A Guide to

LaTeX

and Electronic Publishing

Fourth edition

Helmut Kopka

Patrick W. Daly

Addison-Wesley

Harlow, England • Reading, Massachusetts • Menlo Park, California
New York • Don Mills, Ontario • Amsterdam • Bonn • Sydney • Singapore
Tokyo • Madrid • San Juan • Milan • Mexico City • Seoul • Taipei

Addison Wesley Longman Limited
Edinburgh Gate
Harlow
Essex CM20 2JE
England

and Associated Companies throughout the World.

Cover designed by Designers & Partners, Oxford
Typeset by the authors with the LaTeX Documentation System
Printed in Great Britain by Henry Ling Ltd, at the Dorset Press, Dorchester, Dorset

First published 1993
Second edition 1995
Third edition 1999. Reprinted 1999, 2000
Fourth edition 2004

ISBN ???????????

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library

Library of Congress Cataloging-in-Publication Data
Kopka, Helmut.
A guide to LaTeX : and Electronic Publishing
/ Helmut Kopka, Patrick W. Daly -- 4th ed.
p. cm.
Includes bibliographical references and index.
ISBN 0-201-39825-7
1. LaTeX (Computer file) 2. Computerized typesetting. I. Daly, Patrick W. II. Title.
???????????
??????????? ???????
CIP

Trademark notices

METAFONT™ is a trademark of Addison-Wesley Publishing Company.
TEX™, $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX™, and $\mathcal{A}\mathcal{M}\mathcal{S}$-LATEX™ are trademarks of the American Mathematical Society.
Lucida™ is a trademark of Bigelow & Holmes.
Microsoft®, MS-DOS®, Windows®, Internet Explorer® are registered trademarks of Microsoft Corporation.
PostScript®, Acrobat Reader®, Acrobat logo® are registered trademarks and PDF™ a trademark of Adobe Systems Incorporated.
UNIX® is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company, Limited.
VAX™ and VMS™ are trademarks of Digital Equipment Corporation.
IBM® is a registered trademark and `techexplorer` Hypermedia Browser™ a trademark of International Business Machines Corporation.
Netscape™ and Netscape Navigator™ are trademarks of Netscape Communications Corporation.
TrueType™ is a trademark and Apple® and Macintosh® are registered trademarks of Apple Computer Inc.

Preface

A new edition to *A Guide to LaTeX* begs the fundamental question: Has LaTeX changed so much since the appearance of the third edition in 1999 that a new release of this manual is justified?

The simple answer to that question is 'Well . . . .' In 1994, the LaTeX world was in upheaval with the issue of the new version LaTeX 2ε, and the second edition of the *Guide* came out just then to act as the bridge between the old and new versions. By 1998, the initial teething problems had been worked out and corrected through semi-annual releases, and the third edition could describe an established, working system. However, homage was still paid to the older 2.09 version since many users still employed its familiar syntax, although they were most likely to be using it in a LaTeX 2ε environment. LaTeX has now reached a degree of stability that since 2000 the regular updates have been reduced to annual events, which often appear months after the nominal date, something that does not worry anyone. The old version 2.09 is obsolete and should no longer play any role in such a manual. In this fourth edition, it is reduced to an appendix just to document its syntax and usage.

But if LaTeX itself has not changed substantially since 1999, many of its peripherals have. The rise of programs like pdfTeX and `dvipdfm` for PDF output adds new possibilities, which are realized, not in LaTeX directly, but by means of more modern *packages* to extend the basic features. The distribution of TeX/LaTeX installations has changed, such that most users are given a complete, ready-to-run setup, with all the 'extras' that one used to have to obtain oneself. Those extras include user-contributed packages, many of which are now considered indispensable. Today 'the LaTeX system' includes much more than the basic kernel by Leslie Lamport, encompassing the contributions of hundreds of other people. This edition reflects this increase in breadth.

The changes to the fourth edition are mainly those of emphasis.

1. The material has been reorganized into 'Basics' and 'Beyond the Basics' ('advanced' sounds too intimidating) while the appendices contain topics that really can be skipped by most everyday users. Exception: Appendix H is an alphabetized command summary that many people find extremely useful (including ourselves).

   This reorganizing is meant to stress certain aspects over others. For

example, the section on graphics inclusion and color was originally treated as an exotic freak, relegated to an appendix on extensions; in the third edition, it moved up to be included in a front chapter along with the `picture` environment and floats; now it dominates Chapter 6 all on its own, the floats come in the following Chapter 7, and `picture` is banished to the later Chapter 13. This is not to say that the `picture` features are no good, but only that they are very specialized. We add descriptions of additional drawing possibilities there too.

2. It is stressed as much as possible that LaTeX is a *markup* language, with separation of content and form. Typographical settings should be placed in the preamble, while the body contains only logical markup. This is in keeping with the modern ideas of XML, where form and content are radically segregated.

3. Throughout this edition, contributed packages are explained at that point in the text where they are most relevant. The `fancyhdr` package comes in the section on page styles, `natbib` where literature citations are explained. This stresses that these 'extensions' are part of the LaTeX system as a whole. However, to remind the user that they must still be explicitly loaded, a marginal note is placed at the start of their descriptions.

4. PDF output is taken for granted throughout the book, in addition to the classical DVI format. This means that the added possibilities of pdfTeX and `dvipdfm` are explained where they are relevant. A separate Chapter 10 on PostScript and PDF is still necessary, and the best interface to PDF output, the `hyperref` package by Sebastian Rahtz, is explained in detail. PDF is also included in Chapter 15 on presentation material.

   On the other hand, the other Web output formats, HTML and XML, are only dealt with briefly in Appendix E, since these are large topics treated in other books, most noticeably the *LaTeX Web Companion*.

5. This book is being distributed with the TeXLive CD, with the kind permission of Sebastian Rahtz who maintains it for the TeX Users Group. It contains a full TeX and LaTeX installation for Windows, Macintosh, and Linux, plus many of the myriad extensions that exist.

We once again express our hope that this *Guide* will prove more than useful to all those who wish to find their way through the intricate world of LaTeX. And with the addition of the TeXLive CD, that world is brought even closer to their doorsteps.

Helmut Kopka and Patrick W. Daly
June, 2003

Contents

Appendices

List of Tables

List of Figures

Part I

Basics

1 Introduction

1.1 Just what is LaTeX?

To summarize very briefly:

- LaTeX is a comprehensive set of markup commands used with the powerful typesetting program TeX for the preparation of a wide variety of documents, from scientific articles, reports, to complex books.
- LaTeX like TeX is an open software system, available free of charge. Its core is maintained by the LaTeX3 Project Group but it also benefits from extensions written by hundreds of user/contributors, with all the advantages and disadvantages of such a democracy.
- A LaTeX document consists of one or more source files containing plain text characters, the actual textual content plus markup commands. These include instructions which can insert graphical material produced by other programs.
- It is processed by the TeX program to produce a binary file in DVI (*device independent*) format, containing precise directions for the typesetting of each character. This in turn can be viewed on a monitor, or converted into printer instructions, or some other electronic form such as PostScript, HTML, XML, or PDF.
- A variant on the TeX program called pdfTeX produces PDF output directly from the source file without going through the DVI intermediary. With this, LaTeX can automatically include internal links and bookmarks with little or no extra effort, plus PDF buttons and external links, in addition to graphics in a wide range of common formats.
- TeX activities are coordinated by the TeX Users Group, TUG (`www.tug.org`) who distribute a set of CDs, called TeXLive, annually to its

members, containing a TEX/LATEX installation for various computer types.

The rest of this book attempts to fill in the gaps in the above summary. With the help of the included TEXLive CD for Windows, Macintosh, and Linux, which also contains a directory specific to this book (`\books\Kopka_and_Daly\`), we hope that the user will have additional pleasure in learning the joys of LATEX.

1.2 Markup Languages

1.2.1 Typographical markup

In the days before computers, an author would prepare a *manuscript* either by hand or by typewriter, which he or she would submit to a publisher. Once accepted for publication (and after several rounds of corrections and modifications, each requiring a rewrite of the paper manuscript), it would be sent to a copy editor, a human being who would decorate the manuscript with *markup*, marginal notes that inform the typesetter (another human being) which fonts and spacings and other typographical features should be used to convert it to the final printed form that one expects of books and articles.

Electronic processing of text today follows a similar procedure, except that the humans have been replaced by computer programs. (So far the author has avoided this fate, but they are working on it.) The markup is normally included directly in the manuscript in such a way that it is converted immediately to its output form and displayed on the computer monitor. This is known as WYSIWYG, or 'what you see is what you get'.

However, what you see is not always what you've got. An alternative that is used more and more by major publishers is *markup languages*, in which the raw text is interspersed with indicators 'for the typesetter.' The result as seen on the monitor is much the same as a typewritten manuscript, except that the markup is no longer abbreviated marginal notes, but cryptic code within the actual text. This *source text*, which can be prepared by a simple, dumb *text editor* program, is converted into typographically set output by a separate program.

For example, to code the line

> He took a **bold step** forward.

with HTML, the classical markup language of the World Wide Web, one enters in the source text:

```
He took a <b>bold step</b> forward.
```

In Plain TEX, the same sentence would be coded as:

```
He took a {\bf bold step} forward.
```

The first example is to be processed (displayed) by a Web browser program that decides to set everything between and as bold face. The second example is intended for the TEX program (Section 1.3). The markup in these two examples follow different rules, different syntax, but the functionality is the same.

1.2.2 Logical markup

The above examples illustrate *typographical markup*, where the inserted commands or tags give direct instructions to alter the appearance of the output, here a change of font. An alternative is to indicate the purpose of the text. For example, HTML recognizes several levels of headings; to place a title into the highest level one enters:

```
<h1>Logical Markup</h1>
```

The equivalent LATEX entry would be:

```
\section{Logical Markup}
```

With this *logical markup*, the author concentrates entirely on the content and leaves the typographical considerations to the experts. One merely marks the structure of the document, and has no means of controlling how the logical elements, like section titles, are to be rendered typographically. This information is put into HTML style sheets or LATEX classes and packages, which are external to the actual source file. This means that the entire layout of a document can be overhauled with only minimal or even no alterations to the source file.

Today much effort is being put into XML, the Extensible Markup Language, as the ultimate markup system, since it allows the markup, or tags, to be defined as needed, without any indication of how they are to be implemented. That is left to XSL, the Extensible Stylesheet Language. It must be emphasized that neither XML nor XSL are programs at all; they are specifications for how documents and databases may be marked up, and how the markup tags may be translated into real output. Programs still need to be found to do the actual job.

And that is the fundamental idea behind markup languages: that the source text indicates the logical structure of its contents. Such source files, being written in plain ASCII text, are extremely robust, not being married to any particular software package or computer type.

What does all this have to do with LATEX? In the next Section we outline the development of TEX and LATEX, and go on to show that LATEX, a product of the mid 1980's, is a programmable markup language that is ideally suited for the modern world of electronic publishing.

1.3 TEX and its offspring

The most powerful formatting program for producing book quality text of scientific and technical works is that of Donald E. Knuth (Knuth, 1986a, 1986b, 1986c, 1986d, 1986e). The program is called TEX, which is a rendering in capitals of the Greek letters $\tau\epsilon\chi$. For this reason the last letter is pronounced not as an *x*, but as the *ch* in Scottish *loch* or German *ach*, or as the Spanish *j* or Russian *kh*. The name is meant to emphasize that the printing of mathematical texts is an integral part of the program and not a cumbersome add-on. In addition to TEX, the same author has developed a further program called METAFONT for the production of character fonts. The standard TEX program package contains 75 fonts in various design sizes, each of which is also available in up to eight magnification steps. All these fonts were produced with the program METAFONT. With additional applications, further character fonts have been created, such as for Cyrillic, Chinese, and Japanese, with which texts in these alphabets can be printed in book quality.

The TEX program is free, and the source code is readily available. Anybody may take it and modify it as they like, provided they call the result something other than TEX. This indeed has occurred, and several TEX variants do exist, including pdfTEX which we deal with later in this Chapter. Only Knuth is allowed to alter TEX itself, which he does only to correct any obvious bugs. Otherwise, he considers TEX to be completed; the current version number is 3.14159, and with his death, the code will be frozen for all time, and the version number will become exactly π.

1.3.1 The TEX program

The basic TEX program only understands a set of very primitive commands that are adequate for the simplest of typesetting operations and programming functions. However, it does allow more complex, higher-level commands to be defined in terms of the primitive ones. In this way, a more user-friendly environment can be constructed out of the low-level building blocks.

During a processing run, the program first reads in a so-called *format file* which contains the definitions of the higher-level commands in terms of the primitive ones, and which also contains the hyphenation patterns for word division. Only then does it read in the author's *source file* containing the actual text to be processed, including formatting commands that are predefined in the format file.

Creating new formats is something that should be left to very knowledgeable programmers. The definitions are written to a source file which is then processed with a special version of the TEX program called initex. It stores the new format file in a compact manner so that it can be read in quickly by the regular TEX program.

Although the normal user will almost never write such a format, he or she may be presented with a new format source file that will need to be installed with initex. For example, this is just what must be done to upgrade LATEX periodically. How to do this is described in Appendix B.

1.3.2 Plain TEX

Knuth has provided a basic format named *Plain TEX* to interact with TEX at its simplest level. This is such a fundamental part of TEX processing that one tends to forget the distinction between the actual processing program TEX and this particular format. Most people who claim to 'work only with TEX' really mean that they only work with Plain TEX.

Plain TEX is also the basis of every other format, something that only reinforces the impression that TEX and Plain TEX are one and the same.

1.3.3 LATEX

The emphasis of Plain TEX is still very much at the typesetter's level, rather than the author's. Furthermore, the exploitation of all its potential demands considerable experience with programming techniques. Its application thus remains the exclusive domain of typographic and programming professionals.

For this reason, the American computer scientist Leslie Lamport has developed the LATEX format (Lamport, 1985), which provides a set of higher-level commands for the production of complex documents. With it, even the user with no knowledge of typesetting or programming is in a position to take extensive advantage of the possibilities offered by TEX, and to be able to produce a variety of text outputs in book quality within a few days, if not hours. This is especially true for the production of complex tables and mathematical formulas.

As pointed out in Section 1.2.2, LATEX is very much more a *logical* markup language than the original Plain TEX, on which it is based. It contains provisions for automatic running heads, sectioning, tables of contents, cross-referencing, equation numbering, citations, floating tables and figures, without the author having to know just how these are to be formatted. The layout information is stored in additional *class files* which are referred to but not included in the input text. The predefined layouts may be accepted as they are, or replaced by others with minimal changes to the source file.

Since its introduction in the mid-1980s, LATEX has been periodically updated and revised, like all software products. For many years the version number was fixed at 2.09 and the revisions were only identified by their dates. The last major update occurred on December 1, 1991, with some minor corrections up to March 25, 1992, at which point LATEX 2.09 became frozen.

1.3.4 LaTeX 2ε

The enormous popularity of LaTeX and its expansion into fields for which it was not originally intended, together with improvements in computer technology, especially dealing with cheap but powerful laser printers, had created a diversity of formats bearing the LaTeX label. In an effort to re-establish a genuine, improved standard, the LaTeX3 Project was set up in 1989 by Leslie Lamport, Frank Mittelbach, Chris Rowley, and Rainer Schöpf. Their goal was to construct an optimized and efficient set of basic commands complemented by various *packages* to add specific functionality as needed.

As the name of the project implies, its aim is to achieve a version 3 for LaTeX. However, since that is the long-term goal, a first step towards it was the release of LaTeX 2ε in mid-1994 together with the publication of the second edition of Lamport's basic manual (Lamport, 1994) and of an additional book (Goossens *et al.*, 1994) describing many of the extension packages available and LaTeX programming in the new system. Since then, two further books have appeared, Goossens *et al.* (1997) dealing with the inclusion of graphics and color, and Goossens and Rahtz (1999) explaining how LaTeX may be used with the World Wide Web. Both these topics are also dealt with in this Guide.

Initially updates to LaTeX 2ε were issued twice a year, in June and December, but it has now become so stable that since 2000 the changes are released only once a year, nominally in June.

LaTeX 2ε is now the standard version, and LaTeX 2.09 is considered obsolete, although source files intended for the older version may still be processed with the newer one. In this book, unless otherwise indicated, 'LaTeX' will always mean LaTeX 2ε.

1.3.5 TeX fonts

TeX initially made use of its own set of fonts, called Computer Modern generated by Knuth's METAFONT program. The reason for doing this was that printers at that time (and even today) may contain their own preloaded fonts, but they are often slightly different from printer to printer. Furthermore, they lacked the mathematical character sets that are essential to TeX's main hallmark, mathematical typesetting. So Knuth created pixel fonts that could be sent to every printer ensuring the same results everywhere.

Today the situation with fonts has changed dramatically. Outline fonts (also known as type 1 fonts) are more compact and versatile than the pixel fonts (type 3). They also have a far superior appearance and are drawn much faster in PDF files. The original Computer Modern fonts have been converted to outline fonts, but there is no reason to stick with them, except possibly for the mathematical symbols. It is LaTeX 2ε with its New

Font Selection Scheme that freed TEX from its rigid marriage to Computer Modern.

Fonts are discussed in more detail in Appendix G.

1.3.6 The LATEX bazaar: user contributions

Like the TEX program on which is relies, LATEX is freeware. There may be a prejudice that what is free in not worth anything, but there are other examples in the computer world to contradict this statement. And since the LATEX macros are provided in files containing plain text, there is no problem to exchange, modify, and supplement them. In other words, the user can participate in extending the basic LATEX system.

Taking advantage of a mechanism in LATEX 2.09 that allowed options to the default layouts to be contained in so-called *style option files*, many users began writing their own 'options' to provide additional features to the basic LATEX. They then made these available to other users via the Internet. Many were intended for very specific problems, but many more proved to be of such general usefulness that they have become part of the standard LATEX installation. In this way, the users themselves have built up a system that meets their needs.

With LATEX 2_ε, these user contributions acquired official status: they became known as *packages*, they could be entered directly into the document and not by the back door, guidelines were issued for writing them, and additional commands were introduced to assist package programming. Package files bear the extension `.sty` from LATEX 2.09 days, so that the older style option files may still function as packages today.

Those packages that have established themselves as indispensable for sophisticated LATEX processing are described in this book in those sections where they are most relevant. This does not imply that other packages are less worthwhile, but simply that this book does have to make a selection. Many other packages are described fully in *The LATEX Companion* (Goossens *et al.*, 1994) and it would go beyond the bounds of this book to reproduce it here.

1.3.7 LATEX and electronic publishing

The most significant development in computer usage in the last decade is the rise of the World Wide Web (or the hijacking of the Internet by the glitzy society). LATEX makes its own contribution here with

- programs to convert LATEX files to HTML (Appendix E);
- means of creating PDF output, with hypertext features such as links, bookmarks, active buttons (Chapter 10);

- interfacing to XML both by acting as an engine to render XML documents and with programs to convert LaTeX to XML and vice versa (Appendix E).

All these forms of electronic publishing are alternatives to traditional paper output. We do not expect paper to disappear entirely so quickly, but it is rapidly being replaced by electronic forms, which can always reproduce the paper whenever needed.

1.4 How to use this book

This *Guide* is meant to be a mixture of textbook and reference manual. It explains all the essential elements of the current standard LaTeX 2_ε, but compared to Lamport (1985, 1994), it goes into more detail, offers more examples and exercises, and describes many 'tricks' based on the authors' experiences. It explains not only the core LaTeX installation, but also many of the contributed packages that have become essential to modern LaTeX processing, and thus quasi-standard. We necessarily have to be selective, for we cannot go to the same extend as *The LaTeX Companion* (Goossens *et al.*, 1994), *The LaTeX Graphics Companion* (Goossens *et al.*, 1997), and *The LaTeX Web Companion* (Goossens and Rahtz, 1999), which are still valid *companions* to this book.

The first part of the book is entitled *The Basics*, and deals with the more fundamental aspects of LaTeX: inputting text and symbols, document organization, lists and tables, entering mathematics, and customizations by the user. The second part is called *Beyond the Basics*, meaning it presents concepts which may be more advanced but which are still essential to producing complex, sophisticated documents. The distinction is rather arbitrary. Finally, the appendices contain topics that are not directly part of LaTeX itself, but useful for understanding its applications: installation, error messages, creating packages, World Wide Web, fonts. Appendix H is an alphabetized summary of most of the commands and their use, cross-referenced to their locations in the main text.

1.4.1 Some conventions

In the description of command syntax, typewriter type is used to indicate those parts that must be entered exactly as given, while *italic* is reserved for those parts that are variable or for the text itself. For example, the command to produce tables is presented as follows:

```
\begin{tabular}{col_form}   lines   \end{tabular}
```

The parts in typewriter type are obligatory, while *col_form* stands for the definition of the column format that must be inserted here. The allowed

values and their combinations are given in the detailed descriptions of the commands. In the above example, *lines* stands for the line entries in the table and are thus part of the text itself.

Package: `sample`

Sections describing a package, an extension to basic LaTeX, have the name of that package printed as a marginal note, as demonstrated here for this paragraph. In this way, you are reminded that you must include it with `\usepackage` (Section 3.1.2) in order to obtain the additional features. Without it, you are likely to get an error message about undefined commands.

Sections of text that are printed in a smaller typeface together with the boxed exclamation mark at the left are meant as an extension to the basic description. They may be skipped over on a first reading. This information presents deeper insight into the workings of LaTeX than is necessary for everyday usage, but which is invaluable for creating more refined control over the output.

1.5 Basics of a LaTeX file

1.5.1 Text and commands

The *source file* for LaTeX processing, or simply the *LaTeX file*, contains the *source text* that is to be processed to produce the printed output. Splitting the text up into lines of equal width, formatting it into *paragraphs*, and breaking it into *pages* with page numbers and running heads are all functions of the processing program and not of the input text itself.

For example, words in the source text are strings of letters terminated by some non-letter, such as *punctuation*, *blanks*, or *end-of-lines* (*hard* end-of-lines, ones that are really there, not the *soft* ones that move with the window width); whereas punctuation marks will be transferred to the output, blanks and end-of-lines merely indicate a gap between words. Multiple blanks in the input, or blanks at the beginning of a line, have no effect on the interword spacing in the output.

Similarly, a new paragraph is indicated in the input text by an empty line; multiple empty lines have the same effect as a single one. In the output, the paragraph may be formatted either by indentation of the first line, or by extra interline spacing, but this is not affected in any way by the number of blank lines or extra spaces in the input.

The source file contains more than just text, however; it is also interspersed with markup commands that control the formatting or indicate the structure. It is therefore necessary for the author to be able to recognize what is text and what is a command. Commands consist either of certain single characters that cannot be used as text characters, or of words preceded immediately by a special character, the backslash (\).

The syntax of source text is explained in detail in Chapter 2.

1.5.2 Contents of a LaTeX source file

Every LaTeX file contains a *preamble* and a *body*.

The preamble is a collection of commands that specify the global processing parameters for the following text, such as the paper format, the height and width of the text, the form of the output page with its pagination and automatic page heads and footlines. As a minimum, the preamble must contain the command `\documentclass` to specify the document's overall processing type. This is the first command in the preamble.

If there are no other commands in the preamble, LaTeX selects standard values for the line width, margins, paragraph spacing, page height and width, and much more. By default, these specifications are tailored to the American norms. For European requirements, built-in options exist to alter the text height and width to the A4 standard. Furthermore, there are language-specific packages to translate certain headings such as 'Chapter' and 'Abstract'.

The preamble ends with `\begin{document}`. Everything that follows this command is interpreted as *body*. It consists of the actual text mixed with markup commands. In contrast to those in the preamble, these commands have only a local effect, meaning they apply only to a part of the text, such as *indentation*, *equations*, temporary change of *font*, and so on. The body ends with the command `\end{document}`. This is normally the end of the file as well.

The general syntax of a LaTeX file is as follows:

```
\documentclass[options]{class}
    Further global commands and specifications
\begin{document}
    Text mixed with additional commands of local effect
\end{document}
```

The possible *options* and *classes* that may appear in the `\documentclass` command are presented in Section 3.1.1.

A minimal LaTeX file named `hi.tex` contains just the following lines:

```
\documentclass{article}
\begin{document}
   Hi!
\end{document}
```

1.5.3 Extending LaTeX with packages

Packages are a very important feature of LaTeX. These are extensions to the basic LaTeX commands that are written to files with names that end in `.sty` and are loaded with the command `\usepackage` in the preamble. Packages can be classified by their origin: