

L^AT_EX for Windows – A User’s Perspective

David M. Tulett

Faculty of Business Administration
Memorial University of Newfoundland
St. John’s, NF
Canada, A1B 3X5
dtulett@mun.ca

Introduction

Among operating systems for personal computers, the Microsoft Windows family (95/98/ME/NT/2000) has over 90% of the market, and it is on this market on which this paper focuses. For printing text on to paper, most users of Windows use Microsoft Word. Word can be bought as a stand-alone package, or it can be bought as part of the Office suite package, which contains many popular programs such as Excel and PowerPoint. There are other word processing packages for Windows users, such as Corel WordPerfect, but this package mostly appeals to those who have always used it and have remained loyal to it, or to those who are very price-sensitive. There are also desktop publishing systems such as Corel Ventura, Quark Express, and Adobe PageMaker. These latter programs are quite expensive, and the technical advantages over Word have diminished as Word continues to add new features. These word-processing and desktop publishing systems are all WYSIWYG (what you see is what you get). While to a certain extent all these packages compete with L^AT_EX, in this paper the scope is restricted to a comparison between the hugely-successful Word and L^AT_EX which is by comparison a niche product.

One major advantage of L^AT_EX is that it does a very good job of typesetting mathematics. Even something as simple as $x + y = z$ looks better in L^AT_EX than it does in Word. The difference between the two approaches widens considerably when typesetting something more complex, such as:

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{bmatrix}^{-1} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

The second major advantage (though it may seem like a disadvantage at first) of L^AT_EX over Word is its different paradigm for document creation. In contrast to the WYSIWYG approach, the user of L^AT_EX specifies the structure and lets L^AT_EX handle the design of the document. The advantage of this approach over WYSIWYG has been extensively described – see, for example, Love

[10] for a discussion of L^AT_EX versus Word, and Taylor [12] for a critique of WYSIWYG in general.

This paper addresses the issue of why L^AT_EX has nowhere near the numbers of users that Word has. I know of many people with technical backgrounds who have never tried L^AT_EX and indeed I know of some who have given up on L^AT_EX and have switched to Word. This paper is written from the perspective of an ordinary user of L^AT_EX, not someone who is a computer programmer. The rest of the paper is organized as follows. First, a personal historical background of how I became interested in L^AT_EX is presented. Secondly, a comparison of L^AT_EX and Word is made. Thirdly, we look at other options. Finally, recommendations are given.

Historical Background

The Dark Ages (Before L^AT_EX) In this section I wish to explain how I became interested in L^AT_EX. During my engineering undergraduate education (1971-75), all assignments involving mathematics (except my thesis) could be hand-written, and were. For prose essays, I used an electric typewriter. In 1975, I wrote my undergraduate thesis on a typewriter, using a special ball to handle mathematical symbols. I was glad that this was the only requirement for this technology. After working for several years, I entered a doctoral program in management science, and wrote my dissertation in 1985. While there were developments in the T_EX world at that time, they were not available to me. What was available (on the university’s mainframe) was a program for word-processing which had limited ability to write mathematical text. Writing something like $\alpha = \beta + \gamma$ was easy to do, but creating a display equation was not. For example, the L^AT_EX expression `\[y = \sum_{i=1}^n x_i \]` creates

$$y = \sum_{i=1}^n x_i$$

Making this equation back then required writing an n on one line, $y = \sum x_i$ on a second line, and finally writing $i = 1$ on a third line. This would have to all be set in a non-proportional font and then trial-and-error would have to be used to make sure that all the symbols lined up correctly. It goes without saying that putting the three

lines together produced an equation which looked terrible. At the time I thought that this word processor was a tremendous improvement over my 1975 experience, but of course hoped that something better would become available.

In 1985 I began working for my current employer. In the previous year the Faculty of Business Administration had acquired a large number of Macintosh computers for faculty and staff, and I began to use the Mac for everyday things like making up tests. At first MacWrite was the only word processing program, but then Microsoft Word was released, long before it came to Windows. In 1988 a colleague and I decided to make up a set of notes for a course which we taught. I decided to have a look at what was available.

The Middle Ages (L^AT_EX 2.09) In searching for a software package to help us write our book, I did some reading on the WYSIWYG packages, and saw references to L^AT_EX, which was then in version 2.09. We decided to adopt it, mostly because of its ability to create nice-looking equations. The other advantage of L^AT_EX, that it used logical rather than visual design, seemed at the time to be a disadvantage – after three years of using a Mac it was hard to leave an established paradigm. This was especially true given how we printed the dvi files. We had terminals to the mainframe, which of course let us view the ASCII-based tex file, but there was no way to preview the dvi file. The 300 dpi laser printer was located in another building about 300 metres away.

To learn L^AT_EX I purchased the first edition of Lamport's book [8]. I read it over a weekend to get the general idea of what it was all about, and then read it again trying to learn the content. Even with this preparation, I needed to have the book next to my side for the first several months. (Even today, I often need to consult a book for a particular problem.) By contrast, I learnt how to use Word without ever reading the manual.

During this period with a terminal on my desk, my 512K Mac had become obsolete, and I obtained an IBM PC mostly to use a lot of common business software such as the then-popular Lotus 1-2-3. The university had a site license for a commercially-made version of L^AT_EX, and for about¹ C\$50 I had it installed on my machine. This was still 2.09, but now I could view dvi output on my monitor.

One nagging problem at the time was what to use as an editor for creating the tex file. I tried several things, but eventually settled on using WordPerfect, simply saving the file in ASCII format. (I needed WordPerfect anyway, for communication with non-T_EX users.) Com-

pared with what I had had only a few years earlier, this setup seemed to be the cat's meow.

I had become a proficient user of L^AT_EX and now saw limitations in what had once been a package with so many new things. The biggest limitation was graphics. I can remember trying to approximate a parabola by drawing a sequence of short straight lines. Even straight lines had a small finite set of angles from which to choose. For making a problem involving two-dimensional linear optimization, I would make the objective function and the constraints so that when drawn they would be at angles which L^AT_EX could handle, which is surely the tail wagging the dog!

The Renaissance (L^AT_EX 2_ε) Walking through a bookstore in 1995, I came across the second edition of Lamport's book [9]. After reading it, I followed Lamport's recommendation to obtain *The L^AT_EX Companion* [3], and I inquired at the university about upgrading the software. The company from which the site license had been obtained was no longer selling L^AT_EX, and we were left to fend for ourselves. We obtained a two-CD set of the CTAN archive, and my colleague managed to figure out how to use it despite the lack of instructions. The effort to upgrade L^AT_EX was in my opinion justified by the new `\qbezier` command, but we soon found other useful features, such as the ability to: print on legal size paper; use colour; import graphics; and import new packages. In particular, I was glad to see the **times** and **mathptm** packages had been made for creating words and mathematical characters in Times-Roman fonts. The Computer Modern fonts which I had used up to this point never looked as nice as Times-Roman.

When I obtained Internet access, I discovered a new world, soon finding the web page for TUG, from which many other resources could be accessed. Joining TUG provided among other things annual editions of the T_EX Live CD. A colleague in the mathematics department recommended the Professional File Editor (PFE) to me, and this was a substantial improvement over editing files in WordPerfect. Later I found out about WinEdt, and for only \$40 I then had something even better for creating tex files.

One problem that remained was that I was a L^AT_EX user in a predominantly non-L^AT_EX environment. I could not expect, for example, my students (business majors) to be able to read dvi files. However, on the web page for the commercial vendor PCT_EX, I found a free dvi viewer called DVIScope, which I set up on my computer. This viewer had all sorts of new features, such as magnification and the ability to print only selected pages,

¹ Prices are in US dollars unless indicated as C\$ to mean Canadian dollars. These figures are approximate and may have changed after March, 2001.

and I soon recommended this to my students. However, installing the viewer was not easy. Then I discovered Portable Document Format, and now instead of having the students install the dvi viewer, I recommend the Adobe Acrobat reader, which is useful for other courses and indeed other purposes. Of course, the reader only became useful because of the creation of the **pdf_{tex}** program and other ways of creating pdf files. Related to the use of **pdf_{tex}** are the **hyperref** and **url** packages for creating internal and external references.

To learn how to use some of features described here, I added three more books to my collection: *The L^AT_EX Graphics Companion* [5]; *A Guide to L^AT_EX* [7] for a more recent general-purpose book; and *The L^AT_EX Web Companion* [4]. Along with *L^AT_EX: A Document Preparation System* [9] and *The L^AT_EX Companion* [3], these five books comprise my current L^AT_EX library. An additional resource for the basics of L^AT_EX is the *Not So Short Introduction to L^AT_EX 2_ε* available on the Web from [11].

L^AT_EX versus Word

Are There Problems with L^AT_EX? Much has been said about the long learning curve for L^AT_EX when compared with Word. However, this is not all that fair when the two packages are asked to do different things. If all that one wanted to do was write prose, a half-hour spent on learning L^AT_EX would be sufficient. To write equations, more time is needed to learn L^AT_EX but then more time would be needed to learn the Equation Editor in Word. Specialized packages for L^AT_EX have their own learning curves, but their equivalents in Word (if they exist) require learning too.

Another complaint about L^AT_EX concerns the lack of variety of fonts. This was a very valid concern when all L^AT_EX had was Computer Modern. Now, the set of available fonts is adequate for most purposes. Indeed, when using Word with its very large number of fonts, I only use two of them: Times New Roman (for most things); and Courier (when a non-proportional font must be used). For L^AT_EX the free **times** and **mathptm** fonts are adequate for my purposes.

Every other perceived inadequacy of L^AT_EX has in my opinion been addressed. The T_EXlive CD and new editors such as WinEdt have improved the user friendliness, and all the new packages have greatly improved the functionality. Being able to write to pdf has improved the accessibility of completed L^AT_EX documents to non-L^AT_EX users.

Word – The Ubiquitous (Sub)-Standard At my office Microsoft Windows 2000 is the “standard” operating system, and Microsoft Office 2000 is the standard applications software. These standard products are provided at no charge to the user, and training and other help is

available. Non-standard products like Linux and L^AT_EX are permitted, but at the user’s own expense,² and with no training or help provided. I understand why some level of standardization is necessary - at one time we had Mac-OS, DOS, and Windows 95, and even within one operating system there would be both WordPerfect and Word. In an environment where documents need to be shared, standardization helps bring order out of chaos. At the same time, it’s hard to see how L^AT_EX could flourish in this environment. Among forty-five members of faculty, three use L^AT_EX for Windows, and one uses L^AT_EX for Linux. On a positive note, Adobe Acrobat Reader has recently become a standard, enabling users of L^AT_EX to share files by using pdf format.

At least at the office I am *allowed* to use L^AT_EX. The same cannot be said for some professional societies. The Administrative Sciences Association of Canada (ASAC) *requires* that all articles for its annual conference or for publication in the *ASAC Bulletin* be submitted in either Word or WordPerfect. I recently received a call for papers for the ANZAM/IFSAM VIth World Congress (Australia, 2002) with the same requirement.

Another example of Word being imposed as a standard was when I developed some materials for an on-line course. The details of creating the web pages were being handled by a private company, which needed to have all input in Word. Wanting to be cooperative, I did everything in Word, learning how to use its Equation Editor for the first time. Just as I was finishing this document, everything started to freeze. It seems that I had reached a limit for the number of display equations in a document. I had to convert a couple of display equations to in-text equations to finally finish the document. In a way I’m glad that I had to make this document in Word rather than L^AT_EX, because I had a chance to truly evaluate Word as an alternative to L^AT_EX. To be fair, there are some good points about Word:

1. Word is very easy to learn. Whenever I’ve needed to learn something specific, such as how to make a footnote, all I’ve had to do is use the pull-down help menu.
2. Fonts are plentiful, and can be switched at will.

Overall, however, I believe that L^AT_EX is a superior product for document creation because:

1. L^AT_EX handles mathematics well, both in creation of the document and in printing the final product.
2. For everything but very short documents, logical design beats visual design.

Obtaining Word and L^AT_EX Though there are many retailers of Word, only Microsoft makes it. Word (as part

² To be fair, we are given C\$400 per annum which could be used to buy non-standard software, but this money could also be used to buy books or computer peripherals.

of Office) ships with many high-end personal computers. Many companies have site licenses for Word or Office. For those who need to buy the program, it's expensive for business use, but very inexpensive for academic use. Where I work, Word 2000 can be purchased for C\$117 (about \$75).

L^AT_EX for Windows can be obtained both as free-ware and commercial packages. Every member of TUG (currently \$65 per annum for an individual) obtains the latest version of L^AT_EX from the annual T_EXlive CD. In addition to obtaining L^AT_EX, a text editor is needed (such as the shareware program WinEdt for \$40), and a book such as *A Guide to L^AT_EX* [7] (about \$40) should be obtained. In total, the cost to get started is about \$145.

This cost can be reduced, because one could obtain L^AT_EX from CTAN for free, but I believe that those who benefit from all the work that goes into the development of L^AT_EX should pay something for it, and this something is the annual \$65 cost of belonging to TUG.

At the other end of the scale, one can pay far more than \$65 plus \$40 to obtain a commercial version of L^AT_EX with its own editor. The commercial vendors of L^AT_EX for Windows as listed on the TUG website are (in alphabetical order): MicroPress V_TE_X; PCT_EX; True_EX; and Y & Y.

The premium packages from these suppliers sell for several hundreds of dollars (though some offer older or more basic packages for much less). I do not own any of these packages, and being puzzled as to why someone would buy one, I wrote to all four companies. The most detailed response was from Y and Y [13], in which the following points for buying a commercial system (particularly theirs) were made: ease of setup; access to support; additional features (e.g. cut and paste to PowerPoint); and better fonts. It would be useful if someone were to properly evaluate all four of these commercially available packages and compare them with each other and with what is on the T_EXlive CD, but that is beyond the scope of this paper.

Finding out About Word and L^AT_EX It is probably true to say that every user of L^AT_EX has heard about Word. How many Word users have heard about L^AT_EX? In my experience, most people have never heard of L^AT_EX let alone have any knowledge about it. This situation may upset us, but it shouldn't surprise us. Bookstores offer a multiplicity of books about Word, but only a few high-end bookstores carry anything about L^AT_EX. Schools will have almost always have either Word or WordPerfect, but rarely have L^AT_EX. A new computer may have a word processing package bundled with it, but it won't have L^AT_EX. A word search made in March 2001 on *PC Magazine* shows that the last five mentions of L^AT_EX go back to 1997; the last five mentions of Word go back only to

the last two issues. Clearly, it is easy to never have been exposed to L^AT_EX, and this problem must be addressed.

Even among those who have heard of L^AT_EX I would offer the conjecture that most have never tried it, and I know of some who have tried L^AT_EX only to later abandon it.

Other Options

LyX LyX is a program which tries to combine the typesetting ability of L^AT_EX with the WYSIWYG feel of Word, though LyX call this WYSIWYM (what you see is what you mean). LyX began on the Unix operating system, but has been ported to other operating systems, and in particular it has been ported to Windows by Claus Hentschel (based on previous work by Steven van Dijk). LyX requires that L^AT_EX be installed on the user's computer, and at the present time there is a laborious process to get LyX installed and running. If those who use Word are doing so in part because setup is easy, then I don't believe that they will experiment with LyX. As for the established base of L^AT_EX users, not having WYSIWY(G or M) on the screen is not a serious disadvantage, as one can always use **pdf_{tex}** before the tex file has been completed to see how it looks so far. If the day comes when LyX with all the necessary ancillary programs for Windows comes on a CD with an automatic install feature, then it may well improve the use of L^AT_EX by Windows users, but we're certainly not there yet.

Converters Another approach is to use a "converter". Such a program translates the output from a package such as Word into L^AT_EX, or *vice versa* [6]. Going from L^AT_EX to Word might be useful, for example, for someone who has written a paper in L^AT_EX and now wishes to submit it to a journal or conference which requires submission in Word. Nevertheless, to me it seems like changing wine into water, because all the elegant mathematical typesetting is lost. Going in the Word to L^AT_EX direction, however, could be useful to someone who wants the ease of Word combined with the functionality of L^AT_EX. The Word2_TE_X \$45 shareware program [2] performs such conversions, but not always flawlessly. The examples provided on the website of files in .doc, .tex, and .pdf formats are quite impressive. However, when I used the program to convert a Word file that I had created, the converter made incorrect guesses about the \section and \subsection commands. I would make the conjecture that the Word2_TE_X program works well when the original Word document is well structured (perhaps by using styles), but flounders when the original document has been made completely in the WYSIWYG paradigm in which most Word users operate. The program merely creates tex files; one still needs a L^AT_EX system to create a dvi or pdf file.

XML and Epic There has been much attention paid to the subject of how to write mathematics on the web. This subject, which is extensively described in [4], is one area where both L^AT_EX and Word have problems. My opinion is that for short discussions, the ability to see equations on a browser is useful, but for anything longer than a couple of pages the natural tendency is to want to print the document. This being the case, simply using pdf files (which are easily produced using pdfL^AT_EX) gives far better quality.

Arbortext [1] claims that its Epic E-content Engine is able to translate a wide variety of what they call “legacy” formats (including Word and L^AT_EX) into XML. Also, the Epic Editor creates XML documents from scratch. Epic, the company claims, enables the user to create a single source XML document from which versions for print, Web, and wireless can be made. Since XML will eventually replace HTML, this may be a company to watch. I know of one major corporation which once used L^AT_EX for technical documents, but has switched to Epic.

What Needs to be Done

Do we care if people use alternatives to L^AT_EX? Those who use products like LyX or Epic are using products which have tried to move beyond L^AT_EX. Any improvement to L^AT_EX will probably help these other products too. However, if someone is using Word instead of L^AT_EX then they have something which is deficient in several ways when compared with L^AT_EX. Nevertheless, if that’s what they choose to do, knowing that they could switch to L^AT_EX, then we have to respect that choice. However, I believe that L^AT_EX has few adherents when compared with Word because most people have never heard of it, and those that have may have overestimated its complexity. For these people, I think that we should proclaim what we know to be a better product. Unlike the commercial vendors of L^AT_EX whose profitability would improve if L^AT_EX were more prominent, the rest of us have nothing to gain financially by encouraging the use of L^AT_EX. However, more users might improve the development of L^AT_EX, in particular the L^AT_EX 3 Project.

The T_EX Users’ Group has greatly helped the technical improvement of L^AT_EX. Perhaps TUG needs to focus more of its efforts on the promotion of L^AT_EX. With TUG’s blessing, perhaps the T_EX Live 5 CD could be bundled with all books about L^AT_EX. Indeed, this has already happened with the German edition of *A Guide to L^AT_EX* [7]. We could go even further than this – the CD could be bundled with new computer systems. An editor would be required as well – perhaps PFE would be sufficient at the outset. A pdf file (or even a plain ASCII text file) could contain more information such as how (and why) to join TUG, how to obtain WinEdt or something similar, and a bibliography of books about L^AT_EX.

In summary, I believe that L^AT_EX is superior to Word, especially for documents which contain mathematics. However, for a variety of reasons, Word is many times more popular than L^AT_EX. To increase the number of L^AT_EX users, we need to make it very easy for people to obtain the L^AT_EX software, possibly by widespread distribution of the T_EX Live 5 CD.

References

- [1] Arbortext. Epic, 2001. <http://www.arbortext.com:80/>.
- [2] Kirill A. Chikrii and Anna V. Chikrii, 2001. <http://www.word2tex.com/>.
- [3] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The L^AT_EX Companion*. 1994.
- [4] Michel Goossens, Sebastian Rahtz, et al. *The L^AT_EX Web Companion*. Addison-Wesley, Reading, Massachusetts, 1999.
- [5] Michel Goossens, Sebastian Rahtz, and Frank Mittelbach. *The L^AT_EX Graphics Companion*. Addison-Wesley, Reading, Massachusetts, 1997.
- [6] Wilfried Hennings. Converters between L^AT_EX and PC Textprocessors, 2001. <http://www.kfa-juelich.de/isr/1/texconv/texcnv.html>.
- [7] Helmut Kopka and Patrick W. Daly. *A Guide to L^AT_EX*. Addison-Wesley, Reading, Massachusetts, third edition, 1999.
- [8] Leslie Lamport. *L^AT_EX: A Document Preparation System*. Addison-Wesley, Reading, Massachusetts, first edition, 1985.
- [9] Leslie Lamport. *L^AT_EX: A Document Preparation System*. Addison-Wesley, Reading, Massachusetts, second edition, 1994.
- [10] Tom Love. Why L^AT_EX?, 2000. http://www.h.eng.cam.ac.uk/help/tpl/textprocessing/latex_advocacy.html.
- [11] Tobias Oetiker et al. The not so short introduction to L^AT_EX 2_ε, November 2000. <http://www.tex.ac.uk/tex-archive/info/lshort/english/lshort.pdf>.
- [12] Conrad Taylor. What has WYSIWYG Done to Us?, 1996. <http://www.ideography.co.uk/library/seibold/WYS-ante.html>.
- [13] Y & Y. Personal communication, 2001.