program Listings

You may not need to include any calculus in your lab reports. However you may wish to include program fragments. You can do this using a the `\code` macro defined at the top of this document. The instructor said

The declaration `int x = f(y)` introduces `x` as a variable of type integer and initializes it to the value returned by `f(y)`. The value of `f(y)` is computed when the program runs and is thus called a *dynamic initializer*.

Be aware that the underscore character is treated in a special way by L^AT_EX. If you want to use it in your in line code fragments you must escape it with a backslash. For example, the expression `line_counter + 1` computes a value one more than the current `line_counter`.

For short multi-line listings you should use the `lstlisting` environment provided by the `Listings` package. That package provides many nice services and is even programming language aware. Notice that the text inside the `lstlisting` environment is formatted as is. There is no need to escape underscore characters.

```
while (*s) {
    if (*s == ch) return_value = s;
    s++;
}
```

The `Listings` package allows for a lot of configuration. I refer you to the documentation for all the details.

Any listing larger than three or four lines is usually best put into a figure environment and then referenced in the text where appropriate. Listing 1 shows the technique. Unlike the previous examples, this listing “floats.” This allows L^AT_EX to place it where it fits and it also guarantees that the listing won’t be split across a page boundary. However, you must label the listing and explicitly refer to it in your text using the `\ref` command.

In general, I recommend putting a frame around any large listing so the text of the listing is clearly separated from the normal text of your document. This makes the document significantly easier to read and is especially important if the listing has floated to a location other than where you reference it.

3.4 Verbatim Text

You may sometimes want to include some text in your document that is preformatted but that is not a program listing. The text of a terminal session is one common example. For this purpose you could use L^AT_EX’s `verbatim` environment. However the standard `verbatim` environment has many limitations that make it relatively inflexible. Thus it is often better to use the `Fancyvrb` package which provides a more powerful `verbatim` environment named “`Verbatim`” (notice the upper case ‘V’).

Like the `lstlisting` environment the `Verbatim` environment renders the characters exactly as you type them using a fixed width font. However, the `Verbatim` environment does not attempt to do any syntax highlighting.

Listing 1: Initializing a SLAB Cache

```

kmem_cache_t *gfs_icode;

int init_module(void)
{
    gfs_icode = kmem_cache_create(
        "gfs_inode_icode",
        sizeof(struct gfs_inode_info),
        0,
        SLAB_RECLAIMACCOUNT | SLAB_MEM_SPREAD,
        init_once,
        0);

    if (gfs_icode == 0) {
        printk(KERN_ERR
            "GenericFS: Can't allocate inode cache.\n");
        return -ENOMEM;
    }
    /* The rest of init_module. */
}

```

```

student@hackbox:~/LaTeX$ ls -l
total 712
-rw-r--r-- 1 student student 19655 2011-12-16 07:49 LaTeX.tex
...

```

As with listings any verbatim text that is longer than a few lines should probably be put into a figure and referenced via an appropriate label. As with program listings such verbatim figures should be surrounded by a frame so the text in the floating figure is clearly set off from the normal text. Figure 1 show how it can be done.

Notice how the `Fancyvrb` package can be told to number lines. That is useful if you want to draw your reader's attention to something: Line 12 in Figure 1 is the directory information for the source of this document as it exists at the time of this writing.

The examples above use the '\$' character literally in the Verbatim environment. Its special meaning as the math-mode toggle is disabled. However, because the verbatim environment provided by the `Fancyvrb` package is not standard, your L^AT_EX editor may not understand its effects and thus may become confused about the bare '\$' characters. This is an unfortunate consequence of dealing with a programmable typesetting system. Without executing all the T_EX macros itself, an editor can't always reliably render the raw L^AT_EX text.

```

1 student@hackbox:~/LaTeX$ ls -l
2 total 692
3 -rw-r--r-- 1 student student 10677 2009-12-27 11:27 Block-Diagram.odg
4 -rw-r--r-- 1 student student 14998 2009-12-27 11:27 Block-Diagram.pdf
5 -rw-r--r-- 1 student student 4038 2011-12-15 21:01 lab-template.tex
6 -rw-r--r-- 1 student student 3575 2011-12-15 20:58 LaTeX.aux
7 -rw-r--r-- 1 student student 536 2011-12-15 20:44 LaTeX.bbl
8 -rw-r--r-- 1 student student 811 2011-12-15 20:44 LaTeX.blg
9 -rw-r--r-- 1 student student 12736 2011-12-15 20:58 LaTeX.log
10 -rw-r--r-- 1 student student 671 2011-12-15 20:58 LaTeX.out
11 -rw-r--r-- 1 student student 468854 2011-12-15 20:58 LaTeX.pdf
12 -rw-r--r-- 1 student student 19655 2011-12-16 07:49 LaTeX.tex
13 -rw-r--r-- 1 student student 652 2011-12-15 20:44 references.bib
14 -rw-r--r-- 1 student student 146361 2009-06-08 10:49 Tiki.jpg

```

Figure 1: Full Directory Listing

Table 1: Results of Timing Measurements

<i>Size (KiB)</i>	<i>Normal (ms)</i>	<i>Optimized (ms)</i>
8	2.33	1.19
16	5.28	3.45
32	12.32	8.41

3.5 Tables

L^AT_EX has a powerful table language that allows you to format tables in a number of ways. In this document I only demonstrate simple table handling. I refer to you to a more complete tutorial on L^AT_EX for information about formatting tables [2]. Note that in most cases it is appropriate to put a table in a floating environment so L^AT_EX can place it anywhere necessary. As with all floats, you must then label the table and refer to it explicitly in your text.

For example, Table 1 shows the results of timing both unoptimized and optimized executions of a certain algorithm. The timings are given as a function of the input size.

3.6 Figures

Including graphic figures in L^AT_EX documents tends to be somewhat awkward. The precise graphic file formats that will work depend on the ultimate target format you are generating. Formats such as PNG, JPEG, or PDF are generally the best when producing PDF output. The graphic itself is stored in a separate file and pulled into your document with a suitable command. Figure 2

demonstrates how it works.



Figure 2: “I want what you’re having.”

Notice that it is possible to scale the graphic to an appropriate size. Often dumping a raw graphic into your document looks terrible because it is much larger than is appropriate. Of course one could scale the graphic file outside of *L^AT_EX* as well.

If you need to include some line drawings, create the drawings outside of *L^AT_EX* using some suitable drawing tool. It is best to save the drawings in PDF format (so be sure your tool can do that) since PDF is a vector graphics format (by default). This means the drawings can scale up or down to an arbitrary degree while still looking smooth. It is also easy for *L^AT_EX* to merge PDF graphics into a PDF document. However, since PDF files can’t easily be edited separately you will need to also save your drawings in some editable format as well.

For example, using LibreOffice Draw you can save a small figure, such as the one shown in Figure 3 as an `odg` file and as a `pdf` file. *L^AT_EX* will use the PDF version when it compiles your document, but you can use the ODG version when you have to update or modify the figure.

When creating figures with a tool such as LibreOffice Draw you should also be sure to resize the paper to tightly frame your figure. Otherwise the figure will contain all the blank space on the page as well and look horrible when it is merged into your document.

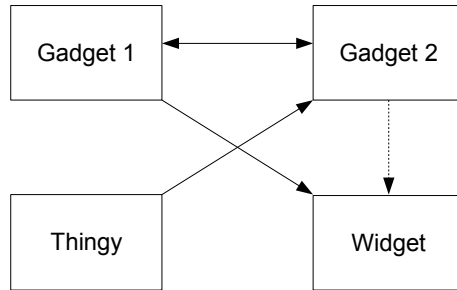


Figure 3: A simple block diagram

3.7 References

As you’ve seen various entities in your document can be labeled with the `\label` command. You can then refer to that entity using a `\ref` command. \LaTeX will automatically replace your reference with an appropriate number. This was done in Section 3.2 where labeled equations were referenced in later paragraphs. Notice that if you are using the `hyperref` package, these cross references are automatically hyper linked.

You can reference URLs in the text of your document with the `\url` command. For example, my personal web site is `https://www.pchapin.org/`. The `hyperref` package makes such URLs clickable in the generated PDF file.

Using an auxiliary tool called BibTeX, you can also format and reference bibliography entries at the end of your document. It is not necessary to do this for lab reports, but if you are interested in using \LaTeX to write a formal paper you may want to look into this feature. This document demonstrates a few citations managed by BibTeX.

4 Writing Services

Writing a high quality document requires attention to matters of spelling, punctuation, usage, and style. Technical people sometimes downplay the importance of such things but that is short sighted. Writing is the means by which you communicate your knowledge to others. If you are unable to write well then everything you know is useless to the rest of society.

\LaTeX itself is not a word processor and does not include any spell checking or grammar checking services. However, your \LaTeX editing environment might support those things. For example, there are VSCode extensions that provide

both grammar and spell checking for L^AT_EX documents, as described in Section 2.4.

5 Compiling L^AT_EX

Once you have completed your document you need to compile it into a PDF. This can be accomplished using the `pdflatex` command. For example if your document is named `lab1.tex` you compile it by doing

```
$ pdflatex lab1.tex
```

L^AT_EX output is fairly noisy but if an error is encountered processing will stop and you'll be prompted to make a choice about how to handle that error. For example you might see

```
! Undefined control sequence.
1.464 interested in using \LaTeX
                                \ to write a formal paper...
?
```

In this example I spelled LaTeX incorrectly and the resulting command is not recognized. At the '?' you can type another '?' to get a list of options but the most usual action is to type 'x' to exit the processor. You can then go back to your document and fix the error.

Because of the way L^AT_EX handles cross references you often have to run the processor more than once to get all the cross references correct. You will see a warning about this if you need to run L^AT_EX again. If you are using BibTeX the usual procedure is to run L^AT_EX three times like this

```
$ pdflatex lab1.tex
$ bibtex lab1
$ pdflatex lab1.tex
$ pdflatex lab1.tex
```

This is necessary so that the bibliography is formatted properly and all citations are properly linked to their bibliographic entries.

When you run L^AT_EX you may see warnings about "Overfull hboxes." These warnings usually arise because of a lines that are too wide to fit inside the page margins. The amount of overflow is given in *points* where each point is 1/72 of an inch. If the distance is only a couple of points you can probably ignore it.

However if the distance is large you should adjust your document to remove the warning.

Typically overfull hboxes come up in program listings or verbatim text. You can often just reformat your program to avoid the problem. For verbatim listings you can sometimes also change the formatting but using a smaller font, as demonstrated in Section 3.4, can be a good option. If the overfull hbox arises in ordinary text it usually means \LaTeX was not able to figure out how to hyphenate some unusual word. You can add hyphenation hints with the `\-` command. See the source of this sentence for an example. Note that hyphenation hints do not force hyphenation; they are only used by \LaTeX if necessary. You can also sometimes just reword a sentence to avoid the difficult words or to move them to a different location.

References

- [1] MiKTeX project page. <http://www.miktex.org/>. Accessed December 2011.
- [2] Tobi Oetiker. The not so short introduction to latex2e. <http://tobi.oetiker.ch/lshort/>. Accessed December 2011.