

Using WordPerfect 5.0 to Create T_EX and L^AT_EX Documents

ANITA Z. HOOVER

University of Delaware
002A Smith Hall
Newark, DE 19716
anita@vax1.acs.udel.edu
ACS03174@UDACSVM

ABSTRACT

This paper addresses the issues associated with using WordPerfect 5.0 on PCs to create T_EX and L^AT_EX documents. WordPerfect macros and keyboard layouts are used as an interface to aid in the input process of T_EX and L^AT_EX control sequences. As a result, a major portion of this paper is devoted to discussing the basic concepts of WordPerfect macros, keyboard layouts, and the macros specifically developed to produce T_EX and L^AT_EX control sequences.

1. Background

In 1984, the University of Delaware's administration decided that WordPerfect would become the faculty's standard word processing package, and so they initiated a grant program which allowed faculty to purchase IBM PCs with WordPerfect at a considerable savings. By 1987 a similar decision had been made for the secretarial staff; near the end of 1988 the actual implementation of this decision was completed, and WordPerfect became the standard word processing package for PCs throughout the University.

During the last year, however, many departments, such as Engineering, Computer Science, Math and Physics, realized that producing technical documents was beyond the scope of WordPerfect, especially those documents that contained complicated mathematics. Many faculty members in these departments had heard of the high-quality mathematical typesetting that T_EX and L^AT_EX produced and thought that either of these packages would be an appropriate tool for producing their documents. It was this interest that brought T_EX and L^AT_EX to the PC environment here at the University of Delaware.* But the problem was to train the secretaries to use these packages. As usual, documents needed to be typed by yesterday, and the secretaries became overwhelmed and frustrated by having to learn both WordPerfect and the proper control sequences for T_EX and/or L^AT_EX in a hurried manner. Many questions began to surface, for example, "Why do I have to type `\alpha` to get α when I used to type only `Alt-A` on my EXXON system?"

In trying to answer such questions, I realized that WordPerfect macros could be used to help speed up the learning and typing process by automatically providing the basic control sequences for setting up a T_EX or L^AT_EX document. The more common math control sequences, such as those that produce lowercase and uppercase Greek characters, are defined in a WordPerfect keyboard layout that contains a number of predefined macros. I should note that it is *not* my intention to tell a new user of T_EX or L^AT_EX to go out and buy a PC with WordPerfect in order to use the macros I developed to produce T_EX

*In trying to reproduce this paper at other sites, it was brought to my attention that it would not T_EX successfully because the `save size` was exceeded. Our site requires a large version of T_EX for producing books, but I did not think this paper would cause any problems. However, upon closer examination I realized that the figures used to create the keyboard layouts (4 and 5) proved to be quite complex and thus required a `save size` of 800 or larger. I decided to use our time-sharing system Vax1 (VAX 8650 running version 4.3 Berkeley UNIX) to make communication easier for submitting this paper, so I have not actually run this on a small system and therefore you may experience difficulties due to memory and processor speed limitations.

and \LaTeX control sequences. However, these macros do provide new users of \TeX and \LaTeX , already familiar with WordPerfect, with a good starting point for creating high-quality technical documents in \TeX and/or \LaTeX format.

2. WordPerfect Macros

A WordPerfect macro provides a way of assigning a name to a series of keystrokes you often repeat. If you need to perform the same operation frequently, you can enter the keystrokes once and save them in the form of a macro. When you want to repeat the operation at a later time, you need only tell WordPerfect the name of the macro (the file where the keystrokes are stored), and WordPerfect will “replay” the keystrokes, automatically performing the task. This concept is common to most software packages that provide macros. For example, WordPerfect macros work much like \TeX and \LaTeX macros, which are shortened forms of your most commonly used control sequences or combined control sequences that you use repeatedly. So it seems that putting \TeX and \LaTeX control sequences into WordPerfect macros is not as unrelated as it may have appeared upon first inspection.

It is quite easy to create a WordPerfect macro if you follow three basic steps:

1. Plan the macro.
2. Save any files before you start writing the macro.
3. Write the macro.

2.1 Plan the Macro

You should have a clear idea of what you want the macro to accomplish before actually writing the macro. If you are creating a simple macro, such as one that produces $\backslash\alpha$, only a few keystrokes may be required. However, if you are creating a more complicated macro, such as prompting the user for an environment name in \LaTeX in order to produce the output shown in Figure 1, it becomes more important to write down the basic steps and actions necessary in order for the macro to work correctly in both new and already existing documents.

```
\begin{environment name entered}
~
\end{environment}
```

Figure 1: Macro output

2.2 Save the Current File

As you create a macro, the functions you use are actually performed on the text on your screen. WordPerfect is a WYSIWYG word processor, so while you can create a macro on a blank screen, having a document on the screen enables you to see the effects of each function you include in the macro. Since your text will be altered by the keystrokes you enter, you should always save your file before creating a macro.

2.3 Write the Macro

1. Press the **Macro Def** key (Ctrl-F10)[†] The message Define Macro: appears in the bottom-left corner of the screen.

[†]Conventions used in this document: Text appearing in typewriter font refers to text you type or a key sequence you press, text appearing in bold is a descriptive name of a macro or key definition. Remember, what one writes is not always what one means, so I hope you will not find typewriter font where bold should be and vice versa. Just keep in mind that when you see the word hold, press, or type, then I want you to carry out the action that follows, otherwise I am just referring to that macro or key definition by name.

2. Name your macro; this can be done in two ways:

- (a) Type a name 1–8 characters long (conforming to the standard rules for naming DOS files) and then press the **Enter** key. This will create a file name with the extension `.WPM`. All named macros for this project were of the form `M-xxxxxx`. This was done in order to eliminate confusion with user-defined macros. For example, `m-be` is a named macro, stored in the file `M-BE.WPM`, that prompts the user for an environment name and produces the output in Figure 1; see Figure 2 for the description and WordPerfect codes of this macro.

Many macros can be created using this method. For this project, named macros were created for the following circumstances:

- If the control sequence is more than three characters long, including the backslash character (`\`).
The reason for limiting what control sequences should be defined as named macros is based on the naming convention chosen for this project. If the control sequence name is too short, it is not advantageous to have to type a longer name for the macro.
- If the control sequence is made up of two separate control sequences, as in a \LaTeX environment; for example, `\begin{equation}` followed by `\end{equation}`.
This would save time typing, plus help prevent errors which arise when an ending environment control sequence is forgotten.

See Table 1 for a listing of the named macros defined for this project and a short description of each.

- (b) Hold down the **Alt** key and press a single letter to create an **Alt** macro. This will create a file name with the extension `WPM`. For example, if you were to hold down the **Alt** key and press the letter **A**, then you would create the macro **Alt-A** and the file name would be `ALTA.WPM`. The basic lowercase Greek characters were assigned to the **Alt** macros in this project. For example, **Alt-A** is a macro that produces the `\alpha` control sequence. Such key assignments are very common on multi-function keyboards. See Figure 3 for the description and WordPerfect codes of this macro.

3. Enter a description of the macro (up to 39 characters). Most macro descriptions for this project stated whether the macro was designed for \TeX and/or \LaTeX and the control sequence(s) it produces.

2.4 Comments

Using the naming method described above, all lowercase Greek characters were defined as **Alt** key macros (**Alt-A** . . . **Alt-Z**). Therefore, uppercase Greek characters had to be given 1–8 character names. However, it became obvious that a WordPerfect keyboard layout could expand the current number of single-keystroke macros. By using a WordPerfect keyboard layout, you can assign macros to the **Ctrl** key as well as the **Alt** key. Therefore, the uppercase Greek characters could be assigned to the **Ctrl** keys, plus other frequently used \TeX and \LaTeX control sequences could be assigned to the other available **Alt** and **Ctrl** keys (for example, **Alt-;** and **Ctrl-**). In order to proceed with this task, we need to examine the following:

- What is a keyboard layout?
- How do you create a keyboard layout?
- How does a keyboard layout work?

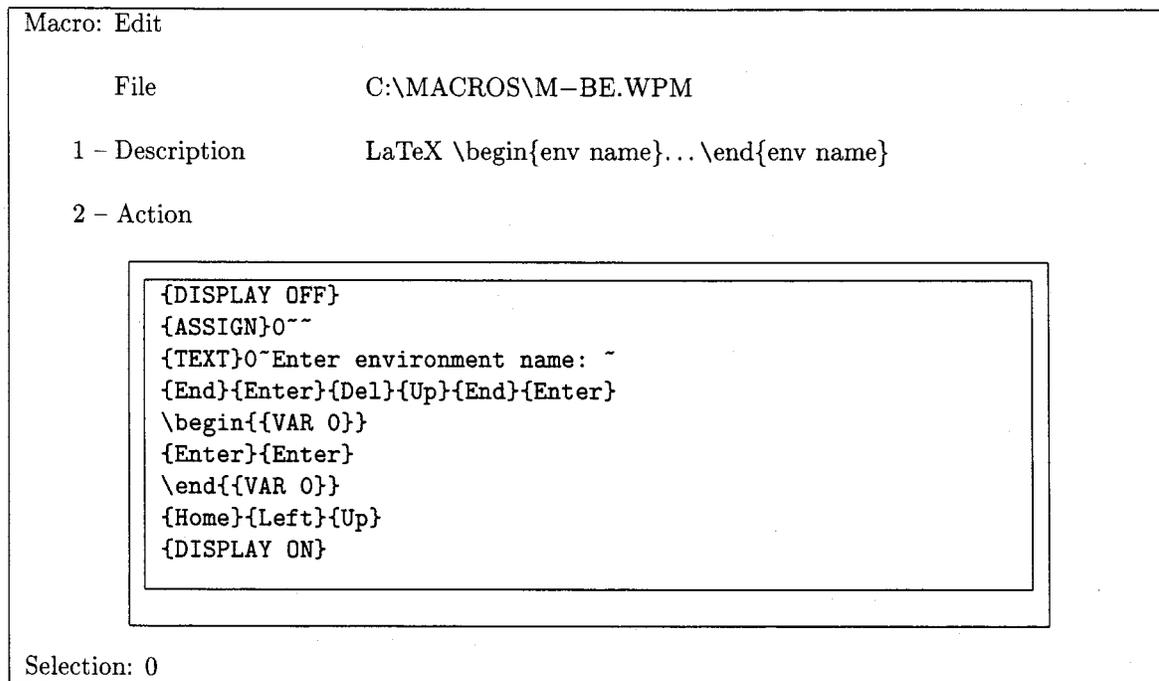


Figure 2: WordPerfect macro for any \LaTeX environment control sequence

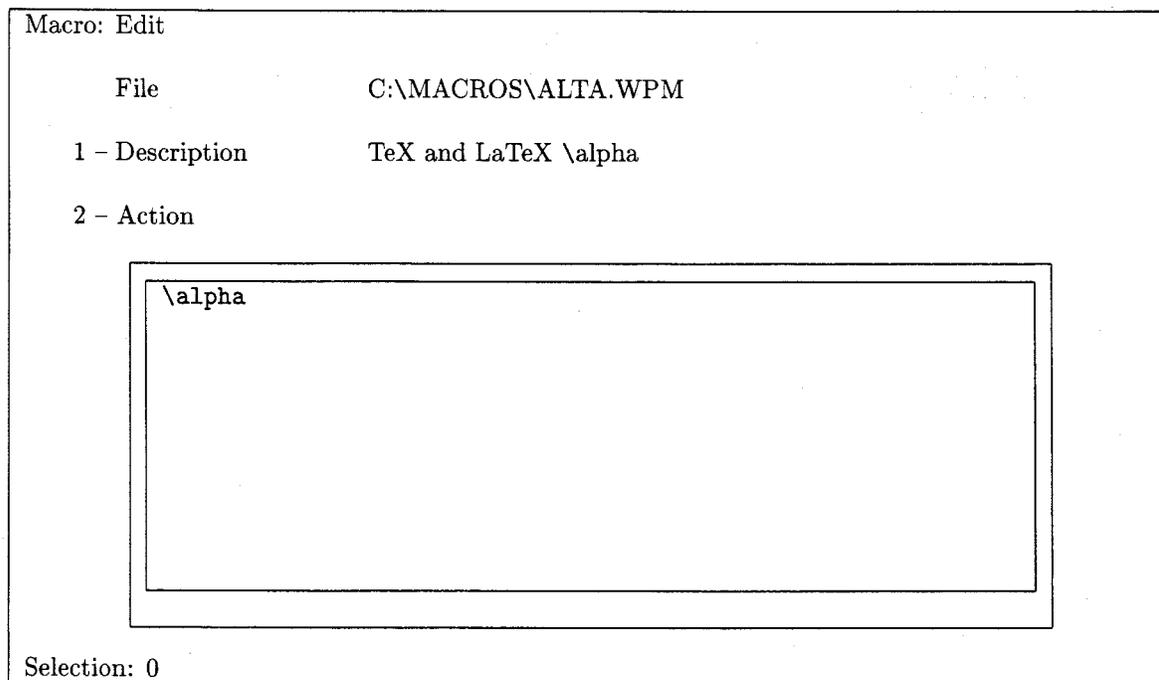


Figure 3: WordPerfect macro for \TeX and \LaTeX \alpha control sequence

\backslash Gamma, Γ	\backslash Delta, Δ	\backslash leftarrow, \leftarrow	\backslash Theta, Θ	\backslash rightarrow, \rightarrow	\backslash Upsilon, Υ	\backslash Xi, Ξ	\backslash uparrow, \uparrow	\backslash downarrow, \downarrow	\backslash ell, ℓ
Q	W	E	R	T	Y	U	I	O	P
\backslash gamma, γ	\backslash delta, δ	\backslash epsilon, ϵ	\backslash theta, θ	\backslash tau, τ	\backslash upsilon, υ	\backslash xi, ξ	\backslash iota, ι	\backslash circ, \circ	\backslash rho, ρ
\backslash nabla, ∇	\backslash Sigma, Σ	\backslash Phi, Φ	\backslash approx, \approx	\backslash Lambda, Λ	\backslash hbar, \hbar	\backslash Pi, Π	\backslash leftrightarrow, \leftrightarrow	\backslash Omega, Ω	
A	S	D	F	G	H	J	K	L	
\backslash alpha, α	\backslash sigma, σ	\backslash phi, ϕ	\backslash mid, $ $	\backslash lambda, λ	\backslash eta, η	\backslash pi, π	\backslash kappa, κ	\backslash omega, ω	
\backslash simeq, \simeq	\backslash equiv, \equiv	\backslash Psi, Ψ	\backslash propto, \propto	\backslash infty, ∞	\backslash sim, \sim	\backslash partial, ∂			
Z	X	C	V	B	N	M			
\backslash zeta, ζ	\backslash chi, χ	\backslash psi, ψ	\backslash times, \times	\backslash beta, β	\backslash nu, ν	\backslash mu, μ			

Ctrl
Normal Key
Alt

Figure 4: Key definitions for Alt-A ... Alt-Z and Ctrl-A ... Ctrl-Z

3. WordPerfect Keyboard Layout

A WordPerfect keyboard layout provides the ability to reassign, or remap, keys on the keyboard. This enables all of the **Alt** and **Ctrl** keys to be mapped to any function you want. In this project, all seventy-seven possible keys combinations have been defined to produce the most frequently used \TeX and/or \LaTeX control sequences in a single keystroke. See Figure 4 for the description of the **Alt-A...Alt-Z** and **Ctrl-A...Ctrl-Z** key definitions for this project. See Figure 5 for all other **Alt** and **Ctrl** keys defined for this project.

It is quite easy to set up a keyboard layout if you follow three basic steps:

1. Plan the key definitions.
2. Create a keyboard layout.
3. Define the keys.

3.1 Plan the Key Definitions

Before you create a keyboard layout, you should plan what keys you will define and what control sequence(s) each key will produce. This is very similar to defining a macro, so you should have a clear idea of what you want the key to accomplish before actually defining it.

3.2 Create a Keyboard Layout

1. Press the **Setup** key (Shift-F1).
2. Press 6 to select the **Keyboard Layout** option.
3. At the **Setup: Keyboard Layout** screen, press 4 to select the **Create** option. The message Keyboard filename: appears in the bottom-left corner of the screen.

Type a name 1-8 characters long (conforming to the standard rules for naming DOS files) and then press the **Enter** key. The keyboard layout for this project has been named **TEXLATEX**

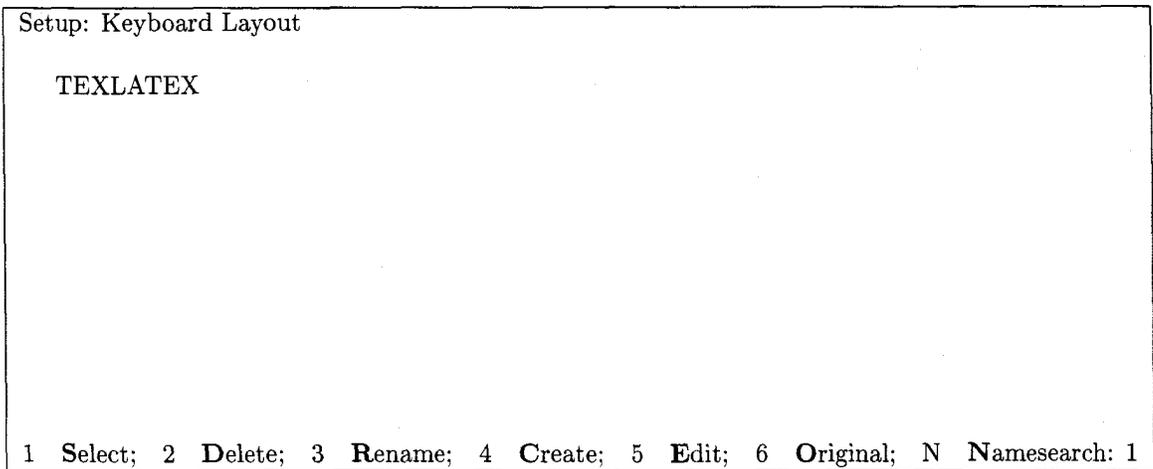


Figure 6: WordPerfect **Setup: Keyboard Layout** screen

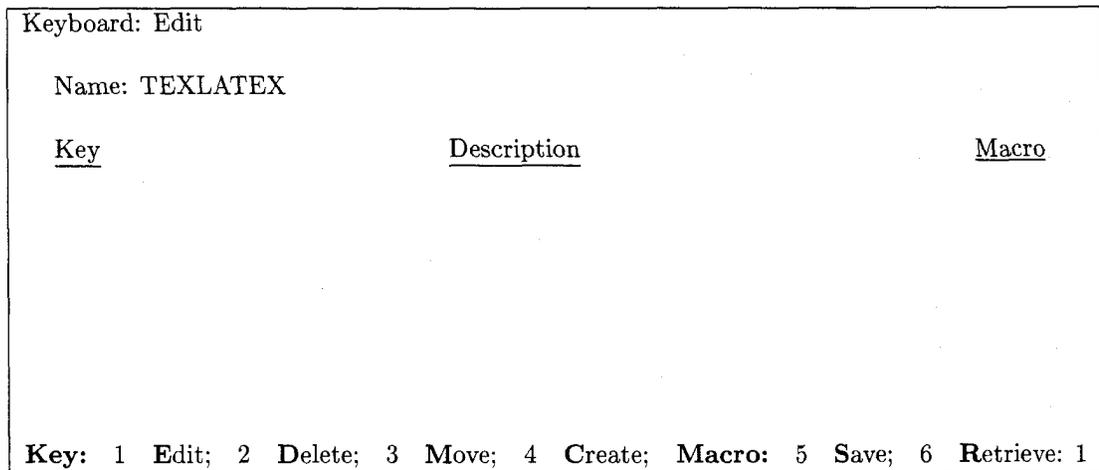


Figure 7: WordPerfect **Keyboard: Edit** screen

3.4 Comments

Now that the keyboard layout is in place, it is necessary to assign to the **Alt** keys the macros that are already stored in macro form as **ALTx.WPM** and to the **Ctrl** keys the macros that are already stored in macro form as **M-xxxxxx.WPM**. This is done quite easily by using the **Macro: Retrieve** option described on the previous page. For example, to assign **ALTA.WPM** as the **Alt-A** key definition, at the **Keyboard: Edit** screen do the following:

1. Press 6 to select the **Macro: Retrieve** option.
2. The message Key: appears in the bottom-left corner of the screen. Hold down the **Alt** key and press the letter **A**.
3. The message Macro: appears in the bottom-left corner of the screen. Type the name of the macro to be assigned to this key and press the **Enter** key; for this example, type **alta**.

You will now see an entry made on the **Keyboard: Edit** screen defining the **Alt-A** key. See Figure 8 for the updated screen.

Keyboard: Edit		
Name: TEXLATEX		
<u>Key</u>	<u>Description</u>	<u>Macro</u>
Alt-A	TeX and LaTeX \alpha	1

Key: 1 Edit; 2 Delete; 3 Move; 4 Create; **Macro:** 5 Save; 6 Retrieve: 1

Figure 8: WordPerfect **Keyboard: Edit** screen with the **Alt-A** key defined

Once the macro has been integrated, you may or may not need to edit it. If you do, it is not difficult: press 1 to select the **Key: Edit** option from the **Keyboard: Edit** screen and notice that the key edit screen for key definition is exactly the same as the macro edit screen for macro definition. See Figure 9 for the description and WordPerfect codes of this key definition.

4. Using WordPerfect Macros and the Keyboard Layout

It is important to keep in mind that the use of these macros and this keyboard layout is no different from the normal WordPerfect environment. However, two steps must be taken in order to use the macros and the keyboard layout:

1. Define where the keyboard layout and macro files will reside.
 - (a) Press the **Setup** key (Shift-F1).
 - (b) Press 7 to select the **Location of Auxiliary Files** option.
 - (c) Press 3 to select the **Keyboard/Macro Files** option.
 - (d) Type the directory name where the macros files and keyboard layout files are located (e.g., C:\MACROS) and press the **Enter** key.
If you already have defined a directory to store macro and keyboard layout files, then you must install the macros and keyboard layout to create T_EX and L^AT_EX documents in that directory.
 - (e) Press 0 ("zero") to return to the **Setup** screen.
2. Select the keyboard layout defined for creating T_EX and L^AT_EX documents.
Macros have been defined to select the **TEXLATEX** keyboard layout automatically and also to switch back to the original keyboard layout which is needed when creating normal WordPerfect documents. See **Changing Keyboard Layouts** in Table 1 for these macros.
 - (a) Press 6 to select the **Keyboard Layout** option.
 - (b) Use the up and down arrows to highlight the **TEXLATEX** keyboard layout. Press 1 to select this keyboard layout.
 - (c) Press 0 to return to your document.

You are now ready to use the keyboard layout and macros.

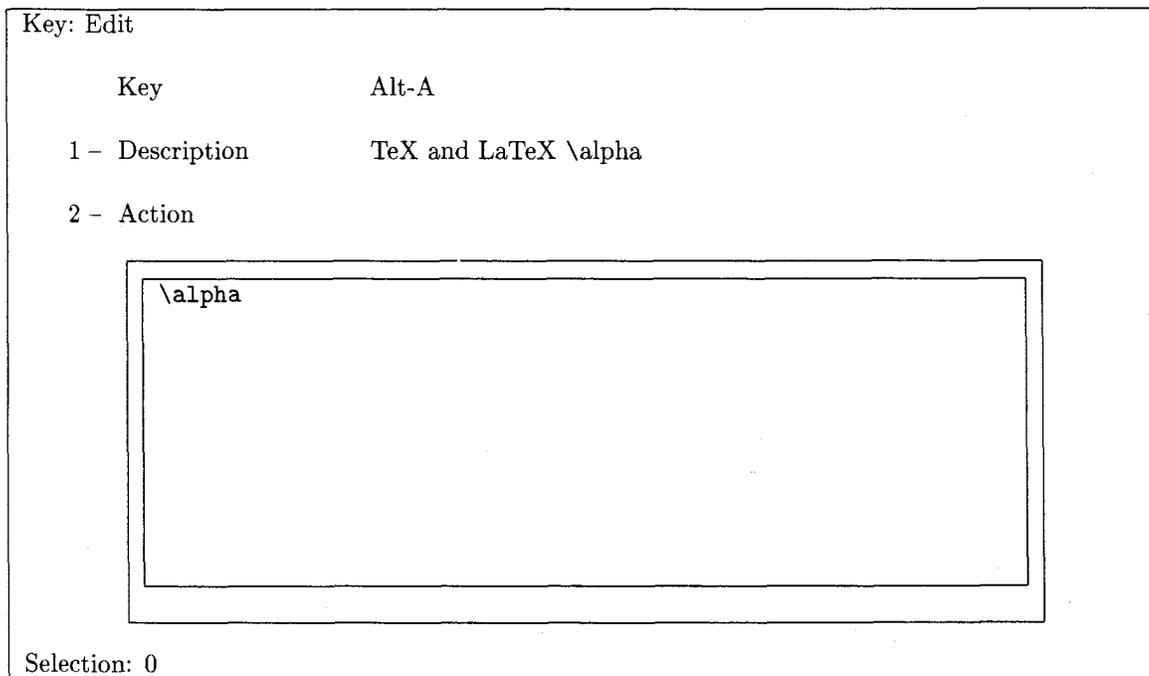


Figure 9: WordPerfect key definition for T_EX and L^AT_EX \alpha control sequence

5. Conclusions

The intent of this project was to help speed up the learning and typing process for setting up a T_EX and/or L^AT_EX document by automatically providing the basic control sequences. The project appears to have accomplished both of these with several interesting points:

1. The control sequences that were assigned to the **Alt** and **Ctrl** keys are common in most word processing systems, so there was very little time required to memorize which control sequences were produced by which key.
2. The amount of typing was drastically reduced as a result of being able to use a single keystroke in place of long control sequence names.
3. Using named macros also saved typing, but more importantly helped prevent errors when a multiple line control sequence such as those used to begin and end L^AT_EX environments were used in a document.
4. Repetitive use of the defined keys and named macros helped in the process of learning the actual control sequences for T_EX and L^AT_EX. Creating a T_EX and/or L^AT_EX document would be possible without the help of these aids; however, the overwhelming consensus of those who have used these WordPerfect macros and the keyboard layout is that the document would take more time to create as a result of having to type out all of the control sequences.
5. Don't get carried away! The control sequences assigned to keys or defined as named macros were carefully chosen in order to optimize typing time, reduce errors, and enhance learning. Make sure you take the time to evaluate whether or not a key definition or named macro for a particular control sequence will enhance the productivity of the user rather than make it worse. For example, why have the user type ALT-F10 and m-11 to produce << when they could type \ll instead.

In general, typing time was reduced, learning capability was enhanced, and the number of errors caused by misspelled control sequences or forgotten control sequences diminished.

5.1 Comments

This project was a success for two reasons.

1. The macro and keyboard layout feature of WordPerfect provided an easy-to-use interface for T_EX and L^AT_EX.

This concept can be applied to other word processing packages, providing technical secretaries with similar advantages who need to use T_EX and L^AT_EX in their jobs.

2. T_EX and L^AT_EX became less intimidating to the technical secretaries, which allowed them to put aside their frustrations and concentrate on their jobs.

This is a very important part of why non-technical users shy away from T_EX and L^AT_EX. We need to reach out and identify such tools to give these users the power and beauty of T_EX and L^AT_EX without the frustration. It will be the non-technical users who will shape the future of T_EX and L^AT_EX and broaden its scope of use.

6. Acknowledgements

I would like to thank Kenneth E. Gadomski of Academic Computing Support for his help in reviewing this paper, Karen M. Kral of Academic Computing Support for sharing her wealth of knowledge about WordPerfect macros and WordPerfect keyboard layouts, and Doris A. Wood and Denise J. Brzoska of the Physics and Astronomy Department for their help in choosing the T_EX and L^AT_EX control sequences that were mapped on the keyboard layout. I would especially like to thank Denise J. Brzoska for the time she spent evaluating the macros and the keyboard layout.

7. Update

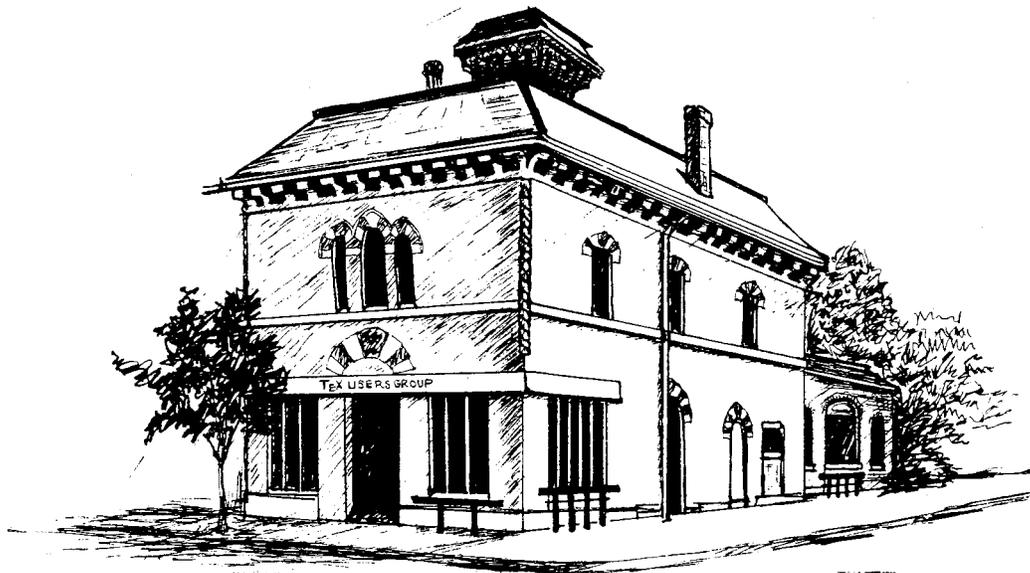
It has been brought to my attention that several key definitions in the keyboard layout did not work. After investigating the problem, I discovered that the keyboard being used was not an IBM keyboard, but in fact a Keytronic Professional Series KB515 (supposedly an IBM compatible). The key definitions that did not work were Ctrl-\, Alt-[, Alt-], Alt-;, Alt-', Alt-', Alt-., Alt-., and Alt-/.

Bibliography

- Academic Computing Support. *Creating and Using WordPerfect 5.0 Macros*. University of Delaware, Newark, October 1988.
- Academic Computing Support. *More About WordPerfect 5.0: Macros*. University of Delaware, Newark, May 1989.
- Knuth, Donald E. *The T_EXbook. Computers and Typesetting* Vol. A. Reading, Mass.: Addison-Wesley, 1986.
- Lamport, Leslie. *L^AT_EX: A Document Preparation System*. Reading, Mass.: Addison-Wesley, 1986.
- McComb, Gordon. *WordPerfect 5.0 Macros and Templates*. New York, NY: Bantam Books, 1988.
- WordPerfect Corporation. *WordPerfect for IBM Personal Computers*. Version 5.0, May 1988.

Math Macros	
m-eq	<code>\begin{equation} ... \end{equation}</code>
m-\$\$	<code>\$\$... \$\$</code>
m-dm	<code>\begin{displaymath} ... \end{displaymath}</code>
m-eqa	<code>\begin{eqnarray} ... \end{eqnarray}</code>
m-sr	<code>\stackrel{ }{ }</code>
m-ob	<code>\overbrace{ }</code>
m-ub	<code>\underbrace{ }</code>
m-ol	<code>\overline{ }</code>
m-ul	<code>\underline{ }</code>
Text Macros	
m-be	<code>\begin{env name you enter} ... \end{env name you enter}</code>
m-ctr	<code>\begin{center} ... \end{center}</code>
m-lst	<code>\begin{enumerate} ... \end{enumerate}</code>
	<code>\begin{itemize} ... \end{itemize}</code>
	<code>\begin{description} ... \end{description}</code>
m-part	<code>\part{ }</code>
m-chap	<code>\chapter{ }</code>
m-sec	<code>\section{ }</code>
m-ssec	<code>\subsection{ }</code>
m-sssec	<code>\subsubsection{ }</code>
m-par	<code>\paragraph{ }</code>
m-spar	<code>\subparagraph{ }</code>
m-ft	<code>\footnote{ }</code>
m-fig	<code>\begin{figure} ... \end{figure}</code>
m-tab	<code>\begin{table} ... \end{table}</code>
m-vs	<code>\vspace{ }</code>
m-hs	<code>\hspace{ }</code>
m-doc	<code>\documentstyle [] { }</code>
	<code>\begin{document} ... \end{document}</code>
m-font	font type changes
m-size	font size changes
m-tex	TeX logo <code>\TeX</code>
m-latex	LaTeX logo <code>\LaTeX</code>
Changing Keyboard Layouts	
m-spec	Selects the TEXLATEX keyboard layout
m-orig	Selects the ORIGINAL keyboard layout

Table 1: WordPerfect macros by name



TeX USERS GROUP
McGill University, Montréal
August 21-24, 1988