

Desktop Publishing:

# Document Preparation with

# X<sub>Y</sub>TEX

Marcel Herbst  
4mation  
herbst@4mat.ch

September 25, 2009

## ABSTRACT

This paper, serving as an “appetizer” for the reader, introduces a particular document preparation system, called X<sub>Ǝ</sub>TeX, a variant of L<sub>Ǝ</sub>TeX. L<sub>Ǝ</sub>TeX was developed as typesetting system for technical documents, but it has also become a proper tool for poets, writers, philologists, social scientists, lawyers or educators, i.e., for all those who produce written material, care for the quality of presentation, and value effective authoring processes. L<sub>Ǝ</sub>TeX is not a so-called WYSIWYG — “What you see is what you get” — system. X<sub>Ǝ</sub>TeX, a newer variant of L<sub>Ǝ</sub>TeX, makes available a system whereby a L<sub>Ǝ</sub>TeX-type input can be used to generate pdf-documents that make use of the broad range of available font families (i.e. those that reside on the computer). The paper gives examples of how to structure documents, to use fonts and font families, and to include figures and pictures. Links are given to literature and software sources for those who are interested to know more.

## CONTENTS

<b>1</b>	<b>Foreword</b>	<b>4</b>
<b>2</b>	<b>Introduction</b>	<b>7</b>
2.1	TEX and L <sup>A</sup> TEX . . . . .	7
2.2	TEX and Unicode . . . . .	8
2.3	Implementation . . . . .	9
<b>3</b>	<b>Document Types and Document Structure</b>	<b>10</b>
3.1	Document Types . . . . .	10
3.2	Document Structure . . . . .	11
<b>4</b>	<b>Fonts, Color and Pictures</b>	<b>12</b>
4.1	fontspec . . . . .	12
4.2	Font Examples . . . . .	14
4.3	Color . . . . .	17
4.4	Pictures . . . . .	17
<b>5</b>	<b>Sources</b>	<b>18</b>
<b>A</b>	<b>Appendix</b>	<b>19</b>
	<b>Bibliography</b>	<b>29</b>

## I FOREWORD

This paper attempts to introduce the reader to a particular document preparation system, called X<sub>Y</sub>TeX, a variant of L<sup>A</sup>TeX<sup>1</sup>. It is addressed to people who work in environments I have some familiarity with, and among these to all those who are currently using Microsoft Word or a similar word processor to prepare their documents — letters, memos, drafts of articles, home assignments, presentations, or books — and are dissatisfied with the way these documents look or with the work processes required. If you are not dissatisfied with the word processor you use, or if you are already using X<sub>Y</sub>TeX or L<sup>A</sup>TeX, this paper is probably not for you.

Desktop publishing has revolutionized the way we and our offices work. We all have become much more productive. But it also had the effect to reduce the typographic quality of the printed — and of the electronic — product. Not professionals are in charge anymore of typesetting much of the printed material that cover our desks or fill our bookshelves, we ourselves, laymen most of the time, are in charge. And it shows. If I download a manual, a research report or a book in the form of a pdf-file from a respected agency, the OECD, for instance, or the European Research Council, or from a publishing house of repute, I am frequently ashamed by what I see. You might say that content matters, not empty formalism. Yes, I agree: content is paramount. But typography is no empty formalism. At least it need — or it should — not be that: it ought to help the reader to comprehend the reading material faster, it should guide him. And in my experience as an author, a clear typographic implementation will greatly enhance the authoring process.

L<sup>A</sup>TeX provides such a clear typographic implementation. It is far superior, and far more powerful, than word processors of the various denominations. And it is, in my view, also preferable to post-processors of text like Quark Xpress or InDesign that are frequently used to typeset books, provided we are focused on the structured documents we have in mind here; flyers, prospectuses, et cetera, with their often free form typography, are not the proper domain of L<sup>A</sup>TeX and its variants. L<sup>A</sup>TeX — and in particular X<sub>Y</sub>TeX — will not only greatly enhance the typographic quality of structured documents, it will also have a positive effect on authoring processes<sup>2</sup>. L<sup>A</sup>TeX is not only a typesetting system for mathematicians or natural scientists, as some people would suggest. True,

---

<sup>1</sup>If you are unfamiliar with a product name contained in this Foreword, don't worry: products are going to be explained in later sections of the paper, or they are not that important.

<sup>2</sup>This is true for a range of TeX-related software currently in development, but since I have not tried those, I shall confine myself here to L<sup>A</sup>TeX and X<sub>Y</sub>TeX. For references to other TeX-related software, see Section 5.

TEX, the basic typesetting software on which L<sup>A</sup>TEX is based, was originally crafted by Donald E. Knuth, the Stanford University information scientist, to typeset Knuth's own technical books [3, 2]. But in the intervening years, L<sup>A</sup>TEX has become a proper tool for poets, writers, philologists, social scientists, lawyers or educators, i.e., for all those who produce written material, care for the quality of presentation, and value effective authoring processes<sup>3</sup>.

I myself stumbled across TEX in the early 1980s. Personal computers were in their infancy, and PC-based word processing was barely superior than writing on an IBM-Selectric typewriter I had the occasion to own in the early 1970s. In fact, for technical reports, the IBM-Selectric setup was still preferable. In contrast, TEX, installed on a PC and coupled with an Epson matrix printer, produced absolutely stunning results using a decent font, including a large array of mathematical or miscellaneous symbols and Greek and Hebrew letters. Soon after I discovered TEX I learned of Leslie Lamport's L<sup>A</sup>TEX [6]. Since then, I have followed the development of L<sup>A</sup>TEX to some extent and I have used the software ever since. When I moved in 1988 to the Swiss Federal Institute of Technology in Zurich (ETH Zürich) to head its Office of Planning & Development, I had L<sup>A</sup>TEX implemented on our Apple Macintosh computers.

L<sup>A</sup>TEX was already used at the ETH Zürich, of course. Just a flight below our offices the Department of Mathematics was residing, and all the secretaries there were proud to be fluent in L<sup>A</sup>TEX. The central administration of ETH Zürich, however, was using some word processor to produce its documents, perhaps already Word, or some other variant of that type of software, I do not recall. When the scope of our tasks grew and when we started to use Oracle as a relational data base, we were looking for more powerful desktop computers and switched to Sun SPARCstations. On these we installed ArborText's Publisher (not available anymore), based on TEX, with a tag-guided input mask, not unlike LyX, a modern document processor equally suited for the audience I have in mind here<sup>4</sup>. The Publisher was powerful and easy to use and, as a TEX front-end, produced fine documents.

One day in the mid 1990s, the central administration of ETH Zürich de-

---

<sup>3</sup>In the proceedings of EuroBachO<sub>TEX</sub> 2007, Arthur Reutenauer [8] mentions two broad TEX communities: "The general scientific community and especially mathematicians and computer scientists"; and "People from humanities, especially in Ancient Greek and linguistics". These two communities *need* TEX i.e. there are not many proper alternatives. All the others have alternatives, if only deficient ones, presumably. My argument is that these additional communities could profit from TEX as well.

<sup>4</sup>See: <http://www.lyx.org/>; please note that all [www](#)-addresses in this paper are "active": click on them and you will reach the respective homepage.

cided to abandon all its Apple computers and Sun SPARCstations and to switch to PC's and concentrate on Microsoft software<sup>5</sup>. In our planning office, we retained some SPARCstations for data management and publication purposes, but apart from these, all the desktop computers were replaced and all internal writing was now based on Microsoft Word. The argument was that one software had to be used because documents were — and had to be — frequently produced or edited collectively, and that software had to be easy to use. That an arrangement was sought where people were able to work collectively on documents made sense; that the software was easy to use made sense as well; that a single software had to be installed was much more questionable:  $\LaTeX$  could have been that single software, but that would have been too much to ask. Granted,  $\LaTeX$  is not that easy to use, but Word isn't either<sup>6</sup> — contrary to what is being said so often.

The following years I was able to observe the ease and efficacy with which higher education administrators of all ranks were working, individually or collectively, on documents using Microsoft Word. I was constantly tempted to correcting other peoples' manuscripts: not the content of these papers, but the way people decided to work with — or *use* — Word. It would have been fruitless to make these corrections. The structured document features of Word were barely properly used because Word made it so easy to change typefaces and type-sizes: a title did not have to be 'declared' as a title of a particular hierarchical level, one could just write something, make it bold, change the font and its size, and center the whole thing. It is clear that under such sloppy application of Word, no table of content and no reference list was ever correct. Large documents ran easily into problems because the software could not handle it<sup>7</sup>. In addition, Word's word and letter spacing is badly deficient; it cannot handle ligatures; hyphenation is so deficient that practically no documents exist without wrong-placed hyphenation marks; to produce documents within a

---

<sup>5</sup>This decision did not affect academia, in that the departments and institutes were still free to choose their own infrastructure.

<sup>6</sup>Experience tells me that with  $\LaTeX$ , novices are operative within a few hours, provided there is somebody in the background to support them (and providing  $\LaTeX$  and document styles are properly installed).  $\LaTeX$  allows for a corporate identity of the office. People can concentrate on content, and their memos, letters, research papers, et cetera, will follow a chosen style. In contrast, users of Microsoft Word will very seldom graduate to become proficient, even after years of working with the product and, consequently, Word documents cannot easily follow the outlines of a corporate identity and are frequently deficient beyond the inherent deficiencies of the software.

<sup>7</sup>In the mid 1990s, ETH Zürich published a research report of some 1,600 pages (one single document containing tables of content; cross-references of sections, tables and figures; and various indexes). This report was (properly) produced with  $\LaTeX$ .

corporate identity style is practically impossible; et cetera.

Word survived, and badly produced documents survived. Most people do not care about printing and publishing standards, or do not have the nerve and the energy to care: they concentrate on what they have to do. But some people do care, and some additional people might care if they see the advantages of a new avenue and better authoring processes, and see the beauty of documents they could produce themselves. If you belong to this group of people, and if you are uninitiated to L<sup>A</sup>TEX or X<sub>Ǝ</sub>TEX, read on.

## 2 INTRODUCTION

This paper does not attempt to serve the role of a tutorial. Many fine books and manuals on TEX-related software are available, and the reader will find some of these listed under Section 5. Rather, the paper ought to serve as an appetizer, so to speak. Read — or glance — through it, and if you find the material appealing, for whatever reasons, you should consult some of the sources that I list.

The title of this paper talks about X<sub>Ǝ</sub>TEX, but since X<sub>Ǝ</sub>TEX is just a new version of L<sup>A</sup>TEX and retains the major characteristics of it, I shall talk about L<sup>A</sup>TEX when I address the generic characteristics of that software, and use the newer term only in reference to the specificities of that new version. Leslie Lamport's — old — L<sup>A</sup>TEX, hence, is a document preparation system of a markup-type based on TEX. It is not a so-called WYSIWYG — “What you see is what you get” — system. WYSIWYG-systems present their document on the computer screen as it will appear on paper (or as a pdf-document). This has some advantages: one can immediately see the outcome of one's typing; and the programs are, in contrast to L<sup>A</sup>TEX, forgiving (L<sup>A</sup>TEX may refuse to run if one makes a mistake). But it also has many disadvantages that make WYSIWYG-systems inherently deficient<sup>1</sup>.

**2.1 TEX and L<sup>A</sup>TEX** TEX itself is a system that revolutionized typesetting. Donald Knuth was so dissatisfied with the way his own books were typeset that he set out to write typesetting programs as well as font collections to support typesetting [4]. Initially, he just wanted to take a short time off to focus on this, but eventually he spent, together with many associates, years of

---

<sup>1</sup>This paper is not a crusade against Word or similar WYSIWYG-programs that have their own merit. It is a paper on X<sub>Ǝ</sub>TEX and its merits. Hence, I shall not list here all the inherent deficiencies of Word et al. (and I shall not — and cannot — list all the difficulties a TEXnovice might encounter).

work to accomplish the task. Knuth's work is now being supported and further developed by a global T<sub>E</sub>X-community<sup>2</sup>.

T<sub>E</sub>X is a system that focuses on the micro-typographical aspects of a document: the way letters or symbols are spaced, ligatures formed, words hyphenated, lines separated. It consists of roughly 900 commands, some 300 of which are basic, and the remaining 600 are combinations of basic commands. T<sub>E</sub>X can be used as a programming language. L<sup>A</sup>T<sub>E</sub>X is a markup-language: it is written in T<sub>E</sub>X and focuses on the macro-typographical aspects of document preparation: the layout of the page, the way chapter or section titles appear, headers and footers, footnotes, etc. L<sup>A</sup>T<sub>E</sub>X is much more simple to use than T<sub>E</sub>X, because T<sub>E</sub>X required the programming of macros which in turn would define the final layout of the document, and only the 'T<sub>E</sub>Xpert' could be expected to take on such a difficult task. L<sup>A</sup>T<sub>E</sub>X like T<sub>E</sub>X, works with two major files: an input file, containing text (i.e content) *and* markup-commands; and an output-file, containing the typeset text of the document. To distinguish text from markup-commands in the input-file, commands will always start with an *escape-character*, a '\'-sign.

In its early versions, L<sup>A</sup>T<sub>E</sub>X was very uniform in appearance. Documents would normally be set in just a few font families, preferably those that were designed by Knuth himself, and while the documents were properly typeset and easy to 'read', L<sup>A</sup>T<sub>E</sub>X-Documents were quickly perceived as looking boring by quite a few. This has changed in recent years. The T<sub>E</sub>X-community wrote style macros to offer a broad range of layout possibilities. Equally important, L<sup>A</sup>T<sub>E</sub>X documents would work with more fonts: font selection provisions, such as through the New Font Selection Scheme (NFSS), made it now possible to steer away from the bland image of L<sup>A</sup>T<sub>E</sub>X documents<sup>3</sup>. What is easy — and perhaps too easy — in the case of so-called 'WYSIWYG' word-processing-systems, namely to change the font, is fairly difficult in the case of L<sup>A</sup>T<sub>E</sub>X. This was definitely one reason why users were that reluctant to adopt L<sup>A</sup>T<sub>E</sub>X.

**2.2 T<sub>E</sub>X and Unicode** A real change, a leap forward, occurred when Jonathan Kew introduced X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X in 2004<sup>4</sup>. First developed for the Apple OS X,

<sup>2</sup>See: [www.tug.org](http://www.tug.org). You will find all you need to know under this address. T<sub>E</sub>X, L<sup>A</sup>T<sub>E</sub>X and X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X are in the public domain.

<sup>3</sup>A range of publishing houses, such as Springer Science + Business Media, use L<sup>A</sup>T<sub>E</sub>X to produce their Books and Journals. While all these publications are properly typeset, with very few exceptions, they are likely having a 'bland' appearance and rely often on an over-used typeface like Times-Roman.

<sup>4</sup>See: [http://scripts.sil.org/cms/scripts/page.php?site\\_id=nrsi&item\\_id=xetex](http://scripts.sil.org/cms/scripts/page.php?site_id=nrsi&item_id=xetex).

and then ported to Linux and Windows machines when Apple started to use Intel processors, X<sub>Y</sub>TEX has made it much simpler to change fonts. X<sub>Y</sub>TEX is a system that combines TEX with the font technologies of the respective operating systems and with unicode. In the old days, TEX accessed on the basis of a 7-bit-implementation at most

$$2^7 = 128$$

glyphs (or characters). Later on this was extended to  $2^8 = 256$  glyphs (the ASCII set). Some additional glyphs could then be constructed by combining characters of the original (ASCII) set. Unicode<sup>5</sup> provides now a practically unlimited encoding for the many characters of the various languages and script systems, and X<sub>Y</sub>TEX makes available a system whereby a L<sup>A</sup>TEX-type input can be used to generate pdf-documents that make use of the broad range of available font families. One of the reasons for this paper is to show this.

With the introduction of X<sub>Y</sub>TEX, X<sub>Y</sub>TEX replaces TEX, and L<sup>A</sup>TEX is being replaced by X<sub>Y</sub> L<sup>A</sup>TEX. With the exception of font calls, all the commands of L<sup>A</sup>TEX (and TEX) are retained.

**2.3 Implementation** To run TEX-related software, various installations in the public domain are being offered that can be obtained through a range of local TEX-Users-Groups<sup>6</sup>. The normal user will also want to have an input editor to write the required input files and to manage his files and projects. On my Mac OS X machines, I am using TEXShop<sup>7</sup>, Richard Koch's TEX editor and previewer. It allows me to: write input files (in a large typeface even a person can read whose eyesight is not — or not any longer — the best); chose various TEX-related programs, to compile these or to typeset text; select style templates, tags or macros; and to view the output file, the typeset text. In the Windows context, I am using Ingo H. de Boer's WinShell<sup>8</sup> and Aleksander Simonic's WinEdt Shell<sup>9</sup>; both editors have roughly the same functionality as that of the TEXShop (and the choice among the three will depend on the computing environment and working preferences).

---

<sup>5</sup>See: <http://unicode.org/>.

<sup>6</sup>See: <http://www.tug.org/>.

<sup>7</sup>See: <http://www.uoregon.edu/~koch/texshop/>.

<sup>8</sup>See: <http://www.winshell.de/>.

<sup>9</sup>See: <http://www.winedt.com/>.

### 3 DOCUMENT TYPES AND DOCUMENT STRUCTURE

L<sup>A</sup>T<sub>E</sub>X-type documents (such as those produced with X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X) fall into classes: books, reports, articles, letters, presentations, et cetera. T<sub>E</sub>Xperts write these class-styles, and one can find dozens of such classes on the web (under <http://www.tug.org/>). Within a document class, various parameters and package options will allow the specification of particular formats or styles. Coupled with the choice of typefaces, one can easily develop one's own 'corporate' style, or one can vary the appearance of documents.

**3.1 Document Types** As we have learned before, a X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X-Document will require an input file that we prepare using an editor (such as T<sub>E</sub>XShop, WinShell, or WinEdt Shell). Let us call this file `example.tex`. This input file will start with a first declarative statement, e.g.:

```
\documentclass[12pt,german,twoside]{article}
```

The statement tells that we have chosen the document-class of an 'article', like this paper (and not of a 'letter', for instance). Within brackets we will find options listed: the text is to be typeset in the size of 12pt, the text is in German (affecting hyphenation), and left and right pages will have their own respective layout. In the following line

```
\usepackage{fontspec,xunicode,fancyhdr}
```

required packages are loaded: the `fontspec` and `xunicode`-packages that we need to access fonts (see: Section 4.1); and the `fancyhdr`-package which we have chosen to set our headers and footers. Other packages can be loaded:

```
\usepackage{euler,graphics,url,babel}
```

The `euler`-package, for instance, to load the Euler-font to typeset mathematics; the `graphics`-package with which to program such logos as X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X, the `url`-package to write (and hyphenate) url-addresses, and the `babel`-package that is required to load a non-English language support (for a language like German). The command on the next line

```
\defaultfontfeatures{Mapping=tex-text}
```

will assure that either L<sup>A</sup>T<sub>E</sub>X-like input can be used to identify a glyph or the character palette of the editor (we shall discuss the details later on). Lastly, font families are to be chosen (apart from the Euler font for mathematical text, if we happen to chose this font; alternatively, another font for mathematics will have to be called or, if no other font is called, the native T<sub>E</sub>X-font will be used).

A roman font (in this case ITC Galliard Pro), with fancy ligatures and fraction constructions, is called as follows:

```
\setromanfont[Ligatures=Rare,Fractions=On]{ITC Galliard Pro}
```

Then we shall have define headers (and empty footers) using the fancyhdr-package:

```
\pagestyle{fancy}
\fancyhf{}
\fancyhead[LE,RO]{\bfseries\thepage}
\fancyhead[RE]{\bfseries\nouppercase <Author>}
\fancyhead[LO]{\bfseries\nouppercase <Title>}}
```

That is the end of specifications required for the document as such. The document itself is specified between two commands, `\begin{document}` and `\end{document}`: title, author, abstract (if we have a scientific paper), table of content, content and appendices, references, index. It should be noted that, in the case a foreign language under `babel` is being chosen, commands like `\abstract` or `\tableofcontents` are automatically translated into the chosen language.

For the novice, all these commands are perhaps pretty confusing, all the more so as L<sup>A</sup>T<sub>E</sub>X (and X<sub>Y</sub>TEX) is a programming language and one single mistake may bring the program to a halt. But modern L<sup>A</sup>T<sub>E</sub>X or X<sub>Y</sub>TEX are not that foreboding, and templates — similar to that presented in Figure 3.1 — are available to guide T<sub>E</sub>Xnovices<sup>1</sup>.

**3.2 Document Structure** L<sup>A</sup>T<sub>E</sub>X, and X<sub>Y</sub>TEX, make it easy to structure documents (see Figure 3.1). To structure documents is an important authoring tool. Before we start to write, we normally have a rough table of content in mind. In the context of a book, we normally know prior to the start of writing which major chapters the book should contain. If, during the course of writing, a new chapter is to be added, or two chapters are to be combined, or one chapter theme is to be eliminated, this poses no problems to L<sup>A</sup>T<sub>E</sub>X.

In the case of a longer paper or book, I normally work with a master document which does not include any sections or text. This master document reads in files which contain the text I would want to include. Between the command `\tableofcontents` and the command `\end{document}` I include a range of `\include{<file_1>} ... \include{<file_n>}` statements. These

<sup>1</sup>Those who are repulsed by the archaic, non-intuitive style of this software may try LyX. In any case, novices will profit from colleagues who are already users of T<sub>E</sub>X-related software, and such users, or T<sub>E</sub>Xperts, one can find in most university settings.

read-in files in which I have composed the text I want to use. Files can be organized hierarchically: a file to be read-in can itself include an `\include` command, et cetera. In this way, I can separate figures or tables which would rip apart my text too much; because  $\LaTeX$  can blank-out text (to the right of a % sign), I can selectively include or exclude a text portion (by preceding or not preceding an `\include` statement with a % sign).

This feature is very practical when working on a larger document over an extended period of time. In the case of a book, one can work with a chapter at the time, without having to deal with the entire document all the time. And as the ideas and document structure evolve, it is very easy to keep abreast with a changing format. With  $\LaTeX$ , the author has always a typographically clean document to work with and, to be honest, I do value this feature a great deal: it helps me in the process of writing and putting my ideas into print.

## 4 FONTS, COLOR AND PICTURES

One of the limitations of  $\LaTeX$  is the use of a broader range of fonts (at least for most people). This gives  $\LaTeX$ -documents, as was mentioned above, a certain uniform appearance (see Section 2.1). Donald E. Knuth, in fact, was very sensitive to the issue of fonts and font design<sup>1</sup>, but very many fonts were not made available. For technical documents this was no major limitation, since a range of fonts were available, apart from Knuth's own design, called Computer Modern [1]. But the normal author, recruited from the many non-technical fields, wished to play with more fonts.

**4.1 fontspec** Will Robertson is the author of a new `fontspec`-package<sup>2</sup> that allows the automatic use of OpenType-fonts: each font contained in the FontBook of Mac OS X, for instance, can be used, including its Apple Advanced Typography (AAT) features. The package supports features such as colors, ligatures, kerning, the presentation of numerals and quotients, subscripts und superscripts, vertical scriptures (such as Chinese), as well as automatic hyphenation (various languages). All in all, 387 languages are supported, from Abaza, Akhazian, Adyghe, Afrikaans to Italian, Hebrew, Javanese, Yiddish, Japanese, to Chinese (Phonetic, Simplified, Traditional), Zande and Zulu<sup>3</sup>.

<sup>1</sup>Knuth designed his font families in such a way so that small fonts hat slightly different shapes than large fonts, a feature found in the old lead-type fonts of font foundries. Most font designs today offer one shape, irrespective of the size of the font.

<sup>2</sup>See: <http://www.ctan.org/tex-archive/macros/xetex/latex/fontspec/>.

<sup>3</sup>The ranking of these languages or scripts is alphabetic according to the OpenType *tag*.

Figure 3.1: Commands and Structure of a Document

```
\documentclass[12pt,draft,german,twoside]{article}
\usepackage{euler,graphics,url,babel}
\usepackage{fontspec,xunicode,fancyhdr}
\defaultfontfeatures{Mapping=tex-text}
\setromanfont[Ligatures=Rare,Fractions=On]{ITC Galliard Pro}
\pagestyle{fancy}
\fancyhf{}
\fancyhead[LE,RO]{\bfseries\thepage}
\fancyhead[RE]{\bfseries\nouppercase <author>}
\fancyhead[LO]{\bfseries\nouppercase <paper title>}

\begin{document}

\title{Arbeiten mit \XeTeX und {\tt fontspec}}
\author{Marcel Herbst\{\bf 4mat}ion}
\date{\today}
\maketitle

\abstract{<Text>}
\tableofcontents
\section{<heading>}
<here comes the text>
\subsection{<heading>}
<here comes the text>
\subsubsection{<heading>}
<here comes the text>
\section{<heading>}
<here comes the text>
...
<et cetera>

\end{document}
```

Figure 4.1: Ligatures

fi, fl, ff, ffi, fj, ct, st

Figure 4.2: Schriftschnitt Didot

Schrift- und Zahlenbeispiel  
1234567890

To address the many glyphs in X<sub>Y</sub>TeX, one can use a character palette (and the mouse), or the keyboard. In the context of TeX and L<sup>A</sup>TeX, commands are used to address glyphs: `\"u` (to produce a ‘ü’), `\'e` (to produce a ‘é’). To assure the compatibility with TeX and L<sup>A</sup>TeX, X<sub>Y</sub>TeX requires in its preamble two commands:

```
\usepackage{xunicode}
\defaultfontfeature{Mapping=tex-text}.
```

A keyboard-input may look as follows: `<alt>--` for ‘—’ or `<alt>-` für ‘-’, etc. Simpler, however, is the normal TeX-form of input, such as ``citation'` for “citation”, ``term'` für ‘term’, `---` for — (dash).

**4.2 Font Examples** Some examples might be illustrative. X<sub>Y</sub>TeX, like TeX, supports ligatures, in that combinations of glyphs, such as `fi`, `fl`, `ff`, `ffi`, `fj`, `ct`, `st`, produce not a succession of characters as the input might suggest, but a corresponding new glyph (see Figure 4.1). In addition, the fraction-slash (a special character) can be used to typeset quotients without the use of the mathematic-mode of TeX (see Figure 4.5). A font like Didot, with the numerals in old style, can be specified as follows:

```
\fontspec[Numbers=OldStyle]{Didot}
```

Akzidenz Grotesk (Medium) — the old classic font of Swiss graphic design-

Figure 4.3: Accented Letters and Special Characters

ä, Ä, ö, Ö, ü, Ü, ß  
 æ, Æ, À, à, â, â, é, è, ç  
 ð, Ð, ł, Ł, Š  
 ©, ®, ™, €, ¥, ‰, §

Figure 4.4: Mathematics

$$\sum_j x_{ij} \geq b_i, \forall i \in I$$

ers, the predecessor of fonts like Helvetica oder Arial, that is distributed by Berthold — can be specified in the following way (see also Figure A.5):

```
\fontspec[Scale=<factor>]{AkzidenzGroteskStd-Md}
```

Accented letters characteristic of the various languages can be typed using the TEX input form just described, they can be keyed-in directly using the keyboard (if the accented letters have a corresponding key on the board), or they can be inserted into the text using the character palette (see Figure 4.3). For each font family, a range of typefaces can be accessed, such as those shown in Figure 4.6. Since X<sub>Y</sub>TEX and X<sub>Y</sub>L<sup>A</sup>TEX share the commands with TEX and L<sup>A</sup>TEX (respectively, with few exceptions), mathematical text can use the same input language (see Figure 4.4). Examples of fonts are to be found in Figures A.1 and A.5, and examples of font sizes in Figures A.2 and A.4.

Figure 4.5: Quotients

$$\frac{1}{2} \quad \frac{1}{4}$$

Figure 4.6: Typefaces

Roman  
*italic*  
**bold Roman**  
*bold Roman italic*  
 ROMAN CAPS

Figure 4.7: Japanese (Kanji numerals)

壹 貳 參 肆 伍 陸 柒 捌 玖 零

Figure 4.8: Yiddish

אײן מאָל האָט מען געמאַכט אין דירה  
 אַ סעודה

Ein mol hot men gemacht in der dir a seyde (transcription).

**4.3 Color** Most documents do not require color, but some do. If you want to include color, you will need to include the color package:

```
\usepackage{color}
```

Color can be defined by referring to the standard colors of red, blue, green or yellow by specifying:

```
\color{<color>}
```

Alternatively, colors can be defined in the preamble of the document:

```
\definecolor{<mycolor>}{rgb}{x,y,z}
```

where  $x$ ,  $y$  and  $z \in [0, 1]$  to define the mixture of red ( $r$ ), green ( $g$ ) and blue ( $b$ ). For instance, the color of Figure A was defined by:

```
\definecolor{SomeColor}{rgb}{0.4, 0.1, 0.9}
```

**4.4 Pictures** In order to include graphic items (diagrams, photographs, etc.) produced outside of L<sup>A</sup>T<sub>E</sub>X, one has to call them with a

```
\includegraphics[scale=<scale factor>]{<file>}
```

command. There is also a starred version of this command that clips the image to a specified bounding box:

```
\includegraphics*[llx, lly] [urx, ury]{<file>}
```

where  $llx$  and  $lly$  are the lower left  $x$  and  $y$  coordinates, respectively, and  $urx$  and  $ury$  are the upper right  $x$  and  $y$  coordinates of the bounding box, respectively. Three examples of including pictures in a document you will find in the appendix:

- a box-plot from my book on *Financing Public Universities: The Case of Performance Funding* (Springer 2007; see: Figure A.8), called by

```
\includegraphics[scale=0.73]{<file>.pdf};
```

- a photograph by Ansel Adams I downloaded from the US Library of Congress (see: Figure A.9) and included with the command

```
\includegraphics[scale=0.295]{<file>.jpg};
```

- and a photo of a coat-rack in our vacation home (see: Figure A.10), and included with the command

```
\includegraphics[scale=0.7]{<file>.jpg}.
```

## 5 SOURCES

There are many sources documenting T<sub>E</sub>X-related material. Among the books, I started out with Leslie Lamport's user's guide [6], of course, but now the novice might consult other publications. I have used publications by Apostolos Syropoulos [9] or Helmut Kopka and Patrick Daly [5] and, of course, the detailed compendium by Frank Mittelbach et al. [7]. You will find many other publications, some of which are presumably more suited for beginners. The journal PracT<sub>E</sub>X<sup>1</sup> contains many articles specifically addressed to people new to the T<sub>E</sub>X-world.

Regarding X<sub>Y</sub>T<sub>E</sub>X, visit [http://scripts.sil.org/cms/scripts/page.php?site\\_id=nrsi&item\\_id=xetex](http://scripts.sil.org/cms/scripts/page.php?site_id=nrsi&item_id=xetex).

---

<sup>1</sup>See: <http://www.tug.org/pracjourn/>.

## A APPENDIX

Figure A.1: Examples of Fonts

Four font designs shall be shown here (using a text of Immanuel Kant, from his *Kritik der praktischen Vernunft*). All the fonts are typeset in 12pt: Gill Sans, Baskerville, Georgia, and Galliard Italic (in this order). The fonts will require a global declaration in the preamble of the document as follows: `\fontspec[<options>]{<font name>}<text>`.

---

Der theoretische Gebrauch der Vernunft beschäftigte sich mit Gegenständen des bloßen Erkenntnisvermögens, und eine Kritik derselben in Absicht auf diesen Gebrauch betraf eigentlich nur das *reine* Erkenntnisvermögen, weil dieses Verdacht erregte, der sich auch hernach bestätigte, daß es sich leichtlich über seine Grenzen, unter unerreichbare Gegenstände oder gar einander widerstreitende Begriffe verlöre.

---

Der theoretische Gebrauch der Vernunft beschäftigte sich mit Gegenständen des bloßen Erkenntnisvermögens, und eine Kritik derselben in Absicht auf diesen Gebrauch betraf eigentlich nur das *reine* Erkenntnisvermögen, weil dieses Verdacht erregte, der sich auch hernach bestätigte, daß es sich leichtlich über seine Grenzen, unter unerreichbare Gegenstände oder gar einander widerstreitende Begriffe verlöre.

---

Der theoretische Gebrauch der Vernunft beschäftigte sich mit Gegenständen des bloßen Erkenntnisvermögens, und eine Kritik derselben in Absicht auf diesen Gebrauch betraf eigentlich nur das *reine* Erkenntnisvermögen, weil dieses Verdacht erregte, der sich auch hernach bestätigte, daß es sich leichtlich über seine Grenzen, unter unerreichbare Gegenstände oder gar einander widerstreitende Begriffe verlöre.

---

*Der theoretische Gebrauch der Vernunft beschäftigte sich mit Gegenständen des bloßen Erkenntnisvermögens, und eine Kritik derselben in Absicht auf diesen Gebrauch betraf eigentlich nur das reine Erkenntnisvermögen, weil dieses Verdacht erregte, der sich auch hernach bestätigte, daß es sich leichtlich über seine Grenzen, unter unerreichbare Gegenstände oder gar einander widerstreitende Begriffe verlöre.*

---

Figure A.2: Variations of Font Sizes

Three examples show the variations of font sizes if scale factors 0.8, 1.0 and 1.2 are applied to the original 12pt Hoefler Text font. The font declaration will now have the following form:

```
\fontspec[Ligatures=Rare,<scale factor>]{Hoefler Text}
```

It should be noted that the line spacing is not (automatically) affected by the scale factor with which font sizes are changed.

---

Der theoretische Gebrauch der Vernunft beschäftigte sich mit Gegenständen des bloßen Erkenntnisvermögens, und eine Kritik derselben in Absicht auf diesen Gebrauch betraf eigentlich nur das *reine* Erkenntnisvermögen, weil dieses Verdacht erregte, der sich auch hernach bestätigte, daß es sich leichtlich über seine Grenzen, unter unerreichbare Gegenstände oder gar einander widerstreitende Begriffe verlöre.

---

Der theoretische Gebrauch der Vernunft beschäftigte sich mit Gegenständen des bloßen Erkenntnisvermögens, und eine Kritik derselben in Absicht auf diesen Gebrauch betraf eigentlich nur das *reine* Erkenntnisvermögen, weil dieses Verdacht erregte, der sich auch hernach bestätigte, daß es sich leichtlich über seine Grenzen, unter unerreichbare Gegenstände oder gar einander widerstreitende Begriffe verlöre.

---

Der theoretische Gebrauch der Vernunft beschäftigte sich mit Gegenständen des bloßen Erkenntnisvermögens, und eine Kritik derselben in Absicht auf diesen Gebrauch betraf eigentlich nur das *reine* Erkenntnisvermögen, weil dieses Verdacht erregte, der sich auch hernach bestätigte, daß es sich leichtlich über seine Grenzen, unter unerreichbare Gegenstände oder gar einander widerstreitende Begriffe verlöre.

---

Figure A.3: Variations of Line Spacings

Three examples show the variations of font sizes if scale factors 0.8, 1.0 and 1.2 are applied to the original 12pt Hoefler Text font. The font declaration have the form as outlined in Figure A.2. However, line spacing has been adjusted by a statement:

```
\begin{spacing}{<factor>} ... \end{spacing}
```

in accordance with the `{scale factor}`.

---

Der theoretische Gebrauch der Vernunft beschäftigte sich mit Gegenständen des bloßen Erkenntnisvermögens, und eine Kritik derselben in Absicht auf diesen Gebrauch betraf eigentlich nur das *reine* Erkenntnisvermögen, weil dieses Verdacht erregte, der sich auch hernach bestätigte, daß es sich leichtlich über seine Grenzen, unter unerreichbare Gegenstände oder gar einander widerstreitende Begriffe verlöre.

---

Der theoretische Gebrauch der Vernunft beschäftigte sich mit Gegenständen des bloßen Erkenntnisvermögens, und eine Kritik derselben in Absicht auf diesen Gebrauch betraf eigentlich nur das *reine* Erkenntnisvermögen, weil dieses Verdacht erregte, der sich auch hernach bestätigte, daß es sich leichtlich über seine Grenzen, unter unerreichbare Gegenstände oder gar einander widerstreitende Begriffe verlöre.

---

Der theoretische Gebrauch der Vernunft beschäftigte sich mit Gegenständen des bloßen Erkenntnisvermögens, und eine Kritik derselben in Absicht auf diesen Gebrauch betraf eigentlich nur das *reine* Erkenntnisvermögen, weil dieses Verdacht erregte, der sich auch hernach bestätigte, daß es sich leichtlich über seine Grenzen, unter unerreichbare Gegenstände oder gar einander widerstreitende Begriffe verlöre.

---

Figure A.4: Fonts of Arbitrary Size

With the *scale factor* (i.e. the option [Scale=<scale factor>]), one can generate fonts of arbitrary size.

---



ct æ  
E

---

Figure A.5: Other Examples of Fonts

---

GalliardStd-Roman GalliardStd-Roman GalliardStd  
*GalliardStd-Italic GalliardStd-Italic GalliardStd-Italic*  
 Hoefler Text Hoefler Text Hoefler Text Hoefler Text  
**FedraSansStd-Medium FedraSansStd-Medium**  
 Baskerville Baskerville Baskerville Baskerville Baskerville  
 American Typewriter American Typewriter American  
**Futura Medium Futura Medium Futura Medium Futura**  
**Georgia Bold Georgia Bold Georgia Bold Georgia**  
**Gill Sans Bold Gill Sans Bold Gill Sans Bold Gill Sans**  
**Arial Bold Arial Bold Arial Bold Arial Bold Arial Bold**  
 COPPERPLATE COPPERPLATE COPPERPLATE COPPERPLATE  
 Courier Courier Courier Courier Courier  
**Courier Bold Courier Bold Courier Bold**  
 Andale Mono Andale Mono Andale Mono Andale  
 Akzidenz-Grotesk Std Light Akzidenz-Grotesk Std Light  
 Akzidenz-Grotesk Std Regular Akzidenz-Grotesk Std Regular  
**Akzidenz-Grotesk Std Medium Akzidenz-Grotesk Std**  
**Akzidenz-Grotesk Std Bold Akzidenz-Grotesk Std Bold**

---

Figure A.6: Color Specimen

red blue  
green black  
turquoise yellow  
grey violet

Figure A.7: Specimen Company Name

**4mation**  
Higher Education Management  
& Publication Support

Figure A.8: Box-Plots produced by the R-System for Statistical Computing

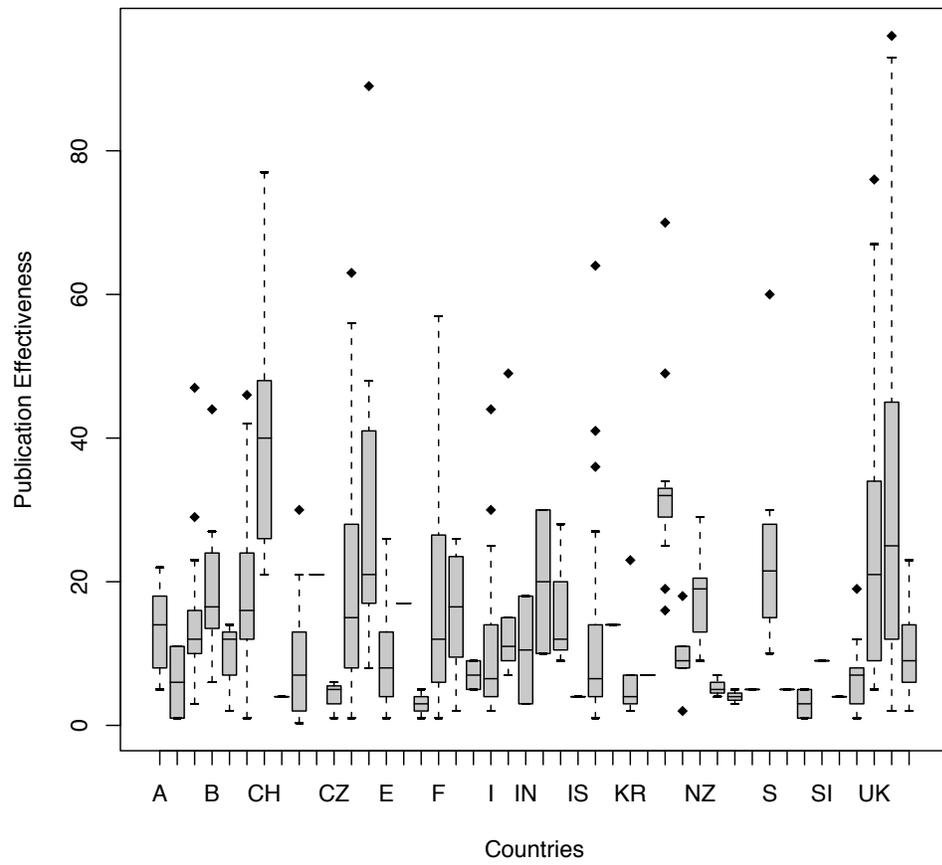
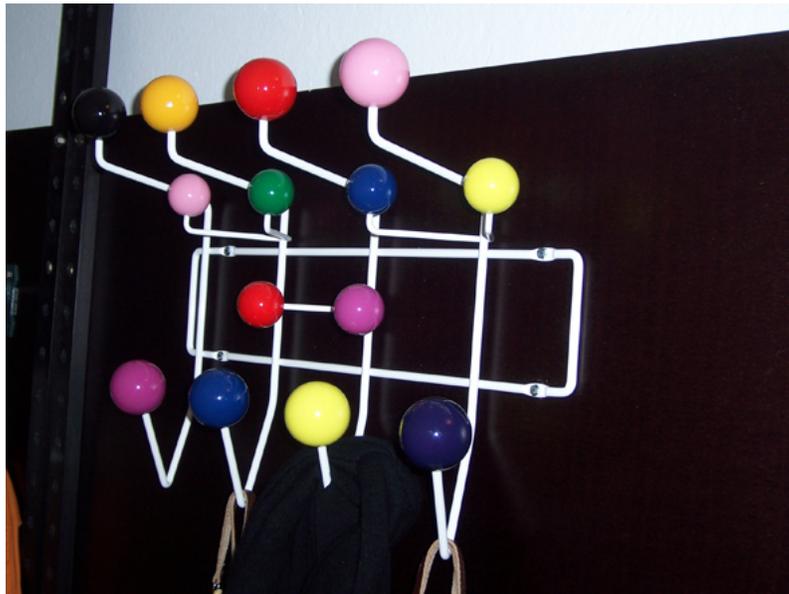


Figure A.9: Tom Kobayashi (Ansel Adams's Photographs of Japanese-American Internment at Manzanar, 1943)



Figure A.10: “Hang it all” (1953) by Charles and Ray Eames



**BIBLIOGRAPHY**

- [1] Alan Hoenig. *TEX Unbound: L<sup>A</sup>T<sub>E</sub>X & TEX Strategies for Fonts, Graphics, & More*. Oxford University Press, 1998.
- [2] Donald E. Knuth. *The TEXbook*, volume A of *Computers & Typesetting*. Addison Wesley, 6th edition, 1986.
- [3] Donald E. Knuth. *The Art of Computer Programming*, volume 1–3. Addison-Wesley Professional, 2nd edition, 1998.
- [4] Donald E. Knuth. *Computers & Typesetting*, volume A–E. Addison-Wesley Professional, 2001.
- [5] Helmut Kopka and Patrick W. Daly. *A Guide to L<sup>A</sup>T<sub>E</sub>X: Document Preparation for Beginners and Advanced Users*. Addison-Wesley Professional, 3rd edition, 2003.
- [6] Leslie Lamport. *L<sup>A</sup>T<sub>E</sub>X — A Document Preparation System — User’s Guide and Reference Manual*. Addison Wesley, 1985.
- [7] Frank Mittelbach, Michel Goossens, Johannes Braams, and David Carlisle. *The L<sup>A</sup>T<sub>E</sub>X Companion (Tools and Techniques for Computer Typesetting)*. Addison-Wesley Professional, 2nd edition, 2004.
- [8] Arthur Reutenauer. A Brief History of TEX, Volume II. *TUGboat*, 29(1):68–72, 2008.
- [9] Apostolos Syropoulos, Antonis Tsolomitis, and Nick Sofroniou. *Digital Typography Using L<sup>A</sup>T<sub>E</sub>X*. Springer, 2001.