How to Typeset an Archaeological Book with LaTeX

Jed Parsons

Department of Classics
University of California, Berkeley

e-mail: `jed@socrates.berkeley.edu`

1.   Introduction

The creation of the Camera-Ready Copy (CRC) for a book on ancient Greek technology and archaeology was prepared with the document preparation system LaTeX. The book, entitled *Hysplex*, by Panos Valavanis and translated into English by Stephen Miller, will be available from the University of California Press in the fall of 1999 (ISBN 0-520-09829-3). This article is intended to demonstrate that the freely-available document preparation systems TeX and LaTeX, while primarily known for their use in mathematical and scientific publication, are suitable tools for Classicists and other humanists for producing well-organized, maintainable, and beautiful texts.

2.   Getting Started with TeX and LaTeX

The typesetting program TeX, and the collection of sophisticated TeX macros called LaTeX, were originally designed with mathematicians and scientists in mind. From the start, they were powerful renderers of complex mathematical equations, but their capacity for rendering languages with non-Latin alphabets was limited. Those limitations have been tremendously reduced by the development of numerous ancillary packages which facilitate the creation of multilingual, structured documents. One can create a document that switches between languages (following hyphenation rules for each language separately), uses cross-references whose pagination is automatically generated, creates custom-formatted bibliographies from bibliographic databases, structures the presentation of the document in a controllable and logical way, and much more.

The functionality of TeX, LaTeX, and the accompanying packages written for
them, makes them a logical choice for humanists now as well as scientists.
For scholars of ancient languages, there are even macro packages for managing
manuscript collations and generating critical editions like those of the Oxford
Classical Texts or Teubner, with automatic line numbering, line referencing in
the critical apparatus(es), etc. Finally, TeX and LaTeX are freely available on
the Internet, or on CDROM for a small cost. The programs can be run on UNIX
(and Linux), Macintosh, Windows, and DOS, to name a few.

TeX and LaTeX (which I shall refer to collectively as LaTeX when the dis-
tinction is unimportant) are not easy to use, it is true. The user does not edit a
document as he or she would when using a word processor. Rather, one writes
markup instructions in a text-only document. This document is then processed
by an interpreter. The resulting output can be viewed or converted to another
file format, such as PostScript or PDF. This requires most writers to adjust the
way they think about composition. It is not my purpose to introduce the reader
to the fundamentals of LaTeX. There are many good books and online tutorials
for this. (See the references on page 31 below.) If you have had no exposure to
LaTeX, browse this document, see if illustrates the sorts of things you want to
do, and if it does, consider looking into some introductory material (see below).
If you have some previous experience, I hope you will find some value in the
more technical aspects of this text.

3.   Some Reasons to Use LaTeX

Why go to the trouble? Above all, for a beautiful book. Word processors, as
they are available today, are little more than glorified typewriters. They have
the ability to generate footnotes and switch fonts, yes, but they have no taste.
Word processors do not perform letterfitting operations and kerning well. They
do not try to balance the gray value (the ratio of black ink to white paper) of a
whole paragraph, shuffling line breaks and minimizing hyphenation on the way.
They do not automatically implement the ligatures intended by the designer
for the font used (e.g., fl, ffi, and perhaps ij, sc, etc.). They do not speak the
language of typographers, and cannot reliably be called upon to *accurately*
set text in points and picas (as any printing house will specify). They do not
manage cross references well (if at all) to text, footnotes, headings, figures,
etc. They do not generally interface well with bibliographic databases and the
like. LaTeX does all these things, and does them well. In sum, word processors
have forgotten (thanks probably both to the intervention of the typewriter
and a market-driven disinterest in relearning) the state of the art of printing
as it was in Gutenberg's day, and they have forgotten the skills developed by

centuries of scribes and calligraphers. The authors of TeX and LaTeX have done their homework: they have consulted with type designers, calligraphers, and typographers alike. The result speaks for itself. Yes, it is harder to use LaTeX than a word processor, but if you care about beauty, there is every reason to make the effort.

4.   Platforms

I run LaTeX on the Linux operating system (which, like LaTeX, is freely available with source code). If you have a commercial operating system, like MacOS or some variety of Windows, you are also in luck. The programs comprised by LaTeX have been ported to may platforms. Indeed, LaTeX documents themselves are inherently portable, since the document encoding is the same for every system. If you have ever received an email attachment formatted by some word processor you don't have, you will appreciate this.

The following sections briefly introduce some basic components of a LaTeX system, with some examples of their use or usefulness.

5.   Software Components of a LaTeX System

LaTeX resides on your system in a directory or folder hierarchy just as do any other applications you have. There are some key components to a basic LaTeX system, which I will briefly introduce here. As a Linux user, I am most qualified to talk about the UNIX utilities which I am familiar with. Similar utilities exist for other platforms. They will probably have slightly different names, but they will function in essentially the same way.

5.1.   `tex` and `latex`

These are the executable programs that interpret your source document and the markup instructions in it. They generate as output a DVI ("device independent") file. This file contains raw instructions for typesetting the page. A further interpreter is necessary to convert the data in this file into a printed output format, such as PostScript or PDF. DVI files are portable between people who have the same font libraries. These files contain specifications such as page size and the location and metrics of every glyph to appear on the page. Those glyphs themselves must be pulled from the font library and set by the final interpreter. (On my system, this is a program called `dvips` which turns DVI files into PostScript.)

LATEX is easily invoked at the shell prompt, taking your input file name as an argument. Input files can instruct other files to be loaded and merged. This makes it easy to keep large books divided into manageable chunks. For *Hysplex*, I created the following files containing parts of the book:

```
half-title.tex
cip.tex
dedication.tex
abbr+bib.tex
prologue.tex
introduction.tex
part1.tex
part2.tex
conclusion.tex
appendix.tex
glossary.tex
summary.tex
```

(There were also other parts of the document that were automatically generated along the way, like the table of contents, list of figures, and index.) All of these files, and my own macro sets, were loaded and arranged in the single file `hx_book.ltx`, so all I had to to to generate the book was type, at my shell prompt,

```
latex hx_book.ltx
```

5.2.   Style Files and Macro Packages

TEX is extensible by anyone who writes macros for it. LATEX is really a (huge) set of TEX macros. Macro packages which provide logical formatting commands may be called Style Files. I regularly use style files that enable me to set polytonic Greek and generate critical editions of texts for teaching. I also often need to format complex tables, include graphic images, construct linguistic trees and diagrams, faithfully represent a fragment of papyrus, etc. There are style files freely available to do all these things.

5.3.   `xdvi`

After LATEX is finished processing your input file (and assuming there are no errors for you to deal with), you receive a message like:

```
(see the transcript file for additional information)
Output written on hx_book.dvi (203 pages, 624220 bytes).
Transcript written on hx_book.log.
```

You probably want to see what your document looks like without having to print it. I do this with a program called `xdvi`, which is the UNIX X-Window system DVI file viewer. There are DVI viewers for other platforms as well. By invoking

```
xdvi hx_book.dvi &
```

I can see, but not edit, the generated output. If there is something to fix, I must edit the LATEX source and recompile it. (You don't have to restart `xdvi` or reload the file; it's sensitive to the change.)

5.4. dvips

To generate printable output, a utility like the UNIX `dvips`, which outputs PostScript code, is useful. To print on my (slightly broken) printer at home, which can only achieve a resolution of 300dpi, I type

```
dvips -D 300 hx_book.dvi
```

which spools a 300dpi PostScript file to the printer. If I want to output a copy at higher resolution, I might type

```
dvips -D 600 -o hx_book.ps hx_book.dvi
```

which instructs `dvips` to create the PostScript file `hx_book.dvi` at 600dpi. I do this regularly with all my documents. To print, I upload the PostScript file to the server at my university (you can use `ftp` for this, or `Fetch` if you are a Mac user; I use `scp`). In my office, I download the file to a Mac, and print it using the Apple LaserWriter Utility on one of my department's spiffy new printers. This way I don't have to mess around with floppy disks. If you have PostScript versions of each font you use, you can embed them in your document. This makes it scalable to any resolution. On my Linux-based LATEX distribution (teTEX), I type

```
dvips -Pcmz -o outfile.ps infile.dvi
```

In our case, we could not do this, since the Greek fonts we used to not yet exist in PostScript format.

5.5. Adobe Distiller and PDFTeX

Another alternative is to generate a PDF file from your DVI file. PDF is Adobe's Portable Document Format, which is rapidly becoming popular for online publishing. The program required to read PDF files, Acrobat Reader, is freely available from Adobe (`http://www.adobe.com`). Adobe's program to encode PDF files is not free, however, but can be purchased for something like US $50 for academic use. A nice alternative for TeX users is to get the free program PDFTeX; for LaTeX there is PDFLaTeX. These programs work very well and make web publishing easy. For more on PDF with LaTeX, see `http://www.math.uakron.-edu/~dpstory/latx2pdf.html`.

6. Logical Document Structure

LaTeX enables you to structure your document in a logical way. That is, you may make abstract divisions and entities, such as "section" and "figure" which are invoked by simple markup commands, and which will all be rendered in the final output according to the rules you specify. Thus, your document might have some encoding like:

```
\section{Observations on the Starting Lines of Various Stadia}

From the data that we have studied to this point and from the
conclusions that we have reached concerning the shape and the
operation of the Classical \emph{hysplex}, we may now proceed
to observations about some characteristics in the starting
lines in other stadia that seem to be relevant to the
\emph{hysplex} mechanism.

\subsection{Mainland Greek Stadia}

\subsubsection{Olympia}

Two constructions were added at the ends of the last phase
of the western \emph{balbis}. These project and make a right
angle with the \emph{balbis}, and they have created problems
for the excavators of the most famous stadium of the ancient
world.\grfn{E. Curtius, F. Adler, \emph{et al}., \emph{Die
Ausgrabungen zu Olympia}~V (1881) 25, pl.~35. Cf.~Romano
fig.~100.}  At one end ...
```

which, when it is parsed by LaTeX, produces the following output (here reduced to 80%).

Observations on the Starting Lines of Various Stadia

From the data that we have studied to this point and from the conclusions that we have reached concerning the shape and the operation of the Classical *hysplex*, we may now proceed to observations about some characteristics in the starting lines in other stadia that seem to be relevant to the *hysplex* mechanism.

Mainland Greek Stadia

OLYMPIA

Two constructions were added at the ends of the last phase of the western *balbis*. These project and make a right angle with the *balbis*, and they have created problems for the excavators of the most famous stadium of the ancient world.[176] At one end

If you're curious about the markup commands, `\section`, `\subsection`, and so forth, are macros that have been defined elsewhere (in my collection of macros associated with the book). When invoked, they format the text as you see, and also generate an entry for the table of contents, keyed to the correct page number. (The table of contents is also generated by macros.) This makes it easy to adjust layout. For example, if the subsection heading needs to be rendered in a larger point size, I can easily implement this change throughout the document by simply changing the declaration of the `\subsection` command slightly. The command `\emph` places the text in braces in italics. It has the added feature of adding a small space to correct for the slant of the italics before following words and tall punctuation marks, but not before periods and commas. You may be see this in the document you are reading now. `\grfn` is my footnote command that is sensitive to the use of Greek. You can't see the footnote in this example, but you can see that the command left a proper footnote mark above the text. Within the braces after the footnote command, you will see the tilde character; that's LaTeX-speak for a non-breakable space. I put these in the text wherever I want to ensure that there is no line break. I would normally put these between initials in peoples' names, too, but here the names are right at the beginning of the note, so I didn't bother.

6.1.   Frontmatter and Backmatter

The University of California Press has its own format that we had to adhere to. For example, front matter was to be numbered with lowercase Roman numerals; the main text and backmatter with Arabic. Furthermore, certain parts of the text were to begin on a new right page; others simply on a new page. A few LaTeX commands made it simple to put this in order:

```
\pagestyle{empty}
\pagenumbering{roman}
\input dedication                          % Dedication, no headers
             \cleardoublepage

\pagestyle{fancy}

\tableofcontents                           % TOC
             \cleardoublepage


                                           % LOF
\listof{hxfloat}{List of Figures}          % Our own float style
                                           % (see millermacs.tex)
             \cleardoublepage

\input prologue
             \clearpage

\input abbr+bib
             \clearpage
```

I've taken out a few formatting trivialities, but that's the bulk of the first part of the main input file `hx_book.ltx`. Simple commands specify the mode of page numeration. The `\input` commands merge in the other documents that comprise the whole book. The table of contents and list of figures are generated by a single command. LaTeX works out all the pagination itself. Finally, the simple commands `\clearpage` and `\cleardoublepage` ensure either a simple page break, or that the next section will start on a new right page. Backmatter is formatted in a similar way.

6.2. Table of Contents

When LaTeX comes across sectioning commands, it inserts the section name in a file it generates for you, wrapped in the proper formatting commands, and the proper page number. In our case, the command `\tableofcontents` generated a page starting like this (again, reduced to 80%):

<div align="center">

Contents

</div>

6.3. List of Figures

LaTeX also possesses a `\listoffigures` command. In our case, the normal LaTeX method for handling figures had to be altered, so we used a package called `float.sty` to customize both our floating figures and our list of figures. I'll say more about this below (page 26).

6.4. Page Numbering

Frontmatter uses Roman numerals, while the rest of the text uses Arabic numerals. LaTeX makes it easy to switch between the two: `\pagenumbering{roman}` and `\pagenumbering{arabic}` are the commands. Macros that generate lists of page numbers, like the table of contents, are given the correct numerals (see above). If you don't want to start with page number 1, reset the page counter.

For Hellenists who want to go the whole nine yards, it is possible to do page numbering using Ionian numerals.

6.5.   Headings

This is one area where LaTeX is wrongly considered difficult to customize. LaTeX provides its own heading macros. The press gave us specific instructions, which we followed by redefining LaTeX's sectioning commands. For example, the `\section` macro was redefined as follows:

```
\renewcommand{\section}{\thispagestyle{plain}
                       \@startsection%
                       {section}%
                       {1}%
                       {0mm}%
                       {-2\baselineskip}%
                       {4\baselineskip}%
                       {\normalfont\Large\centering}
                       }
```

A quick glance at a good LaTeX guide, like Goosens, Mittelbach, and Samarin's *The LaTeX Companion*, and you will easily understand the parameters of this declaration. (Backmatter sections have a slightly different definition in our book.)

6.6.   Running Headers

Hysplex is being published as a volume in a series which has its own heading format as well. Headings my be complicated by the constraint that pages that contain only figure normally do not have running headers. We oscillated between two ways of handling headings, and ultimately opted for the `fancyheadings` style package. We persuaded our editor to let us suppress running headers atop pages that contained only figures. Here are two ways to do this: First, our home-brewed approach, then the method provided by `fancyheadings`.

6.6.1.   The Home-Brewed Approach

First, the declaration of our even and odd page headers:

```
\makeatletter
```

```
\def\ps@textheadings{\ps@headings
   \def\@evenhead{\if@fcolmade%
    \else \thepage\hfil{\itshape\rightmark}\hfil\fi}%
   \def\@oddhead{\if@fcolmade%
    \else\hfil{\itshape\leftmark}\hfil\thepage\fi}%
}
```

This checks to see if any text has been placed on the page. If not, it suppresses the headers. It's followed by the next instructions, which reduce the space allotted to the header to 0 so that the figures may take up the entire available space of the page.

```
\pagestyle{textheadings}

% change spacing on float pages (\@fptop. \@fpsep, \@fpbot)

\setlength{\@fptop}{-1\headheight plus 1fil}
\addtolength{\@fptop}{-\headsep}
\setlength{\@fpbot}{-1\footskip plus 1fil}

\makeatother
```

6.6.2. *Using* `fancyheadings.sty`

It turned out to be easier to work with the built-in macros of `fancyheadings`. If you use this package, you may find the following declarations useful.

```
\pagestyle{fancy}
\setlength{\headrulewidth}{0pt}
\lhead[\fancyplain{}{\thepage}]%        % even pages, left
       {\fancyplain{}{}}%               % odd pp, left
\rhead[\fancyplain{}{}]%                % even pp, right
      {\fancyplain{}{\thepage}}%        % odd pp, right
\chead[\fancyplain{}{\itshape\leftmark}]% section on left
       {\fancyplain{}{\it Hysplex}}
\cfoot{}
```

Periodically, in the document, we had to redefine `\chead` when the running header had to change. (You can define your sectioning macros to do this for you, but for various reasons we opted to do it by hand.)

The `fancyheadings` method for suppressing headers on a float page is well documented. Use the `\iffloatpage` mechanism in your float definitions.

7.   Fonts

One of the notorious hassles with LaTeX is the use of fonts. TeX was written before the advent of PostScript, and is designed to work with its own ancillary font language, METAFONT. METAFONT is itself an extremely powerful tool. Like TeX, it too is an interpreter that parses instructions for drawing fonts, and generates collections of glyphs as output. The METAFONT language has not, as far as I know, been surpassed in the world of font generation tools. Its design language offers the programmer as many user-defined variables as he or she might want; in the modern, PostScript world, such variables are known as *axes*. The recently developed Multiple Master fonts from Adobe, which are regarded as the most versatile PostScript fonts yet, offer three axes of variation.

These days it is possible, though it's sometimes not simple, to use professional PostScript fonts with TeX and LaTeX. It's trivial to use cheap knock-off fonts that have no ligatures, small caps, hanging figures, etc.; it's also easy to use one of the thirty-five standard PostScript fonts (with good ligatures and kerning tables), for which helpful people have provided publicly all the necessary macros and metric files LaTeX needs (see your local CTAN site). Still, to use a professional font that follows the original calligrapher's design is considerably more difficult. In our case, we were lucky enough that all the necessary fonts exist in METAFONT form.

7.1.   Choosing the Right Font

Font selection should be deliberate. The press, when it specified the constraints for the CRC, suggested two possible fonts for us to use: Times New Roman and Palatino. Neither of these fonts appealed to me, though, for use in this project. Considering that the text is a scientific account informed by Classical Greek engineering texts and modern archaeological evidence, it seemed inappropriate to give the page the mixed Baroque and Neo-Humanist texture of Times New Roman, and no better to give it the Humanist face of Palatino. (These fonts are freely available for use with LaTeX, so it would have been no problem to use them.) Happliy, TeX's default font, Computer Modern, seemed an excellent choice. This typeface could be described as having a rationalist pen axis, and Neo-Classical design. Just the thing, I think, for the subject. The Greek font we used, similar to the famed Didot Greek, is perfectly compatible in my opinion.

This example shows the Roman face combined with the Greek orthotic and Greek oblique in a footnote. Note that the eight point footnote fonts are not simply scaled versions of the ten point text face. They have been recompiled by METAFONT with variations in design axis. They are most notably a little wider, by comparison, and darker, with the result that they balance visually

on the page with their bigger, ten-point siblings. Scaled fonts, in contrast, tend to look anemic in footnotes.

91. F. Dürrbach and P. Roussel, *Inscriptions de Délos* (1935) 3.29. In addition the word *hysplex* appears also in other Delian inscriptions: in *IG* XI2 199.23, a the year 274 BC there is mention among other expenses of the payment of on ὕσπληγας ἐπισκευάσαντι" (to Philotas for the construction of the *hysplex*) and im an expense of 5^1/$_2$ drachmas "ὁπλάριον τῇ ὕσπληγι" (for a small part for the *i* XI2 203.96 that is dated to 269 BC, and in the fragmentary *IG* XI2 260.4, a " (torchrace *hysplex*) is mentioned.

The book contains a summary in Greek. Here is a snippet. (Wedge-shaped monotonic accents are also available for use with LATEX.) NB: Greek typographers traditionally indent the first paragraph of a chapter, in contrast with standard American practice.

Το πρόβλημα των πρώιμων εκκινήσεων στους αγώνες δρόμου, πο βρίσχουμε με μεταφορική χρήση σε όλη την αρχαία ελληνική γραμματεί αι. μέχρι το τέλος της αρχαιότητας, αντιμετωπίστηχε αποτελεσματικά νους αξιωματούχους των αγώνων, με την τοποθέτηση συστημάτων α στις balbdec των σταδίων, μπροστά από τους δρομείς. Σε όλα αυτα που ξεκινούν από απλούς ξύλινους μηχανισμούς για να καταλήξου και στην εμφάνιση και στη λειτουργία τους κατασχευές, θα πρέπει

7.2. Advocacy for Fonts: A Plea for Responsibility

Tragically, typefaces are not legally defensible by copyright law. One can, in my country at least, copyright only the name of a font, and (as has recently been shown in a lawsuit between Adobe and a plagiarizer) the computer algorithm that describes a font. The letterforms themselves, which may have taken years for a skilled designer to create, are not regarded by law to be the the works of art they are, and are not given the protection they deserve.

The widespread distribution of cheap knock-off fonts and plagiarism itself hinders the digitization of a font by a firm that has payed for the rights to a good artist's design. A firm can simply not afford to spend its time and effort producing a typeface that will be (legally!) stolen as soon as it is made available. Let me give an example from my own experience. I was searching for a Greek face to accompany a certain Neo-Humanist typeface. In the specimen books, I came across a font by Hermann Zapf named *Heraklit*. The letterforms were once drawn by the calligrapher by hand for a book of Heraclitus' fragments, and then were cut in metal for an edition of the New Testament. I cannot find the words to describe the beauty of this typeface. No digitized version exists,

though it obviously should. I wrote to Mr. Zapf and asked if *Heraklit* would
ever be issued in digital form. He forwarded my letter to the Managing Director
of Linotype. Here is an excerpt from the response I received from him:

> Unfortunately, Heraklit is not available in digital format. It was
> produced by Linotype as a hard metal typeface but it was never
> digitized. These typefaces are amongst the most pirated ones . . . .
> We are not planning to produce Heraklit in the foreseeable future.
> We do like this type face, we have the originals, and we have Her-
> mann Zapf's authorization. But sadly the economics are just wrong.

Of course, the violence does not stop with the plagiarism itself. Plagiarizers
leave out all the "extra" characters: the ligatures, the lining figures, the small
capitals, the swash characters, and whatever else was created by the designer
as part of the work of art as a whole. Plagiarizers do a rough job with contours,
and a rougher job with kerns. So what we are left with are artists and foundries
put out of business by crooks, and shabby effigies of the original design installed
with every word processor in the world. This is a tragedy. I encourage any who
read this to treat fonts as works of art, and respect their designers. Use fonts
that are produced by firms that licensed the designs from the designer. You can
also write a letter, urging your government to regard typefaces as the works of
art they are and give them according legal protection.

8.   Figures

In the production of *Hysplex*, the figures will be pasted in by the publisher in
spaces left in the CRC. Thus all we need to do is leave a space on the page with
the correct caption below it, and the appropriate (probably shorter) caption in
the list of figures. We declared a new type of floating figure, dubbed `hxfloat`.

```
\newfloat{hxfloat}{tp}{lox}
\floatstyle{boxed}
\floatname{hxfloat}{\footnotesize Figure}
```

Let me explain these instructions. First, LaTeX must try to place the new float
either at the top of the page `hxfloat` is invoked on, or on a subsequent page
containing figures only (the meaning of `tp`). Information regarding the location
of figures will be written to a file with the suffix `.lox`. This float will be sur-
rounded by a box (useful at the draft stage, where we want to see the virtual
outline of the figure—in the final version for the printer, the linewidth of the

box will be set to 0). Finally, captions will begin with the word "Figure" in the same type size as the footnote text.

Another command, this one also called `hxfloat`, governs the appearance of the float.

```
\makeatletter
\newcommand\hxfig{%
   \@ifnextchar[%
     \threeargfig
     \twoargfig}
\makeatother

\def\twoargfig#1#2{\threeargfig[#2]{#1}{#2}}

\def\threeargfig[#1]#2#3{%
     \begin{hxfloat}
      \begin{center}
       \fbox{\parbox[b][#2in][s]{30pc}%
              { \vfil
                { \hfil Figure height\ #2 in \hfil }
                \vfil }}
      \end{center}
      \caption[#1]{\footnotesize $\greightpoint$#3%
                                 $\grtenpoint$}
     \end{hxfloat}}
```

This defines a command that takes two or three parameters, the first (optional) parameter being a short caption for the float which will be placed in the list of figures, the second (required) being the height of the figure (in inches), and the third (required) being the full caption to appear below the figure in the text. If you call `\hxfig` with just two arguments, it will behave as though you supplied the value for the second required argument (the full caption) to the optional argument as well. Thus, only if the list of figures needs a different version of the caption do you need to specify it by means of the optional argument. The `\fbox` command generates the box with the right height, with text inside stating the height of the figure. The `\caption` command supplies the text both for the list of figures and for the caption below the figure itself. Elsewhere, we have redefined `\caption` to center the caption horizontally if it is narrower than the width of the textblock, and to format it as a paragraph if it contains more than one line.

9.   Cross References

LaTeX and TeX (using `eplain` with the latter) have powerful tools for labelling and addressing cross references. Using automatic cross references, we don't have to write specifically

```
... see above, page~83, ...
and hope that the document's pagination will never
change---because it will.
Rather, we can write
\begin{verbatim}
... see above, page~\pageref{myref}, ...
```

This will produce in the output the page number associated with the location in the source file where the command `\label{mylabel}` was made. The more general command `\ref` is context sensitive; it knows if the `\label` is in a footnote, on a figure, or in a section or subsection, etc., and will produce the appropriate number. You can use the same label for as many `\ref` and `\pageref` commands as you want. For example,

```
... see above, note~\ref{myref} on page~\pageref{myref}, ...
```

will do what you think it does.

To make it easier for an author or editor to be sure that cross references are pointing at the right block of text, a useful package `showref` may be used. When included, `showref` causes a box to be printed in the margin with the keyword in it wherever a `\label` is invoked, and the keyword to be printed above the resolved reference in the text. When everyone is confident your labels are all properly placed, you can simply uninclude `showref`, and the marginal marks go away.

10.   Bibliography

The author did not provide a bibliography for this book. (There was instead a list of important bibliographic references at the beginning of the book, and numerous cross references bewteen footnotes to help the reader find other references.) For the sake of illustration, I've included bibliographic examples here from another source. There are three aspects to bibliographic references in LaTeX (and `eplain` TeX). First, you must have a database of sources; second, you can use various citation commands to produce references of your stylistic

choosing in the text; third, you can use any number of bibliographic style files, or create your own, to govern the format of your printed bibliography. There are other capabilities I won't go into here, such as the production of multiple bibliographies, the production of bibliographic reference cards for yourself, etc.

The bibliography database has entries like this:

```
@book{vendryes:1902,
   author = {J. Vendryes},
   title = {Recherches sur l'histoire et les effets
            de l'intensit\'e initiale en latin},
   year = 1902,
   publisher = {C. Klinksieck},
   address = {Paris},
   annote = {PA~2119 .V45 1902}
}
```

The database itself is a file terminating in the suffix `.bib`. The first string after the open quote is the keyword. I like keywords of the format `author:year`, because they are easy to remember or guess.

In the text, you use citation commands that reference the keyword of the bibliographic entry, e.g., `\citeN{vendryes:1902}`.

Somewhere in your document source, you will have commands like these.

```
\bibliographystyle{tapa}
\bibliography{bib-file-name}
\nocite{*}     % print all bib entries, not just those cited
```

This will cause the bibliography to be printed here, formatted according to the rules in the bibliographic style file (terminating in `.bst`) which you specify. In this example, I have created a style file with the `makebst` program (freely available) which emulates the bibliography format of the Classics journal TAPA.

The result is text that looks like this (here, a footnote):

---

[7] The standard bibliography includes, for a quantitative solution, Leo (1905), Burger (1928), de Groot (1934), and Cole (1969); for an accentual solution, Lindsay (1893); for a syllable-counting solution, Pighi (1957) (see below note 24). For a recent discussion and assessment of the bibliography, see Kloss (1993).

Some of the bibliography (with `tapa.bst`) looks like this:

Prosdocimi, A. L., 1986. "Sull'accento latino e italico." In A. Etter, ed., *o-o-pe-ro-si, Festschrift für Ernst Risch zum 75. Geburtstag*, pages 601–618. Berlin: de Gruyter.

Soubiran, J., 1988. *Essai sur la versification dramatique des Romains.* Paris: Centre Nationale de la Recherche Scientifique.

Steriade, D., 1988. "Greek Accent: A Case for Preserving Structure." *Linguistic Inquiry* 19(2): 271–314.

11.   Index

Indices are easy to generate with LaTeX and `eplain` TeX using the simple command `\index`. You can arrange a hierarchical index, with nested entries, sorted in any order you want. This last feature was particularly important in this book, where we needed Greek words in the index alphabetized according to English rules.

One problem we encountered was that `\index`, as implemented, does not behave differently when called from within a footnote. Happily, the Usenet group `comp.text.tex` is very active and helpful. Hours after we posted a question to the group, Stefan Ulrich responded with this definition for `\footindex`.

```
\makeatletter
\newcommand\asfoot[1]{\@printfoot#1\end}
\def\@printfoot#1-#2\end{#1 {\it n}\/#2}%

\newcommand*\@wrfootindex[1]{%
\protected@edef\@tempa{\noexpand\protected@write\noexpand\@indexfile{}%
    {\string\indexentry{#1|asfoot}{\noexpand\thepage-\thefootnote}}}%
  \@tempa
   \endgroup\@esphack}


\newcommand\footindex{\@bsphack\begingroup
\@sanitize
\@wrfootindex}
\makeatother
```

Here is a sample of the book's two-column index. Greek readers will find the alphabetization strange, but it was done this way to meet the expectations of English-speaking readers.

When looking at a printed copy, just as you might want to know where you invoked your label and reference commands, you (or someone else) might want to see exactly where you invoked indexing commands. Use the style file `showidx` and with no extra instructions, LaTeX will print your index keywords in the margin on the page in which they were declared.

12.   Crop Marks

The printer who is photosetting your CRC will need cropmarks to center the camera with. There are many macro packages for generating these. We used `cropmark.sty` and set the following parameters:

```
\setlength{\cropwidth}{.2pt}
\setlength{\cropsep}{1pc}
\setlength{\croppadlr}{1pc}
```

13.   Further TeX and LaTeX Resources

13.1.   Online

`http://www.tug.org/`

`http://www.ctan.org/`

Usenet: `comp.text.tex`

13.2.   Printed

Don Knuth, *The TeXbook*. Addison-Wesley.

Leslie Lamport, *LaTeX: A Document Preparation System*. Addison-Wesley.

Michel Goossens, Frank Mittelbach, and Alexander Samarin, *The L^AT_EX Companion*. Addison-Wesley.

Michel Goossens, Sebastian Rahtz, and Frank Mittelbach, *The L^AT_EX Graphics Companion*. Addison-Wesley.

Alan Hoenig, *T_EX Unbound*. Oxford University Press.

Robert Bringhurst, *The Elements of Typographic Style*. Hartley & Marks.