

\LaTeX : a \TeX Workbench for MS-DOS PC's

Wietse Dol, Erik Frambach, Maarten van der Vlerk

Department of Econometrics
University of Groningen
P.O. Box 800
9700 AV Groningen

W.Dol@eco.rug.nl, E.H.M.Frambach@eco.rug.nl, M.H.van.der.Vlerk@eco.rug.nl

Abstract

\TeX and all its companions offer an enormous amount of possibilities. This is both an advantage and a disadvantage. The advantage is that almost anything is possible; the disadvantage is that you need detailed knowledge of all related programs to fully exploit the possibilities.

The MS-DOS program \LaTeX is an attempt to integrate all major \TeX related programs in a shell that shields you from the tedious and frustrating job of setting environment variables and program parameters.

1 Introduction

Preparing documents with \TeX or \LaTeX is by no means a simple job. However, those who persevere are rewarded with a beautifully typeset document. There are at least two reasons why writing with the 'aid' of \TeX may cause some problems.

First of all there is the enormous amount of possibilities that \TeX offers. While this is the reason to use \TeX in the first place, it has the disadvantage that many—at first puzzling and maybe hard to remember—commands have to be used. In principle this is a problem that you will have to live with. A thought to comfort you: many people have tried before you and most of them are by now enthusiastic \TeX users.

The second reason why \TeX may seem difficult to work with is of a more technical nature. Since \TeX itself is only(!) a compiler, it needs several other programs and utilities to make a fully operational text preparation system. For example, at some time between conception and delivery of your document, you might need an editor, a previewer or a printing program. All these programs need setting of environment variables and/or some parameters to cooperate the way you want them to. This is of course just the type of thing you do not want to bother about.

So here is the good news: you don't have to! If you do not like to spend valuable time discovering all these technicalities, try \LaTeX . While in its infant days, it did nothing more than constitute a simple menu for calling the \TeX compiler or an editor and the like. Although it is a relatively young program and is still growing, we feel that by now it is powerful enough to justify the title *workbench*. In the next sections you find a description.

2 Principles of \LaTeX

The \LaTeX system consists of a host of separate programs, e.g. it uses $\text{em}\TeX$ as \TeX implementation, which is public domain but state-of-the-art. It is named after its author, Eberhard Mattes, from the Stuttgart University. You must prepare your \TeX input file with an ASCII editor (we suggest you use QEDIT); the actual \TeX compiler converts this input file into a .dvi file (from DeVice Independent); separate programs generate printed output from the .dvi file and allow you to preview the typeset page on your screen. In addition, there are a great many add-on utilities: a spell-checker; a database program for maintaining bibliographic references; an index generator; a utility to remove \TeX codes; a font generation program, extended graphics support etc.

Some of these programs, especially the various versions of the compiler and of the print and preview programs, require *lots* of parameters and/or environment variables. \LaTeX is designed to shield you from the dirty bits.

As mentioned above, the first objective in creating \LaTeX was to have some sort of integrating menu to call \TeX and the other main programs without the fuss of setting, let alone remembering parameters.

One obvious way to do this is by a *batch file*. However, plain `command.com` batch files tend to be slow, since only one command is read at a time. A much bigger draw-back is its lack of even the most basic commands for an interactive system. A very attractive alternative is provided by 4DOS (shareware by JP Software Inc.), nowadays a well known replacement of `command.com`.

What once started as a small and simple batch file, grew into a collection of batch files that currently comprises over 7000 lines of sometimes fairly sophisticated code. The main reason to program 4TEX in the 4DOS batch language and not in a higher programming language (e.g. Pascal or C) is that by using 4DOS we could create an *open system*, i.e. everyone can modify the workbench to suit personal needs and taste without the need of special tools or extra compilers. Another reason to use the 4DOS batch language is the availability of environment variables and variable functions that enabled us to do things that would require very tedious programming in a higher programming language. Since it is also fast (the complete batch file is read into memory at once), it was an easy decision to implement 4TEX as a 4DOS batch file.

One might object that using 4DOS batch files deprives the old-fashioned `command.com` users from the benefits of 4TEX. We happen to think that this would only be a mild punishment for not recognizing how good 4DOS really is. However, for those who have a good reason not to use 4DOS we also implemented 4TEX to run under `command.com`. This is of course done by loading 4DOS as a secondary shell.

Though 4DOS is very powerful, some routines had to be written in other programming languages, e.g. to support mouse operation and to select files. Some *public domain* and *shareware* programs are used, e.g. MARKNET.EXE, RELNET.EXE and PKUNZIP.EXE.

2.1 Freeware, shareware and commercial software

4TEX uses several public domain (*freeware*) and *shareware* programs.

You may use a shareware copy for an evaluation period of 21 to 30 days. The purpose of this evaluation period is to allow you to determine whether the program meets your needs before purchasing it. Once the evaluation period ends, you agree to either purchase a registered copy of the program, or to stop using it.

Using freeware and shareware only, we are able to distribute 4TEX without violating any law or agreement. As a 4TEX *user* you are supposed to pay for the shareware programs that are listed in the documentation (see Chapter 13, page 79). Remember that 4TEX could never have been build without these programs.

Commercial software is not used, though interfaces to a few programs (e.g. EUROGLot, VERTAAL! and WORD-FINDER) are implemented. Naturally commercial fonts can be added. All DVI-drivers are installed in such a way that this should be easy.

2.2 Setting up 4TEX

After installing 4TEX from the distribution files, some customization is required before 4TEX will run.

The most important files are `texuser.set` and

`system.set`. In these files all parameters that 4TEX needs are set. The file `system.set` contains values for the parameters that are needed for any user (e.g. what options does the compiler need, where are the fonts located), whereas `texuser.set` contains values for the parameters that are user specific (e.g. where TEX files are stored, what screen colors should be used).

As you might have guessed, 4TEX runs perfectly on a network. Because NOVELL NETWARE is the most popular type, 4TEX has an interface to run from such a network. However, only a few lines of code in one of the batch files need to be changed to make 4TEX run on any other type of network.

3 Features

In this section we will describe some features of the 4TEX workbench.

3.1 Add, delete, modify

4TEX aims to be an open system such that every user can add, delete or modify the 4TEX workbench to suit personal needs and taste. For instance 4TEX uses Babel: a simple way to generate TEX format files with multiple languages, and some control sequences to switch languages (and hyphenation patterns) within one TEX document. It is easy to adapt Babel for other/more languages than already used by 4TEX (i.e. English, German, French and Dutch). It is easy to add, modify or delete format files (e.g. 4TEX supports the New Font Selection Scheme).

Of course any user or department has specific printers. 4TEX currently supports matrix printers, LaserJets, DeskJets and PostScript printers. Adding or deleting a printer type should not cause problems. 4TEX lets you choose between local printers, network printers and printing to a file.

At this moment the spell-checker supports the languages Dutch, US English, English, French and German, but any other language can easily be added.

3.2 Help in 4TEX

There are several types of automated assistance available. For each item in every 4TEX menu there is a help screen.

The memory resident program TEXHELP is meant as partial replacement for the L^AT_EX and T_EX manuals. It is a hypertext system that can be called from the editor. For example, if the cursor is at the word `\documentstyle`, pressing the TEXHELP key results in a help screen that refers you to the topic `Document Styles`. Select this topic and you get a help screen that gives information about all valid L^AT_EX document styles and options. Moreover, there are cross references to all options and, among others, the commands `\flushbottom` and `\twocolumn`. It is also possible to jump directly to the index of TEXHELP or to review the last help screen. For most topics, the text

is taken from the L^AT_EX help system for MS-WINDOWS by Michael F. Reid, which is based on George Greenwade's help system for Vax-VMS computers. At this moment there are about 315 documented topics in the TEXHELP database.

Several *example* files exist, varying from a standard L^AT_EX file to more complex subjects, such as tables with asymmetric columns or creating multiple indexes. Example files can be viewed from the editor and, if desired, be inserted in the current document.

Finally, there is on-line T_EX documentation. This consists of documentation of e.g. style files. To save disk space, the documentation is kept in archives that are temporarily decompressed on selection.

3.3 Editing

We have chosen QEDIT (shareware by Semware) as the default editor in 4T_EX. For this editor, we have developed many macros. For example, you can enter a L^AT_EX environment defined by the commands `\begin{env} ... \end{env}` by picking it from a list; insert `\index{this}` behind the word `this` at the cursor position; or call the spell-checker (see below) to check the word at the cursor. Moreover, on loading a text file, the cursor will be placed at the exact location it was when you last edited the file (also available in the View-Block facility, see below).

3.4 Block-View: quick partial compilation

One of the most often heard objections to T_EX is that it is not a so called WYSIWYG system. On the other hand, T_EX users often reply that the acronym WYSIWYG is misleading: it should be WYSIAYG (What You See Is *All* You Get). We do not wish to solve this problem. Instead, 4T_EX offers a feature that we think both sides may like.

Imagine that you have just typed a very complicated formula. The natural way to check whether you did not make a mistake, is to see what it looks like. Or maybe you are just curious. However, to see this single formula you would have to leave the editor, compile the entire document, start the previewer, and find the formula. Then you need to return to the editor, and find the exact position where you left it. If you made a mistake, you have to do this all over again.

In 4T_EX this whole procedure can be performed by the touch of a few keys. Even better, only the preamble of your document and the formula are compiled, which makes it as fast as possible. To view a certain part of a document, mark it as a block in the editor and press the 'view-block' key. Alternatively, if you want to view an entire file, just press the 'view-file' key. If you are making slides, it makes sense to view a complete slide at a time. Therefore, in this case 4T_EX automatically defines the before mentioned block to be the current slide. If T_EX finds an error in the block, the user has the option to be returned to the editor at the line that

contains the mistake.

3.5 Spell-checking

In 4T_EX we use AMSPELL, a public domain program by A. Merckens, to check and correct spelling in T_EX documents. AMSPELL is basically a spell-checker for plain ASCII files, with some special features for dealing with T_EX files.

The basic idea behind this program is to make spell-checking easier by

- providing the context of the possibly misspelled word;
- offering alternatives;
- offering facilities for editing the word;
- automatically replacing misspelled words in your document;
- learning new words.

AMSPELL does not require T_EX commands to be removed from your document (deT_EXing). In fact, it will even interpret the standard accenting commands like `\`, `\'`, `\'` and will automatically use them while replacing misspelled words. When checking a T_EX file, AMSPELL will ignore all text between `$'s` and `$$'s`. Furthermore, AMSPELL will ignore parameters of the L^AT_EX commands `\ref`, `\pageref`, `\cite`, `\nocite`, `\label`, and all text between `\begin{equation}`, `\begin{eqnarray}`, `\begin{eqnarray*}`, `\[` and their counterparts like `\]`. You can change or expand the lists by means of environment variables.

3.6 Extended graphics support

T_EX has been developed with the idea that it should be possible to have a T_EX implementation for every operating system (MS-DOS, VMS, UNIX etc.). Another important feature of T_EX is that documents can be freely exchanged between operating systems (because documents are written in standard ASCII). Graphics, however, are machine dependent and the possibility to include graphics in T_EX or L^AT_EX depend on the operating system and the DVI-driver you are using.

A solution often used for including graphics in T_EX documents is to insert PostScript pictures in the document through `\special` commands. The `\special` command is ignored/passed on by the T_EX compiler but the PostScript DVI-driver knows how to insert the PostScript picture at the right place and in the right size in your document. The disadvantage of this method is of course that you can only include *PostScript* pictures in your document and that you need a PostScript printer to produce output.

The emT_EX DVI-drivers support a `\special` command to include black-and-white bitmapped pictures. Both this feature and the PostScript possibilities are used by 4T_EX to incorporate pictures in T_EX documents.

Graphic files come in many flavors. 4T_EX allows you to view, manipulate and include the following types of pictures in your T_EX documents:

- bitmapped pictures: GIF, TIFF, PCX, BMP, IFF, LBM, IMG, CUT, and PCL;
- vector pictures: HPGL and PostScript.

This is done by using the following freeware software: BM2FONT, HP2XX, PCLTOMSP, and GHOSTSCRIPT.

4TEX automatically chooses the appropriate conversion program, depending on the type of graphic file, or rather the file extension. Any conversion that 4TEX performs will result in either T_EX fonts (accepted by any DVI-driver) or both a PCX and EPS file (for emT_EX and PostScript resp.). Furthermore, a small T_EX file is produced that contains all the necessary commands to incorporate the picture in your document.

4 4TEX in practice

At the Department of Econometrics of the University of Groningen 4TEX is installed on a network, and its usage is logged. The time that 4TEX was running is logged and the number of calls to functions are counted.

To give you an idea of the activity we made a summary for the period of 21 January to 12 March.

<i># calls</i>	<i>function</i>
17202	editor
4273	compiler
3984	block-view
204	4TEX help
942	LaserJet print
9	PostScript print
13	matrix print
2501	preview (not block-view)
230	view log-file
189	spell-check
26	oneword spell-check
689	shell to DOS
175	BibT _E X
47	BiBDBase
216	view picture
6	bitmap to T _E X font
47	HPGL to PCX + EPS
54	PS/EPS to PCX
0	PCL to PCX
5	GraphicsWork Shop
45	MakeIndex
1	DeT _E X
0	UnT _E X
5	Wordlist
4	T _E Xcheck
11	T _E Xcad
0	submit T _E X batch
22	automatic font generation
196	user installed utilities

In principle, these figures speak for themselves. However, let us make a few comments.

The number of calls to the functions editor, compiler, print, and preview do not give information on how 4TEX is used in practice: they indicate actions that have to be taken whether you work in 4TEX or not. They do give information on the amount of T_EX-ing that is going on at our department.

Among the other figures, the number of calls to the function block-view is rather striking. It indicates that this function is a feature indeed.

Another much used facility is shell-to-DOS. We suspect that many people start 4TEX only once per day, and from then on start any other program they need from the DOS-shell (at the expence of only 36K conventional memory). Note that the calls to user installed utilities represent activities that would have to be done in a DOS-shell in other programs.

Whereas without 4TEX most users would not even try to incorporate graphics in their document (other than by using scissors and glue), the total number of calls (328) to functions related to graphics bears witness to how easy this has become in 4TEX.

5 Availability and support

4TEX can be obtained through *anonymous ftp* from the file servers obelix.icce.ruu.nl, directory/pub/erikjan/4tex, and ftp.cs.ruu.nl, directory/pub/TEX/4tex. The system has a modular structure that allows you to install only those parts that you need. Therefore it is important that you read the file `install.txt` first. The installation files are compressed using ARJ.EXE and fit on 1.44 MB 3.5" high density diskettes. You may also take a look at the documentation file before you install anything. `4texdoc.dvi` and `4texdoc.ps` can be found in the same directory.

If you have trouble installing 4TEX or need more information you can send E-mail to

`4TeX-support@eco.rug.nl`.

However, don't expect an answer within the hour—we will try to help you as soon and as best as we can, but 4TEX is an after hours project.

4TEX users can join the 4TEX mailing list. On this list users can pose/answer questions regarding 4TEX usage. New or desired developments and features are also announced and discussed on this list.