editors

ASCII editors for TEX on MS-Windows

Erik Frambach
Faculty of Economics
University of Groningen
E.H.M.Frambach@eco.rug.nl

abstract

There are many good ASCII editor programs available for TEX-users using MS-Windows. However, they all have there strong points and their weaknesses. In this article I will discuss a few of them.

What is ASCII anyway?

All TEX users know that TEX texts are written in ASCII. But what exactly do we mean by ASCII? It's not as obvious as you may think.

Strictly speaking, ASCII (which stands for *American Standard Code for Information Interchange*) is nut much more than the characters a–z, A–Z, 0–9, some punctuation marks, and a few other characters. Basically, everything that you can type in directly from a US keyboard. These characters are represented as codes 32 to 127, and a few more. This is quite sufficient for English text.

But what about accented letters and other characters that don't exist in English? They can get a slot somewhere in the 8-bit *extended ASCII* table that extends up to 255. So we have to define standards for those characters too. However, there are far more 'extended' characters than there are slots available in an 8-bit set. That's why *code pages* were defined. They describe the meaning of characters in a given 8-bit set. As you can imagine there are many of these, and naturally your ASCII editor program and your TEX compiler have to be (made) aware of what code page you are using. Here are some pitfalls that you have to be aware of:

- Code pages in MS-DOS, Windows, Unix and Apple are/can be different.
- Different versions of Windows (US, NL, PL, etc.) behave differently.
- Windows 95 differs from 98, which in turn differs from Windows NT.
- Using TEX *input encoding* or TEX code page conversion through *TCX* filters (a new Web2c feature) can make a big difference.

All in all it's easy if you stick to English, and it gets tricky if you use extended ASCII. Of course you could use TEX commands to specify 'special' characters, but something like `ge{\"\i}nd` becomes almost unreadable. In other cases such transliterations are simply impossible.

So how does this relate to editor programs? Good editor programs 'understand' code pages, e.g. when spell-checking or replacing strings. Recoding into different code pages is sometimes supported.

Another common problem in dealing with ASCII files is the way new lines are specified. On Unix a single CarriageReturn character is used. On MS-DOS and Windows a CarriageReturn followed by a LineFeed characters is used. On an Apple Macintosh a single LineFeed is used. Good editor programs can deal with all these formats. If not, the ASCII text will be nearly impossible to edit.

Criteria

When it comes to choosing an editor program, there are many criteria that may be important. Some are important to any user, others only to professional users. Some features are nice but not essential, some others may obstruct you from doing your job properly or efficiently. Below is a list of criteria that you may want to use when selecting a good editor program for your purposes:

- File size: if you want to be able to edit a huge (say, 10 MB or more) file, the editor has to be able to handle it, and still run sufficiently fast.
- Speed: some editor programs can do thousands of replacements in a fraction of a second, others will take a minute to complete the task.
- Syntax highlighting: if you are writing texts that contain keywords with special meanings (e.g. TEX commands) it can be very helpful if these are displayed in different colors.
- Column manipulation: some editors allow you to cut, paste or move columns of your text, while others can only move lines or paragraphs. If you are editing tables you may need this feature badly.
- Spell-checking: a built-in spell-checker may warn you if you are writing nonsense. However, this feature may slow the editor program down, or require lots of system resources.
- Word wrapping: if you type in text without looking at the screen it's very convenient if the editor program

automatically starts a new line when the current line is full. But you have to be able to switch this feature off in case you need to write long lines.

- Macro language: a good macro language (which is much more than just replaying key sequences) can help you automating tedious tasks.
- Conversion of CR/LF: if line endings can be selected, you will have less problems when you send the file to someone using a different operating system.
- Conversion between different code pages. If you get a file prepared on MS-DOS this can be an essential feature. If an IBM operating system (e.g. VM) using EBCDIC encoding was used to prepare a text you will also need to convert it to ASCII.
- Tab settings: tabs can be used to position texts. A good editor will allow you to set the size of tabs and (if required) it will show where tabs are inserted, and it can convert tabs to spaces.
- Auto save: it makes sense to save your text regularly (e.g. after 1000 key strokes or 10 minutes), so you don't loose everything in case the system crashes for some reason. It's nice if your editor program will do that for you.
- Multi-level undo: sometimes you may want to undo what you just did. Or even undo the last 10 or 100 modifications.
- Edit multiple files simultaneously: sometimes you may need to change something in a lot of files. It's very efficient if you can do that with only a few key strokes instead of file by file.
- Regular expressions: if the program supports these you have a very powerful tool for finding and replacing string in texts. Unfortunately the implementation of regular expressions differs significantly in different editor programs.
- Hexadecimal editing: 'real' programmers will need hexadecimal editing capabilities. Not really a TEX or ASCII issue, though, but very convenient for checking encodings.
- Dynamic Data Exchange: this feature can be used to set up inter-program communication. It can be a powerful tool in dedicated environments.
- Configurability: it helps a lot if you can configure the program just the way you like it.
- Available on multiple operating systems: if you use several systems it's very convenient to be able to use the same editor program on all systems.
- Price: some editor programs are free, others are expensive. However, an expensive program is not necessarily better than a cheap one.
- Trial version: before you buy something you want to see if the product really satisfies your needs.

If you care to think about it, I'm sure you can come up with another 20 criteria or more.

Some candidates

Below I've listed a few Windows (Win32 actually: they may not run on Windows 3.x) editor program that I think are worthy candidates as TEX text editor programs. This doesn't mean that these are the only ones you should consider, but I can assure you I've evaluated about twenty other editor programs that I discarded for various reasons.

E.g., if an editor program was not available from Internet, at the very least as a crippled shareware version, it was not even considered. If it failed to install properly, it was sent to the waste bin immediately. If it crashed within minutes, or if it kept popping up nag screens, it was thrown away at once.

The following editor programs passed the tests with good results and will be discussed in more details later:

- TSE
- PFE
- MED
- WinEdt
- UltraEdit
- Notetab

One more editor program should be mentioned here: Emacs. There are several varieties (NT-Emacs, XEmacs, to name but a few) of this extremely powerful editor program that is available on many platforms. However, because it is discussed in much more detail by Piet van Oostrum in another MAPS article, I will simply refer to that article.

On any Windows system there are always a few editor programs available:

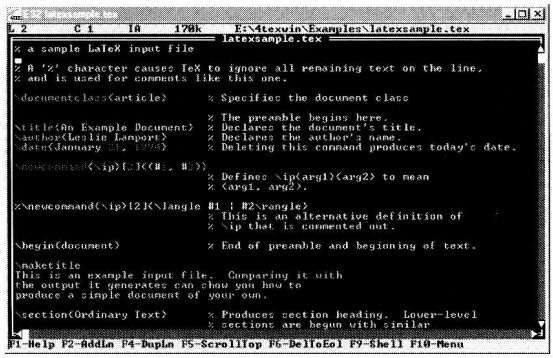
- Notepad
- Wordpad
- Edit

I will discuss the these editor programs as well, and compare them to the others, except for EDIT.COM. This program is remarkably poor: it doesn't even understand long files names. Therefore it doesn't qualify as a Windows editor program.

Pros and cons

In this section I will describe the editor programs and list their pros and cons. The descriptions are not an exhaustive checklists of all criteria, but rather a list of features that are very evident. If a certain feature is not listed, it doesn't mean it's missing, but it's less characteristic or not a very strong point.

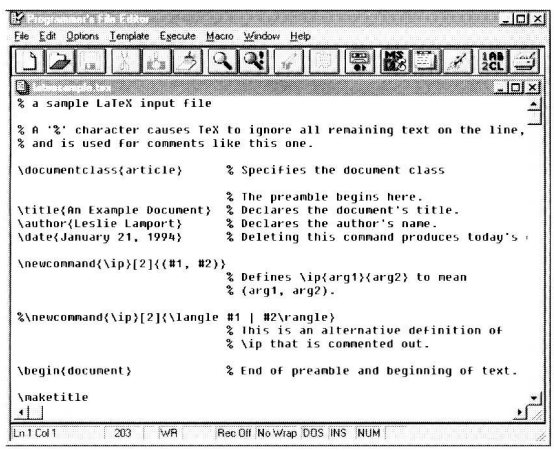
TSE (The Semware Editor)



+ very fast
+ good configurability
+ fine macro language
+ regular expressions (though not complete)
+ edit multiple files
+ German version available
± console application
− no DDE support
− commercial, trial version available
− syntax highlighting, but poor

The TSE editor program can be downloaded from `http://www.semware.com`.

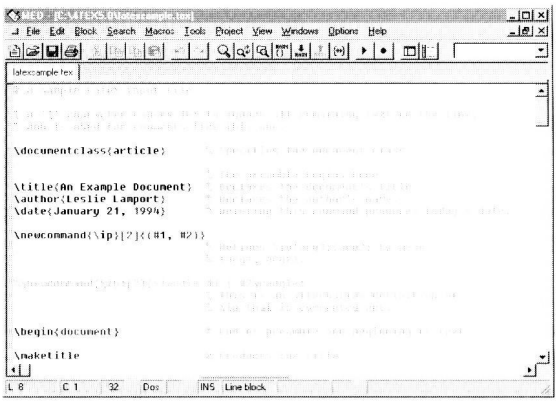
PFE (Programmer's File Editor)



+ highly configurable
+ completely free
+ supports DDE
− no syntax highlighting
− no regular expressions
− no macro language

The PFE editor program can be downloaded from `http://www.lancs.ac.uk/people/cpaap/pfe/`.

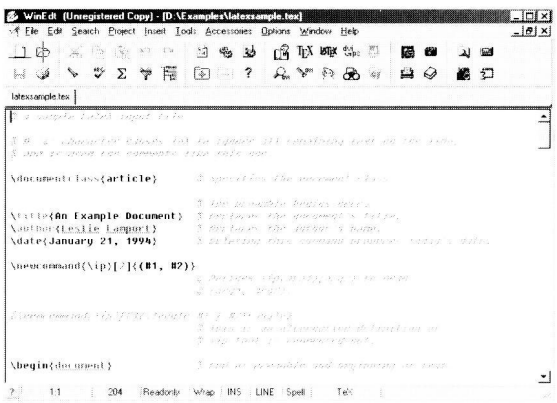
MED (Matthias' Editor)



+ reasonable syntax highlighting
+ support DDE for all commands
+ auto save
+ project manager
+ regular expressions
+ edit multiple files
+ can find matching [(< or) >] but not { }
+ can manipulate columns
+ supports templates
+ can run arbitrary tools (TEX compiler, viewer, BibTEX, etc.)
+ German version available
± cheap shareware
− no macro language

The MED editor program can be downloaded from `http://www.utopia-planitia.de`.

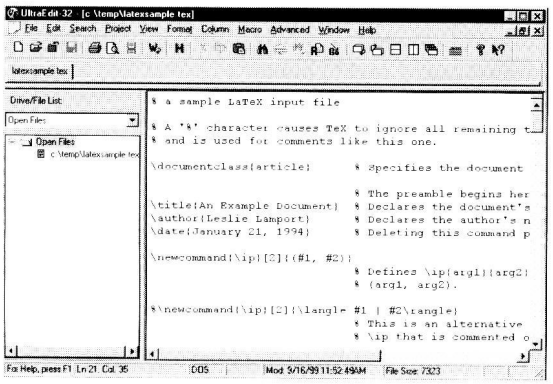
WinEdt



+ splendid syntax highlighting
+ many TEX features built in
+ can manipulate columns
+ understands character sets and TEX notation
+ built-in spell-checker
+ very powerful macro language
+ completely configurable, including menus
+ pre-configured for running MiKTEX
− shareware
− slow start-up, uses lots of system resources

The WinEdt editor program can be downloaded from `http://www.winedt.com/` or from any CTAN server (e.g. `ftp://ftp.ntg.nl`, directory `/tex-archive/systems/win32/winedt/`.

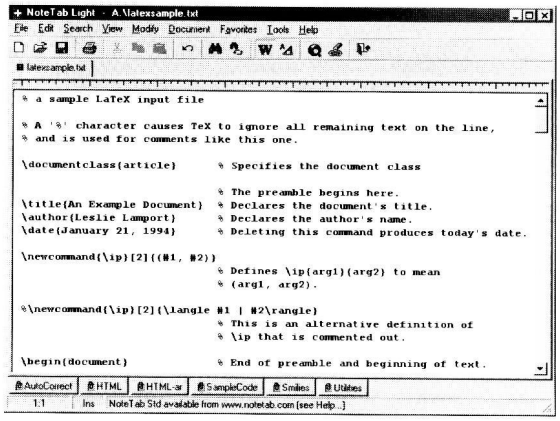
UltraEdit



+ can manipulate columns
+ built-in spell-checker
+ syntax highlighting
+ project manager built in
+ supports regular expressions
+ supports hexadecimal editing
+ supports templates
+ FTP client built in
+ supports file compare
+ many sorting options
+ splendid code page conversions
+ can run external programs and capture output
− shareware

The UltraEdit editor program can be downloaded from `http://www.ultraedit.com`.

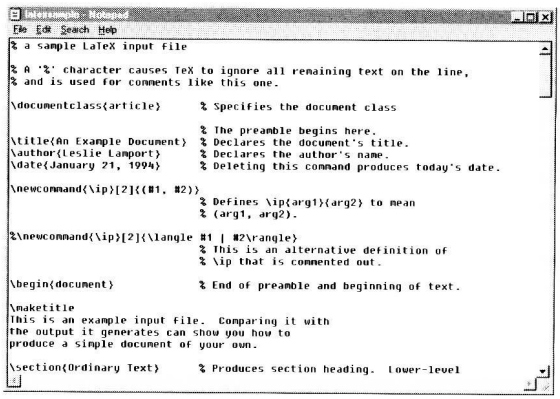
Notetab



+ free 'light' version, trial professional version
+ supports templates
+ good macro language (offers 'clipbooks')
+ nice macros for TEX/LATEX users available
+ supports regular expressions
+ can find matching [({ < or]) } >
+ good HTML editing features
+ supports several conversions
+ good configurability
− shareware

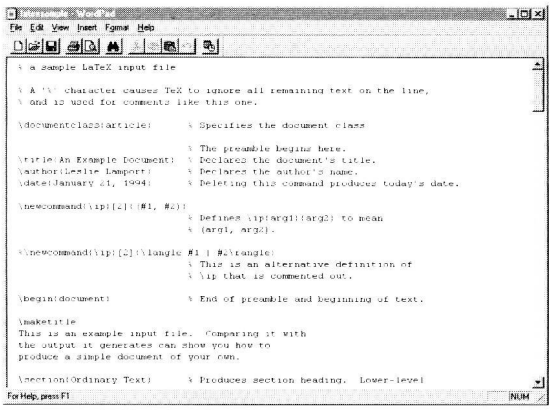
The NoteTab editor program can be downloaded from `http://www.notetab.com`.

Notepad



+ available for 'free' on every Windows system
− very few features
− older versions can't handle files larger than 32 KB
− can only edit one file at a time

Wordpad



+ available for 'free' on every Windows system
+ can handle ASCII, RTF and some MS-Word file formats
− can only edit one file at a time
± is it an editor program or a word processor? or neither? or both?

Conclusions

Looking back at this quick overview some conclusions can be drawn:

- All these editor programs have many features in common, but they are not equal. There are also many bigger and smaller differences.
- All these editor programs have there own strong points and weak points.
- There is no *ultimate* editor that does everything *and* is easy to use *and* is easy to learn *and* is cheap.
- There is no relation between price and performance. Cheap editor programs may well outperform expensive ones. But not necessarily.

Recommendations

So the question remains: what editor program should I choose? Here are a few recommendations to help you decide:

- Make a list of features that are important for your kind of work. Decide how essential these features are.
- Choose an editor that can do more than you currently need. You may want to use other features later.
- Invest time and energy in getting to know the program. You will find that you can do your work a lot more efficient if you know the short-cuts.
- Configure the editor program specifically for the tasks that you have in mind. Macros for often used functions can make your life a lot easier.

Because eventually the best editor program is the one that you are fully accustomed to!