

# L<sup>A</sup>T<sub>E</sub>X kernel programming tips

Péter Szabó

[<pts@fazekas.hu>](mailto:pts@fazekas.hu)

Budapest University of Technology and Economics,  
Department of Computer Science and Information Theory,  
H-1117 Hungary, Budapest, Magyar tudósok körútja 2.

tutorial slides for EuroT<sub>E</sub>X 2006

2006-07-04 14:40–16:00

Debrecen, Hungary

# T<sub>E</sub>X, L<sup>A</sup>T<sub>E</sub>X, e-T<sub>E</sub>X

Software and docs

T<sub>E</sub>X, L<sup>A</sup>T<sub>E</sub>X, e-T<sub>E</sub>X

... and more friends

Who program L<sup>A</sup>T<sub>E</sub>X

Use the source

Read more

And read these, too

Source files

Task to do

Others

- *T<sub>E</sub>X*. The *typesetting system* by Knuth. *The T<sub>E</sub>Xbook* was published in 1983. Related software: METAFONT font compiler. *Computer Modern*, the default font family of T<sub>E</sub>X has been digitalized by Knuth using METAFONT.
- *plain T<sub>E</sub>X*. this is the first T<sub>E</sub>X format (= basic macro package). Written by Knuth. Used for writing The T<sub>E</sub>Xbook.
- *L<sup>A</sup>T<sub>E</sub>X*. structured T<sub>E</sub>X format, even for non-programmers. The latest stable version, L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>, was written by Leslie Lamport in 1993. (Work is still in progress on L<sup>A</sup>T<sub>E</sub>X3, gaining new momentum in 2005.)
- *ε-T<sub>E</sub>X*. T<sub>E</sub>X extended with bidirection writing, justification by horizontal extending of glyphs, and more convenient programming primitive. L<sup>A</sup>T<sub>E</sub>X now runs over ε-T<sub>E</sub>X, but the L<sup>A</sup>T<sub>E</sub>X base system doesn't use its new features.

# ... and more friends

Software and docs

$\TeX$ ,  $\LaTeX$ , e- $\TeX$

... and more friends

Who program  $\LaTeX$

Use the source

Read more  
And read these,  
too

Source files

Task to do

Others

- *pdf $\TeX$* .  $\TeX$  with new features added, including direct PDF generation, more advanced font handling, microtypographic (hz-) tools, PDF page inclusion, new programming primitives (attend Martin Schröder's talk on Friday for more). But we can make PDF even without pdf $\TeX$ , e.g. from DVI using *dvips*, and then from PostScript using *Ghostscript*. These slides were made this way.
- $\Omega$ . revised, reimplemented,  $\TeX$ -compatible with advanced font handling, Unicode support, generic model and special support for non-latin scripts. Work in progress.
- *te $\TeX$* .  $\TeX$  distro for UNIX. Contains all above.
- *$\TeX$ Live*. modern,  $\TeX$  distro with live CD. Multiplatform: Linux, MacOS X, Windows and more.
- *CTAN*. searchable FTP site for all  $\TeX$ -related developments. Get new version of your favorite  $\LaTeX$  package from there.

# Who program L<sup>A</sup>T<sub>E</sub>X

- *the developers of L<sup>A</sup>T<sub>E</sub>X*
- *the developers of L<sup>A</sup>T<sub>E</sub>X packages* (= style). Packages extend and fix L<sup>A</sup>T<sub>E</sub>X functionality.
- *the developers of document classes*. they work for publishing houses, they create the .cls files from the typographic design of the book or article.
- *people localizing L<sup>A</sup>T<sub>E</sub>X*. they make fonts, character encodings, index processors etc. for languages other than English.
- *authors*. they useually write only simple macros, or they just customize packages in order to typeset their work.
- *content management experts*. they write tools for for converting between L<sup>A</sup>T<sub>E</sub>X and other formats (e.g. OpenDocument, HTML, XML, .doc)

Software and docs

T<sub>E</sub>X, L<sup>A</sup>T<sub>E</sub>X, e-T<sub>E</sub>X  
... and more  
friends

Who program  
L<sup>A</sup>T<sub>E</sub>X

Use the source

Read more  
And read these,  
too

Source files

Task to do

Others

# Use the source

Base your solid L<sup>A</sup>T<sub>E</sub>X programming skills on:

- *The Not So Short Introduction to L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>*. This is about using L<sup>A</sup>T<sub>E</sub>X for typesetting, not programming, but this is a good introduction to its syntax and main concepts. Translations available to several languages. <http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf>
- *The T<sub>E</sub>Xbook*. Although it is about plain T<sub>E</sub>X, it explains some really advanced topics about T<sub>E</sub>X and its macro programming language, most of them being relevant to L<sup>A</sup>T<sub>E</sub>X, too. Paragraphs and exercises marked with single and double dangerous bends are especially recommended for thorough reading: these are the most authentic and in-depth explanations about how T<sub>E</sub>X works. Introductory exercise: try to download the T<sub>E</sub>Xbook from CTAN and compile it for yourself.

Software and docs

T<sub>E</sub>X, L<sup>A</sup>T<sub>E</sub>X, e-T<sub>E</sub>X

... and more friends

Who program L<sup>A</sup>T<sub>E</sub>X

Use the source

Read more

And read these, too

Source files

Task to do

Others

# Read more

[Software and docs](#)

[T<sub>E</sub>X, L<sup>A</sup>T<sub>E</sub>X, e-T<sub>E</sub>X](#)

[... and more friends](#)

[Who program L<sup>A</sup>T<sub>E</sub>X](#)

[Use the source](#)

[Read more](#)

[And read these, too](#)

[Source files](#)

[Task to do](#)

[Others](#)

- *The documentation of  $\varepsilon$ -T<sub>E</sub>X*. It documents some important new primitives. L<sup>A</sup>T<sub>E</sub>X now uses  $\varepsilon$ -T<sub>E</sub>X by default, so these powerful primitives are available for the L<sup>A</sup>T<sub>E</sub>X programmer.
- *The manual of pdfT<sub>E</sub>X*. It documents some important new primitives. This will help you understand how the pdftex drivers of graphics.sty and hyperref.sty work. Compilation hint: download the manual folder with the file pdftex-t.tex. Compile it with `texexec -pdf pdftex-t`. If the compilation falls to an infinite loop, abort it when pdfT<sub>E</sub>X finishes running.

# And read these, too

- a comprehensive listing of filename extensions in your favourite L<sup>A</sup>T<sub>E</sub>X book
- documentation of advanced L<sup>A</sup>T<sub>E</sub>X packages: pl. *babel*, *varioref*, *amsmath*, *graphicx*, *hyperref*, *powerdot*, *nath*, *magyar.ldf*. Find the source on CTAN<sup>4</sup>, compile the .dtx files with L<sup>A</sup>T<sub>E</sub>X. Read other peoples' source code.
- Some problems cannot be solved by T<sub>E</sub>X macro programming. Read about other tools in your T<sub>E</sub>X distribution: METAFONT (read *The METAFONTbook*), METAPOST, *kpathsea* (kpse), *afm2tfm*, *fontinst*, *dvips*, *pdfT<sub>E</sub>X*, *dvipdfm* (old, not developed anymore), *BibT<sub>E</sub>X*, *makeindex*.
- A good description of T<sub>E</sub>X *macro expansion*), and its tricky use can be found in the *binhex.tex* package, and David Kastrup's article in the EuroT<sub>E</sub>X 2001 proceedings.

Software and docs

T<sub>E</sub>X, L<sup>A</sup>T<sub>E</sub>X, e-T<sub>E</sub>X

... and more friends

Who program L<sup>A</sup>T<sub>E</sub>X

Use the source

Read more

And read these, too

Source files

Task to do

Others

# Where to look for L<sup>A</sup>T<sub>E</sub>X source files

## Software and docs

### Source files

#### Where to look for L<sup>A</sup>T<sub>E</sub>X source files

#### What L<sup>A</sup>T<sub>E</sub>X loads

#### What L<sup>A</sup>T<sub>E</sub>X loads (2)

#### What it loads secretly

#### What the format contains

#### More about the format

#### Still inside the format

## Task to do

## Others

- *plain text files.* Most files read (and written) by L<sup>A</sup>T<sub>E</sub>X are plain text. Get a text editor and learn how to use it efficiently and productively. Don't hesitate to learn all the keyboard combinations! Advanced editors include: *Vim*, *Emacs*, and even *Kate*. Get a file manager with recursive search functionality; e.g. *Midnight Commander*.
- *the texmf tree.* The source files coming with your T<sub>E</sub>X distribution are placed into the texmf tree. On UNIX, try `/usr/share/texmf*` and `/var/share/texmf`.
- *kpsewhich.* A diagnostic tools for finding a file with a given name in the texmf tree. L<sup>A</sup>T<sub>E</sub>X would find the file at the same place. Sometimes we have to specify the tile type, e.g. 

```
kpsewhich -format="dvips config" config.ps.
```
- *texmf.cnf.* Contains configuration paramters (e.g. memory sizes), and specifications about where to find each file type in the texmf tree.



# What L<sup>A</sup>T<sub>E</sub>X loads

[Software and docs](#)

[Source files](#)

[Where to look for L<sup>A</sup>T<sub>E</sub>X source files](#)

[What L<sup>A</sup>T<sub>E</sub>X loads](#)

[What L<sup>A</sup>T<sub>E</sub>X loads \(2\)](#)

[What it loads secretly](#)

[What the format contains](#)

[More about the format](#)

[Still inside the format](#)

[Task to do](#)

[Others](#)

Compile this `example.tex` document:

```
\documentclass{article}
\usepackage{t1enc}
\usepackage[latin2]{inputenc}
\usepackage[magyar,english]{babel}
\begin{document} Hello, World! \end{document}
```

Look at the console output or examine the `.log` file to find out what files were opened. Use *kpsewhich*.

- *article.cls*. the document class. Defines commands `\section` and `\maketitle`, and all other visual formatting.
- *size10.clo*. Font size and skip setting corresponding to a main text at 10pt size.
- *t1enc.sty*, *fontenc.sty*. map L<sup>A</sup>T<sub>E</sub>X character commands to font positions

# What L<sup>A</sup>T<sub>E</sub>X loads (2)

Software and docs

Source files

Where to look for  
L<sup>A</sup>T<sub>E</sub>X source files

What L<sup>A</sup>T<sub>E</sub>X loads

What L<sup>A</sup>T<sub>E</sub>X loads  
(2)

What it loads  
secretly

What the format  
contains

More about the  
format

Still inside the  
format

Task to do

Others

- *babel.sty*, *babel.def*. loads the macro definitions of Babel, the multilanguage localization framework
- *english.ldf*, *magyar.ldf*. localization to specific languages
- *example.aux*. auxiliary file emitted by the previous run of L<sup>A</sup>T<sub>E</sub>X. Current `\refs` and `\pagerefs` get there values from previous `\labels`, from the `.aux` file. LaTeX regenerates it at each compilation.
- `.bib` and `.bbl` for the bibliography, `.idx` and `.ind` for the index, `.toc`, `.lof` and `.lot` for the table of contents and other lists. These are generated only when their feature is used in the document. Packages may create other files, e.g. *hyperref.sty* creates `.out-ot`, and *powerdot.sty* creates `.bm`.

*texmf.cnf* defines where to load a file from if it is not found in the document compilation folder. To modify any file, copy it to the document folder, and modify there.

# What it loads secretly

Run `strace -e open latex example` on Linux and find out that some other files not mentioned in the `.log` file are also loaded.

- *texmf.cnf*. already seen.
- *tons of ls-R files*. these contain the folder list cache of the texmf tree. If you change some in the tree, don't forget to run *mktextlsr* (as root).
- *aliases*. contains a mapping from aliases to real files. Similar to UNIX symlinks. Usually of historic significance.
- *latex.fmt*, *pdflatex.efmt* etc. This is the L<sup>A</sup>T<sub>E</sub>X format file. It is a binary file which contains precompiled macro definitions (most of them for *latex.ltx*) and hyphenation patterns. The latter were put there in the 80s for performance reasons. Now this is a disadvantage.

Software and docs

Source files

Where to look for L<sup>A</sup>T<sub>E</sub>X source files

What L<sup>A</sup>T<sub>E</sub>X loads

What L<sup>A</sup>T<sub>E</sub>X loads (2)

What it loads secretly

What the format contains

More about the format

Still inside the format

Task to do

Others

# What the format contains

Software and docs

Source files

Where to look for  
L<sup>A</sup>T<sub>E</sub>X source files

What L<sup>A</sup>T<sub>E</sub>X loads

What L<sup>A</sup>T<sub>E</sub>X loads  
(2)

What it loads  
secretly

What the format  
contains

More about the  
format

Still inside the  
format

Task to do

Others

The `initex latex.ini` command regenerates the L<sup>A</sup>T<sub>E</sub>X format (*latex.fmt*). (There is also *pdfinitex*.) The `fmtutil -all` command regenerates all formats, and copies the generated `.fmt` files to their proper place in the `texmf` tree.

The L<sup>A</sup>T<sub>E</sub>X format is generated from these source text files:

- *tex.pool*. T<sub>E</sub>X error messages and other strings – do not edit!
- *latex.ini*. just loads *latex.ltx*
- *latex.ltx*. the main macro definitions of the L<sup>A</sup>T<sub>E</sub>X kernel as a 250 kB T<sub>E</sub>X tight T<sub>E</sub>X source file. Read the corresponding documentation in *base.zip* (already mentioned).
- *texsys.cfg*. contains system-specific parameters (such as format of file names). It is no point to modify it after installation.
- *fonttext.cfg*. just loads *fonttext.ltx*

# More about the format

[Software and docs](#)

[Source files](#)

[Where to look for L<sup>A</sup>T<sub>E</sub>X source files](#)

[What L<sup>A</sup>T<sub>E</sub>X loads](#)

[What L<sup>A</sup>T<sub>E</sub>X loads \(2\)](#)

[What it loads secretly](#)

[What the format contains](#)

[More about the format](#)

[Still inside the format](#)

[Task to do](#)

[Others](#)

- *fonttext.ltx*. loads the base font encoding definition files, and selects *Computer Modern* as the default font family
- *omlenc.sty*, *tlenc.sty*, *otlenc.sty*, *omsenc.sty*: font encoding definition files
- *tlcmr.fd*, *otlcmr.fd*, *otlcmss.fd*, *otlcmmtt.fd*: font definition files of text fonts of the *Computer Modern* family. More .fd files are loaded later automatically by L<sup>A</sup>T<sub>E</sub>X when an unknown `\fontfamily` is selected.
- *fontmath.cfg*. just loads *fontmath.ltx*
- *fontmath.ltx*. selects the *Computer Modern* math fonts as default, defines math symbols and commands (e.g. `\sigma`, but not `\sin` nem).
- *omlcmm.fd*, *omscmsy.fd*, *omxcmex.fd*, *ucmr.fd*. the font definition files of the math fonts of the *Computer Modern* family. Loaded early for performance reasons.

# Still inside the format

[Software and docs](#)

[Source files](#)

[Where to look for L<sup>A</sup>T<sub>E</sub>X source files](#)

[What L<sup>A</sup>T<sub>E</sub>X loads](#)

[What L<sup>A</sup>T<sub>E</sub>X loads \(2\)](#)

[What it loads secretly](#)

[What the format contains](#)

[More about the format](#)

[Still inside the format](#)

[Task to do](#)

[Others](#)

- *preload.cfg*. just loads *preload.ltx*
- *preload.ltx*. preloads some font metrics (TFM) for performance reasons.
- *cmex10.tfm*, *line\*.tfm*, *cmr\*.tfm*, *cmmi\*.tfm*, *cmsy\*.tfm*. loaded above. TFM is a binary format, see docs of METAFONT.
- *hyphen.cfg*. basic, T<sub>E</sub>X format independed macros which support changing languages (more specifically: hyphenation pattern sets)
- *language.dat*. a text file that lists what languages to load hyphenation patterns for. If your favourite language is missing, uncomment it, and regenerate the format.
- *hpyhen.tex*, *frhyph.tex*, *dehyph\*.tex*, *huhyph.tex* and *zerohyph.tex*. hyphenation patterns for languages, in the form of `\patterns` commands. First one is for English by Knuth.
- *ltpatch.ltx*: later L<sup>A</sup>T<sub>E</sub>X patches. Now empty.

# Task 1 and hints

[Software and docs](#)

[Source files](#)

[Task to do](#)

**[Task 1 and hints](#)**

[Task 2 and hints](#)

[Others](#)

*The task:* Change the horizontal space after `\section` number to 1ex, and make it hang to the left.

- Where is `\section` defined? Too many search hits. Is it `\def`, `\newcommand` or `\providecommand`?
- Take only files actually loaded by L<sup>A</sup>T<sub>E</sub>X. Found it: *article.cls*.
- But `\@startsection` is in *latex.ltx*.
- Modify commands `\@sect` and `\@ssect`.
- Add `\tracingmacros1` and `\tracingcommands1` before problems, and examine the .log file.
- Active diagnostics: `\makeatletter`, `\expandafter\show`  
`\csname`, `\typeout{\meaning, \errmessage}`.
- Is the modified version compatible with other packages (who override or don't call `\@sect`)? What about Babel? What about the AMS document classes?

# Task 2 and hints

[Software and docs](#)

[Source files](#)

[Task to do](#)

[Task 1 and hints](#)

[Task 2 and hints](#)

[Others](#)

*The task:* Have page numbering skip the unluckiest number of your life. Then have L<sup>A</sup>T<sub>E</sub>X emit an empty page instead.

- What primitives are used to emit pages? Read the relevant part of The T<sub>E</sub>Xbook. Found them: `\shipout` and `\output`.
- Where does L<sup>A</sup>T<sub>E</sub>X run these commands? Grep in *latex.ltx*. Found `\@outputpage`.
- What is the T<sub>E</sub>X command to increment counters? From The T<sub>E</sub>Xbook: `\@advance`. What are the L<sup>A</sup>T<sub>E</sub>X equivalents? From the definition of `\label: \stepcounter`, `\refstepcounter` and `\setcounter`. Found it: `\stepcounter{page}`.
- Figure out how to increment the counter. Prepend:  
`\ifnum\c@page=13 \stepcounter{page}\fi`
- Copy the whole definition of `\@outputpage`? Add a hook? Most advanced: append to `\cl@page`. Extra `\shipout`.



# String processing

[Software and docs](#)

[Source files](#)

[Task to do](#)

[Others](#)

**[String processing](#)**

[String processing  
– solution](#)

[More topics at will](#)

T<sub>E</sub>X macro expansion is good to build strings from other strings using macros as templates. But what if we want to modify an existing string? There are no built-in tools for that, so we have to write ours. This applies to all T<sub>E</sub>X, not only L<sup>A</sup>T<sub>E</sub>X.

Who needs string processing? Anybody who wants to implement an XML parser. (But try *xmltex* and *passivetex* first before writing your own one.)

As an example, let's try to write a macro `\rmstars` which removes all stars (\*) from a string. The string is specified as an argument in braces, and the result – without the stars and all tokens having catcode 12 – it is put into the macro `\M`. Example invocation: `\rmstars{a * B**cd} \show\M`.

Shouldn't be hard for a Perl programmer (`$M=~s/*/g`), but needs too many tricks in T<sub>E</sub>X. Are you ready to turn the page?

# String processing – solution

Are you sure you want to understand this beauty?

[Software and docs](#)

[Source files](#)

[Task to do](#)

[Others](#)

[String processing](#)

[String processing  
– solution](#)

[More topics at will](#)

```
\def\stripit#1>{\}\def\empty{\}\def\space{ }
\def\rmonestar#1{\ifx#1\hfuzz\empty\else
  \if*\string#1\else#1\fi
  \expandafter\rmonestar\fi}
\begingroup\lccode'!=' \lowercase{\endgroup
\def\oonespace#1 {\ifx\hfuzz#1\empty\else
  #1!\expandafter\oonespace\fi}}
\def\rmstars{%
  \afterassignment\rmstarsb\def\M}
\def\rmstarsb{%
  \edef\M{\expandafter\stripit\meaning\M
    \space\hfuzz\space}
  \edef\M{\expandafter\oonespace\M}
  \edef\M{\expandafter\rmonestar\M\hfuzz}}
```

# More topics at will

Software and docs

Source files

Task to do

Others

String processing

String processing  
– solution

**More topics at will**

- implementing new features (writing L<sup>A</sup>T<sub>E</sub>X packages)
- writing packages accepting options
- changing existing features
- extending the definition of a command
- writing code independent of catcode changes
- .aux file and `\ref` tricks. How to restart footnote numbering on each page? Add a `\label` for each footnote mark, and reset number to 1 if `\pageref` of current and previous footnote differ.

