Foreword

Nearly twenty years after the first ideas for LaTeX2$_\varepsilon$ emerged, the use of LaTeX to produce high-quality technical documents shows no sign of waning. Indeed, over the past 5 or so years there has been if anything an *upturn* in interest in using LaTeX. Better editors, faster computers and the range of powerful LaTeX packages have all contributed to this increased uptake.

For the new user, this vibrancy can appear intimidating. The range of packages available for use with LaTeX is vast, and it is not always obvious which is the 'best of breed.' What new users need therefore is a guide not just to the basics of the LaTeX approach, but also help in navigating this ecosystem so that they can produce the documents they need as rapidly as possible.

Creating well-designed documents is about more than the technical detail of any typesetting system, and so as well as learning LaTeX it is also necessary to understand the wider ideas of good writing and good design if one is to create truly 'beautiful' material.

In *LaTeX and Friends*, Marc van Dongen provides an integrated solution to these inter-related requirements. Treating the presentation of beautiful documents as the key aim of the reader, it offers advice on good practice (both in LaTeX terms and beyond) in the relevant context for the beginner. It also avoids the problem seen in many texts, which fall short in supporting the transition from beginner to advanced user. Thus while new LaTeX users will find the information they need here, so will more established users, making this not only a beginners' guide but also a reference manual for day-to-day LaTeX users.

Joseph Wright

Contents

List of Figures

List of Tables

Preface

This book provides students with an introduction to technical writing and computer presentations with LaTeX, which is the de-facto standard in computer science and mathematics. Seasoned LaTeX users may also use this work as a reference.

The reader will learn techniques for writing large and complex documents, preparing computer presentations, and creating complex graphics in an integrated manner. The book's website, which may be found at `http://csweb.ucc.ie/~dongen/LAF/LAF.html`, has three separate chapters explaining how to use a widely used LaTeX distribution on `Windows`, on `Unix`, and on the `Mac`. These chapters also provide an introduction to some selected integrated development environments (ides).

For teaching purposes I have tried to minimise the number of classes and style files for the reader. This is one of the main reasons why I decided to use the `amsmath` package for the presentation of mathematics, and decided to use `tikz`, `pgfplots`, and `beamer` for the creation of diagrams, data plots, and computer presentations. Another advantage of this approach is that it simplifies the process of creating a viewable/printable output file. Everything should work with `pdflatex`, which is a program that turns LaTeX into `pdf`.

Beginners can be challenged by what are known in the LaTeX community as "verbatim" commands and environments, except when it comes to including, well, verbatim program listings. To avoid errors that are difficult to find and are not always so easy to resolve, these verbatim commands do no appear in this volume. By no means should the decision to omit verbatim commands be a limitation; this work was written without verbatim commands. You likely could write a thesis or dissertation without them too.

M. R. C. van Dongen
Cork
2011

Book Outline

This book has seven parts, some of which are more technical than others. The following is a short outline.

The first two parts are called *Basics* and *Basic Typesetting*. These parts introduce the reader to the basic LaTeX commands for typesetting and cross-referencing. They also explain how to create one or several bibliographies and one or several indexes or glossaries.

The next part is *Tables, Diagrams, and Data Plots*, which is about presenting data in tables, diagrams with the `tikz` package, and data plots with the `pgfplots` package.

Mathematics and Algorithms is the next part. It explains how to typeset mathematics, how to typeset algorithms in pseudo-code, and how to present program listings.

This is followed by *Automation*, which explains how to implement user-defined commands, how to implement option parsing, and how to implement conditional branching. Some readers may wish to skip this part because it is more technical than the other parts.

Miscellany is the next part. It is a collection of optional chapters, some of which are of a more technical nature than others. The first, relatively easy, chapter explains how to create computer presentations with the `beamer` package. It continues with two more technical chapters that explain how to implement user-defined classes and packages and how to use OpenType fonts.

The last part is *References and Bibliography*, which is a collection of indexes, a list of acronyms, a bibliography, and a short typographic jargon reference. Readers not familiar with notions such as characters, glyphs, ligatures, serifs, kerning, fonts, typefaces, points, point size and leading, ems, and ens, are invited to start with the jargon reference before reading the rest of the book.

Overall, the chapters are well balanced but the chapters about typesetting mathematics and presenting diagrams with `tikz` are a bit longer and more detailed. This is why it was decided to split the presentation on typesetting mathematics into two separate chapters. The first of these chapters should be sufficient for most readers. The chapter about presenting diagrams with `tikz` was not split because it was felt that most readers who are interested in some of this chapter would also be interested in the rest.

Acknowledgements

This book would not have been possible without the help of many. First of all, I should like to thank Don Knuth for writing TeX and Leslie Lamport for writing LaTeX—without them the landscape of computer-based typesetting would have been dominated by Bill. I should like to thank Eddie Kohler for writing `otftotfm` and for his help. I am grateful to Till Tantau and colleagues for writing the beautiful `tikz` package and the `beamer` class. Both of them are stars in terms of functionality, productivity, and documentation. Thanks to David Farley and Dario Taraborelli for letting me include the pictures in Figures 4.2 and 16.1. Many thanks to Billy Foley and the University College Cork Art Collection for letting me include the pictures at the back of the part titlepages. I should like to thank Charles P. Schaum, Frank Böhme, George Boyle, Tom Carroll, Finbarr Holland, Rik Kabel, Mico Loretan, Ben McKay, Luca Mercriadri, Oliver Nash, Oleg Paraschenko, Jason Quinlan, Lisa Swenson, Uwe Ziegenhagen, and Wim Michels for useful comments on early drafts. I should also like to thank Paul Blaga, Robin Fairbairns, Peter Flynn, Francisco A. F. Reinaldo, and Boris Veytsman for reviewing the book. Special thanks to Joseph Wright who was so kind to proofread the entire book and to write the foreword. His critical eye spotted many known and unknown errors. Many thanks to Mr Engesser, Ms Glaunsinger, and Ms Fisher at Springer for providing the opportunity to publish this book and for helping me bring this project to a successful end. Finally, I should like to thank all those who have worked on LaTeX and friends, all those who have supported LaTeX and friends, and all who have answered all my LaTeX and METAPOST questions over the last two decades or so. The following are but a few: André Heck, Barbara Beeton, Cristian Feuersänger, Dan Luecking, David Carlisle, David Kastrup, Denis Roegel, Donald Arseneau, D. P. Story, Frank Mittelbach, Frank van Raalte, Hans Hagen, Heiko Oberdiek, Jim Hefferon, John Hobby, Jonathan Fine, Jonathan Kew, Karl Berry, Kees van der Laan, Keith Reckdahl, Kjell Magne Fauske, Mark Wibrow, Nelson Beebe, Peter Wilson, Philipp Lehman, Rainer Schöpf, Ross Moore, Scot Pakin, Sebastian Rahtz, Stephan Hugel, Taco Hoekwater, Thomas Esser, Ulrike Fisher, Victor Eijkhout, Vincent Zoonekynd, Will Robertson, and all the many, many others. Without them the TeX community would have been much worse off.

PART I

Basics

Picture omitted to keep size of `.pdf` small.

Untitled Landscape, oil on paper (1993), 64 × 90 cm
Work included courtesy of Billy Foley and University College Cork Art Collection

PART II

Basic Typesetting

Picture omitted to keep size of `.pdf` small.

Oil and charcoal on canvas (31/08/05), 183 × 223 cm
Work included courtesy of Billy Foley

Chapter 2
Running Text

This chapter explains everything you've always wanted to know about writing text, aligning it, and changing text appearance.

Recall from Chapter 1 that LaTeX is implemented on top of TeX, which is a rewriting machine that turns token streams into token streams. Some of the character tokens in the input stream have a special meaning to TeX. This is studied in Section 2.1. The rest of the chapter is about typesetting. We start with some sections about diacritics, ligatures, dashes, emphasis, footnotes and marginal notes, quotes and quotations. If you're not familiar with these notions then don't worry, because they are explained further on. Also you can visit the typography jargon reference on page 307. This chapter ends with sections about changing the size and the type style of the text, the most important text alignment techniques, and language related issues.

2.1 Special Characters

This section studies ten characters that have a special meaning to TeX. When TeX sees these characters as tokens in the input stream, then it usually does not typeset them but, instead, changes state. The remainder of this section briefly explains the purpose of the tokens and how you typeset them as characters in the output.

Table 2.1 depicts the tokens, their meaning, and the command to typeset them. We have already studied the start-of-comment token (`%`) and the backslash (`\`), which starts control sequences. Typesetting a backslash is done with the commands `\textbackslash` and `\backslash`. The latter command is only used when specifying mathematical formulae, which is described in Chapter 8. The parameter reference token is described in Chapter 11. The alignment tab (`&`) is described in Section 2.19.3. This token usually indicates a horizontal alignment position in array-like structures consisting of rows and columns. The math mode switch token (`$`), the subscript token (`_`), and the explained token (`^`) are described in Chapter 8. The three remaining tokens are described in the remainder of this section.

2.1.1 *Tieing Text*

Remember that LaTeX is a large rewriting machine that repeatedly turns token sequences into token sequences. At some stage it turns a

Table 2.1
The characters in the first column have a special meaning to LaTeX. The purpose of the characters is listed in the column 'Purpose.' The last column lists the command that produces the character. The command `\textbackslash` is used when typesetting normal text. The command `\backslash` is used when typesetting mathematics.

Token	Purpose	Command for Plain Character
#	parameter reference	`\#`
$	math mode switch	`\$`
%	start of comment	`\%`
&	alignment tab	`\&`
~	text tie token	`\textasciitilde`
_	math subscript	`\_`
^	math superscript	`\textasciicircum`
{	start of group	`\{`
}	end of group	`\}`
\	start of command	`\textbackslash` or `\backslash`

token sequence into lines. This is where LaTeX (TeX really) determines the line breaks. The tilde token (~) defines an inter-word space that cannot be turned into a line break. As such it may be viewed as an operator that ties words.

The following shows two important applications of the tilde operator: it prevents unpleasant linebreaks in references and citations.

LaTeX Usage

```
... Figure~\ref{fig:list@format}
  depicts the format of a list.
 It is a reproduction of~\cite[Figure~6.3]{Lamport:94}.
```

It is usually not too difficult to decide where to use the tie operator. The following are some concrete examples, which are taken from [Knuth 1990, Chapter 14].

- References to named parts of a document:
  - `Chapter~12`,
  - `Theorem~1.5`,
  - ....

  Knuth [1990] recommends that you use `Lemmas 5 and~6` because having the 5 at the start of a line is not really a problem.
- Between a person's forenames and between multiple surnames:
  - `Donald~E. Knuth`,
  - `Luis~I. Trabb~Pardo`,
  - `Bartel~Leendert van~der~Waarden`,
  - `Charles~XII`,
  - ....
- Between math symbols in apposition with nouns:
  - `dimension~$d$`,
  - `string~$s$ of length~$l$`,
  - ....

  Here the construct `$`⟨math⟩`$` is used to typeset ⟨math⟩ as an in-line mathematical expression.
- Between symbols in series:
  - `1,~2, or~3`.
- When a symbol is a tightly bound object of a preposition:

  - ⋆ `from 0 to~1,`
  - ⋆ `increase $z$ by~1,`
  - ⋆ ....
- ○ When mathematical phrases are rendered in words:
  - ⋆ `equals~$n$,`
  - ⋆ `less than~$\epsilon$,`
  - ⋆ `modulo~$2$,`
  - ⋆ `for large~$n$,`
  - ⋆ ....
- ○ When cases are being enumerated within a paragraph:
  - ⋆ `Show that function $f( x )$ is (1)~continuous; (2)~bounded.`

2.1.2 *Grouping*

Grouping is a common technique in LaTeX. The opening brace (`{`) starts a group and closing brace (`}`) closes it. Grouping has two purposes. The first purpose of grouping is that it turns several things into one compound thing. This may be needed, for example, if you want to pass several words to a command that typesets its argument in bold face text. The following demonstrates the point.

```
A bold \textbf{word} and
 a bold \textbf letter.
```

A bold **word** and a bold **l**etter.

The second purpose of grouping is that it lets you change certain settings and keep the changes local to the group. The following demonstrates how this may be used to make a local change to the type style of the text inside the group.

```
Normal text here.
{% Start a group.
 \bfseries
 % Now we have bold text.
 Bold paragraphs in here.
}% Close the group.
Back to normal text again.
```

Normal text here. **Bold paragraphs in here.** Back to normal text again.

Inside the group you may have several paragraphs. The advantage of the declaration `\bfseries` is that it defines how the text is typeset until the end of the group. The `\textbf` command just typesets its argument in a bold typeface. The argument may not contain paragraph-breaks.

There is also a low-level TeX mechanism for creating groups. It works just as the braces. A group is started with `\begingroup` and ended with `\endgroup`. These tokens may be freely mixed with braces but {/} pairs and `\begingroup`/`\endgroup` pairs should be properly matched. So `{ \begingroup \endgroup }` is allowed but `{ \begingroup } \endgroup` is not. A brace pair affects whitespace when you're typesetting mathematics but a `\begingroup`/`\endgroup` pair does not.

Table 2.2
Common diacritics

Output	Command	Name
ò	\‘{o}	Acute accent
ó	\’{o}	Grave accent
ô	\^{o}	Circumflex (hat)
õ	\~{o}	Tilde (squiggle)
ö	\"{o}	Umlaut or dieresis
ċ	\.{c}	Dot accent
š	\v{s}	Háček (caron or check)
ŏ	\u{o}	Breve accent
ō	\={o}	Macron (bar)
ő	\H{o}	Long Hungarian umlaut
o͡o	\t{oo}	Tie-after accent
ş	\c{s}	Cedilla accent
ọ	\d{o}	Dot-under accent
o̲	\b{o}	Bar-under accent

Table 2.3
Other special characters

Output	Command	Name
å	\aa	Scandinavian a-with-circle
Å	\AA	Scandinavian A-with-circle
ł	\l	Polish suppressed-l
Ł	\L	Polish suppressed-L
ø	\o	Scandinavian o-with-slash
Ø	\O	Scandinavian O-with-slash
¿	?‘	Open question mark
¡	!‘	Open exclamation mark

2.2 Diacritics

This section studies how to typeset characters with *diacritics,* which are also known as accents. Table 2.2 displays some commonly occurring diacritics and the commands that typeset them. The presentation is based on [Knuth 1990, Chapter 9].

Using `\"{i}` to typeset ï may not work if you're not using a Type 1 font (`T1` font). However, typesetting ï with `\"{\i}` should always work. Here the command `\i` is used to typeset a dotless i (ı). There is also a command `\j` for a dotless ȷ.

Table 2.3 shows some other commonly occurring special characters.

2.3 Ligatures

A ligature combines two or several characters as a special glyph. Examples of English ligatures and their equivalent character combinations are ﬁ (fi), ﬀ (ff), ﬃ (ffi), ﬂ (fl), and and ﬄ (ffl). LaTeX recognises English ligatures and substitutes them for the characters representing them.

Table 2.4 displays some foreign ligatures. The symbol ß (eszett) is

Table 2.4
Foreign ligatures

Output	Command	Name
œ	`\oe`	French ligature œ
Œ	`\OE`	French ligature Œ
æ	`\ae`	Scandinavian ligature æ
Æ	`\AE`	Scandinavian ligature Æ
ß	`\ss`	German ‘Eszett’ or sharp S

```
‘Convention’ dictates that
 punctuation go inside
 quotes, like ‘‘this,’’ but
 some think it’s better
 to do ‘‘this’’.
```

‘Convention’ dictates that punctuation go inside quotes, like “this,” but some think it’s better to do “this”.

Figure 2.1
Quotes

```
‘‘\,‘Fi’ or ‘fum?’\,’’ he asked.\\
‘‘‘Fi’ or ‘fum?’’’ he asked.    \\
‘‘{}‘Fi’ or ‘fum?’{}’’ he asked.
```

“ ‘Fi’ or ‘fum?’ ” he asked.
“‘Fi’ or ‘fum?”’ he asked.
“‘Fi’ or ‘fum?’” he asked.

Figure 2.2
Nested quotations

a ligature of ſs [Bringhurst 2008] and this is reflected in the LaTeX command that typesets the symbol.

Sometimes it is better to suppress ligatures. The following is an example: the `\makebox` command prevents LaTeX from turning the fi in selfish into a ligature, which makes the result much easier to parse: selfish, not selfish.

```
Mr~Crabs is a self\makebox{}ish shellfish.
```

LaTeX Usage

Other words that need “anti-hyphenation” pre-processing are halflife, halfline, selfless, offline, offloaded, and so on.

2.4 Quotation Marks

This section explains how you typeset quotation marks. Figure 2.1 is an example from [Lamport 1994, page 13]. The word ‘Convention’ in this example is in single quotes and the word ‘this’ is in double quotes. The quotes at the start are backquotes (‘ and ‘‘). The quotes at the end are the usual quotes (’ and ’’). Notice that output quote between ‘it’ and ‘s’ is produced using a single quote in LaTeX.

To get properly nested quotations you insert a thin space where the quotes “meet.” Recall that the thin space command (`\,`) typesets a thin space. Figure 2.2 provides a concrete example that is taken from [Lamport 1994, page 14]. Figure 2.2 provides another example. The first line of this example looks much better than the other two. Note that LaTeX parses three consecutive quotes as a pair of quotes followed by one more quote. This is demonstrated by the second line of the output, which looks terrible. The last line of the input avoids the three consecutive quotes by adding an empty group that makes explicit where the double quotes and the single quote meet. Still the resulting output doesn't look great.

Intermezzo. As a general rule, British usage prefers the use of single quotes for ordinary use. This poses a problem if an apostrophe is used for the possessive form: He said 'It is John's book.' This is why it is also acceptable to use double quotes [Trask 1997, Chapter 8].

2.5 Dashes

There are three kinds of dashes: -, –, and —. In LaTeX you get them by typing `-`, `--`, and `---`. The second symbol can also be typeset with the command `\textendash` and the last symbol with the command `\textemdash`. The symbol −, which is used in mathematical expressions such as $a - b$, is not a dash. This symbol is discussed in Chapter 8. The following briefly explains how the dashes are used.

- This is the intra-word dash, which is used to hyphenate compound modifiers such as one-to-one, light-green, and so on [Trask 1997, Chapter 6]. In LaTeX you typeset this symbol as follows: `-`.

– This is the en-dash, which has the width of 1 en. An en is equivalent to half the *current* type size, so an en-dash is shorter in normal text than it is in large text. The en-dash is mainly used in ranges: pages 12 – 15 (from 12 to 15). However, the en-dash is also used to link two names that are sharing something in common: a joint Anglo – French venture [Allen 2001, page 45]. The LaTeX command `\textendash` and the sequence `--` typeset the en-dash. When you typeset an en-dash, it looks better if you add a little space before and after. Remember that `\,` produces a thin space. Use this command for the horizontal space.

LaTeX Usage
```
... pages~12\,--\,15 (from~12 to~15).
```

— This is the em-dash, which has the same width as an em. An em is equal to the *current* type size. The em-dash separates strong interruptions from the rest of the sentence—like this [Trask 1997, Chapter 6]. Bringhurst [2008, page 80] prefers the en-dash to the em-dash. The LaTeX command `\textemdash` and the sequence `---` typeset the em-dash. An em-dash at the start of a line doesn't look very good so you should tie each em-dash to the preceding word.

LaTeX Usage
```
... the rest of the sentence~\textemdash
like this~\parencite[Chapter~6]{Trask:1997}.
```

Figure 2.3 presents an example of the dashes. A few years ago I noticed that sometimes `---` doesn't work with XeTeX (even with `Mapping = tex-text` enabled). However, `\textemdash` always worked.

2.6 Full Stops

LaTeX usually treats a full stop (.) as an end-of-sentence indicator. By

```
The intra-word dash is used to hyphenate
 compound modifiers such as light-green,
 X-ray, or one-to-one. ...
The en-dash is used in ranges: pages~12--15.
The em-dash is used to separate strong
 interruptions from the rest of the
 sentence~--- like this%
 ~\cite[Chapter~6]{Trask:1997}. ...
```

Figure 2.3
Dashes

default, LaTeX inserts a bit more space after the full stop at the end of a sentence than it does between words. It also does this after other punctuation symbols. The `\frenchspacing` command turns this feature off. The command `\nonfrenchspacing` turns the feature on again. When a full stop is not the end of a sentence you need to help LaTeX a bit by inserting the space command (`\␣`) after the full stop.

LaTeX Usage

```
Meet me at 6~p.m.\␣at the Grand Parade.
```

However, when an uppercase letter is followed by a full stop, then LaTeX assumes the full stop is for abbreviation. For example:

LaTeX Usage

```
Donald~E. Knuth developed the {\TeX} system.
```

This convention causes a problem if an uppercase letter really is the end of a sentence. Insert a `\@` before the full stop if this happens.

LaTeX Usage

```
In Frank Herbert’s \emph{Dune} saga,
 the Mother School of the Bene Gesserit
 is situated on the planet Wallach IX\@.
```

LaTeX inherits its habit of putting some extra space after full stops and other punctuation symbols from TeX. Bringhurst [2008, pages 28–30] points out that there really is no reason to add such extra space for modern works. Following Bringhurst’s advice, this document was typeset with `\frenchspacing` enabled.

2.7 Ellipsis

The command `\ldots` produces an ellipsis (…), which is used to indicate an omission. If the ellipsis occurs at the end of a sentence, then you still need to add an end-of-sentence marking full stop. If this happens then Felici [2012, Figure 13.15] recommends that you put the ellipsis close to the preceding text and then add the full stop.

```
Many stories start with
 ‘Once upon a time\ldots.’
They usually end with
 ‘\ldots\␣and they all lived
  happily ever after.’
```

Many stories start with ‘Once upon a time….’ They usually end with ‘… and they all lived happily ever after.’

Figure 2.4
Good borderline punctuation

```
Robert Bringhurst, author of
 \emph{Elements of
       Typographic Style,}
 recommends setting such
 punctuation symbols in
 the brighter type.
\textbf{Do as he}, or
 risk getting ugly type.
```

Robert Bringhurst, author of *Elements of Typographic Style,* recommends setting such punctuation symbols in the brighter type. **Do as he**, or risk getting ugly type.

Figure 2.5
Poor borderline punctuation

```
Robert Bringhurst, author of
 \emph{Elements of
       Typographic Style},
 recommends setting such
 punctuation symbols in
 the brighter type.
\textbf{Do as he,} or
 risk getting ugly type.
```

Robert Bringhurst, author of *Elements of Typographic Style*, recommends setting such punctuation symbols in the brighter type. **Do as he,** or risk getting ugly type.

2.8 Emphasis

Emphasis is a typographic tool for typesetting text in a different typeface. The idea is that this makes the text stand out. Emphasis is especially useful when introducing a new concept, such as in this paragraph.

In some documents, emphasis is implemented by typesetting text in a bold face typeface, by typesetting it in uppercase typeface, or (worse) by underlining the text. LaTeX emphasises text in paragraphs by italicising the text. Trask [1997, page 82] calls this the preferred style for emphasis. The LaTeX command for emphasis is `\emph`.

```
Emphasised \emph{example.}
```

Emphasised *example.*

2.9 Borderline Punctuation

Bold text looks darker than normal, upright text and italicised text look brighter than normal, upright text. When small punctuation symbols get caught between darker and brighter type it is time to pay attention. Robert Bringhurst, author of *Elements of Typographic Style,* recommends setting such punctuation symbols in the brighter type [Bringhurst 2008]. **Do as he**, or risk getting ugly type. Figures 2.4 and 2.5 demonstrate what you get if you follow Bringhurst's advice and what if you don't. The figures do not excel in terms of maintainability because they hardcode the author's name and the title of the work.

2.10 Footnotes and Marginal Notes

It is generally accepted that using footnotes and marginal notes should be used sparingly because they are disruptive. However, proper use of

```
Footnotes\footnote{A footnote is a note
 of reference, explanation, or comment that is
 usually placed below the text on a printed page.}
 can be a nuisance. This is especially true if
 there are many.\footnote{Like here.} The more you see
 them, the more annoying they get.\footnote{Got it?}
```

Footnotes[a] can be a nuisance. This is especially true if there are many.[b] The more you see them, the more annoying they get.[c]

[a] A footnote is a note of reference, explanation, or comment that is usually placed below the text on a printed page.

[b] Like here.

[c] Got it?

Figure 2.6
Using footnotes

marginal notes in documents with wide margins can be very effective.

Not surprisingly, LaTeX provides a command for footnotes and a command for marginal notes. Figure 2.6 demonstrates how to specify footnotes in LaTeX. A *marginal note* or *marginal paragraph* is like a footnote, but placed in the margin as on this page. The command `\marginpar{⟨text⟩}` puts ⟨`text`⟩ in the margin as a marginal note. By passing an optional argument to the command you can put different text on odd (recto/front/right) pages and on even (verso/back/left) pages. The optional argument is used for even pages and the required argument is used for odd pages. If you're using both the optional and required argument then it is easy to remember which is which: the optional argument is to the left of the required argument so it's for the left page; the required argument is for the right page. Note that narrow marginal notes may look better with ragged text, which is text that is aligned to one side only. On the right (left) pages you use ragged right (left) text. Section 2.19.2 explains how to typeset ragged text.

Avoid marginal notes in very narrow margins.

2.11 Displayed Quotations and Verses

The `quote` and `quotation` environments are for typesetting displayed quotations. The former is for short quotations; the latter is for longer quotations. Figure 2.7 shows how you use the `quote` environment. The command `\\` in Figure 2.7 forces a line break.

The `verse` environment typesets poetry and verse. Figure 2.8 shows how you use the environment. In this example, the command `\qquad` inserts two quads. Here a quad is an amount of space that is equivalent to the *current* type size. So if you use a 12 pt typeface then a quad results in a 12 pt space in normal text. The command `\\` inside the `verse` environment determines the line breaks. Remember that the command `\,` before the letter *S* inserts a thin space.

2.12 Line Breaks

In the previous section, the command `\\` inserted a line break in displayed quotations and verses. The command also works inside

Figure 2.7
The quote environment

```
Blah blah blah blah blah blah blah blah blah blah blah.
\begin{quote}
   Next to the originator of a good sentence
    is the first quoter of it. \\
   \emph{Ralph Waldo Emerson}
\end{quote}
Blah blah blah blah blah blah blah blah blah blah blah.
```

Blah blah blah blah blah blah blah blah blah blah blah.

> Next to the originator of a good sentence is the first quoter of it.
> *Ralph Waldo Emerson*

Blah blah blah blah blah blah blah blah blah blah blah.

Figure 2.8
The verse environment

```
The following anti-limerick is
 attributed to W.\,S. Gilbert.
\begin{verse}
  There was an old man of St.~Bees,            \\
  Who was stung in the arm by a wasp;          \\
    \qquad When they asked, ''Does it hurt?''  \\
    \qquad He replied, ''No, it does n't,      \\
  But I thought all the while 't was a Hornet.''
\end{verse}
```

The following anti-limerick is attributed to W. S. Gilbert.

> There was an old man of St. Bees,
> Who was stung in the arm by a wasp;
> When they asked, "Does it hurt?"
> He replied, "No, it does n't,
> But I thought all the while 't was a Hornet."

paragraphs. An optional argument determines the extra vertical space of the line break: `\\[⟨extra vertical space⟩]`. A line break at the end of a page may trigger a page break. If you don't want a page break then you should use the command `\\*`. It is identical to `\\` but it inhibits page breaks.

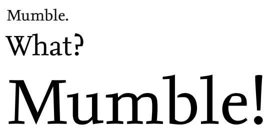
2.13 Controlling the Size

With the proper class and packages there is usually no need to change the type size of your text. However, sometimes it has its merits, e.g., when you're designing your own titlepage or environment. Table 2.5 lists the declarations and environment that change the type size. The preferred "size" for long-ish algorithms and program listings is `\scriptsize`. If you're using a package to typeset listings then the package usually chooses the right size. If not, it probably lets you specify the type size. Figure 2.9 shows how you change the size of text.

Table 2.5
Size-affecting declarations and environments

Declaration	Environment	Example
`\tiny`	`tiny`	Example
`\scriptsize`	`scriptsize`	Example
`\footnotesize`	`footnotesize`	Example
`\small`	`small`	Example
`\normalsize`	`normalsize`	Example
`\large`	`large`	Example
`\Large`	`Large`	Example
`\LARGE`	`LARGE`	Example
`\huge`	`huge`	Example
`\Huge`	`Huge`	Example

```
{\tiny Mumble.       \\
 \begin{normalsize}
       What?
 \end{normalsize}    \\
 \begin{Huge}
       Mumble!
 \end{Huge} }
```



Mumble.
What?
Mumble!

Figure 2.9
Controlling the size

2.14 Seriffed and Sans Serif Typefaces

LaTeX has several commands that change the type style. Before studying these commands it is useful to study the difference between seriffed and sans serif typefaces and when to use them.

A *serif* is a little decoration at the end of some of the strokes of some of the letters. In a *seriffed* typeface the letters have serifs. Seriffed typefaces are sometimes called *roman* typefaces but in LaTeX roman means upright. In a *sans serif* typeface the letters lack serifs.

Most books use a seriffed typeface for the running text [Unger 2007, pp. 167–168] and the most popular typeface for the running text of books and reports is *(Monotype/Linotype) Times Roman* [Felici 2012], a seriffed typeface. Seriffed typefaces are also used for the running text of most papers, theses, and dissertations in science. Turabian [2007, pp. 374–375] recommends that you use a typeface that is designed for text and that you use a size in the range of 10–12 pt, with 12 pt being the preferred size. Admittedly, the being designed for text is a bit vague but Turabian [2007] give two examples, both of which are seriffed.

As lines get longer and longer, seriffed typefaces are easier to read and make fast reading easier [Unger 2007]. Sans serif typefaces may look better on the screen but the ultimate criterion for printed matter is how the text looks in print, so never choose the typeface for your printed text based on how it looks on the screen.

If a typeface family has a seriffed and sans serif typeface of the same type size (point size), then the seriffed typeface usually requires more horizontal space [Unger 2007]. Stated differently, sans serif typefaces are usually more efficient when it comes to saving space. This may be exploited by using sans serif typefaces in captions, in brochures, in short narrow columns, or on road signs [Unger 2007].

If you don't change the typeface then LaTeX will typeset the body of your document in *Computer Modern*. An example of *Computer Modern* may be found in Table 2.6, further on in this chapter.

2.15 Small Caps Letters

Small caps letters are used to typeset acronyms and abbreviations. Their shape is the same as uppercase letter but their height is smaller, which lets them blend in better with the rest of the text. For example, compare NO SHOUTING with no shouting. The latter is easier on the eye.

Adding extra space uniformly to the left and right of characters in a passage of text is called tracking or letterspacing. The extra space that is added per letter is called the tracking space. Tracking passages of small caps text is a common technique to improve the legibility. For example non-spaced small caps is not spaced, whereas spaced small caps is letterspaced.

The command `\textsc` typesets lowercase letters in small caps. The easiest way to automatically letterspace such text is to use the `microtype` package with the option `tracking=smallcaps`. After this all small caps text will be letterspaced.

```
\textsc{No shouting}.
```

No shouting.

The `microtype` package also provides character protrusion (margin kerning) and font expansion. Character protrusion adjusts the characters at the margins of the text. Font expansion uses narrow or wider font versions so as to make the overall appearance of the text more uniform, avoiding long cramped, dark lines with many characters and long loose, bright lines with few characters. As a side-effect, font expansion may also be used to choose better hyphenation points [Schlicht 2010]. This document was typeset using the `microtype` package with the following options.

LaTeX Usage

```
\usepackage[final,tracking=smallcaps,
            expansion=alltext,protrusion=true]{microtype}
```

Bringhurst [2008, page 30] recommends that you add 5–10% of the type size (point size) for the tracking space. The `microtype` package expects the extra tracking in thousands of the type size. The following sets the tracking space to 5% for the `sc` (small caps) shape.

LaTeX Usage

```
\SetTracking{encoding=*,shape=sc}{50}
```

Most `microtype` users agree that the package improves the appearance of their documents.

Declaration	Command	Example
`\mdseries`	`\textmd`	Medium Series
`\normalfont`	`\textnormal`	Normal Style
`\rmfamily`	`\textrm`	Roman family
`\upshape`	`\textup`	Upright Shape
`\itshape`	`\textit`	*Italic Shape*
`\slshape`	`\textsl`	*Slanted Shape*
`\bfseries`	`\textbf`	**Boldface Series**
`\scshape`	`\textsc`	Small Caps Shape
`\sffamily`	`\textsf`	Sans Serif Family
`\ttfamily`	`\texttt`	`Typewriter Family`

Table 2.6
Type style affecting declarations and commands. The last column shows the result in *Computer Modern* (LaTeX's default typeface). The first four lines usually correspond to the default style. The first nine typefaces are proportional. They may have glyphs with different widths, e.g., compare *M* and *i*. Small caps letters are useful for abbreviations. The last typeface is non-proportional, which is useful in program listings.

2.16 Controlling the Type Style

Changing the type size is hardly ever needed in an article, thesis, report, or book. Changing the type style is required much more, but usually this is done automatically by the commands that typeset the title of your document, the section titles, the captions, and so on.

There are ten LaTeX type style affecting declarations. Each declaration has a command that takes an argument and applies the type style of the declaration to the argument. The arguments cannot have paragraph breaks. The declarations and commands are listed in Table 2.6.

Intermezzo. If you really must change the type style of your text then it is probably for a specific purpose. For example, to change the type style of a newly defined word, to change the type style of an identifier in an algorithm, and so on. Rather than hard-coding the style in your input, it is better if you define a user-defined command that typesets your text in the required style and use the command to typeset your text. The command's name should reflect its purpose. For example `\identifier` to typeset an identifier in an algorithm, `\package` to typeset the name of a LaTeX package, and so on. Using this approach improves maintainability. For example, if you want to change the type style of all identifiers in your text then you only need to make changes in the definition of the command that typesets identifiers. Defining your own commands is discussed in Chapter 11.

2.17 Abbreviations

This section is about abbreviations. It provides some guidelines about their spelling and how to typeset them in LaTeX.

2.17.1 *Initialisms*

Abbreviations that are made up of the initial letters of the abbreviated words are called *initialisms*. Non-standard initialism are usually written with a full stop after each part in the abbreviation: Ph.D. (Philosophiae

Figure 2.10 Finer points of typesetting abbreviations

```
In~2010 Prof.~Donald Knuth was invited to the
 annual TeX User Group Conference in San Francisco, Ca.\␣
 to speak about a revolutionary successor to \TeX.
This remarkable system is entirely menu driven and
 incorporates facilities for social networking.
Pronouncing the name involves making the sound of a bell.
```

Doctor), D.Phil. (Doctor of Philosophy), M.Sc. (Master of Science), and so on. However, if the initialisms are standard, then you omit the full stops, so B.B.C. becomes BBC, 4 G.L. (fourth-Generation Language) becomes 4GL, and Ph.D. becomes Ph D (in LaTeX `Ph~D`). Bringhurst [2008, page 48] recommends typesetting abbreviations with more than two uppercase letters in spaced small capitals: SPACED SMALL CAPS. Section 2.15 explains how to get spaced small caps.

Some authors recommend that you letterspace Uniform Resource Locators (URLS), phone numbers, and email addresses because they are not words. See for example [Bringhurst 2008] or [Hedrick 2003].

Abbreviations of personal names such as D. E. K., J. F. K., J. S. B., and the like should not be letterspaced.

2.17.2 *Acronyms*

An *acronym* is an initialism that is pronounced as a word. For example, radar (RAdio Detection And Ranging), sonar (SOund Navigation And Ranging), NASA (National Aeronautics and Space Administration), and EBCDIC (Extended Binary Coded Decimal Interchange Code); but not ACM (Association for Computing Machinery), BBC (British Broadcasting Corporation), and RSVP (Répondez S'il Vous Plaît). Note that not all acronyms are spelt with uppercase letters; if you're not certain, look up the spelling. Since acronyms are just a special form of initialisms, we should follow Bringhurst's advice, and write them with small caps if they are spelt with (two or more) uppercase letters.

2.17.3 *Shortenings*

A word that is abbreviated by taking the first few letters of that word is called a *shortening*. To avoid ambiguity, shortenings are usually written with a full stop at the end of each part. For example, p. (page), proc. (proceedings), sym. (symposium), fig. (figure), Feb. (February), Prof. (Professor), and so on. The abbreviation pp. is for pages.

Remember that LaTeX inserts a little extra white space after a full stop if `\frencspacing` isn't enabled. If an abbreviation is not at the end of a sentence and ends with a full stop then this extra space may look bad. To suppress the extra white space you have to hardcode a space command (`\␣`) after the abbreviation or tie the abbreviation and the following word. Figure 2.10 provides a small example.

2.17.4 *Introducing Abbreviations*

The first time you introduce an abbreviation you should explain it. Most authors first spell out the abbreviation and then provide the abbreviation in parenthesis. The `acronym` package provides some support for defining and referencing abbreviations in a consistent style. This is done using the standard label-referencing technique. The package provides commands for singular and plural versions of abbreviations and for abbreviated and unabbreviated versions.

Page 4 of this book introduces an acronym for integrated development environments. This text was generated by the following input.

LaTeX Input

```
... many \acp{IDE} ...
```

The command `\acp` in this example is provided by the `acronym` package. The command introduces the plural version of an abbreviation. The `acronym` package also provides the `\ac` command, which introduces the singular version of an abbreviation. The argument `IDE` of the `\acp` command is the label of the acronym. Some other part of the input associates the label `IDE` with the abbreviated version 'IDE' and the expanded version 'Integrated Development Environment.' This was (essentially) done as follows:

LaTeX Input

```
\acro{IDE}[\textsc{ide}]%
     {Integrated Development Environment}
```

When this book was generated and the command `\acp` was used in the second last input, this was the first time the label `IDE` was referenced, which is why it resulted in the following output.

LaTeX Output

… many Integrated Development Environments (IDEs) …

The label `IDE` is also referenced in other locations in the input, but when that happens it always results in the abbreviated version of the acronym: IDE. More information about the `acronym` package may be found in the package documentation [Oetiker 2010].

2.17.5 *British and American Spelling*

There are differences between American and British usage in time abbreviations. According to Trask [1997] Americans write 10:05 AM (Ante Meridiem) for five past ten in the morning and 13:15 PM (Post Meridiem) for a quarter past one in the afternoon. British spelling prefers 10.05 a.m. and 13.15 p.m. [Trask 1997]. Felici [2012] notices that Americans have also started using the British form.

For titles such as Mister, Doctor, and so on, British and American usage differ. Britsh usage is the same as for shortenings. For example, Mr Happy, Dr Who, and Fr Dougal McGuire. Americans add the full stop: Mr. Ed, Dr. Quinn, Medicine Woman, and Fr. Bob Maguire.

Table 2.7
Latin abbreviations. The first column lists the abbreviations, the second the original Latin meaning, and the last the English translation. Note that the abbreviations at the bottom of the table are slanted. This is intensional and preferred usage.

Abbreviation	Latin meaning	English meaning
e.g.	exempli gratia	for example
i.e.	id est	that is/in other words
etc.	et cetera	and so forth
viz.	videlicet	that is to say/namely
cf.	confer	compare
et al.	et alii/aliae/alia	and others

2.17.6 *Latin Abbreviations*

This section studies some Latin abbreviations that are commonly used in scientific writing. Table 2.7 presents the more commonly occurring abbreviations, their Latin meaning, and the English translation.

Note that some abbreviations are typeset in italics. This is not by accident: this is how they should be typeset—but conventions may differ from field to field. Also note that the *al* in *et al.* gets a full stop because it is an abbreviation of *aluis* but that the *et* does not get a full stop because it is already spelt out in full. Remember Bringhurst's advice and put the full stop inside the argument of `\emph`: `\emph{et al.}` Finally note that etc. is short for et cetera, not for ectcetra.

Trask [1997] discourages these abbreviations. Trask continues by pointing out that writing statements like the following are wrong because the reader should be invited to consult the reference.

> The Australian language Dyirbal has a remarkable gender system, *cf.* [Dixon 1972].

Trask proposes the following solution.

> The Australian language Dyirbal has a remarkable gender system; see [Dixon 1972].

Abbreviations such as etc., i.e., and e.g. require additional punctuation [Strunk, and White 2000]:

- Abbreviations such as BBC, NBC, etc., are called initialisms.
- Shortenings, i.e., abbreviations that are formed by taking the first letters of the abbreviated word, usually end with a full stop.
- Abbreviations are not always spelt te same, e.g., Ph.D. and Ph D.

2.17.7 *Units*

The Système International d'Unités/International System of Units (SI) provides rules for consistent typesetting of quantities of units. Heldoorn [2007] provides a summary of these rules. The following is a summary of the main rules.

- The base unit symbols are printed in upright roman: g (gram), m (metre), t (tonne), ... Exceptions are unit symbols that are spelt in Greek and the symbols for inch, degrees, seconds, and so on.

```
Fill in the missing word.\\
Fill in the missing
 \phantom{word}.
```

Fill in the missing word.
Fill in the missing   .

Figure 2.11
The `\phantom` command

- ○ The first letter of the unit symbol is uppercase if it is derived from a proper name: Å (Ångström), N (Newton), Pa (Pascal), ...
- ○ The plural form of the base unit symbol is the same as the singular.
- ○ The base unit symbols do not receive an end-of-abbreviation full stop.

Needless to say, it is important that you typeset quantities of units correctly and consistently. The hard way is doing it by hand. The easy way is doing it with LaTeX.

At the moment of writing the most popular package for specifying SI units is the `siunitx` package [Wright 2011].

- ○ It provides support to configure how the SI units are typeset. For example, kg m s^{-1}, versus kg m s$^{\text{-1}}$, versus kg m/s, and so on.
- ○ It provides commands to typeset quantities of units: `\SI[mode=text]{1.23}{\kilogram}` will give you 1.23 kg and `\SI{1.01}{\kilogram}` will typeset 1.01 kg in the default typesetting mode.
- ○ The package provides macros to typeset lists of quantities in a given unit. For example `\SIlist{0.1;0.2;1.0}{\milli\metre}` gives you 0.1, 0.2 and 1.0 mm if the default typesetting mode is `text`. If you add the option `list-final-separator={, and~}` then you get 0.1, 0.2, and 1.0 mm.
- ○ By default, unit symbols are typeset using the default math roman font but you can also use different fonts.

Discussing the entire `siunitx` package is beyond the scope of this book. The interested reader is referred to the package documentation [Wright 2011] for further information.

2.18 Phantom Text

Some commands don't typeset anything with ink but do affect the horizontal and vertical spacing. The following is the first of three useful versions.

`\phantom{⟨stuff⟩}`

This command "typesets" its argument using invisible ink. The dimensions of the box are the same as the dimensions required for typesetting ⟨`stuff`⟩. ☑

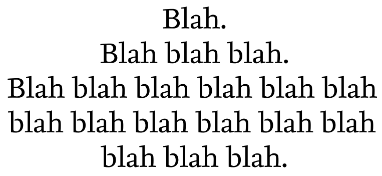
Figure 2.11 demonstrates how you use the command. The `\hphantom` and `\vphantom` commands are horizontal and vertical versions of the `\phantom` command. The following explains how they work.

`\hphantom{⟨stuff⟩}`

This is the horizontal version of the `\phantom` command. The command creates a box with zero height and the same width as its argument, ⟨`stuff`⟩. ☑

Figure 2.12
The center environment

```
\begin{center}
  Blah.\\
  Blah blah blah.

  Blah blah blah blah blah
  blah blah blah blah blah
  blah blah blah blah blah.
\end{center}
```



Blah.
Blah blah blah.
Blah blah blah blah blah blah
blah blah blah blah blah blah
blah blah blah.

\vphantom{⟨stuff⟩}

This is the vertical version of the `\phantom` command. The command creates a box with zero width and the same height as its argument, ⟨`stuff`⟩. It is especially useful for getting the right size for delimiters such as parentheses in mathematical formulae that span multiple lines. This is explained in more detail in Section 8.8.1. ☑

2.19 Alignment

This section studies three commands and two environments that change the text alignment. The first command centres text. The second and third command align text to the left and to the right. The first of the environments is the `tabular` environment, which typesets row-based content with horizontal alignment positions (columns). The last environment is the `tabbing` environment. This environment lets you define horizontal alignment (tab) positions and lets you position text relative to these alignment positions.

2.19.1 *Centred Text*

The `center` environment centres text. The example in Figure 2.12 demonstrates the environment. The example is inspired by Iggy Pop.

2.19.2 *Flushed/Ragged Text*

The `flushleft` environment and the `\raggedright` declaration typeset text that is aligned to the left. Likewise, the `flushright` environment and `\raggedleft` declaration typeset text that is aligned to the right. The example in Figure 2.13 shows the effect of the `flushleft` environment.

2.19.3 *Basic `tabular` Constructs*

The `tabular` environment typesets text with rows and alignment positions for columns. The environment also has siblings called `tabular*` and `array`. The `tabular*` environment works similar to `tabular` but it takes an additional argument that determines the width of the resulting construct. This environment is explained in Section 2.19.5. The

```
\begin{flushleft}
  Blah.\\
  Blah blah blah.

  Blah blah blah blah blah
  blah blah blah blah blah
  blah blah blah blah blah.
\end{flushleft}
```

Blah.
Blah blah blah.
Blah blah blah blah blah blah
blah blah blah blah blah blah
blah blah blah.

Figure 2.13
The `flushleft` environment

`array` environment can only be used in math mode. The `tabular` and `tabular*` environments can be used in both text and math mode.

The remainder of this section introduces the `tabular` environment. This introduction should more than likely suffice for day-to-day usage. A more detailed presentation is provided in Section 2.19.5.

In its simplest form the `tabular` environment is used as follows.

LaTeX Usage

```
\begin{tabular}[⟨global alignment⟩]
               {⟨column alignment⟩}
    ⟨text⟩ & ⟨text⟩ & … & ⟨text⟩ \\
    …
    ⟨text⟩ & ⟨text⟩ & … & ⟨text⟩ \\
    ⟨text⟩ & ⟨text⟩ & … & ⟨text⟩
\end{tabular}
```

The body of the environment contains a sequence of rows that are delimited by linebreaks (`\\`). Each row is a sequence of alignments tab-delimited ⟨`text`⟩. The *i*-th ⟨`text`⟩ in a row corresponds to the *i*-th column. The following explains the arguments of the environment:

⟨`global alignment`⟩
This optional argument determines the vertical alignment of the environment. Allowed values are `t` (align on the top row), `c` (align on the centre), or `b` (align on the bottom row). The default value of this argument is `c`. ☑

⟨`column alignment`⟩
This argument determines the column alignment and additional decorations. For day-to-day usage, the following options are relevant.

- **`l`** This option corresponds to a left-aligned column.
- **`r`** This corresponds to a right-aligned column.
- **`c`** This corresponds to a centred column.
- **`p{⟨width⟩}`** This option corresponds to a top-aligned ⟨`width`⟩-wide column that is typeset as a paragraph in the "usual" way. Some commands such as `\\` are not allowed at the top level.
- **`|`** This option does not correspond to an actual column but results in additional decoration. It results in a vertical line drawn at at the "current" position. For example, if ⟨`column alignment`⟩ is `l|cr` then there will be a vertical line separating the first two columns. Using this option is discouraged because the vertical lines usually distract. ☑

The `tabular` environment also defines the following commands,

Figure 2.14
Using the tabular environment

```
\begin{tabular}{l|crp{3.1cm}}
  \hline
     1 &   2 &   3
       & Box me in,
         but not too
         tight, please.
\\\hline
    11 &  12 &  13 & Excellent.
\\ 111 & 112 & 113 & Thank you!
\\\hline
\end{tabular}
```

1	2	3	Box me in, but not too tight, please.
11	12	13	Excellent.
111	112	113	Thank you!

which may be used inside the environment. You can only use these commands at the start of a row.

\hline

This command inserts a horizontal rule. The command may only be used at the *start* of a row. ☑

\cline{⟨number$_1$⟩}{⟨number$_2$⟩}

This draws a horizontal line from the start of column ⟨number$_1$⟩ to the end of column ⟨number$_2$⟩. ☑

\vline

This results in a vertical line. The command may only be used if the column is aligned to the left, to the right, or to the centre. ☑

Figure 2.14 presents a simple example of the tabular environment. The example shows all alignments and the paragraph feature.

Note that line breaks are inserted automatically inside p-type columns. Line breaks are not allowed in columns aligned with l, r, or c.

Intermezzo. The column alignment option | and the commands \hline, \cline, and \vline are irresistible to new users. This may be because most examples of the tabular environment involve the option and these commands. It is understandable that new users want to repeat this, especially when they're not aware that using the option and the commands in moderation is better because the grid lines are dazzling and distracting. Chapter 6 provides some guidelines on how to design good tables.

Regular $m \times n$ tables with the same alignment in the same column are rare. The following command lets you join columns within a row and override the default alignment.

\multicolumn{⟨number⟩}{⟨column alignment⟩}{⟨text⟩}

This inserts ⟨text⟩ into a *single* column that is formed by combining the next ⟨number⟩ columns in the current row. The alignment of the column is determined by ⟨column alignment⟩. This command is especially useful for overriding the default alignment in column headings of a table. An example is presented in the next section. ☑

2.19.4 *The booktabs Package*

The booktabs package adds some extra functionality to the tabular environment. The package discourages vertical grid lines. Using the booktabs package results in better looking tables.

The following are some of the package's features.

- It provides different commands for different rules.
- It provides different rules that may have different widths.
- It provides commands for temporarily/permanently changing the width of the rules.
- It has a command that adds extra line space.
- It is compatible-ish with the colortbl package, which is used to specify coloured tables.

The booktabs package provides the following commands. The first four commands take an option that specifies the width of the rule. The first four commands can only be used at the start of a row.

\toprule[⟨width⟩]

This typesets the full horizontal rule at the top the table. ☑

\bottomrule[⟨width⟩]

This typesets the full horizontal rule at the bottom of the table. ☑

\midrule[⟨width⟩]

This typesets the remaining full horizontal rules in the table. ☑

\cmidrule[⟨width⟩]{⟨number$_1$⟩-⟨number$_2$⟩}

This typesets a partial horizontal rule. The rule is supposed to be used in the middle of the table. It ranges from the start of column ⟨number$_1$⟩ to the end of column ⟨number$_2$⟩. ☑

\addlinespace[⟨height⟩]

This command is usually used immediately after a line break and it inserts more vertical line space to the height of ⟨height⟩. ☑

Figure 2.15 demonstrates how to use the booktabs-provided rule commands. The resulting output is presented in Figure 2.16. Notice that the inter-linespacing is much better than the output in Figure 2.14. Also notice the different widths of the rules.

2.19.5 *Advanced tabular Constructs*

Using basic tabular constructs usually suffices for day-to-day typesetting. This section explains the techniques that give you the power to typeset more advanced tabular constructs.

The following starts by presenting two addition column options. This is followed by some style parameters that control the default size and spacing of the tabular, tabular*, and array environments. The column options are as follows.

*{⟨number⟩}{⟨column options⟩}

This inserts ⟨number⟩ copies of ⟨column options⟩. For example, *{2}{lr} is equivalent to lrlr. ☑

Figure 2.15
Input of booktabs package

```
\begin{tabular}[c]{lrrp{47mm}}
  \toprule \multicolumn{l}{r}{\textbf{Destination}}
         & \multicolumn{l}{r}{\textbf{Duration}}
         & \multicolumn{l}{r}{\textbf{Price}}
         & \multicolumn{l}{r}{\textbf{Description}}
\\\midrule
   Cork City
   &  7 Days & \euro 300.00
   & Visit Langer Land. Price includes visits
     to Rory Gallagher Place and de Maarkit.
\\ Dingle
   &  8 Days & \euro 400.00
   & Have fun with Fungie.
\\\bottomrule
\end{tabular}
```

Figure 2.16
Output of booktabs package. The input of this figure is listed in Figure 2.15. Clearly, booktabs rules rule.

Destination	**Duration**	**Price**	**Description**
Cork City	7 Days	€300.00	Visit Langer Land. Price includes visits to Rory Gallagher Place and de Maarkit.
Dingle	8 Days	€400.00	Have fun with Fungie.

@{⟨text⟩}

This is called an @-expression. It inserts ⟨text⟩ at the current position. This is useful if you want to add certain text or symbols at the given position. For example @{.} inserts a full stop at the current position.

LaTeX normally inserts some horizontal space before the first column and after the last column. It inserts twice that amount of space between adjacent columns. However, this space is suppressed if an @-expression precedes or follows a column option. For example, if ⟨column alignment⟩ is equal to @{}ll@{}l@{} then this suppresses the horizontal space before the first column, after the last column, and between the second and last column. The length \tabcolsep controls the extra horzontal space that is inserted. The value of the command is half the width that is inserted between columns.

A horizontal spacing command in an @-expression controls the separation of two adjacent columns. For example, @{\hspace{⟨width⟩}} inserts a horizontal ⟨width⟩-wide space.

Finally, @-expressions may also adjust the default column separation. The \extracolsep{⟨width⟩} adds additional horizontal ⟨width⟩-wide space between subsequent columns. *However,* additional width is never inserted before the first column. The \extracolsep{\fill} inserts the maximum possible amount of horizontal space. This is useful if you want to extend the width to the maximum possible width.

The columns in the second table in Figure 2.17 are spread out evenly with an @-expression. The third table adds the usual space to

```
\begin{tabular*}{3cm}{@{}lcr@{}}
  \toprule M & M & M \\\bottomrule
\end{tabular*}
\begin{tabular*}{3cm}
      {@{\extracolsep{\fill}}%
       lcr%
       @{\hspace{0pt}}}
  \toprule M & M & M \\\bottomrule
\end{tabular*}
\begin{tabular*}{3cm}
      {@{\hspace{\tabcolsep}}%
       @{\extracolsep{\fill}}%
       lcr%
       @{\hspace{\tabcolsep}}}
  \toprule M & M & M \\\bottomrule
\end{tabular*}
```

```
M   M   M          M      M      M      M     M     M
```

Figure 2.17
Controlling column widths with an @-expression. The output is spaced out for clarity.

the start of the first and the end of the last column. The first table is added for comparison. ☑

The following commands control the default appearance of `tabular`, `tabular*`, and `array` environments.

`\arraycolsep`
This length is equal to half the default horizontal distance between adjacent columns in the `array` environment. This amount of space is also equal to the default horizontal space inserted before the first column and after the last column. ☑

`\tabcolsep`
This is half the default horizontal distance between adjacent columns in the `tabular` and `tabular*` environments. Again, this is equal to the default horizontal space that is inserted before the first column and after the last column. ☑

`\arrayrulewidth`
This length is the width of the lines resulting from a `|` in the ⟨`column options`⟩ argument and the lines resulting from the commands `\cline`, `\hline`, and `\vline`. ☑

`\doublerulesep`
This is the distance between two adjacent lines resulting from a `||` in the ⟨`column options`⟩ argument or two adjacent lines resulting from the `\hline` command. ☑

`\arraystretch`
This command determines the distance between successive rows. It defaults to 1 and “multiplying” it by x results in rows that are x times further apart. So, by redefining this command to 0.50 you halve the row distance. Redefining commands is explained in Chapter 11. ☑

Figure 2.18
The `tabbing` environment

```
\begin{tabbing}
From \=here to \=there \\
   \>and \>then\\
   \>\>all\\
   \>the \>way\\
back \>to \>here.
\end{tabbing}
```

```
From here to there
     and     then
             all
     the     way
back to      here.
```

2.19.6 *The `tabbing` Environment*

The `tabbing` environment is useful for positioning material relative to user-defined alignment positions. The remainder of this section describes some basic usage of the environment. The reader is referred to [Lamport 1994, pages 201–203] for more detailed information.

The `tabbing` environment can only be used in *paragraph mode* (the "usual mode"). It produces lines of text with alignment in columns based upon *tab positions.*

`\=`

Defines the next tab (alignment) position. ☑

`\\`

Inserts a line break and resets the next tab position to the value of `left_margin_tab`. ☑

`\kill`

Throws away the current line but remembers the tab positions defined with `\=`. ☑

`\+`

Increments `left_margin_tab`. ☑

`\-`

Decrements `left_margin_tab`. ☑

`\>`

Move to the next tab stop. ☑

Figures 2.18 and 2.19 present two examples of the `tabbing` environment. The examples do not demonstrate the full functionality of the environment.

2.20 Language Related Issues

As suggested by its title, this section is concerned with language related issues. The remaining three sections deal with hyphenation, foreign languages, and spelling.

2.20.1 *Hyphenation*

LaTeX's (TeX's really) automatic hyphenation is second to none. However, sometimes even TeX gets it wrong. There are two ways to overcome such problems.

- The command `\-` in a word tells LaTeX that it may hyphenate the word

```
\begin{tt}\begin{tabbing}
AAA\=AAA\=AAA\=AAA \kill
FUNC euc( INT a,
          INT b ): INT \\
BEGIN \+ \\
  WHILE (b != 0) DO \\
  BEGIN \+ \\
    INT rem = a MOD b;\\
    a = b; \\
    b = rem; \- \\
  END \\
  RETURN a; \- \\
END;
\end{tabbing}\end{tt}
```

```
FUNC euc( INT a, INT b ): INT
BEGIN
   WHILE (b != 0) DO
   BEGIN
      INT rem = a MOD b;
      a = b;
      b = rem;
   END
   RETURN a;
END;
```

Figure 2.19
Advanced `tabbing`

```
\usepackage[dutch,british]{babel}
⋮
\selectlanguage{dutch}
% Dutch text here.
 Nederlandse tekst hier.

\selectlanguage{british}
% Engelse tekst hier.
 English text here.
```

Figure 2.20
Using the `babel` package

at that position.

```
Er\-go\-no\-mic has three hyphenation positions.
```
LaTeX Usage

- ○ Specifying the same hyphenation patterns is messy and prone to errors. Using the `\hyphenation` command is a much cleaner solution. This command takes one argument, which should be a comma-separated list of words. For each word you can put a hyphen at the (only) possible, desired, or allowed hyphenation positions. You may use the command several times. The following is an example.

```
\hyphenation{fortran,er-go-no-mic}
```
LaTeX Usage

2.20.2 *Foreign Languages*

The `babel` package supports multi-lingual documents. The package supports proper hyphenation, switches between different languages in one single document, definition of foreign languages, commands that recognise the “current” language, and so on. Figure 2.20 provides a minimal example. Rik Kabel kindly informed that XƎTEX users use the `polyglossia` package instead of `babel`. One of the advantages of the `polyglossia` package is that it automatically loads the `bidi` package when bi-directional scripts are used.

2.20.3 *Spell-Checking*

LaTeX does not support automatic spell-checking. Note that spell-checking isn't trivial anyway because commands may generate text. Text may come from external files, so make sure you spell-check your bibliography files.

However, most modern Integrated Development Environments (IDEs) have a spell checker. The `ispell` program, which can be run from the command line, has a LaTeX spell-check mode. The `-t` flag tells the command that the input is LaTeX.

Unix Session

```
$ ispell -l -t -S input.tex | sort -u
```

PART III

Tables, Diagrams, and Data Plots

Picture omitted to keep size of `.pdf` small.

Oil and charcoal on canvas (24/01/08), 132 × 213 cm
Work included courtesy of Billy Foley

PART IV

Mathematics and Algorithms

Picture omitted to keep size of `.pdf` small.

Oil paint and charcoal on canvas (05/09/09), 152 × 213 cm
Work included courtesy of Billy Foley

PART V

Automation

Picture omitted to keep size of `.pdf` small.

Oil paint and charcoal on canvas (17/06/09), 152 × 213 cm
Work included courtesy of Billy Foley

PART VI

Miscellany

Picture omitted to keep size of `.pdf` small.

Oil and charcoal on canvas (06/10/10 no 2), 64 × 91 cm
Work included courtesy of Billy Foley

Chapter 17

Using LaTeX on Windows

This chapter explains how to install a free and widely available LaTeX distribution called TeX Live on `Windows`, how to use a widely used LaTeX IDE called `TeXworks`, and a BibTeX IDE called `JabRef`.

TeX Live is easy to install and use. For example, this book was typeset with a standard TeX Live installation without any special configurations. Another advantage of using TeX Live is that it works on `Windows`, on `Unix`, and on the `Mac`. This is also one of the reasons for choosing the LaTeX and BibTeX IDEs. However, the IDEs were also chosen because they are easy to use.

The remainder of this chapter starts by explaining how to uninstall your previous TeX Live installation and how to install TeX Live. This is followed by some sections explaining how to use `TeXworks` and `JabRef`. The last section explains how to install class and package files.

17.1 Removing Your Previous Installation

There should not be any reason to uninstall your previous TeX Live installation. However, if you are stuck for disk space removing the previous installation may help. Notice that you may need administrator rights if you want to remove the previous installation.

Uninstalling the previous installation is done by running the command `tlmgr uninstall` from the command line. To do this open the `Start` menu, select `Run`, and enter `command`. In the window that opens you can type in commands from the command line. You should type in the command `tlmgr uninstall`:[1]

Windows Usage

```
C:\> tlmgr uninstall
```

17.2 Installing TeX Live

This section explains how to install TeX Live. If you have internet access, you can download a TeX Live installer from `http://tug.org/texlive/acquire-netinstall.html`. If you don't have internet access or if internet access is slow, you can install TeX Live from a *TeX Collection* Digital Versatile Disk (DVD). You may get a copy of the

[1] The text `C:\Users\user>` in the example is the *prompt*, which is output by the Operating System (OS) when it's ready to accept a command. Prompts differ from system to system and from user to user, so your prompt may be different.

dvd from a friend, from a TEX user group, or (if you have it) from your local IT support. A list of all TEX user groups may be found at `http://www.tug.org/usergroups.html`. Installing from DVD is similar to installing from the internet. To simplify the presentation, this section assumes you have internet access.

17.2.1 *Proxy Configuration*

To install TEX Live over the internet you need internet access. If access to the internet is via a proxy then you must tell the installation program about the address of the proxy. You do this by creating an environment variable that holds the value of the proxy's address. To create the environment variable you do the following. (You may need administrator rights to do this.)

1. Left click `Main Menu`.
2. Right click `My Computer`.
3. Select `Properties`.
4. Select `Advanced`.
5. Click the `Environment Variables` button.
6. In the `System Variables` section click the `New` button. This should open the `New System Variable` dialogue box.
   a. Enter the text `http_proxy` in the `Variable Name` text box.
   b. Enter the URL and the port number of your proxy in the `Variable Value` text box. For example `http://http-proxy:8080` but this may depend on your computer's configuration.
   c. Click the `Ok` button until you're done.

17.2.2 *Downloading the Installer*

The recommended method for installing TEX Live is using the *network installer,* which downloads the TEX Live distribution as part of the installation process. To download the network installer, go to `http://tug.org/texlive/acquire-netinstall.html`, click on the *install-tl.zip* link, and save the installer on your desktop.

17.2.3 *Running the Installer*

The main installation process is not much more difficult than downloading the installer. The following are the main details.

1. Rest your mouse pointer over the installer icon on your desktop. Right click and select `extract all`. Select all the default options to finish the extraction process. After this process you should have a folder called `install-tl-⟨number⟩`. The ⟨number⟩ corresponds to the year and the version of the installation.
2. Go to the installation folder by double clicking `install-tl-⟨number⟩`.
3. Double click `install-tl.bat` (it may appear as `install-tl`) to start

the installation process. Accept all default configurations to complete the installation.

The TeX Live installation process should have created a sub-menu called `TeX Live` ⟨`year`⟩ under `Start`→`All Programs`, where ⟨`year`⟩ is the year of the TeX Live distribution. The sub-menu contains some useful utility programs, some of which are explained further on.

Except for some minor configuration, which is explained in the next section, you are now done with the installation process.

17.2.4 *Minor Configuration*

Usually, TeX Live does a pretty good job when it comes to selecting the defaults. However, it may get get things wrong when it comes to the default paper size, which is set to A4. If you require an 8.5 × 11 inches paper size, the easiest way to make this the default paper size is running the command `tlmgr paper letter` from the command line. There is also a Graphical User Interface (GUI) for `tlmgr` but it is not explained to simplify the presentation.

Run `tlmgr` from the command line works as before: Open the `Start` menu, select `Run`, and enter `command`; in the window that opens type in the command `tlmgr paper letter`:

Windows Usage

```
C:\Users\user> tlmgr paper letter
```

17.2.5 *Testing the Installation*

At this stage it is a good idea to test the installation. To test your installation you could create a LaTeX input file and process it with `pdflatex`. However, it may be easier to compile one of the files that come with TeX Live. For example, you should be able to compile the file `small2e.tex` by executing the command `pdflatex small2e.tex` from the command line.

Windows Usage

```
C:\Users\user> pdflatex small2e.tex
```

Thanks to the TeX Live Team, the test passes with flying colours.

17.3 First Steps with `TeXworks`

`TeXworks` is a LaTeX IDE that is based on an award-winning LaTeX `Mac` IDE called `TeXShop`. `TeXworks` has a simple user interface, is easy to use and is installed as part of TeX Live. `TeXworks` also works on `Unix` and on the `Mac` (as `TeXShop`). These are the main reasons why `TeXworks` is the only LaTeX IDE that is explained in this chapter.

This section explains how to create and compile the LaTeX input in Figure 17.1. The LaTeX input file depends on a tiny bibliography file called `LAF.bib`, which is listed in Figure 17.2. To save time, you may download both files from `http://csweb.ucc.ie/~dongen/LAF/`

Figure 17.1 LaTeX input for bibliography example

```
\documentclass[11pt]{book}
\usepackage[style=authoryear,
            block=space,
            language=british]{biblatex}
\renewcommand*{\bibopenparen}{[}
\renewcommand*{\bibcloseparen}{]}
\renewcommand*{\bibnamedash}{\rule[0.48ex]{3em}{0.14ex}\space}
\addbibresource{LAF}

\begin{document}
  \chapter{Philip Glass}
  \begin{refsection}
  Philip Glass’ \emph{Analog}˜\parencite{Analog} \ldots
  \end{refsection}
  \chapter{Steve Reich}
  \begin{refsection}
  Steve Reich’s \emph{Come Out}~\parencite{Come@Out} \ldots
  \emph{Music for 18~Musicians}~\parencite{18@musns} \ldots
  \end{refsection}
  \printbibliography[section=1,title=Philip Glass Discography]
  \printbibliography[section=2,title=Steve  Reich Discography]
\end{document}
```

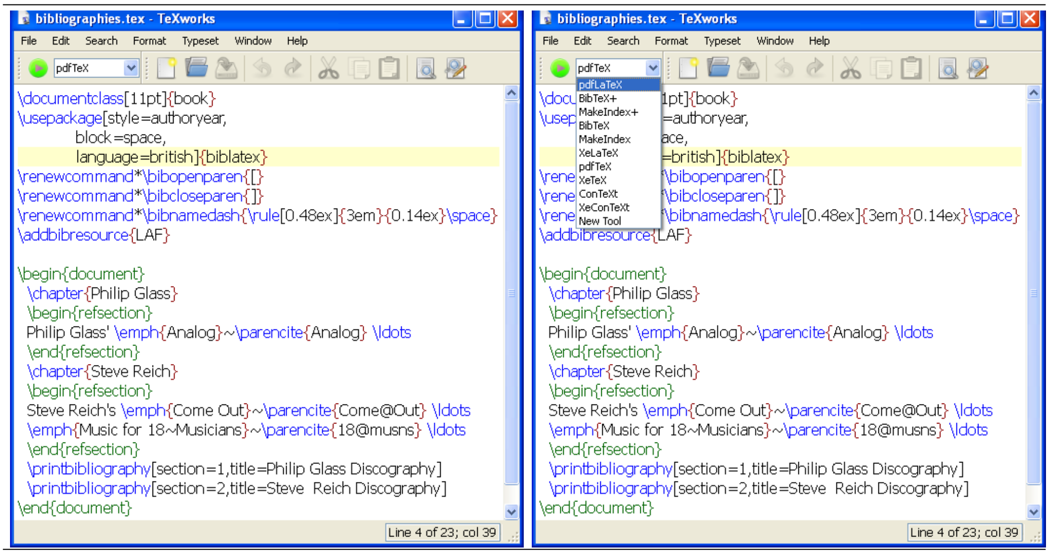
Figure 17.2 BibTeX input for bibliography example

```
@Misc{Analog,
 author  = {Glass, Philip},
 title   = {Analog},
 year    = {2006} }
@Misc{18@musns,
 author  = {Reich, Steve},
 title   = {Music for 18~Musicians},
 year    = {1976} }
@Misc{Come@Out,
 author  = {Reich, Steve},
 title   = {Come Out},
 year    = {1966} }
```

`Extras/bibliographies.tex` and `http://csweb.ucc.ie/~dongen/LAF/Extras/LAF.bib`.

Start by running `TeXworks` and by loading the input program. The following explores the most important aspects of the `TeXworks` GUI.

- The `TeXworks` GUI is shown in the picture on the left of Figure 17.3.
- The green circle with the grey triangle in it is the *typesetting button.*
- To the right of the typesetting button is a drop-down list, which is called the *typesetting engine list.* You use it to select the current *typesetting engine,* which “typesets” your document.
- The typesetting engine is run when you click the typesetting button.



```
\documentclass[11pt]{book}
\usepackage[style=authoryear,
        block=space,
        language=british]{biblatex}
\renewcommand*\bibopenparen{[}
\renewcommand*\bibcloseparen{]}
\renewcommand*\bibnamedash{\rule[0.48ex]{3em}{0.14ex}\space}
\addbibresource{LAF}

\begin{document}
 \chapter{Philip Glass}
 \begin{refsection}
 Philip Glass' \emph{Analog}~\parencite{Analog} \ldots
 \end{refsection}
 \chapter{Steve Reich}
 \begin{refsection}
 Steve Reich's \emph{Come Out}~\parencite{Come@Out} \ldots
 \emph{Music for 18~Musicians}~\parencite{18@musns} \ldots
 \end{refsection}
 \printbibliography[section=1,title=Philip Glass Discography]
 \printbibliography[section=2,title=Steve  Reich Discography]
\end{document}
```

Figure 17.3
Setting the typesetting engine. The picture on the left shows TeXworks with pdfTeX as the current typesetting engine. The picture on the right shows how the typesetting engine is changed to pdfLaTeX by selecting it from the drop-down list to the right of the typesetting button.

The drop-down list always shows the current typesetting engine. In the picture on the left of Figure 17.3 the typesetting engine is pdfTeX.

- There are different typesetting engines: pdfLaTeX, BibTeX, MakeIndex, pdfTeX, ... For example, when pdfLaTeX is the typesetting engine, pdflatex is used to typeset the document, when BibTeX is the typesetting engine, bibtex is used to "typeset" the document, and so on. Note that typesetting engines engines are defined at the TeXworks level: they drive executable programs. For example, the pdfLaTeX engine drives the pdflatex executable.
- You may change the current typesetting engine by selecting it from the typesetting engine list. This is shown in the picture on the right of Figure 17.3, which selects pdfLaTeX.

You now know the basics of the TeXworks GUI. You might as well start using it:

1. This book uses a pdflatex philosophy, so make sure you pdfLaTeX is selected in the typesetting engine list.
2. Proceed by typesetting the document, which you may do by clicking the typesetting button.
3. When you click the button, this should create your output document and should pop up a pdf viewer that displays the output document. Here it is assumed you downloaded the input program and didn't change it. If you have errors, then you may have to resolve the errors and typeset the document until all errors are gone.

Congratulations. You have just overcome the first obstacle on the short path of becoming a TeXworks expert.

On closer inspection the output document isn't perfect. For example, the citation commands in the text only list the logical citation labels, and there is no bibliography. The following section explains how to resolve these problems.

17.4 Creating Bibliographies with `TeXworks`

This section explains how to create proper bibliographies in combination with `TeXworks`. We shall continue with the example from the previous section. Make sure you have saved the BibTEX input from Figure 17.2 in a file called `LAF.bib`. It is recommended you download the BibTEX input from `http://csweb.ucc.ie/~dongen/LAF/Extras/LAF.bib`.

17.4.1 *Creating One Single Bibliography*

As explained in the first chapter, running `pdflatex` alone cannot guarantee a consistent bibliography. If you want to create your bibliography with BibTEX, you also have to run `bibtex`. In this section you learn how to create a bibliography with `TeXworks`. For simplicity, we shall continue the example from the previous section. Start `TeXworks`, load the example, and typeset it once with the `pdfLaTeX` engine.

Most documents have only one bibliography. Remember from Chapter 1 that you compile such documents, by first running `pdflatex`, then running `bibtex`, and then running `pdflatex` two more times. You already know how to run `pdflatex` from within `TeXworks`. If only you could execute `bibtex` from within `TeXworks` .... Fortunately, this is possible. You can run `bibtex` by typesetting your document with the `BibTeX` engine. You see why `TeXworks` is easy?

1. Start by making `BibTeX` your typesetting engine.
2. Execute `bibtex` by clicking the typesetting button.
3. Continue by making `pdfLaTeX` your typesetting engine again.
4. Complete the process by typesetting your document two more times. You do this by clicking the typesetting button twice.

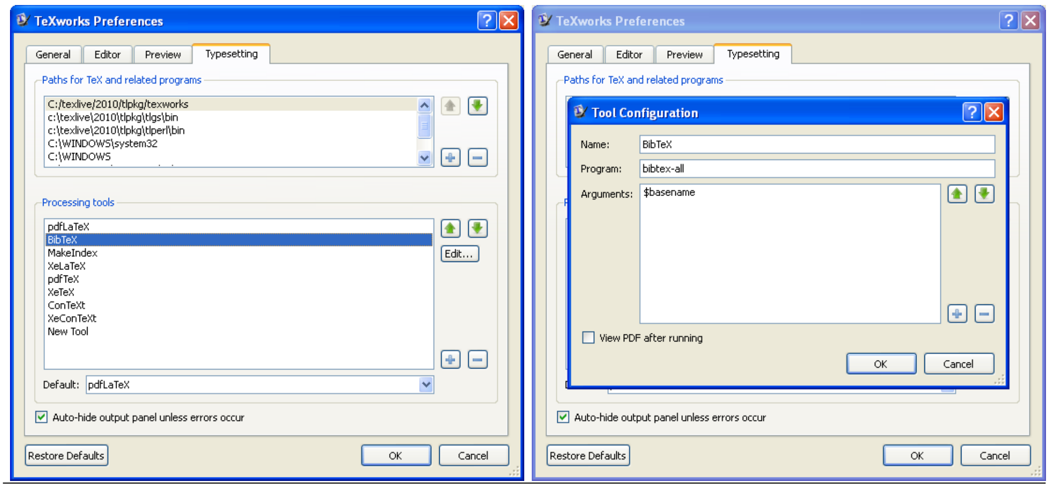
For documents with one bibliography, the previous procedure is enough to get perfect bibliographies. Unfortunately, our LaTEX input file has more than one bibliography. When you typeset such documents with the `BibTeX` engine, `TeXworks` only runs `bibtex` on one auxiliary bibliography file. In the next section we shall configure `TeXworks` so it will run `bibtex` on all auxiliary bibliography files.

17.4.2 *Creating Several Bibliographies*

As explained in the previous section the `BibTeX` typesetting engine only runs `bibtex` on one auxiliary bibliography file. In this section we shall configure the engine so that `bibtex` is executed for each relevant file. After the configuration the `BibTeX` typesetting engine should work correctly for any number of bibliographies.

To configure the `BibTeX` typesetting engine, you do the following.

1. Select `Edit`→`Preferences`. This pops up the `Preference` menu.



Paths for TeX and related programs:
C:/texlive/2010/tlpkg/texworks
c:\texlive\2010\tlpkg\tlgs\bin
c:\texlive\2010\tlpkg\tlperl\bin
C:\WINDOWS\system32
C:\WINDOWS

Processing tools:
pdfLaTeX
BibTeX
MakeIndex
XeLaTeX
pdfTeX
XeTeX
ConTeXt
XeConTeXt
New Tool

Default: pdfLaTeX

☑ Auto-hide output panel unless errors occur

Name: BibTeX
Program: bibtex-all
Arguments: $basename

☐ View PDF after running

Figure 17.4

Configuring TeXworks for multiple bibliographies. The TeXworks Preference menu on the left pops up when you click edit→preferences and then select the Typesetting tab. Selecting BibTeX and clicking Edit pops up the menu on the right. This is where you configure the BibTeX engine. The Program text field determines the program that is executed for the BibTeX engine, and the Arguments text field determines its argument.

2. Proceed by selecting the Typesetting tab. This activates the Typesetting Menu. This is shown on the left of Figure 17.4.
3. In the Processing Tools list select BibTeX. This pops up the Tool Configuration menu for the BibTeX typesetting engine. This is shown on the right of Figure 17.4.
4. The Program text box in the Tool Configuration menu determines the program that TeXworks executes when you typeset your document with the BibTeX engine. Enter bibtex-all.bat; we shall implement this user-defined script later.
5. The Arguments text box in the Tool Configuration menu determines which arguments are passed to our user-defined script. Enter $basename.
6. Accept the current configuration by repeatedly clicking the Ok buttons until the Preference menu is gone.
7. At this stage, you're almost done. All that's left is implementing the user-defined script, which should have the following commands. You may download the script from http://csweb.ucc.ie/~dongen/LAF/Extras/windows/bibtex-all.bat. (The 1 in %1*.aux is the digit one.)

Windows Usage

```
for %%f in (%1*.aux) do (
    bibtex %%~nf
)
```

8. Save the script as bibtex-all.bat in a folder that is in your PATH, and make sure the script is executable.

You should now be able to create your bibliographies by typesetting your document once with the BibTeX typesetting engine and twice with the pdfLaTeX typesetting engine.

17.5 Creating Indexes with TeXworks

In this section we shall create an output document with indexes. We shall do this using the LaTeX input that is depicted in Figure 17.5. You

Figure 17.5
LaTeX input for indexes example

```
\documentclass[11pt]{article}
\usepackage{multind}

\makeindex{programs}
\makeindex{authors}
\begin{document}
  Knuth\index{authors}{Knuth} is the author of
   \TeX\index{programs}{TeX@\TeX}.

  \printindex{programs}{Index of Programs}
  \printindex{authors}{Index of Authors}
\end{document}
```

may download the input from `http://csweb.ucc.ie/~dongen/LAF/Extras/indexes.tex`.

Creating the indexes is done by typesetting the document with the `MakeIndex` engine. Unfortunately, this will only create one proper index, so you have to configure the `MakeIndex` engine. You may do this with a similar technique as the technique for configuring the `BibTeX` typesetting engine. The following explains the configuration process.

1. Select `Edit`→`Preferences`. This activates the `Preference` menu.
2. Proceed by selecting the `Typesetting` tab.
3. In the `Processing Tools` list select `MakeIndex` and click the `Edit` button. This should activate the `Tool Configuration` menu for the `MakeIndex` typesetting engine.
4. Enter `makindex-all` in the `Program` text box; we shall implement this user-defined script later.
5. There is no need to enter anything in the `Arguments` text box. The reason for this is that the names of the index files don't depend on the LaTeX document name, so `makeindex` has to process all `.idx` files.
6. Accept the current configuration by repeatedly clicking the `Ok` buttons until the `Preference` menu is gone.
7. Implementing the user-defined script completes the configuration. The following is the script. You may download the script from `http://csweb.ucc.ie/~dongen/LAF/Extras/windows/makeindex-all.bat`.

   Windows Usage

   ```
   for %%f in (*.idx) do (
       makeindex %%~nf
   )
   ```
8. Save the script as `makeindex-all.bat` in the same folder as `bibtex-all.bat`.

You should now be able to create your indexes by (1) typesetting your document with the `pdfLaTeX` typesetting engine, (2) typesetting it with the `MakeIndex` typesetting engine, and (3) typesetting it two more times with the `pdfLaTeX` typesetting engine.

17.6 The LaTeXmk Typesetting Engine

The previous two sections explained how to configure TeXworks for multiple bibliographies and multiple indexes and how to compile your document in a "traditional" manner. A fairly recent development is compiling your documents with *latexmk*, which is equivalent to a make script for LaTeX.

At the time of writing this chapter, latexmk was sufficiently advanced that it could typeset most documents automatically. You just run latexmk -pdf on your LaTeX source file, and the software will execute pdflatex, bibtex, and makeindex on all relevant files until it reaches a fixpoint (until there are no more changes). Needless to say, using latexmk is much easier than running several different programs by hand. The only disadvantage of the latexmk approach is that latexmk may take a bit longer. Further information about latexmk may be found on http://www.phys.psu.edu/~collins/software/latexmk-jcc/.

TeXworks also has a LaTeXmk typesetting engine. As suggested by the name, the engine compiles your input document with latexmk. If you are the kind of person that wants to do things the easy way, then this may be the typesetting engine for you.

17.7 Bibliography Management with JabRef

JabRef [*JabRef*] is an open source BibTeX IDE. JabRef requires Java Virtual Machine (JVM) version 1.5 or newer and should run on any machine that has a JVM running. If you don't have a JVM on your machine, visit the JabRef Frequently Asked Question (FAQ) at http://jabref.sourceforge.net/faq.php for instructions on how to best install it.

JabRef has an advanced BibTeX editor, a search engine that lets you search for patterns in the bibliography, an import facility for various other bibliography formats, automatic key generation, and a search facility that lets you search Search Medline, Citeseer, IEEEXplore and arXiv on line.

The remainder of this section is a very short introduction to JabRef, which should be just enough to create a new BibTeX file and edit it. This should be enough to get started; you may learn more about JabRef from its built-in help function.

17.7.1 *Installing JabRef*

Installing JabRef is easy. To install JabRef, visit the JabRef download page at http://jabref.sourceforge.net/download.php and click one of the download button for the latest stable version. Click on the link JabRef-⟨number⟩-setup.exit and save it on your desktop. Go to your desktop and double-click on JabRef-⟨number⟩-setup.exit to start the installation process. Accept all the defaults.

When the installation process is finished, you should have a JabRef icon on your desktop and a JabRef entry in the Start→All Programs

Figure 17.6 Sample `JabRef` bibliography input

```
@Book{Knuth:1990,
  title     = {The {\TeX}book},
  publisher = {Addison\textendash Wesley},
  year      = {1990},
  author    = {Knuth, Donald E.},
  isbn      = {0-201-13447-0},
  url       = {http://www.ctan.org/tex-archive/systems/knuth/tex/}
}
@Article{Knuth:78,
  author    = {Knuth, Donald E.},
  title     = {The Average Time for Carry Propagation},
  journal   = {Indagationes Mathematicae},
  year      = {1978},
  volume    = {40},
  pages     = {238-242}
}
@Book{Lamport:94,
  title     = {{\LaTeX}: \emph{A Document Preparation System}},
  publisher = {Addison\textendash Wesley},
  year      = {1994},
  author    = {Lamport, Leslie},
  isbn      = {0-021-52983-1}
}
```

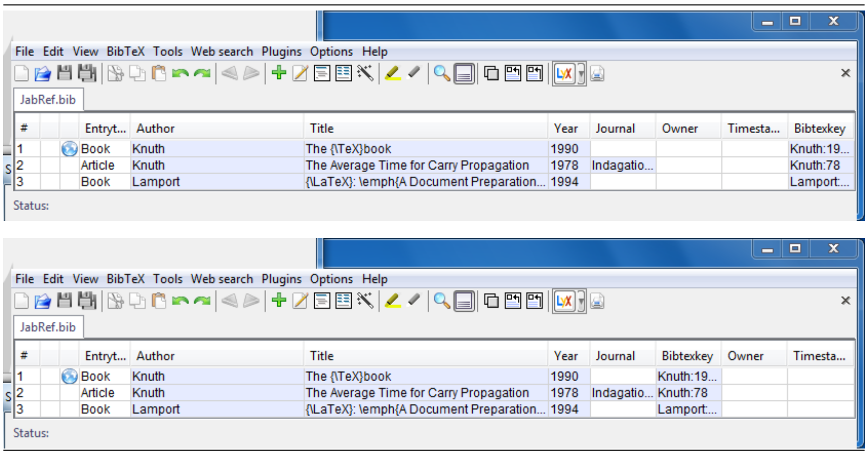
menu. Further more, all your BibTeX files should be associated with `JabRef` so double-clicking any BibTeX file should open it with `JabRef`.

17.7.2 *Using `JabRef`*

This section explains how to create and modify a BibTeX file with `JabRef`. It uses the tiny BibTeX file that is depicted in Figure 17.6. You may download the file from `http://csweb.ucc.ie/~dongen/LAF/Extras/JabRef.bib` and save it as `JabRef.bib`.

Start `JabRef` and open the bibliography database `JabRef.bib`. You should now see the picture at the top of Figure 17.7. The IDE is rather intuitive and shows the different kinds of entries in the BibTeX file: there are two books and one article. The globe on the left side of the entry for the TeXbook corresponds to the `url` field in the BibTeX file. Clicking on the globe leads to the URL in the `url` field. The BibTeX keys are listed in the column `Bibtexkey`. If this column is bit too far to the right to your liking, then you can drag it to a different position. This is shown in the picture at the bottom of Figure 17.7. Other columns may be repositioned similarly.

To create a new entry for an article, select `BibTeX`→`New entry`, and select `Article` as the entry type. Clicking the green plus icon is an alternative selecting `BibTeX`→`New entry`. In the menu that pops up, enter the fields for some real or fictional article. The picture at the top of Figure 17.8 shows the result of filling out the fields required


File Edit View BibTeX Tools Web search Plugins Options Help
JabRef.bib

#	Entryt...	Author	Title	Year	Journal	Owner	Timesta...	Bibtexkey
1	Book	Knuth	The {\TeX}book	1990				Knuth:19...
2	Article	Knuth	The Average Time for Carry Propagation	1978	Indagatio...			Knuth:78
3	Book	Lamport	{\LaTeX}: \emph{A Document Preparation...	1994				Lamport...

Status:

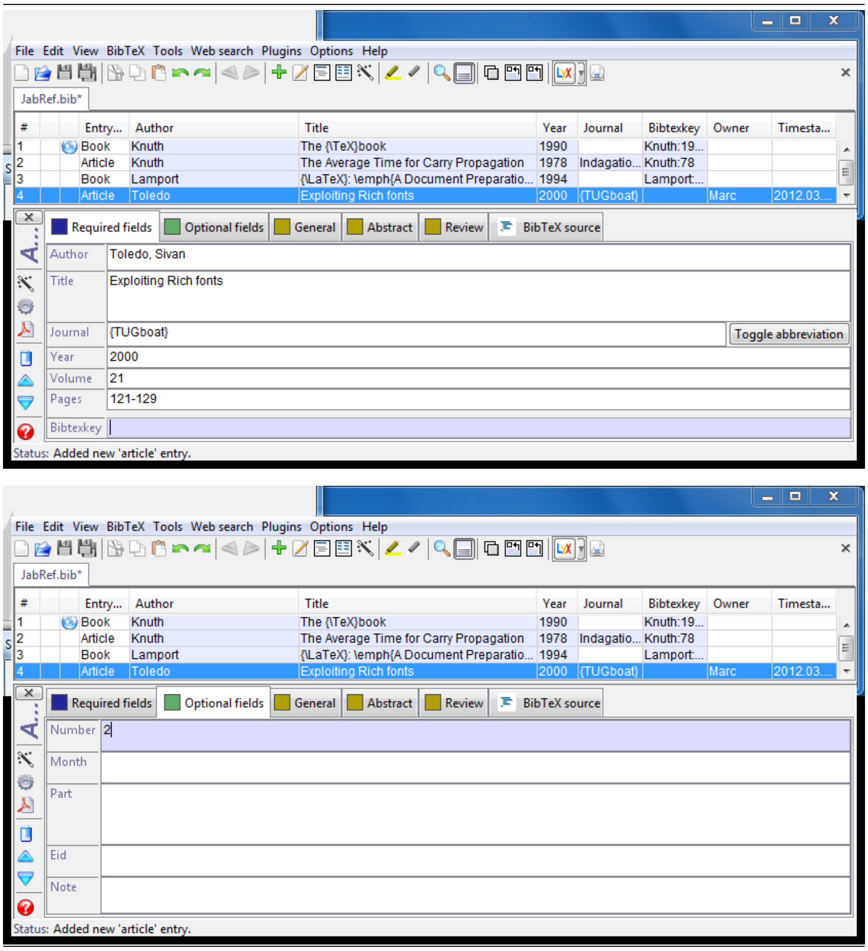
File Edit View BibTeX Tools Web search Plugins Options Help
JabRef.bib

#	Entryt...	Author	Title	Year	Journal	Bibtexkey	Owner	Timesta...
1	Book	Knuth	The {\TeX}book	1990		Knuth:19...		
2	Article	Knuth	The Average Time for Carry Propagation	1978	Indagatio...	Knuth:78		
3	Book	Lamport	{\LaTeX}: \emph{A Document Preparation...	1994		Lamport...		

Status:

Figure 17.7

`JabRef` IDE. The picture on the top shows the original IDE. In the picture on the bottom, the `Bibtexkey` column is dragged to just before the `Owner` column.



#	Entry...	Author	Title	Year	Journal	Bibtexkey	Owner	Timesta...
1	Book	Knuth	The {\TeX}book	1990		Knuth:19...		
2	Article	Knuth	The Average Time for Carry Propagation	1978	Indagatio...	Knuth:78		
3	Book	Lamport	{\LaTeX}: \emph{A Document Preparatio...	1994		Lamport...		
4	Article	Toledo	Exploiting Rich fonts	2000	{TUGboat}		Marc	2012.03....

Author: Toledo, Sivan
Title: Exploiting Rich fonts
Journal: {TUGboat}
Year: 2000
Volume: 21
Pages: 121-129
Bibtexkey:

Status: Added new 'article' entry.

#	Entry...	Author	Title	Year	Journal	Bibtexkey	Owner	Timesta...
1	Book	Knuth	The {\TeX}book	1990		Knuth:19...		
2	Article	Knuth	The Average Time for Carry Propagation	1978	Indagatio...	Knuth:78		
3	Book	Lamport	{\LaTeX}: \emph{A Document Preparatio...	1994		Lamport...		
4	Article	Toledo	Exploiting Rich fonts	2000	{TUGboat}		Marc	2012.03....

Number: 2
Month:
Part:
Eid:
Note:

Status: Added new 'article' entry.

Figure 17.8

Creating a new BibTeX entry. The figure at the top shows the `Required fieds` tab with filled out text fields for the required entries of an article. The figure on the bottom shows the `Optional fields` tab with the filled out text field for the number of the journal.

fields `Author`, `Title`, `Journal`, `Year`, `Volume`, and `Pages`. You may fill in the `Bibtexkey` field with an explicit key. If you're stuck for a choice, `JabRef` will generate a key for you if you click the magic wand icon, which is to the left of the `Title` text field.

If your article is in a journal that has a number, you have to fill out the optional `Number` entry. You may do this by clicking the `Optional fields` tab and entering the number in the `Number` text field. The picture at the bottom of Figure 17.8 shows the result. You may save the current BibTEX database, including the new article entry. by selecting `File`→`Save database` or by clicking the (single) floppy disk icon.

17.8 Installing Classes and Packages

This section explains the two basic class and package installation techniques in TEX Live. The first and preferred technique uses the `tlmgr` program, which is provided by TEX Live. The second technique is "installing by hand," which is prone to errors.

Installing class and package files is impossible unless you know a little bit about how TEX Live organises its files, and how TEX Live executables locate these files. The first three sections provide a quick introduction to the required background. The last two sections explain how to install class and package files. Throughout, the word package is a synonym for class or package.

17.8.1 *The TEX Directory Structure*

TEX Live distributions are huge and with the current technology of operating systems it is impossible to quickly locate a file that is located underneath some directory. What is more, TEX Live distributions cannot guarantee unique file names. For example, an auxiliary file called `aux.tex` may occur in more than one directory.

The TEX Users Group (TUG) Working Group on a TEX Directory Structure TWG-TDS [2004] defined a TEX Directory Structure (TDS), which is specifically designed to overcome problems related to the efficient location of files in a TEX distribution. TEX Live has adopted the TDS and many classes and packages assume an underlying TDS.

Note that a TDS is a property of a directory and several directories may have a TDS. For example, if you accept all defaults when you install TEX Live on `Windows`, then the global TEX Live files are in the directory `%SYSTEMDRIVE%\texlive\⟨year⟩` , where `⟨year⟩` is the year of the TEX Live distribution. At the directory's root is a large directory called `texmf-dist`, which has *a* TDS. Effectively, this is where TEX Live keeps most of its class and package files. Likewise, TEX Live assumes the existence of a user directory called `%USERPROFILE%\texmf` , which should also have a TDS.

Within a TDS-compliant directory, each type of file is expected in a specific place. Figure 17.9 depicts some of the aforementioned TDS `texmf-dist`. At the root level the figure shows the directories `bibtex`, `doc`, `fonts`, and `tex`. As suggested by the names, the directory `bibtex`

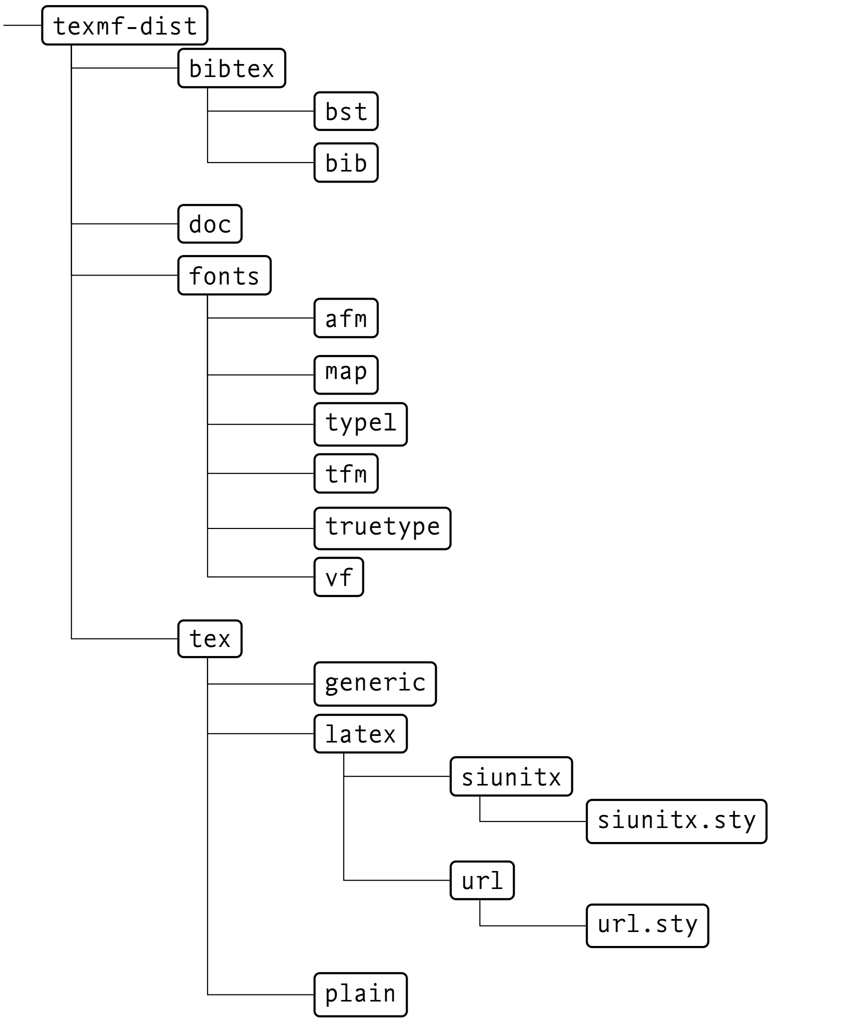


Figure 17.9
TeX directory structure (TDS). The directory `texmf-dist` at the root of the depicted directory tree is where TeX Live keeps its files. The directories inside `texmf-dist`, form a hierarchical structure that is common to other TDS-compliant directories. At the top level the directories `bibtex`, `doc`, `fonts`, and `tex` have a prescribed structure. For example, the directory `texmf-dist\tex\latex` has one directory for each standard installed package. The package files are then organised inside their directory. For example, the `url` package file is located in the file `tex\latex\url\url.sty` inside `texmf-dist`. The files for the `siunitx` package are located in the subdirectory `tex\latex\siunitx\siunitx.sty`.

is for BibTeX related files, `doc` is for documentation, `fonts` is for font files, and `tex` is for TeX files. LaTeX files should be put in the directory dirtex/latex, which should have a subdirectory for each package. The figure shows two such package directories: `siunitx` and `url`. The main purpose of each package directory is storing “its” package files.

17.8.2 *Updating an `ls-R` Database*

When a TeX Live executable such as `pdflatex` has to locate a package, it uses a library called `kpathsea` [Berry, Weber, and Hoekwater 2011], which uses several different techniques to locate the package. One of these techniques is looking up the package location in a plain-text `ls-R` *database,* of which there may be several.

The name of an `ls-R` database is always the same: `ls-R`. Because the name of each `ls-R` database is unique, there can be no more than one `ls-R` database in a given directory. We say that an `ls-R` database is *rooted* in the directory that it is in. Likewise, we say that a directory is the *root* of its `ls-R` database. The main purpose of an `ls-R` database is to store the locations of classes, packages, and other files in the database's root directory and descendant directories.

Remember that you may install a package with `tlmgr` or by hand. When `tlmgr` installs a package, `tlmgr` updates all the relevant `ls-R` databases. When a package is installed in one of these `ls-R` directories (or one of its descendant directories), this usually requires updating the `ls-R` database. You may do this by executing the following command.

Windows Usage

```
C:\> texhash ⟨root directory⟩
```

Here ⟨`root directory`⟩ is the root directory of the `ls-R` database. If `root directory` is large, running the command may take a while. Note that you need write access to the database, so you may need administrator access to do this.

17.8.3 *How `kpathsea` Finds its Files*

As explained earlier on in this chapter, TEX Live executables delegate the task of locating classes, packages, and other files to the `kpatshea` library [Berry, Weber, and Hoekwater 2011]. This section explains this process in more detail. To simplify the presentation, this section only discusses how `kpathsea` locates package files. The process for LATEX (`.tex`), class (`.cls`), and other types of files is similar [Berry, Weber, and Hoekwater 2011].

When `pdflatex` formally asks `kpathsea` for the location of a given package called ⟨`package`⟩, the library starts by constructing a *search path*. Berry, Weber, and Hoekwater [2011, section 3.2] explains how the search path is constructed.

A `kpathsea` search path is a list of colon-separated *path elements*. Each path element consists of a single annotated directory. The annotation determines how the directory should be searched for ⟨`package`⟩.

As expected, `kpathsea` processes the path elements in the search path from left to right until it finds the location of ⟨`package`⟩ or fails.

1. When `kpathsea` processes the next path element, it starts by extracting the directory, ⟨`directory`⟩, from the path element.
2. If one of the prefixes of ⟨`directory`⟩ is the root of an `ls-R` database, then `kpathsea` tries to look up the location of ⟨`package`⟩ in that `ls-R` database [Berry, Weber, and Hoekwater 2011, page 27]. If this operation succeeds, then `kpathsea` returns the resulting location. (Berry, Weber, and Hoekwater [2011] implies that there should be no more than one `ls-R` database along any path leading from any directory to any leaf directory. Furthermore, the name of the `ls-R` database should be in the internal `kpathsea` variable `TEXMFDBS`. This was confirmed by Karl Berry in private communication.)
3. Otherwise, `kpathsea` considers the annotation of the path element.
4. If the path element is of the form `!!`⟨`directory`⟩, `kpathsea` stops processing the path element.
5. If the path element is of the form ⟨`directory`⟩`//`, `kpathsea` completes

processing the path element by searching ⟨directory⟩ recursively for ⟨package⟩.

6. If the path element is of the form ⟨directory⟩, kpathsea completes processing the path element by searching ⟨directory⟩ without recursion. It is assumed the directory has a TDS.
7. (Other path element forms are not allowed.)

The following command outputs the kathsea search path for pdflatex and style files.

Windows Usage

```
C:\> kpsewhich -progname pdflatex -show-path .sty
```

When you run this, many of the path elements in the output have directories that are annotated with the !! prefix. If you install a new package in one of these directories (or descendant directories thereof) then the ls-R database in that directory *must* be updated. If you forget updating the database, the package may never be found.

17.8.4 *Installing Packages with* `tlmgr`

As already mentioned, the tlmgr program is provided as part of TEX Live. The main purpose of tlmgr is updating and installing packages and classes that are part of TEX Live. The program can also update itself. You don't need administrator access to use the program.

If you want to install a class or package that is part of TEX Live then installing them with tlmgr is by far the easiest. Not only will tlmgr install the class or package for you but it will also update any relevant ls-R database. The following is how you install classes and packages with tlmgr from the command line. There is also a tlmgr GUI but this is not explained here.

tlmgr install ⟨names⟩
This installs the classes and packages in the list ⟨names⟩. There is no need to add extensions in ⟨names⟩. ☑

There is also a tlmgr GUI but this is not explained here.

The following are some other useful kpathsea options for day-to-day use.

tlmgr -help
This provides a short explanation of how to use tlmgr. For complete information, visit http://tug.org/texlive/doc/tlmgr.html. ☑

tlmgr update ⟨names⟩
This updates the classes and packages in the list ⟨names⟩. There is no need to add extensions in ⟨names⟩. ☑

tlmgr update --all
This updates all classes and packages. ☑

tlmgr update --self
This updates tlmgr itself. This option may be required if your TEX Live distribution is too old. ☑

tlmgr update --self --all
This updates tlmgr and all classes and packages. ☑

The tlmgr program can also make backups, restore backups, and lots of other things. The package documentation at `http://tug.org/texlive/doc/tlmgr.html` provides further information.

17.8.5 *Installing Packages by Hand*

The previous section showed that installing packages with `tlmgr` is easy. Unfortunately, not all packages are supported by TeX Live, some have to be installed by hand.

Installing classes and packages by hand requires a bit more work than installing with `tlmgr`. Fortunately, it is not much more difficult if you install them in your *local `textmf` tree*—a TDS that is reserved for user-defined class and package files [Berry 2011, page 11].

You may get the location of your (local) `texmf` tree by executing the command `kpsewhich var-value=TEXMFHOME` from the command line. By default TeX Live uses `%USERPROFILE%\texmf` for your `texmf` tree.

Remember from the second last section that the command `kpsewhich -progname pdflatex -show-path .sty` shows the search path for package files. If TeX Live is installed with all the defaults, this command should list your local `texmf` tree in a path element that is suffixed with `//` annotation. Recall that the `//` suffix means that `kpathsea` searches the directory recursively. If all you have to do is installing a single file then creating a subdirectory for the package in your local `texmf` tree and putting the file in that directory should be enough. If the package you're installing is based on a TDS then unpacking the files in the root of your local `texmf` tree should suffice. After this you may, but this is not required update the `ls-R` database at the root of your local `texmf` tree. (Remember that you may do this with the command `texhash`.)

Unfortunately, not all packages are provided as a TDS and you may have to uncompress the files, build the package documentation, move all files to their destination directories, and update the `ls-R` database. The following shows how to do this for a package called ⟨`package`⟩.

1. Create a temporary directory. Download, save, and if necessary extract the package files in this temporary directory.
2. Usually there is a `.ins` file. If there is, run `pdflatex` on it.
3. If there is a `.dtx` file then you must create the class or package documentation. You do this by running `pdflatex` once or twice on the `.dtx` file.
4. What remains is moving the files to the correct location in the local `texmf` tree. In the following it is assumed the local `texmf` tree is in the default TeX Live location: `%USERPROFILE%\texmf` . Start by creating the destination directories for the LaTeX files and for the documentation.

   Windows Usage
   ```
   C:\> mkdir %USERPROFILE%\texmf\tex\doc\⟨package⟩
   C:\> mkdir %USERPROFILE%\texmf\tex\latex\⟨package⟩
   ```
5. Move all LaTeX files to the proper destination.

Extension	Destination in `%USERPROFILE%\texmf\fonts`
`.afm`	`afm\⟨vendor⟩\⟨font name⟩`
`.map`	`map\pdftex\⟨vendor⟩`
`.pfb`	`type1\⟨vendor⟩\⟨font name⟩`
`.tfm`	`tfm\⟨vendor⟩\⟨font name⟩`
`.ttf`	`truetype\⟨vendor⟩\⟨font name⟩`
`.vf`	`vf\⟨vendor⟩\⟨font name⟩`

Table 17.1
TEX Live font directories. The first column lists the extension of the font files. The second column lists the destination directory for these files. The destination is relative to `%USERPROFILE%\texmf\fonts`.

Windows Usage

```
C:\> move *.tex %USERPROFILE%\texmf\tex\latex\⟨package⟩
C:\> move *.sty %USERPROFILE%\texmf\tex\latex\⟨package⟩
C:\> move *.cls %USERPROFILE%\texmf\tex\latex\⟨package⟩
C:\> move *.fd  %USERPROFILE%\texmf\tex\latex\⟨package⟩
```

6. Move all documentation files to the proper destination.

Windows Usage

```
C:\> move *.dvi %USERPROFILE%\texmf\tex\doc\⟨package⟩
C:\> move *.pdf %USERPROFILE%\texmf\tex\doc\⟨package⟩
C:\> move *.ps  %USERPROFILE%\texmf\tex\doc\⟨package⟩
```

7. If the installation involves fonts then you also have to move the font-related files to their proper destination. Table 17.1 lists the destination directories for the font files. The name ⟨`vendor`⟩ is the name of the vendor/supplier of the font and ⟨`font`⟩ is the name of the font. Some vendor directory names are standardised [Berry 1990] but any name is allowed.

8. At this stage it's safe to remove your temporary directory.

9. Complete the installation by updating the `ls-R` database in your local `texmf` directory.

Windows Usage

```
C:\> texhash %USERPROFILE%\texmf
```

Chapter 18
Using LaTeX on Unix

This chapter explains how to install a free and widely available LaTeX distribution called TeX Live on Unix, how to use a widely used LaTeX IDE called TeXworks, and a BibTeX IDE called JabRef.

TeX Live is easy to install and use. For example, this book was typeset with a standard TeX Live installation without any special configurations. Another advantage of using TeX Live is that it works on Windows, on Unix, and on the Mac. This is also one of the reasons for choosing the LaTeX and BibTeX IDEs. However, the IDEs were also chosen because they are easy to use.

The remainder of this chapter starts by explaining how to uninstall your previous TeX Live installation and how to install TeX Live. This is followed by some sections explaining how to use TeXworks and JabRef. The last section explains how to install class and package files.

18.1 Removing Your Previous Installation

If this is the first time you're installing TeX Live, or if your current TeX Live installation is from a previous year, you may proceed to the next section. Otherwise, you have to uninstall your previous version. You may need administrator permissions to remove the previous installation, so make sure you log in as `root` if you need them.

- If your TeX Live was installed by your OS then you should remove it with your OS package manager. For example, on `ubuntu` you could uninstall TeX Live by running `apt-get remove texlive`.
- If your TeX Live installation was installed using the TeX Live installer, then you may remove the installation by running the command `tlmgr uninstall` from the command line.

Unix Usage

```
# tlmgr uninstall
```

18.2 Installing TeX Live

This section explains how to install TeX Live. If you have internet access, you can download a TeX Live installer from `http://tug.org/texlive/acquire-netinstall.html`. If you don't have internet access or if internet access is slow, you can install TeX Live from a *TeX Collection* DVD. You may get a copy of the DVD from a friend, from a TeX user group, or (if you have it) from your local IT support. A list of all TeX user groups may be found at `http://www.tug.org/usergroups.html`. Installing from DVD is similar to installing from the internet. To simplify the presentation, this section assumes you have internet access.

18.2.1 *Downloading the Installer*

The recommended method for installing TeX Live is using the *network installer,* which downloads the TeX Live distribution as part of the installation process. To download the network installer, go to `http://tug.org/texlive/acquire-netinstall.html`, click on the *install-tl-unx.tar.gz* link, and save the installer on your desktop.

18.2.2 *Running the Installer*

The main installation process is not much more difficult than downloading the installer. The following are the main details.

Throughout it is assumed that you have `root` access. We shall install TeX Live in the directory `/usr/local/texlive`. Make sure you are logged in as `root`.

When you install TeX Live as root you want to make sure all users have access to all files in the TeX Live distribution. The easiest way to achieve this is by setting the `umask` to `022`. This gives `root` read, write, and execute (search) permissions, and all other users read and execute (search) permissions. This step is crucial, don't forget it.

Unix Usage

```
# umask 022
```

Next you create the parent directory for the TeX Live installation and make it the working directory.

Unix Usage

```
# mkdir -p /usr/local/texlive
# cd /usr/local/texlive
```

The next step is uncompressing the installer that was saved on your desktop. As part of this step you may as well remove the file from your desktop.

Unix Usage

```
# tar xfz ~/Desktop/install-tl-unx.tar.gz
# rm ~/Desktop/install-tl-unx.tar.gz
# ls
install-tl-⟨number⟩
```

The number at the end of the install directory is a combination of the year and the version. Proceed by making the install directory your working directory.

Unix Usage

```
# cd install-tl-⟨number⟩
# ls
LICENSE.CTAN README.usergroups readme-html.dir     tlpkg
LICENSE.TL   index.html        readme-txt.dir
README       install-tl        release-texlive.txt
```

The script `install-tl` is what you're after. Run it.

Unix Usage
```
# ./install-tl
```

After a few seconds the script should show a text-based installation menu similar to the menu in Figure 18.1.

Usually, the default installation should work, which is why it's safe to start the installation process. You start the installation process by entering `I`⟨`return`⟩. The installation process may take anything from a few minutes to an hour or so. The duration depends on the speed of your internet connection. Eventually, the installation should finish. When this happens, type `Q`⟨`return`⟩ to quit the installer and *make sure* you log off as `root`.

18.2.3 *Setting the Environment Variables*

Finally, you must set some search path related variables. This should be done at the user level so make sure you log off as `root`.

The TeX Live installer puts executables such as `latex`, `pdflatex`, and others in the executable directory `/usr/local/texlive/`⟨`year`⟩`/bin/i386-linux`, where ⟨`year`⟩ is the year of the distribution. TeX Live is installed in `/usr/local/texlive` and the distribution is from 2011. Therefore, the executable directory is given by `/usr/local/texlive/2011/bin/i386-linux`. One of the configuration tasks is adding this directory to the `PATH` environment variable.

To ease the burden of setting the path related variables, we shall define an auxiliary variable that holds the path to the TeX Live installation. You should define this variable in your `.login` file.

Unix Usage
```
export TEXDIR=/usr/local/texlive/2011
```

Next you change the `PATH` variable in your `.login` file. You do this by adding the following lines to your `.login` file. Note that this should be *after* the definition of the variable `TEXDIR`.

Unix Usage
```
export PATH=${TEXDIR}/bin/i386-linux:${PATH}
```

You only have to add this line once. For subsequent TeX Live installations, changing the value of the year in `TEXDIR` should suffice.

Next you modify the manual and info paths. This should also be done after the definition of `TEXDIR`. You should keep these lines for subsequent TeX Live installations.

Unix Usage
```
export MANPATH=${TEXDIR}/texmf/doc/man:${MANPATH}
export INFOPATH=${TEXDIR}/texmf/doc/info:${INFOPATH}
```

Except for some minor configuration, which is explained in the next section, you are now done with the installation process.

18.2.4 *Minor Configuration*

Usually, TeX Live does a pretty good job when it comes to selecting the defaults. However, it may get get things wrong when it comes to the

Figure 18.1
The TEX Live installation menu on Unix

```
 <B> platforms: 1 out of 19

 <S> installation scheme (scheme-full)
     84 collections out of 85, disk space required: 2638 MB

 Customizing installation scheme:
   <C> standard collections
   <L> language collections

 <D> directories:
   TEXDIR (the main TeX directory):
     /usr/local/texlive/2011
   TEXMFLOCAL (directory for site-wide local files):
     /usr/local/texlive/texmf-local
   TEXMFSYSVAR (directory for variable and automatically generated data):
     /usr/local/texlive/2011/texmf-var
   TEXMFSYSCONFIG (directory for local config):
     /usr/local/texlive/2011/texmf-config
   TEXMFVAR (personal directory for variable and automatically generated data):
     ~/.texlive2011/texmf-var
   TEXMFCONFIG (personal directory for local config):
     ~/.texlive2011/texmf-config
   TEXMFHOME (directory for user-specific files):
     ~/texmf

 <O> options:
   [ ] use letter size instead of A4 by default
   [X] allow execution of restricted list of programs via \write18
   [X] create all format files
   [X] install macro/font doc tree
   [X] install macro/font source tree

 <V> set up for portable installation

Actions:
 <I> start installation to hard disk
 <H> help
 <Q> quit

Enter command:
```

default paper size, which is set to A4. If you require an 8.5 × 11 inches paper size, the easiest way to make this the default paper size is running the command `tlmgr paper letter` from the command line. There is also a GUI for `tlmgr` but it is not explained to simplify the presentation.

Unix Usage

```
user@machine:~$ tlmgr paper letter
```

18.2.5 *Testing the Installation*

At this stage it is a good idea to test the installation. Note that this requires logging off an on. If you don't log off and on then your OS will not know the new values of the environment variables. To test your installation you could create a LaTeX input file and process it with `pdflatex`. However, it may be easier to compile one of the files that come with TeX Live. For example, you should be able to compile the file `small2e.tex` by executing the command `pdflatex small2e.tex` from the command line.

Unix Usage

```
user@machine:~$ pdflatex small2e.tex
```

Thanks to the TeX Live Team, the test passes with flying colours.

18.3 First Steps with `TeXworks`

`TeXworks` is a LaTeX IDE that is based on an award-winning LaTeX `Mac` IDE called `TeXShop`. `TeXworks` has a simple user interface and is freely available. `TeXworks` also works on `Windows` and on the `Mac` (as `TeXShop`). These are the main reasons why `TeXworks` is the only LaTeX IDE that is explained in this chapter.

To install `TeXworks`, visit `http://www.tug.org/texworks/#Getting_TeXworks` and follow the installation instructions.

This section explains how to create and compile the LaTeX input in Figure 18.2. The LaTeX input file depends on a tiny bibliography file called `LAF.bib`, which is listed in Figure 18.3. To save time, you may download both files from `http://csweb.ucc.ie/~dongen/LAF/Extras/bibliographies.tex` and `http://csweb.ucc.ie/~dongen/LAF/Extras/LAF.bib`.

Start by running `TeXworks` and by loading the input program. The following explores the most important aspects of the `TeXworks` GUI.

- The `TeXworks` GUI is shown in the picture on the left of Figure 18.4.
- The green circle with the grey triangle in it is the *typesetting button.*
- To the right of the typesetting button is a drop-down list, which is called the *typesetting engine list.* You use it to select the current *typesetting engine,* which "typesets" your document.
- The typesetting engine is run when you click the typesetting button. The drop-down list always shows the current typesetting engine. In the picture on the left of Figure 18.4 the typesetting engine is `pdfTeX`.
- There are different typesetting engines: `pdfLaTeX`, `BibTeX`, `MakeIndex`,

Figure 18.2
LATEX input for bibliography example

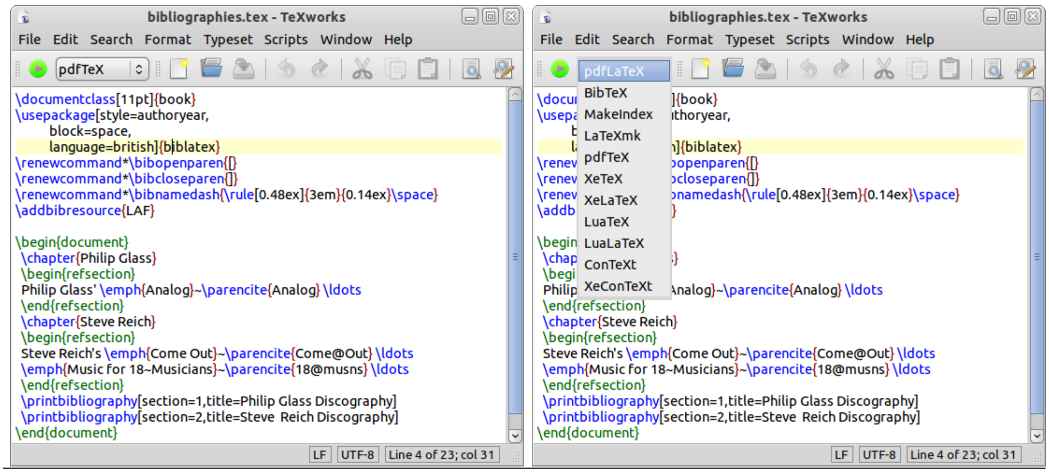
```
\documentclass[11pt]{book}
\usepackage[style=authoryear,
            block=space,
            language=british]{biblatex}
\renewcommand*{\bibopenparen}{[}
\renewcommand*{\bibcloseparen}{]}
\renewcommand*{\bibnamedash}{\rule[0.48ex]{3em}{0.14ex}\space}
\addbibresource{LAF}

\begin{document}
  \chapter{Philip Glass}
  \begin{refsection}
  Philip Glass' \emph{Analog}~\parencite{Analog} \ldots
  \end{refsection}
  \chapter{Steve Reich}
  \begin{refsection}
  Steve Reich's \emph{Come Out}~\parencite{Come@Out} \ldots
  \emph{Music for 18~Musicians}~\parencite{18@musns} \ldots
  \end{refsection}
  \printbibliography[section=1,title=Philip Glass Discography]
  \printbibliography[section=2,title=Steve  Reich Discography]
\end{document}
```

Figure 18.3
BIBTEX input for bibliography example

```
@Misc{Analog,
 author  = {Glass, Philip},
 title   = {Analog},
 year    = {2006} }
@Misc{18@musns,
 author  = {Reich, Steve},
 title   = {Music for 18~Musicians},
 year    = {1976} }
@Misc{Come@Out,
 author  = {Reich, Steve},
 title   = {Come Out},
 year    = {1966} }
```

pdfTeX, ... For example, when pdfLaTeX is the typesetting engine, pdflatex is used to typeset the document, when BibTeX is the typesetting engine, bibtex is used to "typeset" the document, and so on. Note that typesetting engines engines are defined at the TeXworks level: they drive executable programs. For example, the pdfLaTeX engine drives the pdflatex executable.

- You may change the current typesetting engine by selecting it from the typesetting engine list. This is shown in the picture on the right of Figure 18.4, which selects pdfLaTeX.


bibliographies.tex - TeXworks
File Edit Search Format Typeset Scripts Window Help
pdfTeX
LF UTF-8 Line 4 of 23; col 31
bibliographies.tex - TeXworks
File Edit Search Format Typeset Scripts Window Help
pdfLaTeX
BibTeX
MakeIndex
LaTeXmk
pdfTeX
XeTeX
XeLaTeX
LuaTeX
LuaLaTeX
ConTeXt
XeConTeXt
LF UTF-8 Line 4 of 23; col 31

```
\documentclass[11pt]{book}
\usepackage[style=authoryear,
      block=space,
      language=british]{biblatex}
\renewcommand*\bibopenparen{[}
\renewcommand*\bibcloseparen{]}
\renewcommand*\bibnamedash{\rule[0.48ex]{3em}{0.14ex}\space}
\addbibresource{LAF}

\begin{document}
 \chapter{Philip Glass}
 \begin{refsection}
 Philip Glass' \emph{Analog}~\parencite{Analog} \ldots
 \end{refsection}
 \chapter{Steve Reich}
 \begin{refsection}
 Steve Reich's \emph{Come Out}~\parencite{Come@Out} \ldots
 \emph{Music for 18~Musicians}~\parencite{18@musns} \ldots
 \end{refsection}
 \printbibliography[section=1,title=Philip Glass Discography]
 \printbibliography[section=2,title=Steve  Reich Discography]
\end{document}
```

Figure 18.4

Setting the typesetting engine. The picture on the left shows TeXworks with pdfTeX as the current typesetting engine. The picture on the right shows how the typesetting engine is changed to pdfLaTeX by selecting it from the drop-down list to the right of the typesetting button.

You now know the basics of the TeXworks GUI. You might as well start using it:

1. This book uses a pdflatex philosophy, so make sure you pdfLaTeX is selected in the typesetting engine list.
2. Proceed by typesetting the document, which you may do by clicking the typesetting button.
3. When you click the button, this should create your output document and should pop up a pdf viewer that displays the output document. Here it is assumed you downloaded the input program and didn't change it. If you have errors, then you may have to resolve the errors and typeset the document until all errors are gone.

Congratulations. You have just overcome the first obstacle on the short path of becoming a TeXworks expert.

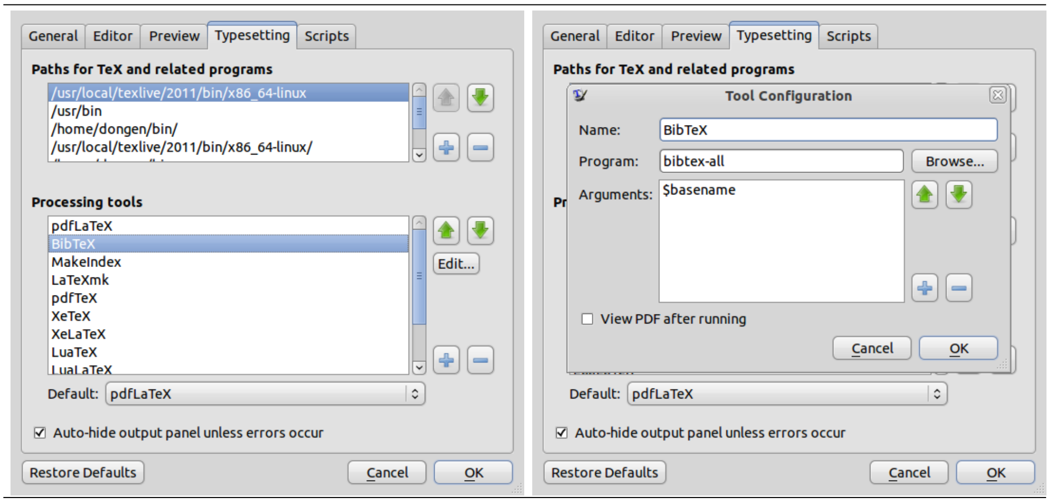
On closer inspection the output document isn't perfect. For example, the citation commands in the text only list the logical citation labels, and there is no bibliography. The following section explains how to resolve these problems.

18.4 Creating Bibliographies with TeXworks

This section explains how to create proper bibliographies in combination with TeXworks. We shall continue with the example from the previous section. Make sure you have saved the BibTEX input from Figure 18.3 in a file called LAF.bib. It is recommended you download the BibTEX input from http://csweb.ucc.ie/~dongen/LAF/Extras/LAF.bib.

18.4.1 *Creating One Single Bibliography*

As explained in the first chapter, running pdflatex alone cannot guarantee a consistent bibliography. If you want to create your bibliography with BibTEX, you also have to run bibtex. In this section you learn



Paths for TeX and related programs

/usr/local/texlive/2011/bin/x86_64-linux
/usr/bin
/home/dongen/bin/
/usr/local/texlive/2011/bin/x86_64-linux/

Processing tools

pdfLaTeX
BibTeX
MakeIndex
LaTeXmk
pdfTeX
XeTeX
XeLaTeX
LuaTeX
LuaLaTeX

Default: pdfLaTeX

☑ Auto-hide output panel unless errors occur

Tool Configuration

Name: BibTeX
Program: bibtex-all
Arguments: $basename

☐ View PDF after running

Figure 18.5

Configuring TeXworks for multiple bibliographies. The TeXworks Preference menu on the left pops up when you click edit→preferences and then select the Typesetting tab. Selecting BibTeX and clicking Edit pops up the menu on the right. This is where you configure the BibTeX engine. The Program text field determines the program that is executed for the BibTeX engine, and the Arguments text field determines its argument.

how to create a bibliography with TeXworks. For simplicity, we shall continue the example from the previous section. Start TeXworks, load the example, and typeset it once with the pdfLaTeX engine.

Most documents have only one bibliography. Remember from Chapter 1 that you compile such documents, by first running pdflatex, then running bibtex, and then running pdflatex two more times. You already know how to run pdflatex from within TeXworks. If only you could execute bibtex from within TeXworks .... Fortunately, this is possible. You can run bibtex by typesetting your document with the BibTeX engine. You see why TeXworks is easy?

1. Start by making BibTeX your typesetting engine.
2. Execute bibtex by clicking the typesetting button.
3. Continue by making pdfLaTeX your typesetting engine again.
4. Complete the process by typesetting your document two more times. You do this by clicking the typesetting button twice.

For documents with one bibliography, the previous procedure is enough to get perfect bibliographies. Unfortunately, our LaTeX input file has more than one bibliography. When you typeset such documents with the BibTeX engine, TeXworks only runs bibtex on one auxiliary bibliography file. In the next section we shall configure TeXworks so it will run bibtex on all auxiliary bibliography files.

18.4.2 *Creating Several Bibliographies*

As explained in the previous section the BibTeX typesetting engine only runs bibtex on one auxiliary bibliography file. In this section we shall configure the engine so that bibtex is executed for each relevant file. After the configuration the BibTeX typesetting engine should work correctly for any number of bibliographies.

To configure the BibTeX typesetting engine, you do the following.

1. Select Edit→Preferences. This pops up the Preference menu.
2. Proceed by selecting the Typesetting tab. This activates the Typesetting Menu. This is shown on the left of Figure 18.5.
3. In the Processing Tools list select BibTeX. This pops up the Tool Configuration menu for the BibTeX typesetting engine. This is shown on the right of Figure 18.5.
4. The Program text box in the Tool Configuration menu determines the program that TeXworks executes when you typeset your document with the BibTeX engine. Enter bibtex-all; we shall implement this user-defined script later.
5. The Arguments text box in the Tool Configuration menu determines which arguments are passed to our user-defined script. Enter $basename.
6. Accept the current configuration by repeatedly clicking the Ok buttons until the Preference menu is gone.
7. At this stage, you're almost done. All that's left is implementing the user-defined script, which should have the following commands. You may download the script from http://csweb.ucc.ie/~dongen/LAF/Extras/unix/bibtex-all.

   Unix Usage

   ```
   #!/bin/sh
   for FILE in $@*.aux; do
       bibtex ${FILE}
   done
   ```

8. Save the script as bibtex-all in a directory that is in your PATH, and make sure the script is executable.

You should now be able to create your bibliographies by typesetting your document once with the BibTeX typesetting engine and twice with the pdfLaTeX typesetting engine.

18.5 Creating Indexes with TeXworks

In this section we shall create an output document with indexes. We shall do this using the LATEX input that is depicted in Figure 18.6. You may download the input from http://csweb.ucc.ie/~dongen/LAF/Extras/indexes.tex.

Creating the indexes is done by typesetting the document with the MakeIndex engine. Unfortunately, this will only create one proper index, so you have to configure the MakeIndex engine. You may do this with a similar technique as the technique for configuring the BibTeX typesetting engine. The following explains the configuration process.

1. Select Edit→Preferences. This activates the Preference menu.
2. Proceed by selecting the Typesetting tab.
3. In the Processing Tools list select MakeIndex and click the Edit button. This should activate the Tool Configuration menu for the MakeIndex typesetting engine.
4. Enter makindex-all in the Program text box; we shall implement this user-defined script later.

Figure 18.6 LaTeX input for indexes example

```
\documentclass[11pt]{article}
\usepackage{multind}

\makeindex{programs}
\makeindex{authors}
\begin{document}
  Knuth\index{authors}{Knuth} is the author of
   \TeX\index{programs}{TeX@\TeX}.

  \printindex{programs}{Index of Programs}
  \printindex{authors}{Index of Authors}
\end{document}
```

5. There is no need to enter anything in the `Arguments` text box. The reason for this is that the names of the index files don't depend on the LaTeX document name, so `makeindex` has to process all `.idx` files.
6. Accept the current configuration by repeatedly clicking the `Ok` buttons until the `Preference` menu is gone.
7. Implementing the user-defined script completes the configuration. The following is the script. You may download the script from `http://csweb.ucc.ie/~dongen/LAF/Extras/unix/makeindex-all`.

Unix Usage

```
#!/bin/sh
for FILE in *.idx; do
    makeindex ${FILE}
done
```

8. Save the script as `makeindex-all` in the same directory as `bibtex-all`. Make sure the script is executable.

You should now be able to create your indexes by (1) typesetting your document with the `pdfLaTeX` typesetting engine, (2) typesetting it with the `MakeIndex` typesetting engine, and (3) typesetting it two more times with the `pdfLaTeX` typesetting engine.

18.6 The LaTeXmk Typesetting Engine

The previous two sections explained how to configure `TeXworks` for multiple bibliographies and multiple indexes and how to compile your document in a "traditional" manner. A fairly recent development is compiling your documents with *`latexmk`*, which is equivalent to a `make` script for LaTeX.

At the time of writing this chapter, `latexmk` was sufficiently advanced that it could typeset most documents automatically. You just run `latexmk -pdf` on your LaTeX source file, and the software will execute `pdflatex`, `bibtex`, and `makeindex` on all relevant files until it reaches a fixpoint (until there are no more changes). Needless to say, using `latexmk` is much easier than running several different programs by hand. The only disadvantage of the `latexmk` approach is that `latexmk` may take a bit longer. Further information about latexmk may be found on http://www.phys.psu.edu/~collins/software/latexmk-jcc/.

TeXworks also has a LaTeXmk typesetting engine. As suggested by the name, the engine compiles your input document with latexmk. If you are the kind of person that wants to do things the easy way, then this may be the typesetting engine for you.

18.7 Bibliography Management with JabRef

JabRef [*JabRef*] is an open source BIBTEX IDE. JabRef requires JVM version 1.5 or newer and should run on any machine that has a JVM running. If you don't have a JVM on your machine, visit the JabRef FAQ at http://jabref.sourceforge.net/faq.php for instructions on how to best install it.

JabRef has an advanced BIBTEX editor, a search engine that lets you search for patterns in the bibliography, an import facility for various other bibliography formats, automatic key generation, and a search facility that lets you search Search Medline, Citeseer, IEEEXplore and arXiv on line.

The remainder of this section is a very short introduction to JabRef, which should be just enough to create a new BIBTEX file and edit it. This should be enough to get started; you may learn more about JabRef from its built-in help function.

18.7.1 *Installing JabRef*

Installing JabRef is easy. To install JabRef, visit the JabRef download page at http://jabref.sourceforge.net/download.php and click one of the download button for the latest stable version. Click on the link JabRef-⟨number⟩.jar and save it in a suitable place. From now on you may start JabRef by executing the command java -jar ⟨jar file⟩, where ⟨jar file⟩ is the path leading to the file JabRef-⟨number⟩.jar that you've just saved.

18.7.2 *Using JabRef*

This section explains how to create and modify a BIBTEX file with JabRef. It uses the tiny BIBTEX file that is depicted in Figure 18.7. You may download the file from http://csweb.ucc.ie/~dongen/LAF/Extras/JabRef.bib and save it as JabRef.bib.

Start JabRef and open the bibliography database JabRef.bib. You should now see the picture at the top of Figure 18.8. The IDE is rather intuitive and shows the different kinds of entries in the BIBTEX file: there are two books and one article. The globe on the left side of the entry for the TEXbook corresponds to the url field in the BIBTEX file. Clicking on the globe leads to the URL in the url field. The BIBTEX keys are listed in the column Bibtexkey. If this column is bit too far to the right to your liking, then you can drag it to a different position. This

Figure 18.7 Sample `JabRef` bibliography input

```
@Book{Knuth:1990,
  title     = {The {\TeX}book},
  publisher = {Addison\textendash Wesley},
  year      = {1990},
  author    = {Knuth, Donald E.},
  isbn      = {0-201-13447-0},
  url       = {http://www.ctan.org/tex-archive/systems/knuth/tex/}
}
@Article{Knuth:78,
  author    = {Knuth, Donald E.},
  title     = {The Average Time for Carry Propagation},
  journal   = {Indagationes Mathematicae},
  year      = {1978},
  volume    = {40},
  pages     = {238-242}
}
@Book{Lamport:94,
  title     = {{\LaTeX}: \emph{A Document Preparation System}},
  publisher = {Addison\textendash Wesley},
  year      = {1994},
  author    = {Lamport, Leslie},
  isbn      = {0-021-52983-1}
}
```

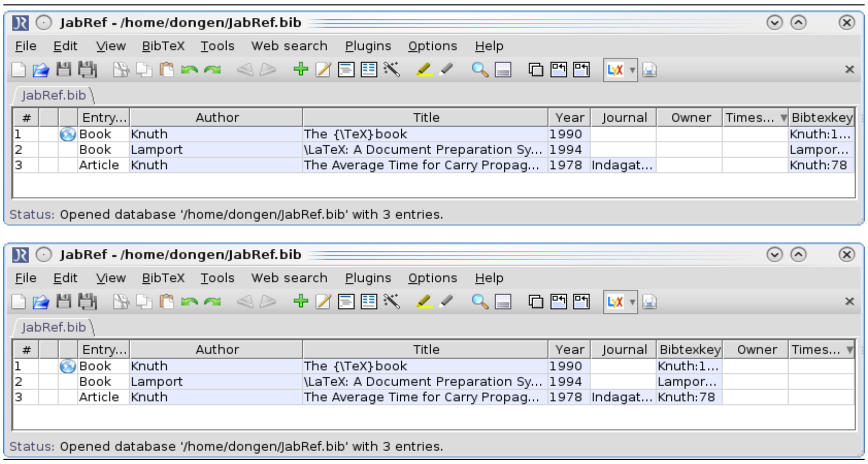
is shown in the picture at the bottom of Figure 18.8. Other columns may be repositioned similarly.

To create a new entry for an article, select `BibTeX`→`New entry`, and select `Article` as the entry type. Clicking the green plus icon is an alternative selecting `BibTeX`→`New entry`. In the menu that pops up, enter the fields for some real or fictional article. The picture at the top of Figure 18.9 shows the result of filling out the fields required fields `Author`, `Title`, `Journal`, `Year`, `Volume`, and `Pages`. You may fill in the `Bibtexkey` field with an explicit key. If you're stuck for a choice, `JabRef` will generate a key for you if you click the magic wand icon, which is to the left of the `Title` text field.

If your article is in a journal that has a number, you have to fill out the optional `Number` entry. You may do this by clicking the `Optional fields` tab and entering the number in the `Number` text field. The picture at the bottom of Figure 18.9 shows the result. You may save the current BibTeX database, including the new article entry. by selecting `File`→`Save database` or by clicking the (single) floppy disk icon.

18.8 Installing Classes and Packages

This section explains the two basic class and package installation techniques in TeX Live. The first and preferred technique uses the `tlmgr` program, which is provided by TeX Live. The second technique is "installing by hand," which is prone to errors.


JabRef - /home/dongen/JabRef.bib
File Edit View BibTeX Tools Web search Plugins Options Help
JabRef.bib

#	Entry...	Author	Title	Year	Journal	Owner	Times...	Bibtexkey
1	Book	Knuth	The {\TeX}book	1990				Knuth:1...
2	Book	Lamport	\LaTeX: A Document Preparation Sy...	1994				Lampor...
3	Article	Knuth	The Average Time for Carry Propag...	1978	Indagat...			Knuth:78

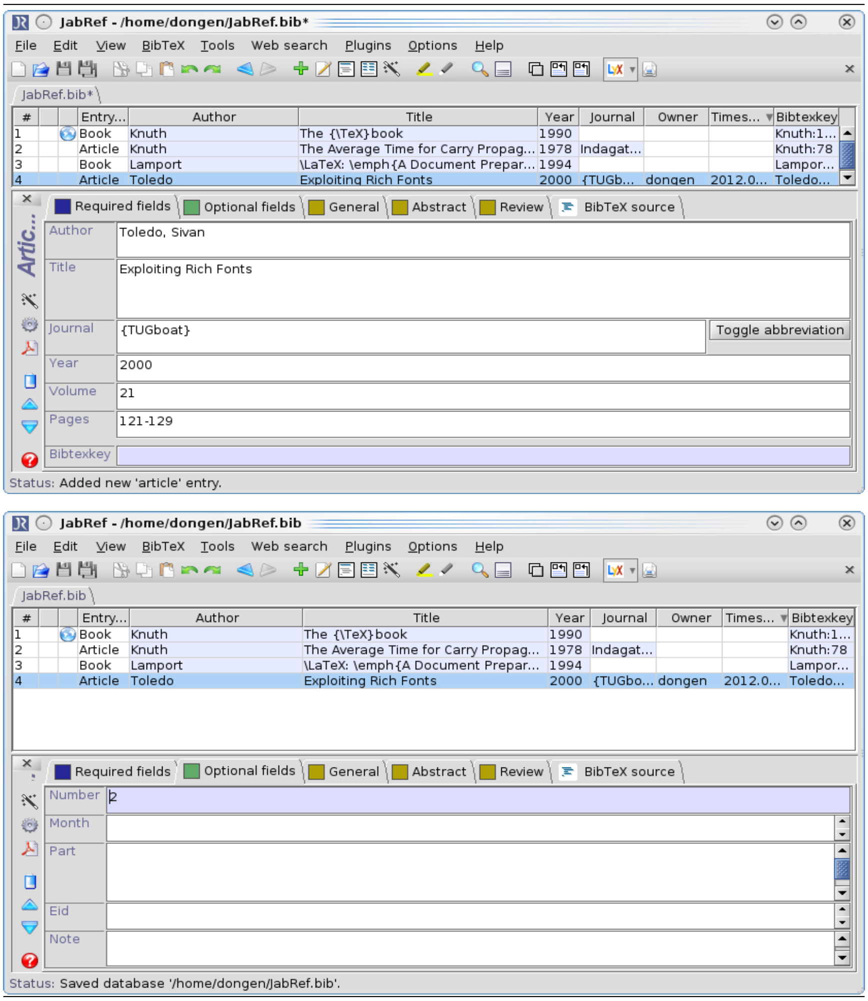
Status: Opened database '/home/dongen/JabRef.bib' with 3 entries.

#	Entry...	Author	Title	Year	Journal	Bibtexkey	Owner	Times...
1	Book	Knuth	The {\TeX}book	1990		Knuth:1...		
2	Book	Lamport	\LaTeX: A Document Preparation Sy...	1994		Lampor...		
3	Article	Knuth	The Average Time for Carry Propag...	1978	Indagat...	Knuth:78		

Status: Opened database '/home/dongen/JabRef.bib' with 3 entries.

Figure 18.8

JabRef IDE. The picture on the top shows the original IDE. In the picture on the bottom, the Bibtexkey column is dragged to just before the Owner column.



#	Entry...	Author	Title	Year	Journal	Owner	Times...	Bibtexkey
1	Book	Knuth	The {\TeX}book	1990				Knuth:1...
2	Article	Knuth	The Average Time for Carry Propag...	1978	Indagat...			Knuth:78
3	Book	Lamport	\LaTeX: \emph{A Document Prepar...	1994				Lampor...
4	Article	Toledo	Exploiting Rich Fonts	2000	{TUGb...	dongen	2012.0...	Toledo...

Author: Toledo, Sivan
Title: Exploiting Rich Fonts
Journal: {TUGboat}
Year: 2000
Volume: 21
Pages: 121-129
Bibtexkey:

Status: Added new 'article' entry.

JabRef - /home/dongen/JabRef.bib
File Edit View BibTeX Tools Web search Plugins Options Help
JabRef.bib
Required fields
Optional fields
General
Abstract
Review
BibTeX source

#	Entry...	Author	Title	Year	Journal	Owner	Times...	Bibtexkey
1	Book	Knuth	The {\TeX}book	1990				Knuth:1...
2	Article	Knuth	The Average Time for Carry Propag...	1978	Indagat...			Knuth:78
3	Book	Lamport	\LaTeX: \emph{A Document Prepar...	1994				Lampor...
4	Article	Toledo	Exploiting Rich Fonts	2000	{TUGbo...	dongen	2012.0...	Toledo...

Number: 2
Month:
Part:
Eid:
Note:

Status: Saved database '/home/dongen/JabRef.bib'.

Figure 18.9

Creating a new BibTEX entry. The figure at the top shows the Required fieds tab with filled out text fields for the required entries of an article. The figure on the bottom shows the Optional fields tab with the filled out text field for the number of the journal.

Installing class and package files is impossible unless you know a little bit about how TEX Live organises its files, and how TEX Live executables locate these files. The first three sections provide a quick introduction to the required background. The last two sections explain how to install class and package files. Throughout, the word package is a synonym for class or package.

18.8.1 *The TEX Directory Structure*

TEX Live distributions are huge and with the current technology of operating systems it is impossible to quickly locate a file that is located underneath some directory. What is more, TEX Live distributions cannot guarantee unique file names. For example, an auxiliary file called `aux.tex` may occur in more than one directory.

The TUG Working Group on a TEX Directory Structure TWG-TDS [2004] defined a TDS, which is specifically designed to overcome problems related to the efficient location of files in a TEX distribution. TEX Live has adopted the TDS and many classes and packages assume an underlying TDS.

Note that a TDS is a property of a directory and several directories may have a TDS. For example, if you accept all defaults when you install TEX Live on `Unix`, then the global TEX Live files are in the directory `/usr/local/texlive/⟨year⟩` , where ⟨`year`⟩ is the year of the TEX Live distribution. At the directory's root is a large directory called `texmf-dist`, which has *a* TDS. Effectively, this is where TEX Live keeps most of its class and package files. Likewise, TEX Live assumes the existence of a user directory called `~/texmf` , which should also have a TDS.

Within a TDS-compliant directory, each type of file is expected in a specific place. Figure 18.10 depicts some of the aforementioned TDS `texmf-dist`. At the root level the figure shows the directories `bibtex`, `doc`, `fonts`, and `tex`. As suggested by the names, the directory `bibtex` is for BIBTEX related files, `doc` is for documentation, `fonts` is for font files, and `tex` is for TEX files. LATEX files should be put in the directory lstex/latex, which should have a subdirectory for each package. The figure shows two such package directories: `siunitx` and `url`. The main purpose of each package directory is storing "its" package files.

18.8.2 *Updating an `ls-R` Database*

When a TEX Live executable such as `pdflatex` has to locate a package, it uses a library called `kpathsea` [Berry, Weber, and Hoekwater 2011], which uses several different techniques to locate the package. One of these techniques is looking up the package location in a plain-text `ls-R` *database,* of which there may be several.

The name of an `ls-R` database is always the same: `ls-R`. Because the name of each `ls-R` database is unique, there can be no more than one `ls-R` database in a given directory. We say that an `ls-R` database is *rooted* in the directory that it is in. Likewise, we say that a directory is

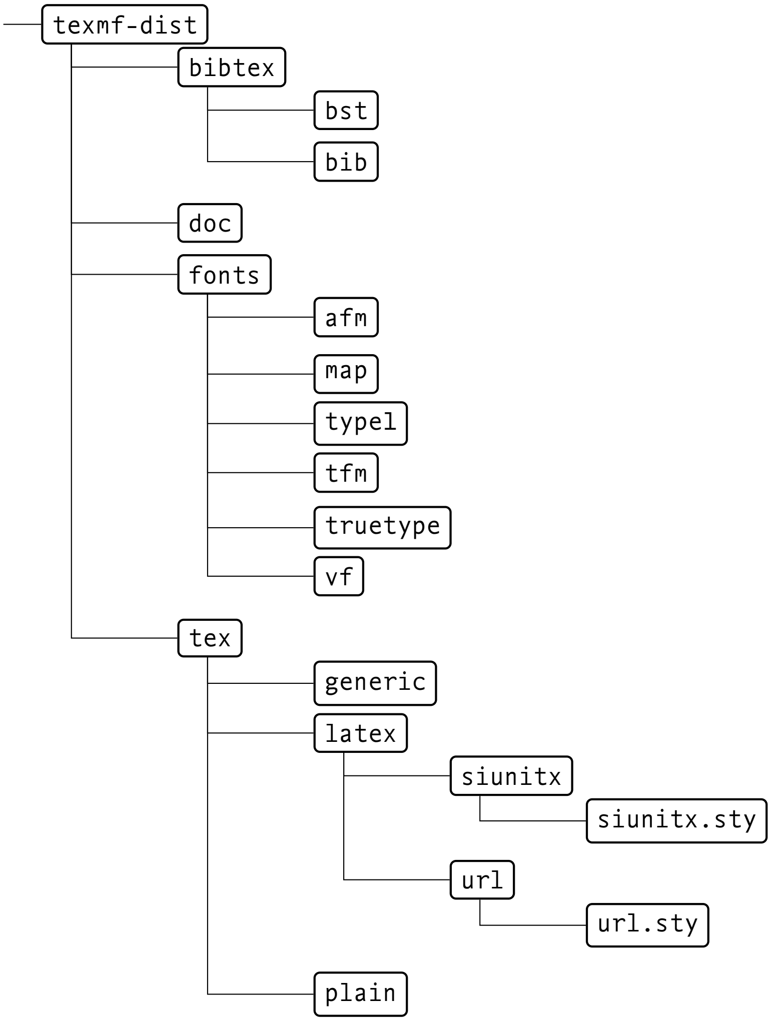


Figure 18.10

TeX directory structure (TDS). The directory `texmf-dist` at the root of the depicted directory tree is where TeX Live keeps its files. The directories inside `texmf-dist`, form a hierarchical structure that is common to other TDS-compliant directories. At the top level the directories `bibtex`, `doc`, `fonts`, and `tex` have a prescribed structure. For example, the directory `texmf-dist/tex/latex` has one directory for each standard installed package. The package files are then organised inside their directory. For example, the `url` package file is located in the file `tex/latex/url/url.sty` inside `texmf-dist`. The files for the `siunitx` package are located in the subdirectory `tex/latex/siunitx/siunitx.sty`.

the *root* of its `ls-R` database. The main purpose of an `ls-R` database is to store the locations of classes, packages, and other files in the database's root directory and descendant directories.

Remember that you may install a package with `tlmgr` or by hand. When `tlmgr` installs a package, `tlmgr` updates all the relevant `ls-R` databases. When a package is installed in one of these `ls-R` directories (or one of its descendant directories), this usually requires updating the `ls-R` database. You may do this by executing the following command.

Unix Usage

```
# texhash ⟨root directory⟩
```

Here ⟨`root directory`⟩ is the root directory of the `ls-R` database. If `root directory` is large, running the command may take a while. Note that you need write access to the database, so you may need administrator access to do this.

18.8.3 *How `kpathsea` Finds its Files*

As explained earlier on in this chapter, TeX Live executables delegate the task of locating classes, packages, and other files to the `kpatshea` library [Berry, Weber, and Hoekwater 2011]. This section explains this process in more detail. To simplify the presentation, this section only discusses how `kpathsea` locates package files. The process for LaTeX (`.tex`), class (`.cls`), and other types of files is similar [Berry, Weber, and Hoekwater 2011].

When `pdflatex` formally asks `kpathsea` for the location of a given package called ⟨`package`⟩, the library starts by constructing a *search path*. Berry, Weber, and Hoekwater [2011, section 3.2] explains how the search path is constructed.

A `kpathsea` search path is a list of colon-separated *path elements*. Each path element consists of a single annotated directory. The annotation determines how the directory should be searched for ⟨`package`⟩.

As expected, `kpathsea` processes the path elements in the search path from left to right until it finds the location of ⟨`package`⟩ or fails.

1. When `kpathsea` processes the next path element, it starts by extracting the directory, ⟨`directory`⟩, from the path element.
2. If one of the prefixes of ⟨`directory`⟩ is the root of an `ls-R` database, then `kpathsea` tries to look up the location of ⟨`package`⟩ in that `ls-R` database [Berry, Weber, and Hoekwater 2011, page 27]. If this operation succeeds, then `kpathsea` returns the resulting location. (Berry, Weber, and Hoekwater [2011] implies that there should be no more than one `ls-R` database along any path leading from any directory to any leaf directory. Furthermore, the name of the `ls-R` database should be in the internal `kpathsea` variable `TEXMFDBS`. This was confirmed by Karl Berry in private communication.)
3. Otherwise, `kpathsea` considers the annotation of the path element.
4. If the path element is of the form `!!`⟨`directory`⟩, `kpathsea` stops processing the path element.
5. If the path element is of the form ⟨`directory`⟩`//`, `kpathsea` completes processing the path element by searching ⟨`directory`⟩ recursively for ⟨`package`⟩.
6. If the path element is of the form ⟨`directory`⟩, `kpathsea` completes processing the path element by searching ⟨`directory`⟩ without recursion. It is assumed the directory has a TDS.
7. (Other path element forms are not allowed.)

The following command outputs the `kathsea` search path for `pdflatex` and style files.

`Unix` Usage

```
# kpsewhich -progname pdflatex -show-path .sty
```

When you run this, many of the path elements in the output have directories that are annotated with the `!!` prefix. If you install a new package in one of these directories (or descendant directories thereof) then the `ls-R` database in that directory *must* be updated. If you forget updating the database, the package may never be found.

18.8.4 *Installing Packages with `tlmgr`*

As already mentioned, the `tlmgr` program is provided as part of TeX Live. The main purpose of `tlmgr` is updating and installing packages and classes that are part of TeX Live. The program can also update itself. You don't need administrator access to use the program.

If you want to install a class or package that is part of TeX Live then installing them with `tlmgr` is by far the easiest. Not only will `tlmgr` install the class or package for you but it will also update any relevant `ls-R` database. The following is how you install classes and packages with `tlmgr` from the command line. There is also a `tlmgr` GUI but this is not explained here.

`tlmgr install` ⟨`names`⟩

This installs the classes and packages in the list ⟨`names`⟩. There is no need to add extensions in ⟨`names`⟩. ☑

There is also a `tlmgr` GUI but this is not explained here.

The following are some other useful `kpathsea` options for day-to-day use.

`tlmgr -help`

This provides a short explanation of how to use `tlmgr`. For complete information, visit `http://tug.org/texlive/doc/tlmgr.html`. ☑

`tlmgr update` ⟨`names`⟩

This updates the classes and packages in the list ⟨`names`⟩. There is no need to add extensions in ⟨`names`⟩. ☑

`tlmgr update --all`

This updates all classes and packages. ☑

`tlmgr update --self`

This updates `tlmgr` itself. This option may be required if your TeX Live distribution is too old. ☑

`tlmgr update --self --all`

This updates `tlmgr` and all classes and packages. ☑

The `tlmgr` program can also make backups, restore backups, and lots of other things. The package documentation at `http://tug.org/texlive/doc/tlmgr.html` provides further information.

18.8.5 *Installing Packages by Hand*

The previous section showed that installing packages with `tlmgr` is easy. Unfortunately, not all packages are supported by TeX Live, some have to be installed by hand.

Installing classes and packages by hand requires a bit more work than installing with `tlmgr`. Fortunately, it is not much more difficult if you install them in your *local `textmf` tree*—a TDS that is reserved for user-defined class and package files [Berry 2011, page 11].

You may get the location of your (local) `texmf` tree by executing the command `kpsewhich var-value=TEXMFHOME` from the command line. By default TeX Live uses `~/texmf` for your `texmf` tree.

Remember from the second last section that the command `kpsewhich -progname pdflatex -show-path .sty` shows the search path for package files. If TeX Live is installed with all the defaults, this command should list your local `texmf` tree in a path element that is suffixed with `//` annotation. Recall that the `//` suffix means that `kpathsea` searches the directory recursively. If all you have to do is installing a single file then creating a subdirectory for the package in your local `texmf` tree and putting the file in that directory should be enough. If the package you're installing is based on a TDS then unpacking the files in the root of your local `texmf` tree should suffice. After this you may, but this is not required update the `ls-R` database at the root of your local `texmf` tree. (Remember that you may do this with the command `texhash`.)

Unfortunately, not all packages are provided as a TDS and you may have to uncompress the files, build the package documentation, move all files to their destination directories, and update the `ls-R` database. The following shows how to do this for a package called ⟨`package`⟩.

1. Create a temporary directory. Download, save, and if necessary extract the package files in this temporary directory.
2. Usually there is a `.ins` file. If there is, run `pdflatex` on it.
3. If there is a `.dtx` file then you must create the class or package documentation. You do this by running `pdflatex` once or twice on the `.dtx` file.
4. What remains is moving the files to the correct location in the local `texmf` tree. In the following it is assumed the local `texmf` tree is in the default TeX Live location: `~/texmf`. Start by creating the destination directories for the LaTeX files and for the documentation.

   Unix Usage

   ```
   # mkdir -p ~/texmf/tex/doc/⟨package⟩
   # mkdir -p ~/texmf/tex/latex/⟨package⟩
   ```
5. Move all LaTeX files to the proper destination.

   Unix Usage

   ```
   # mv *.tex *.sty *.cls *.fd ~/texmf/tex/latex/⟨package⟩
   ```
6. Move all documentation files to the proper destination.

   Unix Usage

   ```
   # mv *.dvi *.pdf *.ps ~/texmf/tex/doc/⟨package⟩
   ```
7. If the installation involves fonts then you also have to move the font-related files to their proper destination. Table 18.1 lists the destination directories for the font files. The name ⟨`vendor`⟩ is the name of the vendor/supplier of the font and ⟨`font`⟩ is the name of the font. Some vendor directory names are standardised [Berry 1990] but any name is allowed.
8. At this stage it's safe to remove your temporary directory.
9. Complete the installation by updating the `ls-R` database in your local `texmf` directory.

Extension	Destination in `~/texmf/fonts`
`.afm`	`afm/⟨vendor⟩/⟨font name⟩`
`.map`	`map/pdftex/⟨vendor⟩`
`.pfb`	`type1/⟨vendor⟩/⟨font name⟩`
`.tfm`	`tfm/⟨vendor⟩/⟨font name⟩`
`.ttf`	`truetype/⟨vendor⟩/⟨font name⟩`
`.vf`	`vf/⟨vendor⟩/⟨font name⟩`

Table 18.1

TeX Live font directories. The first column lists the extension of the font files. The second column lists the destination directory for these files. The destination is relative to `~/texmf/fonts` .

Unix Usage

```
# texhash ~/texmf
```

PART VII

References and Bibliography

Picture omitted to keep size of `.pdf` small.

Oil and charcoal on canvas (08/10/10), 64 × 91 cm
Work included courtesy of Billy Foley

Typographic Jargon

This is a short reference of typographic jargon. The presentation is not top-down but lists the jargon in alphabetical order. More detailed information may be found in good typography books such as, for example, [Felici 2012].

baseline The baseline is the virtual reference line that the characters are written on. Parts of some of the letters may stick out below the baseline. For example, letters like *g*, *j*, and *p* stick out below the baseline. Figure 18.11 provides an example.

bounding box The bounding box of a character is the virtual rectangle containing the character. Each box has a reference point, a height, a width, and a depth. See Figure 18.12 for an example. The boxes are designed to combine their characters on the baseline by chaining their boxes. This explains why some characters are not contained entirely by their bounding boxes. Figure 18.13 shows how the boxes from Figure 18.12 are combined.

character A character is a member of a typeface.

em The relative length unit em is a synonym for the current type size.

em space An em space is a horizontal space with a length of 1 em.

en An en is a relative length unit that is equal to half an em.

font A computer font is the software that defines a typeface.

font family A font family is a collection of fonts that define a typeface family.

glyph A gyph is a certain form of a character. Different characters usually have different glyphs but the converse is not true. For example, the letter *a* has several different glyphs: a, A, a, A, *a*, *A*, a, A, *a*, *a*, and so on.

italic typeface An italic typeface is a slanted typeface that is usually based on calligraphic handwriting.

kerning Kerning refers to adjusting the space between adjacent glyph pairs that are too close or too far apart. Kerning is needed because the bounding box of a glyph may not always be compatible with the bounding box of any other glyph. When this happens kerning adjusts the position of the glyphs by moving the relative position of their bounding boxes. Figure 18.15 shows how kerning works. In this example kerning reduces the distance of the glyph pairs but kerning may also increase the distance. Kerning may also move glyphs up or down.

Figure 18.11
Baseline and mean line. The solid line is the baseline, which the letters/glyphs are written on. The dashed line is the mean line. The mean line limits the height of the nonascending lowercase letters. Uppercase letters and letters such as *f*, *h*, and *k* stick out above the mean line. Letters such as *g* and *p* stick out below the baseline. The distance between baseline and mean line is the x-height.

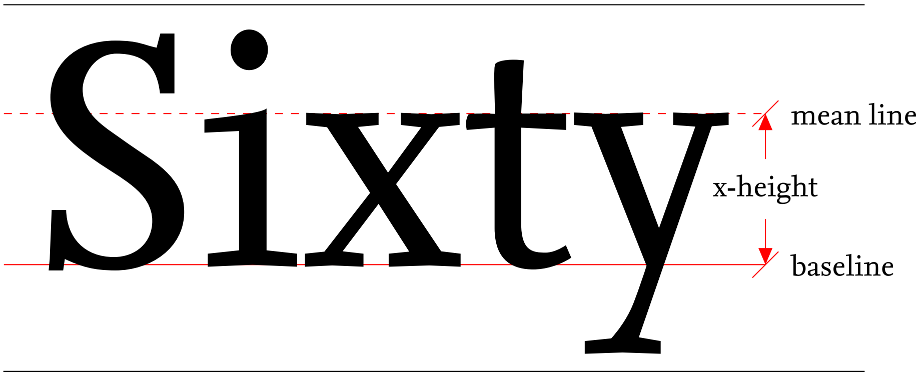


Figure 18.12
Bounding boxes. Each letter is contained in its bounding box, which is drawn with a dashed line. Words are created by sequencing letter boxes (see Figure 18.13). Letters may extend beyond the box outline if this makes it easier to sequence the boxes. The circle on the baseline is the box's reference point. Each box has a height, a width, and a depth. The height (depth) measures the part above (below) the baseline.

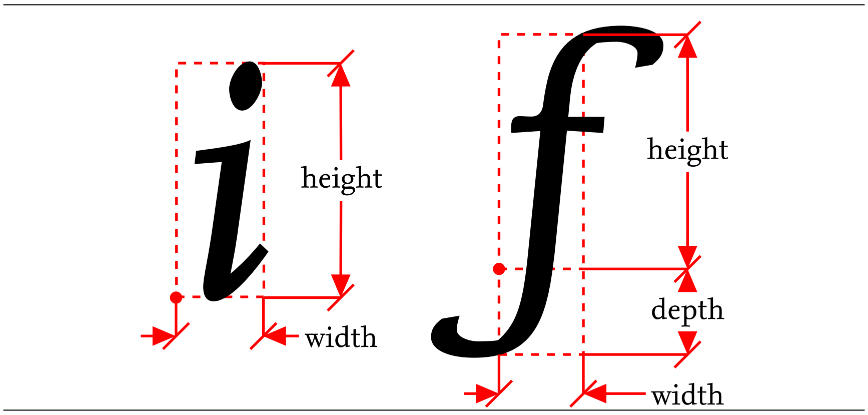


Figure 18.13
Word formation. The word 'if' is formed by sequencing the bounding boxes of the letters *i* and *f*. The right side of the bounding box of the *i* touches the left side of the bounding box of the *f* and the reference point of the bounding boxes are on the same horizontal line.

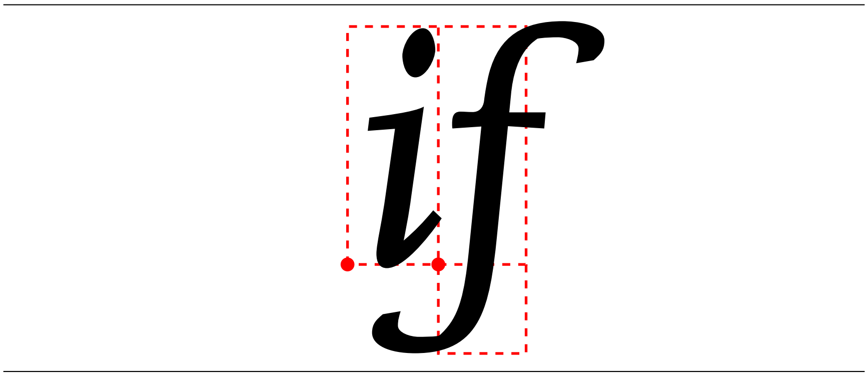
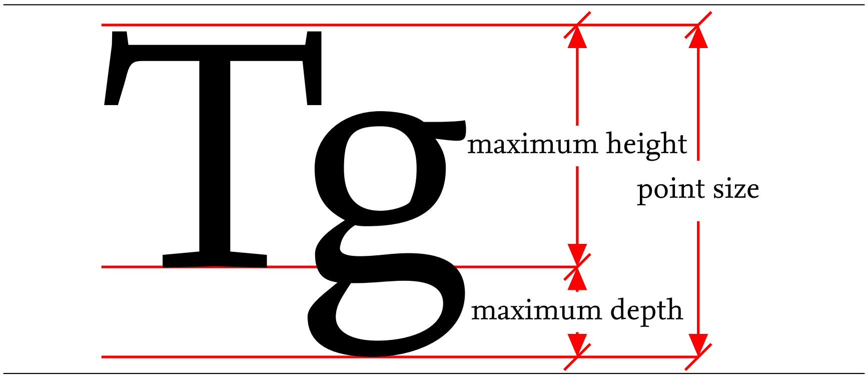


Figure 18.14
Computing the point size. The point size is determined by the sum of the largest possible height and the largest possible depth of the letters in the font at the current size.



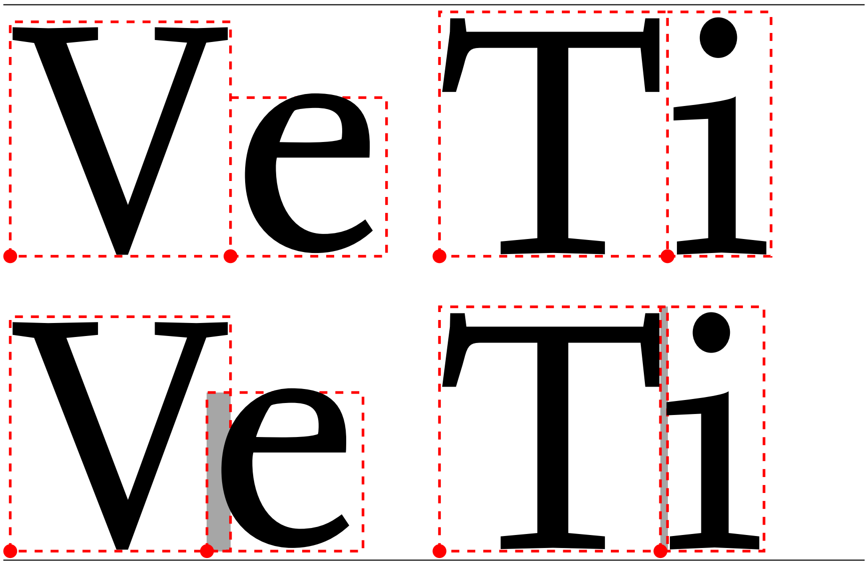


Figure 18.15
Kerning. The distance between the letters in the pairs 'Ve' and 'Ti' on the first line is too large. In the second line this is resolved by adjusting the distances within the pairs. This is called kerning. In this example, kerning reduces the distance between the reference points of the bounding boxes, which makes the bounding boxes overlap. The overlap is shown in grey.

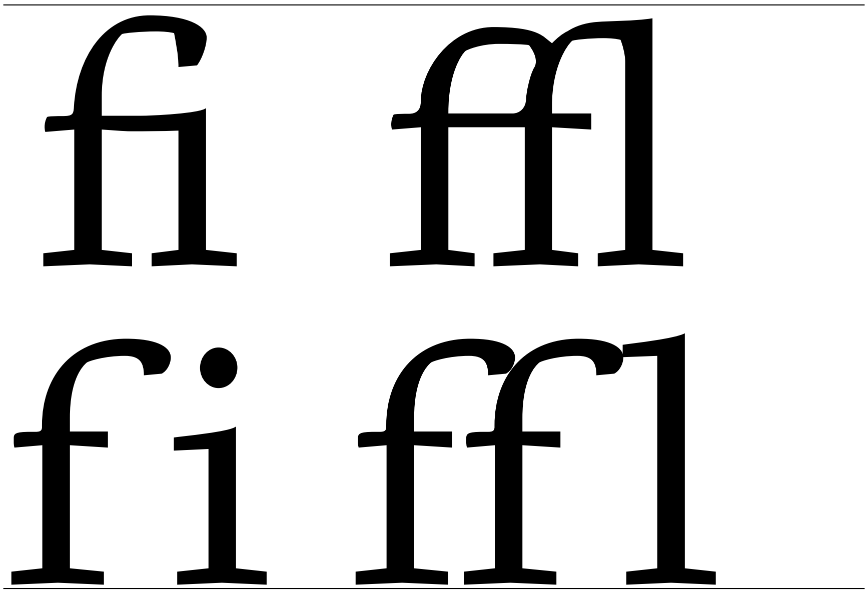


Figure 18.16
Ligatures. This figure shows two example of ligatures. At the top are the ligatures and at the bottom the letters that form the ligatures.

leading The leading or line spacing is the distance between the baselines of two subsequent lines in the text.

letterspacing Letterspacing or tracking refers to the uniform amount of space that is added to the left and right of the characters in a passage of text.

ligature A ligature is a glyph that represents a combination of two or more individual letters, digits, or punctuation marks. Figure 18.16 shows two examples of ligatures.

line spacing See leading.

mean line The mean line is the virtual demarcation line for the height of the lowercase letters without ascenders. See Figure 18.11 for an example.

oblique typeface See slanted typeface.

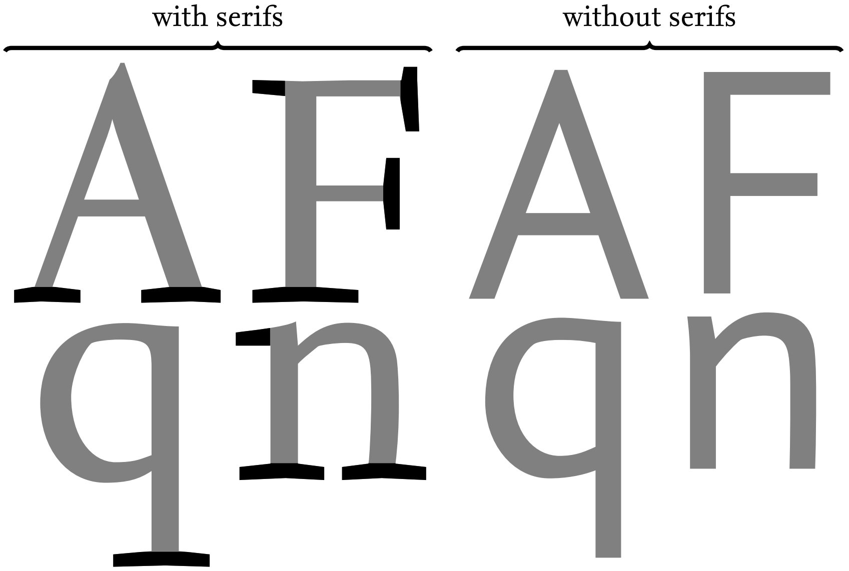


Figure 18.17 Seriffed versus sans serif letters. The letters on the left have serifs, which are drawn in black. The rest of the letters is drawn in grey. The letters on the right lack serifs.

roman typeface A roman or upright typeface is a typeface whose glyphs have an upright shape. Usually the main text of papers and books is typeset in a roman typeface.

slanted typeface A slanted or oblique typeface is a typeface whose glyphs are slanted to the right. The resulting glyphs may be obtained by mathematically transforming the roman glyphs. However, some typefaces have a different design.

point A TeX point is an absolute length unit that is equivalent to 1⁄72.27 inch. The point unit is denoted pt, so 72.27 pt is equal to 1 inch.

point size The point size or type size is a relative length unit. The point size depends on the font and current size. You get the point size of the font (at the given size) by taking the letters of the font and adding the largest height and the largest depth of the bounding boxes of the letters. Figure 18.14 illustrates the definition.

pt See point.

quad A LaTeX quad is a relative horizontal length unit that is equivalent to 1 em [Knuth 1990, page 314].

sans serif typeface A sans serif typeface is a typeface whose glyphs lack serifs. See Figure 18.17 for an example.

small caps Small caps letters have a similar shape as uppercase letters but they aren't as high so they blend in better with lowercase letters: abcd...xyx ABCD...XYZ.

serif A serif is a small decoration at the end of the strokes of some of the glyphs. See Figure 18.17 for an example.

seriffed typeface A seriffed typeface is a typeface (some of) whose characters have serifs.

thin space A thin space is usually set to ¼ em or about half the word space [Felici 2012]. In LaTeX a thin space is normally ⅙ em [Knuth 1990, page 167].

tracking See letterspacing.

type The word type literally means form and refers to the metal blocks that were used for printing. Nowadays it refers to printable characters.

typeface A typeface is a collection of letters, digits, punctuation symbols, and other characters that share a common design and common features.

typeface family A typeface family is a collection of typefaces that share some features. Many typeface families come with a roman shape, an italic shape, a slanted shape, and so on. Examples of typeface families are *Computer Modern Roman, Computer Modern Sans Serif, Gentium, Linux Libertine Serif/Times New Roman, TeX Gyre Heros/Helvetica, TeX Gyre Pagella/Palatino Linotype,* and so on.

type size See point size.

typewriter typeface A typewriter or monospaced typeface is a typeface whose glyphs have equal width.

x-height The x-height is the distance between the mean line and the baseline. By design the distance coincides with the height of the lowercase *x*. See Figure 18.11 for an example.

Bibliography

Abrahams, P. A., K. A. Hargreaves, and K. Berry [2003]. *TeX for the Impatient.* Addison–Wesley. URL: ftp://tug.org/tex/impatient.

Allen, Robert [2001]. *Punctuation.* Oxford University Press. ISBN: 0-19-860439-4.

American Mathematical Society [5th Feb. 2002]. *User's Guide for the amsmath Package.* Version 2.0.

——— [5th Feb. 2002]. *Using the amsthm Package.* Version 2.20.

Arnold, Doug [25th Jan. 2010]. *The LaTeX for Linguists Home Page.* URL: http://www.essex.ac.uk/linguistics/external/clmt/latex4ling/.

Arseneau, Donald [20th Jan. 2010]. *url.sty.* Version 3.3. Edited as a LaTeX document by Robin Fairbairns.

Aslaksen, Helmer [1993]. "Ten TeX Tricks for the Mathematician". In: *TUGboat* 14. A modern version of this paper is available from http://www.math.nus.edu.sg/aslaksen/cs/tug-update.pdf, 135–136.

Bazargan, Kaveh, and C. V. Radhakrishnan [2007]. "Removing vertical stretch—mimicking traditional typesetting with TeX". In: *TUGboat* 28.1, 133–136. URL: http://www.tug.org/TUGboat/tb28-1/tb88bazargan.pdf.

Beccari, Claudio [1997]. "Typesetting Mathematics for Science and Technology According to ISO 31/XI". In: *TUGboat* 18.1, 39–48. URL: http://www.tug.org/TUGboat/tb18-1/tb54becc.pdf.

Becker, T., and V. Weispfenning [1993]. *Gröbner Bases* A Computational Approach to Commutative Algebra. Graduate Texts in Mathematics. Springer. ISBN: 3-540-97971-9.

Berry, Karl [1990]. "Filenames for Fonts". In: *TUGboat* 11.4.

——— ed. [1st June 2011]. *The TeX Live Guide* TeX Live 2011. URL: http://tug.org/texlive.

Berry, Karl, Olaf Weber, and Taco Hoekwater [1st Apr. 2011]. *Kpathsea Library.* Version 6.0.1. URL: htpp://tug.org/.kpathsea.

Bigwood, Sally, and Melissa Spore [2003]. *Presenting Numbers, Tables, and Charts.* Oxford University Press. ISBN: 0-19-860722-9.

Borowski, Eprahim J., and Jonathan M. Borwein [2005]. *Collins Dictionary of Mathematics.* Collins. ISBN: 0-00-780080-0.

Bovani, Michel [30th Jan. 2005]. *Fourier GUTenberg.*

Breitenbucher, Jon [2005]. “LaTeX at a Liberal Arts College”. In: *PracTeX Journal* 3. URL: http://tug.org/pracjourn/2005-3/breitenbucher.

Bringhurst, Robert [2008]. *The Elements of Typographic Style.* Hartley & Marks. ISBN: 0-988179-206-3.

Buchsbaum, Arthur, and Francisco Reinaldo [2007]. “A Tool for Logicians”. In: *The PracTeX Journal* 3. URL: http://tug.org/pracjourn/2007-3/buchsbaum.

Burt, John [2005]. “Using poemscol for Critical Editions of Poetry”. In: *PracTeX Journal* 3. URL: http://tug.org/pracjourn/2005-3/burt.

Carlisle, David [5th Mar. 1999a]. *The enumerate Package.* Version 3.00.

——— [16th Mar. 1999b]. *The keyval Package.* Version 1.13.

——— [28th May 2001]. *The dcolumn Package.* Version 1.06.

Carlisle, D. P. [14th Nov. 2005]. *Packages in the graphics Bundle.*

Carlisle, D. P., and S. P. Q. Rahtz [16th Feb. 1999]. *The graphicx Package.* Version 1.0f.

Clapham, Christopher, and James Nicholson [2005]. *Oxford Concise Dictionary of Mathematics.* Third Edition. Oxford University Press. ISBN: 978-0-19-860742-7.

Dashboard, Humanities Division [8th May 2008]. *LaTeX—Humanities Computing.* URL: https://coral.uchicago.edu:8443/display/humcomp/LaTeX.

Dearborn, Elizabeth [2006]. “TeX and Medicine”. In: *PracTeX Journal* 4. URL: http://tug.org/pracjourn/2006-4/dearborn.

Eijkhout, V. [2007]. *TeX by Topic, A TeXnician’s Reference.* Addison–Wesley. ISBN: 0-201-56882-9. URL: http://www.eijkhout.net/tbt/.

Felici, James W. [2012]. *The Complete Manual of Typography* A Guide to Setting Complete Type. Second Edition. Adobe Press. ISBN: 978-0-321-77326-5.

Fenn, Jürgen [2006]. “Managing Citations and Your Bibliography with BibTeX”. In: *PracTeX Journal* 4. URL: http://www.tug.org/pracjourn/2006-4/fenn/.

Feuersänger, Christian [5th Aug. 2010a]. *Manual for Package* PGFPLOTS. Version 1.4.1.

——— [5th Aug. 2010b]. *Manual for Package* PGFPLOTSTABLE. Version 1.4.1.

Fine, Jonathan [1992]. “Some Basic Control Macros for TeX”. In: *TUGboat* 13.1.

Fiorio, Christophe [14th Dec. 2004]. *algorithm2e.sty—package for algorithms.* Version 3.3.

Flynn, Peter [2007]. “Rolling your own Document Class: Using LaTeX to keep away from the Dark Side”. In: *TUGboat* 28.1. URL: http://www.tug.org/TUGboat/Articles/tb28-1/tb88flynn.pdf.

Garcia, Aracele, and Arthur Buchsbaum [2010]. “About LaTeX tools that students of Logic should know”. In: *PracTeX Journal* 1. In Portugese. URL: http://tug.org/pracjourn/2010-1/garcia.

Giacomelli, Roberto [12th July 2009]. *calctab Package.* Version 0.6.1.

Goldberg, Jeffrey P., and H.-Martin Münch [1st Feb. 2011]. *The lastpage Package.* Version 1.2g.

Goossens, M., S. Rahtz, and F. Mittelbach [1997]. *The LaTeX Graphics Companion: Illustrating documents with TeX and PostScript.* Addison–Wesley. ISBN: 0-201-85469-4.

Graham, R. L., Donald E. Knuth, and O. Patashnik [1989]. *Concrete Mathematics: A Foundation for Computer Science.* Addison-Wesley. ISBN: 0-201-14236-8.

Hedrick, Charles [15th Jan. 2003]. *Guidelines for Typography in* NBCS. URL: `http://www.nbcs.rutgers.edu/~hedrick/typography/typography.janson-syntax.107514.pdf`.

Heinz, Carsten, and Brooks Moses [22nd Feb. 2007]. *The `Listings` Package.* Version 1.4.

Heldoorn, Marcel [2nd Dec. 2007]. *The `SIunits`—package* Consistent application of SI Units. Version 1.36.

Høgholm, Morten [28th July 2008]. *The `breqn` Package.* Version 0.98.

——— [17th Jan. 2011]. *The `xfrac` Package.* Version 0.3b.

Høgholm, Morten et al. [12th Feb. 2011]. *The `mattools` Package.* Version 1.10.

Kern, Uwe [21st Jan. 2007]. *Extending LaTeX's color facilities: `xcolor`.* Version 2.11.

Knuth, Donald E. [1990]. *The TeXbook.* Addison–Wesley. ISBN: 0-201-13447-0. URL: `http://www.ctan.org/tex-archive/systems/knuth/tex/`.

Kohler, Eddie [2011]. *`otftotfm` Manual.* Version 2.90. URL: `http://www.lcdf.org/type/otftotfm.1.html`.

Krab Thorub, Kresten, Frank Jensen, and Chris Rowley [22nd Aug. 2007]. *The `calc` Package,* Infix Notation Arithmetic in LaTeX.

Lamport, Leslie [1994]. *LaTeX: A Document Preparation System.* Addison–Wesley. ISBN: 0-021-52983-1.

LaTeX3 Project [12th Mar. 1999]. *LaTeX2ε for Class and Package Writers.* URL: `http://www.latex-project.org/guides/clsguide.pdf`.

LaTeX3 Project [9th Feb. 2000]. *LaTeX2ε Font Selection.* URL: `http://www.latex-project.org/guides/fntguide.pdf`.

Lehman, Philipp. *The `biblatex` Package.* Version 1.5.

——— [2004]. *The Font Installation Guide* Using Postscript Fonts to their Full Potential in LaTeX. Version 2.14.

Longborough, Brent [28th Aug. 2011]. *`tagging.sty`.* Version 1.0.

Midtiby, Henrik Skov [21st Apr. 2011]. *The `todonotes` Package.*

Miede, André [24th Jan. 2010]. *The Classic Thesis Style.*

Mittelbach, Frank, Robin Fairbairns, and Werner Lemberg [6th Jan. 2006]. *LaTeX Font Encodings.*

Mühlich, Matthias [23rd Feb. 2006]. *The `CoverPage` Package.* Version 1.01.

Oberdiek, Heiko [23rd Dec. 2010]. *The `keyvalue` Package.* Version 3.10.

Ochsenmeier, Erwin [2011]. *`FourSenses.net`.* At the moment of writing, this site is being moved from `http://theotex.blogspot.com/`. URL: `http://en.foursenses.net/`.

Oetiker, Thomas [8th Sept. 2010]. *An Acronym Environment for LaTeX2ε.* Version 1.36.

Oetiker, Tobias et al. [2007]. *The Not so Short Introduction to LaTeX2ε*. URL: `http://tobi.oetiker.ch/lshort/`.

Owens, John D. [2006]. "The Installation and use of OpenType Fonts in LaTeX". In: *TUGboat* 27.2, 112–119.

Pakin, Scot [2005]. *The Comprehensive LaTeX Symbol List*. URL: `http://tug.ctan.org/info/symbols/comprehensive/symbols-letter.pdf`.

Patieri, Lórenzo [2010]. *Customizing `classicthesis` with the `arsclassica` Package*.

Peyton Jones, S., and J. Hughes [1999]. `Haskell 98`*: A Non-strict, Purely Functional Language*. `http://www.haskell.org/onlinereport/`.

Peyton Jones, Simon L., John Huges, and John Launchberry [1993]. "How to Give a Good Research Talk". In: *ACM SIGPLAN Notices* 28.11, 9–12. URL: `http://research.microsoft.com/en-us/um/people/simonpj/papers/giving-a-talk/giving-a-talk.htm`.

Rahts, Sebastian [1993]. "Essential NFSS2, Version 2". In: *TUGboat* 14.2, 132–137.

Reckdahl, Keith [2006]. *Using Imported Graphics in LaTeX and pdfLaTeX*. URL: `ftp://ftp.tex.ac.uk/tex-archive/info/epslatex.pdf`.

Reichert, Axel [4th Aug. 1998]. *`units.sty`—`nicefrac.sty`*. Version 0.9b.

Robertson, Will [26th Feb. 2011]. *The `fontspec` Package*. Version 2.1f.

Scharrer, Martin [2011-07-23]. *List of Internal LaTeX2ε Macros useful to Package Authors*. Version 0.4.

Schlicht, R. [10th Jan. 2010]. *The `microtype` Class*. Version 2.4.

Schneider, Tom [27th Apr. 2011]. *LaTeX Style and BibTeX Bibliography Formats for Biologists: TeX and LaTeX Resources*. URL: `http://www-lmmb.ncifcrf.gov/~toms/latex.html`.

Senthil, Kumar M. [2007]. "LaTeX Tools for Life Scientists (BioTeXniques?)" In: *PracTeX Journal* 4. URL: `http://tug.org/pracjourn/2007-4/senthil`.

Smith, Peter. *LaTeX for Logicians*. URL: `http://www.logicmatters.net/latex-for-logicians/`.

Strunk, W., and E. B. White [2000]. *The Elements of Style*. Fourth Edition. Macmillan Publishing. ISBN: 0-205-30902-x.

Talbot, Nicola [15th Nov. 2009]. *`datatool` v 1.01: Database and data manipulation*. Version 2.03.

Tanksly, Charlie. *LaTeX: A Guide for Philosophers*. URL: `http://www.charlietanksley.net/latex/`.

——— *PhilTeX Forums* A Place for Philosophers to Learn about LaTeX. URL: `http://www.charlietanksley.net/philtex/forum/`.

Tantau, Till [25th Oct. 2010]. *TikZ & PGF*. Version 2.00-cvs.

Tantau, Till, Joseph Wright, and Vedran Miletić [12th July 2010]. *The* BEAMER *Class*. Version 3.10.

Taraborelli, Dario [2010]. *The Beauty of LaTeX*. URL: `http://nitens.org/taraborelli/latex`.

Team, `JabRef` Development. *`JabRef`*.

Tellechea, Christian [8th Apr. 2011]. *`spreadtab`*. Version 0.3c.

TWG-TDS [23rd June 2004]. *A Directory Structure for TeX Files*. Version 1.1. URL: `http://tug.org/tds/tds.pdf`.

Thomson, Paul A. [2008a]. “Clinical Trials Management on the Internet—I. Using LaTeX and SAS to Produce Customized Forms”. In: *PracTeX Journal* 3.

——— [2008b]. “Clinical Trials Management on the Internet—II. Using LaTeX, PostScript, and SAS to Produce Barcode Label sheets”. In: *PracTeX Journal* 3. URL: `http://tug.org/pracjourn/2008-3/thompson2`.

Toledo, Sivan [2000]. “Exploiting Rich Fonts”. In: *TUGboat* 21.2, 121–129.

Trask, R. L. [1997]. *Penguin Guide to Punctuation.* Penguin Books. ISBN: 0-140-51366-3.

Tufte, Edward R. [2001]. *The Visual Display of Quantitative Information.* Second Edition. Graphics Press LLC. ISBN: 978-0-9613921-4-7.

Turabian, Kate L. [2007]. *A Manual for Writers of Research Papers, Theses, and Dissertations.* Seventh. University of Chicago Press. ISBN: 978-0-226-82337-9.

Turner, Ken [1st Nov. 2010]. *BibTeX Style Examples.* URL: `http://www.cs.stir.ac.uk/~kjt/software/latex/showbst.html`.

Unger, Gerard [2007]. *While You're Reading.* Mark Batty Publisher. ISBN: 978-0-9762245-1-8.

Van Oostrum, Piet [2nd Mar. 2004]. *Page Layout in LaTeX.*

Veytsman, Boris, and Leila Akhmadeeva [2006]. “Drawing Medical Pedigree Trees with TeX and PSTricks”. In: *PracTeX Journal* 4. URL: `http://tug.org/pracjourn/2006-4/veytsman`.

Voß, Herbert [2010]. *Math Mode.* URL: `ftp://cam.ctan.org/tex-archive/info/math/voss/mathmode/Mathmode.pdf`.

Wilson, Peter [13th Feb. 2011]. *The `fonttable` Package.* Version 1.6b. Maintained by Will Robertson.

Wilson, Peter, and Lars Madsen [6th Mar. 2011]. *The Memoir Class.*

Wright, Joseph [1st May 2010]. *`pgfopts`—LaTeX Package Options with `pgfkeys`.* Version 2.0.

——— [15th June 2011]. *`siunitx`—A Comprehensive (SI) units package.* Version 2.2i.

Zeidler, Eberhard, ed. [1996]. *Oxford User's Guide to Mathematics.* Typeset in LaTeX by Bruce Hunt. Oxford University Press. ISBN: 0-19-850763-1.

Acronyms and Abbreviations

AMS	American Mathematical Society
API	Application Programming Interface
APL	A Programming Language
CTAN	Comprehensive TeX Archive Network
DVD	Digital Versatile Disk
CD	Compact Disk
FAQ	Frequently Asked Question
GUI	Graphical User Interface
IDE	Integrated Development Environment
JVM	Java Virtual Machine
ISBN	International Standard Book Number
OS	Operating System
SI	Système International d'Unités/International System of Units
TDS	TeX Directory Structure
TUG	TeX Users Group
URL	Uniform Resource Locator
VM	Virtual Machine
WYSIWYG	What You See is What You Get

Indexes

LATEX and TEX Commands

Environments

Classes

Packages

Languages and External Commands

Environment Variables

Colophon

THIS BOOK was created with `pdflatex` in a standard TeX Live installation. The cover, the spine, and pages i–iv were produced by the publisher. The main text was typeset with the `book` class, using *FF Nexus* at 11/13×27 as the font family, and scaling its typewriter fonts to 83%.

The page, figure, and table layout were implemented with a user-defined package. The same holds for the `itemize`, `enumerate`, and `description` environments.

I had two main concerns when designing the page layout. First, I wanted figures and tables that could run into the margins. With program listings this is almost always needed; also this would let me typeset input and output side by side. My second concern was that I wanted the figure and table captions to the side. That way, long explanations would not be so disruptive.

The artwork at the back of the part titlepages is included courtesy of Billy Foley, a Cork-based artist and member of the Cork Artists Collective. The landscape on page 2 is included courtesy of the University College Cork Art Collection. More of Billy Foley's work may be found on `www.billyfoley.com`.

The `amsmath` and `amssymb` packages were used to help typeset the mathematics. The `mathastext` package was used with the option `italic` to make sure that the numbers and letters in mathematical expressions were typeset in *Nexus*. The result is not always perfect but overall it looks pleasing. The bibliography was typeset with the `biblatex` package. The `microtype` package was used with the options `tracking=smallcaps`, `expansion=true`, and `protrusion=true`.