

In the Company of T_EX and Friends or Automating the processing of T_EX Files

K J Dryllerakis

*4, Rue du Grand Cerf
Bruxelles, B-1000
Belgium*

Setting Up the Stage

Our story begins in England at the end of the '80s. We find ourselves in a University environment. Our protagonist, a science researcher, has been built from the essence of Hollywood adventurers much like Indiana Jones. His world of adventures is not that of lost treasures, buried ancient cities and modern machinations to recover them; it is the magical world of Computers. A world filled with mysteries, inventions, tools, blueprints and intrigues. His current pursuit is the exploration of T_EX, the kingdom of computerised typesetting. In his adventures, the protagonist has nothing to be jealous of his fellow Hollywood adventurers. He experiences all sets of heroic emotions: enthusiasm in the apocalypse of new trails and possibilities, courage in setting targets, fear when the object of desire is not easily conquered, despair of failure and misfortunes when around the corner something unexpected greets him.

The Actors

He soon understands that T_EX is the only serious alternative at that time for typesetting any text that contains even basic mathematics. It is royalty free to use, truly public domain, it is used by his fellow researchers typing away complicated formulæ, and its rivals –meagre word processors– do not offer a serious alternative to consider. It is true, the machinery available does limit the choice: an ATARI with a dual floppy system allowing the simultaneous use of two floppy disks represents the section's computer treasure. Yet T_EX is there, able to produce the required output, even on a noisy 8-pin dot-matrix printer. And for the protagonist, the quest begins. His first try, producing a thesis of around 100 pages

filled with mathematical forms and expressions, is in plain \TeX . A single invocation of the \TeX typesetter to process the thesis requires more than 3 minutes of his time (if no errors are encountered, of course). But it is all over after a single pass, and necessary diagrams are added later by hand.

The Inner Workings

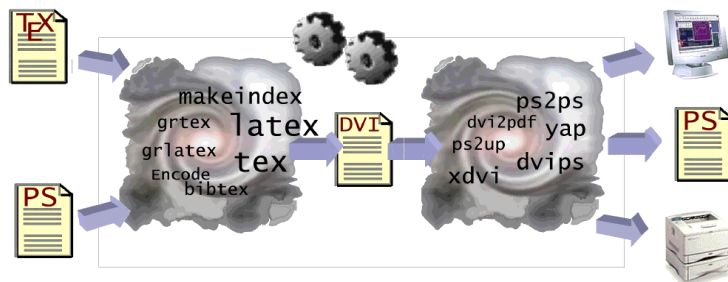
\TeX is all about producing an encoded device-independent file (code named DVI-file) from a text source, which contains the actual text to be printed sprinkled with commands on how to arrange it on a page. It is a true programming language, a tool that gives its user the freedom to describe in a formal way how his work should be finally presented on paper. In computer terminology, it encapsulates within the same source file both the object level data (the text) and the code (\TeX commands) for producing the required output. This property is what followers consider its strength and rivals its weakness. Like any other tool, its main purpose is to facilitate the completion of the task at hand. Yet, \TeX is a complicated tool with many controls, settings and idiosyncrasies. Not everyone using \TeX has the patience and interest to infiltrate behind the scenes, to learn and use this tool imaginatively to its full potential. And this was the case for quite a number of academics (the main clients) which were obliged by the general flow to make use of it.

Let there be Friends

As the usage of computers became more expanded in processing text documents in the 90s, the usage of \TeX had further developed to engulf the numerous facilities that any serious authoring activity requires. These include pre-defined document templates, table of contents, indexes, references to figures, texts, bibliographical citations, inclusion of graphical images, output to a portable printer format. Most of these extra tools, now faithful companions of \TeX , have been built in its own spirit and image. They include \LaTeX (a document preparation tool) $\text{BIB}\text{\TeX}$ (for bibliographical citations), $\text{SLI}\text{\TeX}$ (for the preparation of slides), makeindex (for building indexes), xfig (and X-windows drawing program), $\text{GREEK}\text{\TeX}$ and $\text{GRI}\text{\TeX}$ (for typesetting Greek texts), dvips (for converting DVI files to Postscript format files) as well as numerous utilities (like ps2ps , ghostview , *etc.*) for handling, transforming and printing Postscript files.

Darkening Skies

Very soon, processing \TeX files was no longer an easy and straightforward task, despite the fact that the workstations available to the research community had by then made invoking and running \TeX a quick and painless processes. The usage of numerous tools required multiple invocations of the \TeX processor over the same source file, alternating with calls to \TeX 's companions. The necessity of including images and diagrams also required at that time the production of a

Figure 20: Conceptual diagram of the processing of T_EX files

Postscript output file. Even the resulting Postscript file frequently needed post-processing, to filter for example the odd pages, or to create a booklet, or even to create a two-up or four-up proofing copy. Furthermore, an army of auxiliary files littered the disk area that the original T_EX files occupied. It was for our protagonist, as for most T_EXers, a strenuous and copious process indeed. The original enthusiasm –being able to produce a high quality document with the available technical means– was gradually substituted by the dark cloud of tediousness and mechanical repetition in producing this document: editing the source file, invoking `tex` or `latex`, then T_EX's companions, then again `tex`, correct all the minor errors which prevented T_EX from producing a `dvi` file, until you could see your recent modifications to the document on screen. And the cycle began again and again.

Analyse and Conquer

Nevertheless, a careful look reveals that the processing of T_EX files is more tedious than complicated. When tedious repetition of methodical tasks is anticipated, it is a direct call for computer automation. At last! A new light appears. A novel quest starts. Amazement; bewilderment; consideration of possibilities. This is the part where the protagonist, happy and excited, takes a step back to examine the situation and to plan his actions.

The processing of T_EX files is essentially a two-step process. The source files (T_EX source code, Postscript images) are ground together through successive invocations of T_EX and friends to produce a `dvi` file. This file is further processed, ending up either on a screen, printed on a piece of paper or saved in a portable format, be it Postscript or PDF (*see* Figure 1).

It's all Greek to me

But, further complications lay ahead for our protagonist. With the introduction of GREEKT_EX, yet another step was necessary. Texts written in the Greek ISO encoding (8859-7) were not compatible with the internal representation of

GREEK \TeX which was created to allow input files in ancient Greek using the standard Latin alphabet. A pre-processing phase to convert the file to GREEK \TeX character encoding was necessary. A number of utilities appeared (like `gr2gr` by A. Haritsis) that could perform the necessary conversions and these were added to the companions of \TeX .

Preparing for Battle

Technology exists to serve us. It is present to facilitate our tasks, to make our work easier and more fun to perform. What should a tool be able to do to help us in the processing of \TeX files? Brewed in the ideals of UNIX, our protagonist will readily suggest the creation of a filter, a converter, a tool that would apply transformations to a set of input \TeX files in order to produce a portable, device-independent file. A simple command line utility that taking as an argument the name of a \TeX file will analyse it, process it accordingly and output –where requested and in the appropriate format– the device-independent file. Since technology has allowed us the speed of multiple \TeX invocations at little cost, it should be a straightforward and clean task.

The blueprint to success

An algorithm is the strategic plan for winning a battle. Generic enough, it does not talk of soldiers but of battalions, it does not talk of guns but of artillery. For the processing of \TeX files, the following is the blueprint that will lead the protagonist to succeed in his quest.

- Target 1: *Analyse the source*

input:

A source file to be processed by \TeX

output:

Information on how the file should be processed

- Understand character encoding (ISO 8859–1/ISO 8859–7)
- Understand format file required (\TeX , \LaTeX , $\mathcal{A}\mathcal{M}\mathcal{S}$ - \LaTeX , $\LaTeX 2\epsilon$, GREEK \TeX , NFSS \LaTeX , GR \LaTeX , SL \TeX *etc.*)
- Understand if BIB \TeX is necessary
- Understand if `makeindex` is to be used
- Understand if special packages are used which require specialised treatment (e.g. chapter bibliography package were all included files must be filtered through BIB \TeX)

- Target 2: *Prepare the scene*

input:

The source file, information on how it should be processed

output:

A temporary working area and a decoded version of the \TeX file if necessary.

- Create a temporary area to work
- If the source file is encoded, perform decoding operation
- Repeat this step for all included files if necessary
- Perform any other pre-T_EX transformations
- Target 3: *Process the T_EX file to produce a DVI file*
 - input:**
The source file, information on how it should be processed
 - output:**
A device-independent (DVI) file
 - process the source file using the appropriate format (L_AT_EX, A_MS-L_AT_EX, G_RI_LA_TE_X, etc.) once
 - for each of T_EX's friends which are required (BIB_TE_X, makeindex etc.), let each of them perform its job and then grind the source file again through the necessary T_EX format to integrate the changes.
 - if L_AT_EX believes the further processing is necessary due to changed references, let the source file pass through another invocation of T_EX.
If an error occurs during any of these stages, present the user with the log file on screen, let him review the messages generated by T_EX and friends, cleanup the temporary work area and relax after a job well done.
- Target 4: *Process the DVI file*
 - input:**
A device independent file (DVI), information on how it should be processed
 - output:**
A file in portable format, either Postscript or PDF
 - convert the DVI file to either Postscript or PDF, as required
 - further process the Postscript file if required (convert it to booklet format, two-up format etc.) by using more of T_EX's companions
- Target 5: *Final Operations*
 - input:**
A file in portable format, information on how it should be processed
 - output:**
output of the file in the suitable medium (printer, screen, file).
 - Direct the portable-format file (using the necessary utilities) to the appropriate medium, be it the screen of quick viewing, a printer or a file
 - cleanup the temporary working area and relax

Using the blueprint

Following this blueprint, a program which can rightfully be called T_EX2PS or T_EX2PDF may be created. Such a utility will not only provide a finishing line for

our hero's quest, but it will also help numerous other $\text{T}_{\text{E}}\text{X}$ ers through their everyday life with $\text{T}_{\text{E}}\text{X}$. It is true that its conception and realisation reflects the ideals of a command-line world; it is a silent converter of $\text{T}_{\text{E}}\text{X}$ files into portable format files or a quick co-ordinator that will reflect on your monitor the changes you just made to your source file. Such a tool, probably makes more sense in a workgroup environment where it also reflects the system-wide availability of $\text{T}_{\text{E}}\text{X}$ tools. It is set-up once and numerous people (from $\text{T}_{\text{E}}\text{X}$ perts to novices) can benefit from its presence. It will repeat the same tedious steps of invoking, time after time, $\text{T}_{\text{E}}\text{X}$ and friends without ever complaining. Isn't this what computers thrive at without ever experiencing emotions, like being tired, frustrated, angry. Which is more than we can say for us human users. Indeed, our protagonist has managed to experienced all the range of human emotions while working with $\text{T}_{\text{E}}\text{X}$ for more than a decade.

The end?

Is this the end of our story? Well, the end depends on each of us. Each person will formulate his own ending. For some, this ending will be a happy one. For others, a sad and indifferent. It all depends on the reader's reaction to what was written here about $\text{T}_{\text{E}}\text{X}$ using $\text{T}_{\text{E}}\text{X}$. Some will use such a tool to help them automate what they could mechanically perform themselves. Others will just use it so as not to learn the mechanics of the processing of $\text{T}_{\text{E}}\text{X}$ files. The important factor is to learn to use the power computers offer to perform the tasks we have set for ourselves at a high level of quality.

A copy of the author's implementation of the co-ordinator utility using the standard UNIX shell (`/bin/sh`) can be found at <ftp://obelix.ee.duth.gr/pub/TeX> under the name `dotex`. It is adapted to the $\text{T}_{\text{E}}\text{X}$ subsystem that was employed at the Department of Computing, Imperial College in London with further customisations to allow it to run under NT using the recent porting of the GNU bash shell by CYGNUS (<http://www.cygnum.com>). It is not fully aware of $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}2_{\text{e}}$ and the newer additions to $\text{T}_{\text{E}}\text{X}$'s friends, but should be easy to customise according to needs.