

What is it about all those *T_EXs

Abstract

This short article describes the different ‘layers’ in a T_EX system, the differences between T_EX engines, extensions, macro packages, and distributions. I hope to take away some of the confusions that people new to T_EX and less technically inclined people have when they are confronted with terms like `pdftex`, `texlive`, `tetex`, `miktex`, `pdflatex` and so on.

Keywords

T_EX, LaTeX, miktex, tetex, T_EXLive, etex, pdftex, pdfetex

Introduction

When you want to use T_EX on your computer you are quite soon confronted with a lot of software parts which have names that end on ‘tex’. If you are lucky you have a computer which has T_EX pre-installed. Otherwise you will have to choose what to install. You might get a DVD with one install-button that installs a complete system for you, but more often you will have to choose which components you want to install. And if your T_EX installation has become outdated you may come to the point that you have to install a new system. Sometimes it is just an upgrade of the existing installation but there comes a point when that is no longer supported and then you have to decide what to choose. Now this choice can be quite confusing. Do you have to use ‘MiK_T_EX’ or ‘T_EXLive’, ‘TeXShop’ or ‘TeXnicCenter’? And what are these things? Why doesn’t installing TeXShop on a Mac give you a working T_EX system? And if you want to process your files do you choose ‘latex’ or ‘miktex’? We will see that the last question is the wrong one but for new users it seems a logical thing to ask. Below I will try to put some order into all these terms so that you will be able to place the different things in their proper place and to know which questions have meaning.

The beginning

In the beginning there was T_EX. This is the original program designed and programmed by Donald Knuth. You prepared your input according to the specifications in The T_EX Book. This style of T_EX documents is called

‘plain T_EX’. You would run `tex` from the command line (there were no graphical user interfaces in that time) on this input and a DVI file would be produced. There was a program to preview the DVI file and one or more programs to send it to a suitable printer. And there was a companion program `metafont` but that was only used by some font freaks. And that was it.

Further developments

After the original T_EX implementation several new developments have occurred.

- *Different syntax forms for your input document.* Plain T_EX is quite simple but out of the box it doesn’t support more advanced features, like cross references, automatically numbered sections and so on. You can program these yourself but it is tedious and it makes exchange of documents with other authors more difficult. It would be nice if you could use the work that others have done. Such collections of definitions and pre-programmed features are called ‘macro packages’. The most well known of these are LaTeX and ConT_EXt. Please note that these macros do not require changes to the T_EX program but are a kind of additional input files for T_EX, extending the input language T_EX understands with new – usually higher-level – commands.
- *Extensions of the T_EX program.* There are several limitations in the T_EX program that make it hard for writers of macro packages to program some advanced features. Another wish people had is to produce PDF files directly rather than, or better, in addition to, the arcane DVI format. Or to make easier use of the fonts that are present in modern operating systems. For a good programmer it is not very difficult to enhance the T_EX program to overcome these difficulties. However, Knuth does want T_EX to be stable, so he doesn’t change it anymore. He allows others to make changed versions, however, as long as they are not called T_EX. Therefore all kinds of extensions of T_EX have appeared under names such as eT_EX (additional facilities for macro programmers), pdfT_EX (PDF output as an option), or XeT_EX (support for

operating systems fonts).

- *Engines.* Most of the macro packages can be combined with most of the (extended) programs mentioned above. Often the combinations are pre-packaged in such a way that they can be executed as a specific program. We call such a program an ‘engine’. (Some people might use the word ‘engine’ only for the above mentioned programs without reference to a preloaded macro package.) An engine can be for example `latex`, `pdftex` or `pdflatex`.
- *Additional programs.* Having only T_EX and `metafont` wasn’t sufficient. Many other supporting programs have been added, such as programs to manipulate bibliographies, picture processing, support of different languages, and so on. Well known ones are `bibtex` for bibliography processing and `makeindex` for the sorting and formatting of alphabetical indices.
- *Graphical user interfaces* to make working with T_EX and the additional programs easier. Nowadays most computer users do not feel comfortable with command line interfaces. Therefore there are graphical programs, sometimes called IDE’s (integrated development environments). With these you can edit your input documents, often with syntax coloring. And they contain menus and buttons for seamlessly activating the major programs that are part of a T_EX processing cycle, including viewing errors, previewing the document and printing it. Examples are `TeXShop`, `TeXnicCenter`, and `Texmaker`
- *Many other additions* are useful, for example support for many fonts, additional macro packages as additions to `LaTEX`, `ConTEXt` and others, and so on. Together such a T_EX system easily contains tens of thousands of files. Managing these files and keeping up with new releases of them and keeping them consistent is a major job. Therefore there are *distributions* which consist of a carefully selected and tested collection of files and programs that together form a working environment for the processing of T_EX documents.

In the rest of this article I will deal with the above mentioned parts in a ‘top-down’ manner, i.e., starting with the things the user will encounter first, and then digging down to a more detailed view of how these things are built up internally. As we go into more details some of the above information will be repeated below.

Distributions

A distribution is a complete collection of T_EX-related programs of all thinkable sorts, macro packages, doc-

umentation files, supporting programs, etc. They usually come on a CDROM or DVD (most don’t fit on a CDROM anymore these days) or can be downloaded from the Internet.

If you install a distribution you have in principle a working and very complete T_EX system, but usually not a GUI. The advantage of installing a complete distribution rather than collecting the parts yourself is that it most probably has been thoroughly checked and all the parts have versions that are supposed to work together.

Popular distributions are:

- `MiKTEX`. Built and distributed by Christian Schenk for MS-Windows. Some parts of it have been ported to GNU/Linux and Mac OS X. This is probably the most popular distribution for MS-Windows. The great advantage of `MiKTEX` is that it has an advanced ‘package manager’. This is a part that automatically installs missing parts if they are needed. So you can choose to install a rather small subset of `MiKTEX` initially, and let the package manager automatically download and install additional needed parts. Of course you must be connected to the Internet for this to happen. The package manager can also be used to upgrade outdated parts or install new parts manually.
- `teTEX`. This used to be a large distribution for Unix-like systems packaged by Thomas Esser. Many GNU/Linux systems still have this as their default distribution. However, it is no longer maintained. It has been replaced by:
- `TEXLive`. This distribution was originally based on `teTEX` but the choice of what it included was different from `teTEX`. It also supports MS-Windows, GNU/Linux and Mac OS X, and is therefore a good choice when portability is important. It is released on a CDROM/DVD set and freely distributed (about yearly) to the members of most T_EX user groups, like NTG. It can also be downloaded from the Internet.
- `MacTEX` is a distribution for Mac OS X which is a subset of `TEXLive` that can be downloaded from the Internet.

GUI's

A GUI (Graphical User Interface) is a program or collection of programs to make your life as a T_EX user easier. They contain an editor for entering and modifying the T_EX input file(s), often with syntax highlighting. And they usually have buttons to run the proper engine on the file, display it on your screen, print it, and run supporting programs like bibliography and index processing.

They usually don’t include the T_EX programs and

macro packages but suppose that you have installed these from a distribution. So for example installing TeXShop in your Mac doesn't help you very much if you don't also install a complete TeX system like TeXLive or MacTeX.

Popular GUI's are TeXnicCenter and WinShell on MS-Windows, TeXShop on Mac OS X and TeXmaker for GNU/Linux as well as MS-Windows and Mac OS X. On Unix and GNU/Linux systems many people use the Emacs editor as a GUI for TeX.

Macro packages

We have already mentioned the macro packages plain, LaTeX and ConTeXt. A macro package consists of a collection of command definitions and some such, and are contained in one or more text files. For efficiency reasons they are preprocessed and stored in a binary file called a 'format file' or 'fmt' file. A TeX program (in the extended sense) can easily load a format file, either by specifying it as a command parameter or by 'cloning' the engine program file and giving it a name corresponding to the name of the format file. So for example the program latex was the TeX engine with the LaTeX format preloaded, and pdflatex is the pdfTeX engine preloaded with the LaTeX format.

Even if you have written your own macro package it is easy to generate preloaded engines and give them a required program file name.

XeTeX can also be loaded with a LaTeX format and the resulting program is called xelatex.

Engines

The TeX program that translates your input file to a DVI or PDF file we have called the TeX engine. The program file that contains the original engine is usually just called tex (or tex.exe on MS-Windows systems). The TeX engine doesn't change very much as Knuth has decided that he will not make any more enhancements to it despite the limitation the program has. And as it is mature bugfixes are also rare.

Other people, however wanted to add new features. This is allowed by the license, but the resulting program must not be called 'TeX'. Such a program is also an engine, but then not for TeX but for an extension of TeX. Strictly speaking they should therefore not be called TeX-engines, although in sloppy speaking this is often done.

The most important extensions of TeX are (or were):

- eTeX. This extension of TeX adds typesetting for languages that are written right-to-left, like Hebrew and Arabic. Left-to-right and right-

to-left typesetting can be intermixed. It also expands the number of 'registers' that TeX has for storing numbers, dimensions and so, which, with complicated packages sometimes were exhausted. There are also some other extensions. LaTeX and ConTeXt nowadays are dependent on some of these extensions.

- pdfTeX. This is an extension that adds direct PDF output besides DVI output. It was developed in a PHD research by Hàn Thế Thành. It also contains microtypographics extensions which were the main subject of his thesis.
- pdfeTeX. This used to be a combination of eTeX and pdfTeX, but now pdfTeX contains the eTeX extensions itself thereby obviating a separate pdfeTeX engine. And as this contains all the possibilities of the original TeX and eTeX, it has become the default engine for most TeX applications. Only plain TeX uses the original engine from Knuth.
- XeTeX adds the possibility of using your operating system's fonts in a simple way, and it supports Unicode input and the use of OpenType fonts. It incorporates the eTeX extensions and some of the pdfTeX extensions. It has a dvi backend but works closely with dvipdfmx to produce PDF. The program file is usually called xetex.
- luaTeX. This is a relatively new extension, still in development, which adds the Lua programming language to the pdfTeX engine. This gives an enormous flexibility. It also supports OpenType fonts. It is meant to become the successor of pdfTeX. The program file is called luatex.
- Others. There are other engines, such as Omega and Aleph but they are experimental and often buggy and therefore not much used. Aleph functionality is being merged into luatex.

The engines are the first layer on which a TeX system is built. Most people use a preloaded engine, however, which is a combination of a TeX (extension) program and a macro package. We will take an engine preloaded with LaTeX as an example.

Nowadays always the pdfTeX engine is used (as described in the previous section including the eTeX extensions). Both the program latex and the program pdflatex are engines consisting of pdftex preloaded with the LaTeX macro package. The difference is that the program latex is parametrised to generate DVI output and pdflatex to generate PDF output. But note that both can also produce the other output if the document source would choose so.

The program context is the pdftex engine preloaded with the ConTeXt format, but it can also use other engines, such as XeTeX. ConTeXt comes in dif-

ferent language flavors, like `cont-en` for the English version and `cont-nl` for the Dutch version. `context` is usually not called directly, however, but by means of the program `texexec`.

Summary

engines:

`tex`, `pdftex`, `xetex`, `luatex`

macro packages:

Plain, L_AT_EX, ConT_EXt

engines with preloaded macro packages:

`tex`, `latex`, `pdflatex`, `xelatex`, `context`

GUI's:

TeXnicCenter, WinShell, TeXShop, Texmaker

distributions:

MiK_TE_X, te_TE_X (obsolete), T_EXLive, MacT_EX.

I would appreciate to get feedback to be able to improve on this overview. If you have additions or feel that some things are unclear please email me at the address below.

Piet van Oostrum
piet@vanoostrum.org