Making \TeX\ Work

Norman Walsh

O’Reilly & Associates, Inc.
90 Sherman Street Cambridge, MA 02140
norm@ora.com

20 April 1994

Abstract

This article contains the Table of Contents, Preface, and Chapter 4 of \textit{Making \TeX\ Work} by Norman Walsh; published by O’Reilly and Associates (ISBN 1-56592-051-1).

No effort has been made to preserve the ‘look and feel’ of the book in this text, and some simplifications have been performed (particularly with respect to special formatting issues and package names).

---

1 Contents ‘Making \TeX\ Work’

1.1 Table of Contents

<table>
<thead>
<tr>
<th>Title</th>
<th>Author</th>
<th>Publisher</th>
<th>Date</th>
<th>ISBN</th>
<th>Pages</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Making \TeX\ Work</td>
<td>Norman Walsh</td>
<td>O’Reilly and Associates</td>
<td>April, 1994</td>
<td>1-56592-051-1</td>
<td>518</td>
<td>US $29.95</td>
</tr>
</tbody>
</table>

---

1: An Introduction to \TeX\  

1 The Big Picture  

What Is \TeX\?  

\TeX\ for Beginners  

Boxes and Glue  

Control Sequences  

Special Characters  

Text Formatting Versus Word Processing  

What About Desktop Publishing?  

What About troff?  

What About SGML?  

How \TeX\ Works  

\TeX\Xing a Simple Document  

Editing your document  

Running \TeX\  

Using macros  

Using fonts  

Previewing or printing \TeX\ documents  

\TeX\Xing More Complex Documents  

Building a Table of Contents  

Figure References  

indexes and Glossaries  

Bibliographies  

Special Things

2 Editing  

What Can an Editor Do?  

Quoting Automatically  

Matching Braces  

Inserting Common Control Sequences  

Running \TeX\ Automatically  

GNU Emacs  

Starting \TeX\ Mode  

Typing in Emacs  

Running \TeX\ Automatically  

AUC-\TeX\  

---
3 Running \TeX

What Do You Run? .................................................. 47
What Files Does \TeX Need? .................................... 48
Pool Files ...................................................... 48
Format Files ................................................... 49
User Files ....................................................... 49
The Command Line .............................................. 51
Command-line Options ......................................... 51
Command-line Cautions ......................................... 52
Misinterpretation of the command line ...................... 52
Filenames and \TeX ............................................. 53
Multiple documents ............................................. 54
\TeX Without Options .......................................... 55
What About Errors? ............................................. 56
Log Files ....................................................... 58
Interpreting \TeX Error Messages .............................. 59
Naming a File \TeX Cannot Find ............................... 59
Misspelling a Control Sequence Name ...................... 60
Failure to Close an Environment ............................. 60
Math Outside of Math Mode ................................... 61
Missing Fonts ................................................ 62
Everything Else .............................................. 62
The Question Mark Prompt .................................... 62

4 Macro Packages
Installation: Making Format Files .......................... 67
Hyphenation Patterns ........................................... 68
General-purpose Formats ....................................... 68
Plain \TeX ................................................... 70
Building the Plain \TeX format ............................... 71
Extended Plain \TeX .......................................... 71
Building the Extended Plain \TeX format .................... 72
I\TeX2\versus I\TeX ........................................ 72
I\TeX2\versus I\TeX ........................................ 73
Building the I\TeX2 format .................................. 74
I\TeX ...................................................... 75
Building the I\TeX format with the OFSS .................... 76
AMSTeX ................................................... 76
Building the AMSTeX format ................................ 77
AMSTeX ................................................... 79
Building the AMSTeX format ................................ 79
Lollipop ..................................................... 79
Building the Lollipop format ................................ 80
\TeXinfo .................................................. 81
Other Formats ................................................ 83
Special-purpose Formats ....................................... 84
Si\TeX .................................................... 87
No\TeX ................................................... 87
Seminars ..................................................... 88
\TeX in Color .............................................. 88
Setting Up Color ............................................. 88
Using Color ................................................ 89
Now I've Got Color, but I Need Black and White! ........ 90
Color Under I\TeX2 .......................... ............................. 90
Color Is Subtle ............................................ 90
Further Reading ............................................. 91

II: Elements of a Complex Document

5 Fonts
What \TeX Needs To Know ..................................... 96
Selecting a Font in \TeX ...................................... 97
Which Character Is Which? .................................. 98
The Issue of Size ............................................ 98
Expressing Design Size in \TeX ................................ 99
Expressing Magnification in \TeX ............................ 100
Standard Magnifications ...................................... 100
Where Do TFM Files Come From? ............................ 101
The New Font Selection Scheme .............................. 102
Selecting Fonts with the New Font Selection Scheme .... 104
Low-level interface to NFSS .................................. 105
Defining Fonts with NFSS2 ................................ 106
Declaring a family ......................................... 107
Declaring shapes ............................................ 107
Storing Font Definitions .................................... 109
Changing the Defaults ...................................... 109
NFSS Pitfalls ............................................... 110
PostScript Fonts Under NFSS ................................ 111
Adjustments to Scale ......................................... 111
When Things Go Wrong ...................................... 112
When \TeX Complains ........................................ 112
When the DVI Driver Complains ............................. 113
Can't find PK file ......................................... 113
Accents don’t work or the wrong characters are printed .... 114
Printer prints the right characters but the wrong font .... 114
Bijlage T
Making \TeX Work

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encoding Vectors</td>
<td>115</td>
</tr>
<tr>
<td>Virtual Fonts</td>
<td>116</td>
</tr>
<tr>
<td>Automatic Font Generation by DVI Drivers</td>
<td>118</td>
</tr>
<tr>
<td>Math Fonts in \TeX</td>
<td>120</td>
</tr>
<tr>
<td>Concrete Examples</td>
<td>121</td>
</tr>
<tr>
<td>METAPOST Fonts</td>
<td>121</td>
</tr>
<tr>
<td>PostScript Type 1 Fonts</td>
<td>121</td>
</tr>
<tr>
<td>Using a new PostScript font in \TeX (for PostScript printers)</td>
<td>123</td>
</tr>
<tr>
<td>Using a new PostScript font in \TeX (for non-PostScript devices)</td>
<td>126</td>
</tr>
<tr>
<td>HP LaserJet Softfonts</td>
<td>127</td>
</tr>
<tr>
<td>TrueType Fonts</td>
<td>128</td>
</tr>
<tr>
<td>6 Pictures and Figures</td>
<td>129</td>
</tr>
<tr>
<td>Different Kinds of Images</td>
<td>129</td>
</tr>
<tr>
<td>Bitmapped Images</td>
<td>130</td>
</tr>
<tr>
<td>Scalable Images</td>
<td>130</td>
</tr>
<tr>
<td>Device Independence Revisited</td>
<td>131</td>
</tr>
<tr>
<td>Using Only \TeX</td>
<td>131</td>
</tr>
<tr>
<td>Plain \TeX</td>
<td>132</td>
</tr>
<tr>
<td>\LaTeX \TeX</td>
<td>132</td>
</tr>
<tr>
<td>The epic and ecpic styles</td>
<td>134</td>
</tr>
<tr>
<td>The bezier style</td>
<td>135</td>
</tr>
<tr>
<td>Other styles</td>
<td>136</td>
</tr>
<tr>
<td>Other approaches</td>
<td>136</td>
</tr>
<tr>
<td>PiCT\TeX</td>
<td>137</td>
</tr>
<tr>
<td>XY-pic</td>
<td>139</td>
</tr>
<tr>
<td>Dra\TeX</td>
<td>141</td>
</tr>
<tr>
<td>Using a Little Help</td>
<td>144</td>
</tr>
<tr>
<td>MFPic</td>
<td>144</td>
</tr>
<tr>
<td>Fig2MF</td>
<td>144</td>
</tr>
<tr>
<td>METAPOST</td>
<td>145</td>
</tr>
<tr>
<td>Using a Little More Help</td>
<td>145</td>
</tr>
<tr>
<td>PSTricks</td>
<td>145</td>
</tr>
<tr>
<td>\TeX\xdraw</td>
<td>145</td>
</tr>
<tr>
<td>tpic</td>
<td>145</td>
</tr>
<tr>
<td>Using a Lot of Help</td>
<td>146</td>
</tr>
<tr>
<td>Electronic Cut-and-paste</td>
<td>146</td>
</tr>
<tr>
<td>Scalable Image Formats</td>
<td>146</td>
</tr>
<tr>
<td>Bitmapped Image Formats</td>
<td>149</td>
</tr>
<tr>
<td>DVI driver \special</td>
<td>149</td>
</tr>
<tr>
<td>bm2font</td>
<td>149</td>
</tr>
<tr>
<td>pbntopk</td>
<td>149</td>
</tr>
<tr>
<td>Inserting PostScript Images into \TeX</td>
<td>150</td>
</tr>
<tr>
<td>Manipulating Images</td>
<td>151</td>
</tr>
<tr>
<td>Image Magick</td>
<td>153</td>
</tr>
<tr>
<td>PBMplus</td>
<td>154</td>
</tr>
<tr>
<td>xv</td>
<td>155</td>
</tr>
<tr>
<td>xloadimage</td>
<td>156</td>
</tr>
<tr>
<td>Image Alchemy</td>
<td>156</td>
</tr>
<tr>
<td>ColorView</td>
<td>157</td>
</tr>
<tr>
<td>Jpeg4</td>
<td>157</td>
</tr>
<tr>
<td>pmjpeg</td>
<td>157</td>
</tr>
<tr>
<td>tx2p2c</td>
<td>157</td>
</tr>
<tr>
<td>Ghostscript</td>
<td>157</td>
</tr>
<tr>
<td>GoScript</td>
<td>161</td>
</tr>
<tr>
<td>hp2xx</td>
<td>161</td>
</tr>
<tr>
<td>Image Editors</td>
<td>162</td>
</tr>
<tr>
<td>xfig</td>
<td>162</td>
</tr>
<tr>
<td>idraw</td>
<td>164</td>
</tr>
<tr>
<td>tgif</td>
<td>165</td>
</tr>
<tr>
<td>bitmap/pixmap</td>
<td>165</td>
</tr>
<tr>
<td>Other Bitmap Editors</td>
<td>166</td>
</tr>
<tr>
<td>texcad/xtexcad</td>
<td>167</td>
</tr>
<tr>
<td>Screen Dumps</td>
<td>168</td>
</tr>
<tr>
<td>7 International Considerations</td>
<td>171</td>
</tr>
<tr>
<td>Typesetting in Any Language</td>
<td>171</td>
</tr>
<tr>
<td>Reading Input Files</td>
<td>172</td>
</tr>
<tr>
<td>Appearances can be deceiving</td>
<td>174</td>
</tr>
<tr>
<td>Changing the Rules</td>
<td>174</td>
</tr>
<tr>
<td>Printing the Result</td>
<td>177</td>
</tr>
<tr>
<td>The Babel Styles</td>
<td>178</td>
</tr>
<tr>
<td>Building Multilingual Babel Formats</td>
<td>179</td>
</tr>
<tr>
<td>\TeX Pitfalls</td>
<td>180</td>
</tr>
<tr>
<td>Very Complex Languages</td>
<td>181</td>
</tr>
<tr>
<td>Japanese</td>
<td>181</td>
</tr>
<tr>
<td>ASCII Nihongo \TeX</td>
<td>181</td>
</tr>
<tr>
<td>NTT JT\TeX</td>
<td>182</td>
</tr>
<tr>
<td>Poor Man’s Japanese \TeX</td>
<td>182</td>
</tr>
<tr>
<td>J\em\TeX</td>
<td>184</td>
</tr>
<tr>
<td>Chinese</td>
<td>185</td>
</tr>
<tr>
<td>Poor Man’s Chinese \TeX</td>
<td>185</td>
</tr>
<tr>
<td>Arabic</td>
<td>185</td>
</tr>
<tr>
<td>Hebrew</td>
<td>186</td>
</tr>
<tr>
<td>8 Printing</td>
<td>189</td>
</tr>
<tr>
<td>Printing Fonts</td>
<td>189</td>
</tr>
<tr>
<td>Built-in Fonts</td>
<td>190</td>
</tr>
<tr>
<td>Font substitution in \em\TeX</td>
<td>190</td>
</tr>
<tr>
<td>Font substitution in dvips</td>
<td>190</td>
</tr>
<tr>
<td>Font substitution in other drivers</td>
<td>191</td>
</tr>
<tr>
<td>External Fonts</td>
<td>191</td>
</tr>
<tr>
<td>Bitmapted Fonts</td>
<td>192</td>
</tr>
<tr>
<td>Scalable Fonts</td>
<td>195</td>
</tr>
<tr>
<td>Font Printing Pitfalls</td>
<td>195</td>
</tr>
<tr>
<td>Printing Pictures and Figures</td>
<td>196</td>
</tr>
<tr>
<td>Unsolvable Problems</td>
<td>196</td>
</tr>
<tr>
<td>Solvable Problems</td>
<td>196</td>
</tr>
<tr>
<td>Pictures Using Only \TeX</td>
<td>197</td>
</tr>
<tr>
<td>METAPOST Figures</td>
<td>197</td>
</tr>
<tr>
<td>Scalable Images</td>
<td>197</td>
</tr>
<tr>
<td>Bitmap Images</td>
<td>197</td>
</tr>
<tr>
<td>Selected Drivers</td>
<td>197</td>
</tr>
<tr>
<td>\em\TeX Drivers</td>
<td>198</td>
</tr>
<tr>
<td>dvilj2</td>
<td>200</td>
</tr>
<tr>
<td>dvips</td>
<td>200</td>
</tr>
<tr>
<td>DVILASER/HP</td>
<td>201</td>
</tr>
<tr>
<td>DVILASER/PS</td>
<td>203</td>
</tr>
<tr>
<td>PTI Laser/HP and PTI Jet</td>
<td>205</td>
</tr>
<tr>
<td>PTI Laser/PS</td>
<td>205</td>
</tr>
<tr>
<td>dvipsone</td>
<td>206</td>
</tr>
<tr>
<td>9 Previewing</td>
<td>209</td>
</tr>
<tr>
<td>Previewing Under X11</td>
<td>210</td>
</tr>
<tr>
<td>Previewing with xdvi</td>
<td>210</td>
</tr>
<tr>
<td>Previewing with X\TeX</td>
<td>213</td>
</tr>
</tbody>
</table>
10 Online Documentation

Something Is Lost ........................................ 234
\TeX{}Xinfo ............................................. 235
\LaTeX{}X2HTML ........................................ 236
Lame\TeX{} ............................................... 238
detex ..................................................... 240
dvispell .................................................. 240

11 Introducing METAFONT ................................. 241

What to Run? .............................................. 242
What Files Does METAFONT Need? ........................ 243
Pool Files ............................................... 243
Base Files ............................................... 243
METAFONT Programs .................................... 244
Command-line Options .................................. 244
Building a Base File .................................. 245
Running METAFONT ..................................... 245
Picking a Mode .......................................... 245
Proofing mode ........................................... 246
Selecting a Size ........................................ 246
Making a GF Font ....................................... 247
Making a PK Font ....................................... 247
What About Errors? ..................................... 248
Other Errors ............................................ 251
Output at Very High Resolutions ....................... 251
Output at Very Low Resolutions ........................ 252

12 Bibliographies, Indexes, and Glossaries .......... 253

\BibTeX{} .................................................. 253
How \BibTeX{} Works ...................................... 254
Building a Bibliography Database ...................... 255
Database entries ........................................ 255
Entry types ............................................. 256
Abbreviations .......................................... 257
Preamble ............................................... 257
Comments ............................................... 258
Special characters .................................... 258
Bibliography Styles .................................... 258
Special-purpose styles .................................. 259
Bibliography Database Tools ............................ 260
bibsort .................................................. 260
biblook .................................................. 260
bibclean ............................................... 260
citetags ............................................... 261
citefind ............................................... 261
bibextract ............................................. 261
lookbibtex ............................................. 261
bibdestringify ........................................ 261
edb ..................................................... 261
Emacs Bib\TeX{} mode .................................... 261
bibview ............................................... 262
xbibtex ............................................... 262
bibdb .................................................. 263
Tib ..................................................... 264
Making Indexes ........................................ 265
Index Entries .......................................... 265
Index Format .......................................... 266
Special-purpose styles ................................ 266
Making Glossaries ..................................... 266

III: A Tools Overview ..................................... 269

13 Non-commercial Environments ......................... 271

Web2C .................................................... 271
\emTeX{} .................................................. 272
Where to Start ......................................... 274
Unpacking the Archives ................................ 274
Setting Up the Environment .............................. 275
Testing the Installation ................................ 275
Installing Fonts ........................................ 276
Testing font installation ................................ 276
Making \emTeX{} Smaller .................................. 277
Deleting extra documentation .......................... 277
Deleting extra executables .............................. 277
Deleting extra printer drivers .......................... 279
Basic Customization .................................... 280
Environment variables .................................. 280
Setting up configuration files ........................ 282
Installing the Beta Test Versions of \emTeX{} .......... 282
Running \emTeX{} 386 in Windows ........................ 283
Automatic Font Generation with \emTeX{} ................ 284
Installing \emTeX{} on a Network ......................... 284
texas .................................................... 285
sb\TeX{} ............................................... 286
g\TeX{} ............................................... 287

14 Commercial Environments ............................. 289

\mu\TeX{} by ArborText .................................. 289
Y\&Y\TeX{} ............................................. 290
Textures ............................................... 291
Turbo\TeX{} ............................................ 291
PCTeX .................................................. 292
PCTeX For Windows ..................................... 294
Scientific Word ........................................ 295

15 \TeX{} on the Macintosh ................................. 299

CMac\TeX{} ............................................. 299
Direct\TeX{} .......................................... 300
Oz\TeX{} ............................................... 301
TeXtures ............................................... 302
1.3 List of Tables

1-1 Special Characters in Plain TeX ................ 9
1-2 How to Typeset Special Characters ............ 9
2-2 AUC-TeX Math Operators in Mathematics .......... 30
2-3 TeX Modes of Interaction ................... 40
5-1 Standard Weight and Width Designations ........ 103
5-2 Weight and Width Are Combined to Form Series ... 103
5-3 Standard Abbreviations of Font Shape .......... 104
5-4 User-level Font Selection Control Sequences in NFSS2 ................ 104
5-5 Encoding Schemes Supported by NFSS2 ............ 106
5-6 Default Fonts .................................. 109
6-1 Graphics Manipulation Packages ................. 152
6-6 Graphics Editing Packages ........................ 162
7-1 Standard Control Sequences for Symbols from Other Character Sets .... 173
7-2 New Control Sequences Proposed by TWGMMLC ...... 173
7-3 Language Switch Macros Proposed by TUG ...... 176
8-1 Common DVI Drivers ............................. 198
8-3 Other DVILASER/HP Utilities .................. 202

1.4 List of Examples

1-1 An Example of a TeX Document ............... 6
1-2 A typical bibliography database entry ........... 21
2-1 A Simple Driver File .......................... 31
2-2 Local Variables in an Emacs Buffer ............ 32
2-3 A Contrived Error ............................. 38
2-4 Revision Control Macros for TeX Documents using RCS ................ 42
3-1 The Document BADFONT.TEX .................. 57
4-1 Plain TeX Input .................................. 70
4-2 \texttt{HypEx} Input .............................. 74
4-3 \texttt{HypEx} Input File ........................ 76
4-4 AMSTeX Input File .............................. 77
4-5 Lollipop Input File .............................. 79
4-6 \texttt{Xinfo} Input ............................... 81
4-7 The ChemTeX Source for Caffeine ............... 85
4-8 The ChemStruct Source for the Lithium Cation ... 85
4-9 The MusicTeX Source for Figure 4-7 ............ 86
5-1 Font-shape Declaration with NFSS2 ............ 108
5-2 Font-shape declaration with NFSS2 (simplified) .... 108
5-3 The PSTeXfont script .......................... 119
6-1 The Input for the Plain TeX Diagram .......... 132
6-2 The LATEX Input for Figure 6-2 ............... 133
6-3 The epic Input for Figure 6-3 .................. 134

8-4 Other DVILASER/PS Utilities ................ 204
8-5 Other dvipson Utilities ....................... 207
9-1 Common Previewers .............................. 210
9-2 TeX Preview Utilities ......................... 221
9-3 DVIWinde Utilities .............................. 229
11-1 Some Popular METAFONT Fonts on the CTAN Archives ................ 242
12-1 Types of Entries with Required and Optional Fields ............ 256
13-1 Summary of the emTeX Distribution ............ 273
13-2 Fonts Libraries Available for emTeX on CTAN ................ 276
13-3 Other emTeX Executables ....................... 278
13-4 The Beta Test Files for emTeX ................ 282
14-1 TurboTeX Programs .............................. 291
14-2 TurboTeX METAFONT Programs .................. 292
15-1 Summary of the CMacTeX Distribution at CTAN .......... 300
15-2 Summary of the OzTeX Distribution at CTAN301 ........ 307
15-3 Font Tools in the BSR package ................ 307
16-1 The Public Domain DVI Driver Family ............ 315
16-2 RTF Translators ............................... 324
16-3 Additional Scripts in PS Utilis ................. 330
B-1 The Computer Modern Roman Font Encoding .......... 346
B-2 The Computer Modern Math Italic Font Encoding .... 347
B-3 The Cork Font Encoding ........................ 348
B-4 The Adobe Standard Font Encoding ............... 350
2 Preface ‘Making \TeX\ Work’

\TeX\ is a tool for creating professional quality, typeset pages of any kind. It is particularly good, perhaps unsurpassed, at typesetting mathematics; as a result, it is widely used in scientific writing. Some of its other features, like its ability to handle multiple languages in the same document and the fact that the content of a document (chapters, sections, equations, tables, figures, etc.) can be separated from its form (typeface, size, spacing, indentation, etc.) are making \TeX\ more common outside of scientific and academic circles.

Designed by Donald Knuth in the late 1970s, more than a decade of refinement has gone into the program called \TeX\ today. The resulting system produces publication-quality output while maintaining portability across an extremely wide range of platforms.

Remarkably, \TeX\ is free. This fact, probably as much as any other, has contributed to the development of a complete \TeX\ system by literally thousands of volunteers. \TeX, the program, forms the core of this environment and is now supported by hundreds of tools.

2.1 Why Read This Book?

This book is for anyone who uses \TeX. Novices will need at least one other reference, because this book does not describe the nuts and bolts of writing documents with \TeX\ in any great detail.

If you are new to \TeX, there is much to learn. There are many books that describe how to use \TeX. However, the focus of this book is mostly at a higher level. After digesting Chapter 1, *The Big Picture*, you should be able to proceed through the rest of the book without much difficulty even if you have never seen \TeX\ before. So, if you are a system administrator interested in learning enough about these programs to install and test them for your users, you should be all set. If you are interested in learning how to write documents with \TeX, this book will be helpful, but it will not be wholly sufficient.

Why do you need this book at all? Although many individual components of the \TeX\ system are well documented, there has never before been a complete reference to the whole system. This book surveys the entire world of \TeX\ software and helps you see how the various pieces are related.

A functioning \TeX\ system is really a large collection of programs that interact in subtle ways to produce a document that, when formatted by \TeX, prints the output you want. All the different interactions that take place ultimately result in less work for you, the writer, even though it may seem like more work at first. Heck, it may be more work at first, but in the long run, the savings are tremendous.

Many books about \TeX\ refer the reader to a ‘local guide’ for more information about previewing and printing documents and what facilities exist for incorporating special material into documents (like special fonts and pictures and figures). In reality, very few local guides exist.

The \TeX\ environment is now mature and stable enough to support a more ‘global guide.’ That is what this book attempts to be. It goes into detail about previewing and printing, and incorporating special fonts, about adding pictures and figures to your documents, and about many other things overlooked by other books.

Because fonts play a ubiquitous role in typesetting, this book is also about METAFONT, the tool that Donald Knuth designed for creating fonts.

2.2 Scope of This Book

Here’s how the book is laid out.

Part I: An Introduction to \TeX.

Chapter 1, *The Big Picture*. If you don’t know anything about \TeX\ at all, this chapter will help you get started. If you’re a system administrator charged with the task of installing and maintaining \TeX\ tools, you’ll get enough information to do the job.

Chapter 2, *Editing*. An overview of some environments that can make working with \TeX\ documents easier. It describes some editors that ‘understand’ \TeX, how to integrate \TeX\ into a ‘programmer’s editor,’ spellchecking, revision control, and other aspects of \TeX\ editing.

Chapter 3, *Running \TeX*. The mechanics of running \TeX, the program. It discusses what things \TeX\ needs to be able to run, how to start \TeX, command-line options, leaving \TeX, and recovery from errors.

Chapter 4, *Macro Packages*. Overview of \TeX\ macro packages. This chapter describes how to make a format file, the major general-purpose writing packages, some special...
purpose writing packages, how to make slides for presentations, and how to handle color in \TeX.

**Part II: Elements of a Complex Document.**

Chapter 5, *Fonts.* This chapter explores the issues that need to be addressed when using fonts. Many of these issues are not particularly \TeX-related, but \TeX is very flexible, and it’s important to understand the tradeoffs that must be made. This chapter also examines some \TeX-specific issues: font selection, files that \TeX needs, automatic font generation, and virtual fonts.

Chapter 6, *Pictures and Figures.* How many ways are there to include pictures and figures in a \TeX document? Lots and lots of ways. This chapter examines them all.

Chapter 7, *International Considerations.* \TeX is well qualified to do international typesetting. This chapter looks at the issues that are involved: representing international symbols in your input file, what \TeX produces, getting the right fonts, multiple languages in the same document, and macro packages and style files that solve some of these problems. Some strategies for dealing with very difficult languages (like Japanese and Arabic) are also explored.

Chapter 8, *Printing.* What goes in has to come out. This chapter tells the what, where, why, and how of printing your documents.

Chapter 9, *Previewing.* Save paper; preview your documents before you print them.

Chapter 10, *Online Documentation.* Online documentation is becoming increasingly popular. This chapter explores different ways that both typeset and online documentation can be produced from the same set of input files.

Chapter 11, *Introducint METAFONT.* Sometimes it is necessary or desirable to create a special version of a standard \TeX font. Maybe you really need the standard 10pt font at 11.3pt. This chapter will tell you how to work with existing METAFONT fonts. It won’t tell you how to create your own fonts; that’s a whole different story.

Chapter 12, *Bibliographies, Indexes, and Glossaries.* Maintaining a bibliographic database can be a great timesaver. This chapter looks at the Bib\TeX program and other tools for building and using bibliographic databases. It also discusses the creation of indexes and glossaries.

**Part III: A Tools Overview.**

Chapter 13, *Non-commercial Environments.* Many \TeX environments are freely available. This chapter describes public domain, free, and shareware versions of \TeX.

Chapter 14, *Commercial Environments.* A large, complex system like \TeX can be overwhelming (although I hope less so after you read this book ;-) ). One of the advantages of selecting a commercial implementation of \TeX is that some form of customer support is usually provided. Still other commercial implementations offer features not found in any free releases. This chapter describes several commercial \TeX releases.

Chapter 15, *\TeX on the Macintosh.* Many issues discussed in this book apply equally to all platforms, including the Macintosh platform, but the Mac has its own special set of features. This chapter looks at some versions of \TeX and other tools designed specifically for use on the Mac.

Chapter 16, *\TeX Utilities.* This chapter lists many of the utilities available in the CTAN archives and provides a brief description of what they do.

Appendix A, *Filename Extension Summary.* Lots of files can be identified by their extensions. This appendix lists the extensions that are most often seen in conjunction with \TeX and describes what the associated files contain.

Appendix B, *Font Samples.* Examples of many METAFONT fonts available from the CTAN archives.

Appendix C, *Resources.* A complete list of the resources described in this book.

Appendix D, *Long Examples.* This appendix contains examples (scripts, batch files, programs) that seemed too long to place in the running text.

**Bibliography.** Where I learned what I know. Also, where you can look for more information about specific topics.

### 2.3 Conventions Used in This Book

The following typographic conventions are used in this book:

*Italic* is used for filenames, directories, user commands, program names, and macro packages (except where the distinctive logo type is used). Sometimes italics are used in the traditional way for emphasis of new ideas and important concepts.

*Typewriter* is used for program examples, FTP sites, \TeX control sequences, and little bits of \TeX syntax that appear in running text (for example, typewriter text in a reference to the \verb|picture| environment is a clue that \verb|\picture| literally uses the word ‘picture’ to identify this environment).

*Typewriter Bold* is used in examples to show the user’s actual input at the terminal.

Typewriter*Italic* identifies text that requires a context-specific substitution. For example, *filename* in an example would be replaced by some particular filename.

Footnotes are used for parenthetical remarks. Sometimes, lies are spoken to simplify the discussion, and the footnotes restore the lie to the truth.1

Filename extensions, like `.tex` in `book.tex`, are shown in uppercase letters when referring to a particular type of file. For example, a `\TeX Font Metric` or `TFM` file would be `somefile.fmm`. The actual extension of the file may be different (upper or lowercase, longer or shorter) depending on your operating environment.

---

1 And sometimes they don’t ;-)
When the shell prompt is shown in an example, it is shown as a dollar sign, $\. You should imagine that this is your system’s prompt, which might otherwise be %, C>, [C:\], or a dialog box.

When spaces are important in an example, the ’ ‘ character is used to emphasize the spaces. Effective, isn’t it?

In some places, I refer to specific keys that you should press. When it’s important that I mean pressing particular keys and not typing something, I emphasize the keys. For instance, an example that includes Enter means that you should literally press the Enter or Return key. The sequence Ctrl-D means that you should press and hold the ‘Control’ and ‘d’ keys simultaneously. Control-key combinations aren’t case sensitive, so you needn’t press the shift key.

2.4 How to Get \TeX

\TeX{} and the other programs mentioned in this book are available from a number of places. It’s impossible to list all of the places where you might find any given tool, but there is one place where you will almost certainly find every tool: the Comprehensive \TeX{} Archive Network (CTAN).

This network is a fully-mirrored anonymous FTP hierarchy on three continents. Always use the FTP site that is geographically closest to you. The following table lists the current members of CTAN as of July, 1993:

<table>
<thead>
<tr>
<th>Geographic Location</th>
<th>Site 1</th>
<th>Site 2</th>
<th>IP Address</th>
<th>Top-Level Directory</th>
</tr>
</thead>
<tbody>
<tr>
<td>United States</td>
<td>ftp.shsu.edu</td>
<td>ftp.tex.ac.uk</td>
<td>192.241.115.10</td>
<td>tex-archive</td>
</tr>
<tr>
<td>Germany</td>
<td>ftp.uni-stuttgart.de</td>
<td></td>
<td>129.69.8.13</td>
<td>tex-archive</td>
</tr>
</tbody>
</table>

You may also access the CTAN archives by electronic mail if you do not have FTP access. For up-to-date instructions about the mail server, send the single-line message help to fileserv@shsu.edu.

2.4.1 Where Are the Files?

Every CTAN mirror site has the same well-organized directory structure. The top-level directory also contains a complete catalog of current files organized by name, date, and size. The catalogs are named FILES.bydate, FILES.byname, and FILES.bysize, respectively, in the top level directory. The top-level directory contains the following subdirectories:

The archives at ftp.shsu.edu and ftp.tex.ac.uk also support gopher access to the archives. The UK gopher supports indexed access to the archives. A World Wide Web (hypertext) interface to the archives is available from:

http://jasper.ora.com/CTAN/can.html

This interface includes brief descriptions of many packages and the ability to perform keyword and date searches.

2.4.2 Getting Software Without FTP

The electronic alternatives to FTP, described in the section ‘Getting Examples From This Book’ of this chapter are also viable alternatives for getting software from the CTAN archives.

In addition, there are a number of ways to get distributions through nonelectronic channels. The names and addresses of these sources are listed in Appendix C, Resources.

You can get many of the popular \TeX{} distributions on diskette from the \TeX{} Users Group (TUG). Emacs, Ghostscript, and other packages by the Free Software Foundation (FSF) are available on tape directly from the FSF. You may also find large bulletin board systems that support \TeX{} (for example, Channel1 in Cambridge, MA).

2.4.3 Getting Examples From This Book

All of the substantial code fragments and programs printed in this book are available online. The examples in this book are all in Perl, a language for easily manipulating text, files, and processes. I decided to use Perl simply because it is available for every platform discussed in this book. It is the only ‘universal’ scripting language that will work under MS-DOS, OS/2, UNIX, and the Macintosh. All of the scripts in this book can be converted to a different scripting language (the various UNIX shells or something like 4DOS’s extended batch language for MS-DOS and OS/2) if you prefer. I’ve tried to write the Perl scripts in a straightforward way so that any given task won’t be too difficult.

The examples are available electronically in a number of ways: by FTP, FTPMAIL, BITFTP, and UUCP. The cheapest, fastest, and easiest ways are listed first. If you read from the top down, the first one that works is probably the best. Use FTP if you are directly on the Internet. Use FTPMAIL if you are not on the Internet but can send and receive electronic mail to Internet site (this includes CompuServe users). Use BITFTP if you send electronic mail via BITNET. Use UUCP if none of the above work.

NOTE: The examples were prepared using a UNIX system. If you are running UNIX, you can use them without modification. If you are running on another platform, you may
need to modify these examples to correct the end-of-line markers. For example, whereas under UNIX every line ends with a line feed character (the carriage return is implicit), under DOS every line must end with explicit carriage return and line feed characters.

**FTP** To use FTP, you need a machine with direct access to the Internet. A sample session is shown below.

```$ ftp ftp.uu.net  
Connected to ftp.uu.net.  
Name (ftp.uu.net:prefect): anonymous  
331 Guest login ok, send e-mail address as password.  
Password: prefect@guide.com  
230 Guest login ok, access restrictions apply.  
ftp> cd /published/oreilly/nutshell/maketexwork  
250 CWD command successful.  
ftp> get README  
200 PORT command successful.  
150 Opening ASCII mode data connection for README (xxxx bytes).  
226 Transfer complete.  
local: README remote: README  
xxxx bytes received in xxx seconds (xxx Kbytes/s).  
ftp> binary  
200 Type set to I.  
(Repeat get commands for the other files.  They are listed in the README file.)  
ftp> quit  
221 Goodbye.  
$```

**FTPMAIL** FTPMAIL is a mail server available to anyone who can send electronic mail to, and receive it from, Internet sites. This includes most workstations that have an email connection to the outside world and CompuServe users. You do not need to be directly on the Internet.

Send mail to `ftpmail@decwrl.dec.com`. In the message body, give the name of the anonymous FTP host and the FTP commands you want to run. The server will run anonymous FTP for you and mail the files back to you. To get a complete help file, send a message with no subject and the single word `HELP` as the message body.

The README file should appear some time later (up to a day or more) in the directory `/usr/spool/uucppublic/yourname`. If you don’t have an account, but would like one so that you can get electronic mail, contact UUNET at 703-204-8000.

If you don’t have a UUNET account, you can set up a UUCP connection to UUNET in the United States using the phone number 1-900-468-7727. As of this writing, the cost is 50 cents per minute. The charges will appear on your next telephone bill. The login name is `uucp` with no password. Your entry may vary depending on your UUCP configuration.

**Gopher** If you are on the Internet, you can use the `gopher` facility to learn about online access to examples through the O’Reilly Online Information Resource. Access `gopher.ora.com` as appropriate from your site.

### 2.5 Versions of \TeX

The most recent versions of \TeX and METAFONT are version 3.1415 and version 2.71, respectively. Version 3 of \TeX introduced several new features designed to improve support for non-English languages (including the use of
8-bit input and some refinements to hyphenation control. If you use an older version of \TeX, you should upgrade.

Donald Knuth has specified that \TeX's version number converges to \( \pi \), therefore version 3.1415 is only the fourth minor revision after version 3. The next minor revision will be version 3.14159. Similarly, METAFONT's version number converges to \( e \) (2.7182818284\ldots).

2.6 Implementations and Platforms

The interface that \TeX presents to the writer is very consistent. Most of the examples described in this book are applicable to every single implementation of \TeX. However, \TeX is not a closed system. It is possible to step outside of \TeX to incorporate special elements into your document or take advantage of the special features of a particular environment. These extensions can dramatically restrict the portability of your documents.

Many of the topics covered in this book offer alternatives in those areas that are less portable. Therefore, it is natural to ask what implementations are really covered.

Before outlining which implementations are covered, let me suggest that this book will be useful even if you are using an implementation not 'officially' covered here. The reality of the situation is this: many, many tools have been ported to \TeX. Many of the tools mentioned in this book are available on platforms that are not specifically discussed. Time and equipment constraints prevented Amiga, Atari, NeXT, VMS, and Windows NT implementations of \TeX from being specifically addressed in this edition of the book.

2.6.1 UNIX

UNIX is probably the most common \TeX platform. The emphasis in this book is on UNIX workstations running X11, producing output for PostScript and HP LaserJet printers.

Linux and other personal computer implementations of UNIX are not addressed specifically; however, with the successful port of X11 to Linux, I’m confident that every UNIX tool here can be, or has been, ported to Linux (and probably other PC UNIX environments).

The only implementation of the \TeX program for UNIX considered in any detail is the free implementation distributed in web2c. This distribution is described in the section called ‘Web2C’ in Chapter 13, Non-commercial Environments. Most of the other UNIX tools discussed here are also free.

2.6.2 MS-DOS

With very few exceptions, the tools in this book are available under MS-DOS. Because PCs are very popular, a lot of effort has gone into porting UNIX tools to MS-DOS. Some packages, however, require a 386SX (or more powerful) processor. For the most part, I focus on PCs running MS-DOS only; however, Microsoft Windows and DesqView are not entirely ignored.

There are quite a few options when it comes to selecting an implementation of the \TeX program under MS-DOS. Several free implementations are discussed as well as some commercial implementations. For more information about these implementations, consult Chapter 13, Non-commercial Environments, and Chapter 14, Commercial Environments.

2.6.3 OS/2

In this book, OS/2 is treated primarily as a superset of MS-DOS. When possible, I look at OS/2-specific versions of each utility, but rely on MS-DOS as a fall-back.

Extensions to \emph{\TeX} for OS/2 are explored, as are editing environments such as \emph{epm}. The multi-threaded nature of OS/2 allows more complete porting of UNIX tools. When better ports are available for OS/2, they are discussed.

2.6.4 Macintosh

The Macintosh is very different from the systems described above. Chapter 15, \TeX on the Macintosh, discusses the Macintosh environment in detail.

There are four implementations of \TeX for the Macintosh. Three are freely available, and one is commercial: CMacTeX is free, Oz\TeX and Direct\TeX are shareware, and Textures is a commercial package from Blue Sky Research.

2.7 We’d Like to Hear From You

We have tested and verified all of the information in this book to the best of our ability, but you may find that features have changed (or even that we have made mistakes!). Please let us know about any errors you find, as well as your suggestions for future editions, by writing:

O’Reilly & Associates, Inc.
103 Morris Street, Suite A
Sebastopol, CA 95472
1-800-998-9938 (in the US or Canada)
1-707-829-0515 (international/local)
1-707-829-0104 (FAX)

You can also send us messages electronically. To be put on the mailing list or request a catalog, send email to:

nuts@ora.com (via the Internet)
uunet!ora!nuts (via UUCP)

To ask technical questions or comment on the book, send email to:

bookquestions@ora.com (via the Internet)

2.8 Acknowledgments

This book would not exist if I had not received support and encouragement from my friends and colleagues, near and far. I owe the deepest debt of gratitude to my wife, Deborah, for patience, understanding, and support as I progressed through what is easily the most all-consuming task I have ever undertaken.
Chapter 4 from book: Macro Packages

Everyone who uses \TeX uses a macro package (also called a ‘format’). A macro package extends \TeX to provide functionality that is suited to a particular task or set of tasks.

This chapter provides a summary of \TeX macro packages. General-purpose packages designed to typeset a wide range of documents—articles, books, letters, and reports—are examined first. The general-purpose packages described are Plain \TeX, Extended Plain \TeX, \LaTeX, \LaTeXe, AMSTeX, AMSLaTeX, Lollipop, and TeXinfo. After surveying the general-purpose packages, several special-purpose packages designed to handle specific tasks—typesetting transparencies, music, chemistry or physics diagrams, etc.—are described. The special-purpose packages surveyed are Sli\TeX, Foil\TeX, Seminar, Music\TeX, ChemStruct, and Chem\TeX.

There are a lot of overlapping features and similar commands in the general-purpose packages. To understand why this is the case, consider how a new macro package comes into existence. An ambitious person, who is very familiar with \TeX, decides that there are some things she would like to express in her documents that are difficult to express with existing formats. Perhaps, for example, no existing format produces documents that match the precise specifications required for publishing in her field, or perhaps she has in mind a whole new document structuring paradigm. A more mundane possibility is simply that she has been customizing an existing format for some time and now feels it has enough unique features to be useful to others.

In any event, a new format is born. Now, if this format is designed for a very specific task, writing multiple-choice mathematics exams, for example, it might not have very many general-purpose writing features. On the other hand, if it is designed for writing longer, more general documents (e.g., history textbooks or papers to appear in a particular journal) then there are a number of features that it is likely to include; provisions for numbered lists, cross references, tables of contents, indexes, and quotations are all examples of features common to many documents.

To support these common features, many macro packages have similar control sequences. This stems from the fact that they are all built on top of a common set of primitives and that macro package authors tend to copy some features of other packages into their own.

You may find that you’d like to use the features of several different packages in the same document. Unfortunately, there is no provision for using multiple formats to process a single document. The features required to process most documents are shared by all of the general-purpose formats, however. You are more likely to need multiple macro packages if you want to use a special format to construct a diagram or figure and incorporate it into a document. Chapter 6, *Pictures and Figures*, describes several ways to take ‘electronic snippets’ of one document and insert them into another, which is one possible solution to this problem.

If you’re beginning to feel a little lost, have no fear. Most general-purpose formats are sufficient for most documents. And there’s no reason why every document you write has to be done with the same format. Many people find \LaTeX and Plain \TeX sufficient, but if you’re writing an article for a particular journal and someone has written a format specifically for that journal’s documents, by all means use it. It is more likely, however, that someone has written a style file which tailors \LaTeX to the requirements of the journal.
In addition to describing some common macro packages in this chapter, I’ll describe how to build format files for them. If the packages that you want to use have already been installed at your site, you can ignore the installation sections.

The packages that you find most convenient will depend on the tasks you perform and how well each package suits your work style. The list of packages in this chapter is not meant to be all-inclusive, nor is it my intention to suggest which packages are best. Use the ones you like, for whatever reasons.

I can hear some of you already, ‘I don’t really need a macro package,’ you say, ‘I can roll my own with just \TeX.’

And you are absolutely correct.

I don’t recommend it, however. It’s akin to using your compiler without any of the built-in functions. Most \TeX primitives offer little support by themselves for writing documents.

A New Perspective: \LaTeX\textsuperscript{2}c (and its successor, which will be \LaTeX\textsuperscript{3}) seek to address many of the problems mentioned above by defining a core \LaTeX\ format with extension packages to provide custom features.

The \LaTeX\textsuperscript{2}c system provides a single format file that supports \LaTeX, AMSTeX, and Si\TeX. For the time being, \LaTeX\textsuperscript{2}c is described separately in this chapter along with the other formats. Be aware, however, that \LaTeX\textsuperscript{2}c is now the standard \LaTeX\ and you should migrate to it as soon as possible.

### 3.1 Installation: Making Format Files

A format file, as described in Chapter 3, \textit{Running \TeX}, is a special ‘compiled’ version of a macro package. The \xind\ program interprets all of the control sequences in a macro package and writes the corresponding memory image into a file. Loading a format file is much faster than loading individual macro packages in your document because \TeX\ does not have to interpret any of the control sequences while it is also processing text.

In general, all format files should be stored in the same directory.\footnote{At the time of this writing, it’s actually still in test-release, but it may be available as a standard release by the time you read this.} If you install \TeX\ in a directory called \texttt{tex}, then formats typically go in a directory called \texttt{formats} in the \texttt{tex} directory. This is not universally the case because you need separate format directories for big and small \TeX\s.\footnote{On Macintosh systems and other environments that don’t have directories, format files are typically stored in their own folders (or other metaphorically appropriate place :-).}

Usually, an environment variable indicates where the format files are located. Environment variables are a common way of customizing your interaction with programs. They are usually set in your \texttt{AUTOEXEC.BAT} file for MS-DOS, \texttt{CONFIG.SYS} for OS/2, or the \texttt{rc} file for a shell (i.e., \texttt{.cshrc}, \texttt{.kshrc}) on UNIX systems. Any good reference book for your operating system or shell will describe how to use environment variables.

A common name for the environment variable that indicates where \TeX\ formats are located is \texttt{TEXFORMATS}. Implementations that provide big and small \TeX\s need another variable to indicate the directory that contains formats for big \TeX.\footnote{The distinction between big and small \TeX\ is described in the section called ‘What Do You Run?’ in Chapter 3.}

\xind\ is not always a separate program. Some implementations of \TeX\ combine the functionality of \xind\ and \xind\ into a single program and use a special switch at runtime to determine which function to perform. In this chapter, all of the examples use the program name \texttt{initex} to identify \xind\. If you use an implementation that doesn’t provide a separate \xind\ program, you should use the \TeX\ program with the \xind\ switch instead. For example, for \texttt{em\TeX}, use \texttt{tex /i} instead of \texttt{initex} when you build a format.

Like \TeX, \xind\ needs to be able to find input files. Usually, this is accomplished by searching the directories listed in the \texttt{TEXINPUTS} environment variable. Place the input files that \xind\ needs in a directory on the \texttt{TEXINPUTS} path before running \xind\ unless otherwise directed. The \texttt{TEXINPUTS} environment variable is discussed in the ‘User Files’ section of Chapter 3.

### Hyphenation Patterns

In order for \TeX\ to correctly hyphenate words, every format file must contain a set of hyphenation patterns. The patterns are part of an algorithmic solution to the problem of breaking a word into syllables for hyphenation.

The details of the hyphenation algorithm (given in Appendix H of \textit{The \TeXbook}) are too complex to describe here, but two aspects of this solution deserve particular emphasis. First, using patterns means that a dictionary of hyphenated words is not necessary.\footnote{A small set of exceptions is maintained because the algorithm isn’t perfect.} This saves a lot of space and time. Second, by loading different sets of patterns, \TeX\ can achieve equal success at hyphenating any language—even English :-). There are actually at least two sets of hyphenation patterns for English, one for British English and one for American English. Chapter 7, \textit{International Considerations}, describes how to load multiple sets of hyphenation patterns for typesetting multilingual documents.

### 3.2 General-purpose Formats

This section describes several macro packages that are designed for formatting standard documents like articles or books. In order to provide some form of comparative measure, each macro package is used to create the same document, a one-page report that looks like Figure 1. I constructed this example to demonstrate a few common elements in a document: several sizes of headings, a paragraph of text, inline and displayed mathematics, and a few...
fonts. There are lots of other things that aren’t shown (tables, figures, footnotes, etc.), and these elements vary as much as any other in the different macro packages.

Observant readers will notice that the examples are shown in the Computer Modern fonts while the rest of this book uses different fonts. There are a number of reasons why the example is shown in Computer Modern. For one thing, all of the formats discussed here use the Computer Modern fonts by default. Using a different set of fonts would only add more complexity to each example. A more subtle problem is that I do not have appropriate mathematics fonts for Garamond. There are a number of complex issues involving the use of fonts in \TeX. They are discussed in Chapter 5, *Fonts*.

### Chapter 1

**Unsolved Problems**

#### 1.1 Odd Perfect Numbers

A number is said to be *perfect* if it is the sum of its divisors. For example, 6 is perfect because $1+2+3=6$, and 1, 2, and 3 are the only numbers that divide evenly into 6 (apart from 6 itself). It has been shown that all even perfect numbers have the form $2^{p-1}(2^p-1)$ where $p$ and $2^p-1$ are both prime.

The existence of odd perfect numbers is an open question.

/%}

---

**Figure 1: Sample page**

#### 3.2.1 Plain \TeX

Plain \TeX is the format written by Donald Knuth during the development of \TeX. It is described fully in *The \TeX*book.*

Plain \TeX ties together the \TeX primitives in a way that makes it practical to work in \TeX. If you do not have a computer programming background, you may find Plain \TeX a little bit intimidating. It is definitely a ‘roll your own’ environment. Although it demonstrably contains all of the functionality required to write everything from letters to books, there is very little ‘user-friendly’ packaging around the internals of \TeX.

Aside from user interface considerations, which are highly subjective, Plain \TeX lacks some functionality when compared to other formats. There is no provision in Plain \TeX for automatically numbered sections, labelled figures, tables of contents, indexes, or bibliographies. Any of these functions can be constructed in Plain \TeX if you are willing to invest the time and energy required to write your own macros, but they are not built into Plain \TeX.

If you enjoy writing your own macros or plan to produce novel types of documents, a firm grasp of Plain \TeX will allow you to write anything in \TeX. A firm grasp of Plain \TeX also makes it easier to understand and modify other formats (like LATEX) that are built on top of Plain \TeX.

In addition, Plain \TeX is the only format that is always distributed with \TeX. The other formats discussed in this chapter are freely available but do not come with \TeX.

The Plain \TeX input that produces the report in Figure 1 is shown in Example 1.

**Example 1: Plain \TeX Input**

```latex
% Format: Plain
\font\chapfont=cmbx12 scaled 1728
\font\titlefont=cmbx12 scaled 2073
\font\secfont=cmbx12 scaled 1200
\parskip=\baselineskip
\parindent=0pt
\hscale=5in
\hoffset=.75in
\leftline{\chapfont Chapter 1}
\vskip36pt
\leftline{\titlefont Unsolved Problems}
\vskip36pt
\leftline{\secfont 1.1\ Odd Perfect Numbers}
\vskip12pt
A number is said to be \textit{perfect} if it
\begin{center}
\begin{tabular}{c}
\vspace{-4pt}
\end{tabular}
\end{center}
\begin{center}
the sum of its divisors. For example, 6 is
\end{center}
\begin{center}
perfect because $1+2+3=6$, and 1, 2, and 3 are
\end{center}
\begin{center}
the only numbers that divide
\end{center}
\begin{center}
evenly into 6 (apart from 6 itself).
\end{center}
It has been shown that all even perfect numbers have the form
\begin{center}
$2^{p-1}(2^p-1)$
\end{center}
where $p$ and $2^p-1$ are both prime.

The existence of odd perfect numbers is an open question.
\bye
```

Building the Plain \TeX format To build Plain \TeX, you need only two files: plain.tex and hyphen.tex. These files are distributed with \TeX so they should be available as soon as you have installed \TeX. The hyphen.tex file is language-dependent. Readers who frequently work with non-English text should read Chapter 7 for more information about obtaining non-English hyphenation patterns.

The command:

```bash
$s initex plain \dump
```

---

7Really observant readers may have noticed that it’s a version of Garamond ;-)
will create the Plain \TeX format. Move the resulting files, plain.fmt and plain.log, into your \TeX formats directory.

3.2.2 Extended Plain \TeX

Extended Plain \TeX extends Plain \TeX in a number of useful ways without forcing you to use any particular ‘style’ of output. The argument is this: although Plain \TeX really doesn’t provide all of the features that you need (tables of contents, cross references, citations, enumerated lists, convenient access to verbatim input, etc.), many of these features don’t have any direct impact on the appearance of your document. Unfortunately, other general-purpose macro packages like \LaTeX, which do provide these features, tend to force you to accept their notion of what the typeset page should look like.

Extended Plain \TeX is an attempt to solve that problem. It provides many behind-the-scenes features without providing any general page layout commands (like \texttt{chapter} or \texttt{section}), which means that these features can be used inside Plain \TeX without much difficulty and without changing the layout of the typeset pages.

Building the Extended Plain \TeX format  
To build the Extended Plain \TeX format, you need the plain.tex and hyphen.tex files required to build the Plain format as well as the eplain.tex file distributed with Extended Plain \TeX. Make and install the Plain \TeX format first, then change to the directory that contains the Extended Plain \TeX distribution.

The command:

\verb+$initex \plain eplain \dump$ will create the Extended Plain \TeX format. Move the resulting files, eplain.fmt and eplain.log, into your \TeX formats directory.

3.2.3 \LaTeXX_2 \textit{Versus} \LaTeXX

The tremendous popularity of \LaTeX in the \TeX community has had an unfortunate side effect: because it is a very familiar and flexible format, many people have used it as the basis for extensions of one sort or another. This has resulted in a wide range of (slightly) incompatible formats and a lot of frustration.

This situation is being rectified by a new release of \LaTeXX, currently called \LaTeXX_2. The new release replaces the existing dialects of \LaTeXX (\LaTeX with and without NFSS, Slit\TeX, AMS\LaTeX, etc.) with a single core system and a set of extension packages. \LaTeXX_2 includes a compatibility mode which will allow it to continue to format existing documents without change (provided that they do not rely on local modifications to the \LaTeXX format, of course). Local modifications can also be incorporated into the \LaTeXX_2 system as extension packages, making \LaTeXX_2 a complete replacement for all existing versions of \LaTeXX and packages closely derived from \LaTeXX.

The most significant and least compatible difference between \LaTeXX and \LaTeXX_2 is the font selection scheme. There are many control sequences for selecting fonts in \LaTeXX. Some control the typeface (\texttt{\textbackslash rm, \textbackslash tt, \textbackslash sc, etc.}), some the size (\texttt{\textbackslash small, \textbackslash normalsize, \textbackslash large, etc.}), and some the appearance (\texttt{\textbackslash it, \textbackslash bf, \textbackslash em, etc.}).

Under the Old Font Selection Scheme (OFSS), the control sequences for selecting a font completely override any font selection already in place. Consider, for example, the control sequences \texttt{\textbackslash it and \textbackslash bf, which switch to italic and boldface}. Using \texttt{bf\it} produces italic text, and \texttt{bf\it} produces boldface text, and \texttt{bf\it} produces bold-faced-italic text (which is probably what you wanted).

Under the New Font Selection Scheme, typeface (called \textit{family} in NFSS parlance), appearance (called \textit{series} and \textit{shape}), and size are viewed as orthogonal components in font selection. Because these parameters are independent, selecting an italic appearance with the \texttt{\textbackslash it} control sequence switches to italic in the current typeface and size. Under the NFSS, \texttt{\textbackslash bf\it} does select boldface-italic in both the current typeface and size (if it is available).

\LaTeXX_2 supports only the NFSS, version 2 (called NFSS2). For more than a year, the NFSS (initially version 1, and more recently version 2) has been available as an extension for \LaTeXX 2.09. However, in light of stable test releases of \LaTeXX_2, the NFSS2 package for \LaTeXX 2.09 has been withdrawn. The discussion of NFSS in this book applies equally well to \LaTeXX 2.09 with NFSS2, but it is described in terms of \LaTeXX in an effort to be more applicable in the future. The NFSS is discussed in more detail in Chapter 5.

3.2.4 \LaTeXX_2

Leslie Lamport’s \LaTeXX format is probably the most commonly used \TeX format. It is described in \textit{\LaTeX: A Document Preparation System} and many other \TeX books. \LaTeXX_2 is the new standard \LaTeXX. It is described in \textit{The \LaTeXX Companion}. The next edition of \LaTeXX: A Document Preparation System will also describe \LaTeXX_2.

At the time of this writing, \LaTeXX_2 is available only in a test release, but by the time you read this, it is likely to be available as the new standard \LaTeXX. It is described first in this chapter to emphasize that you should begin using \LaTeXX_2 as soon as possible. Once \LaTeXX_2 is out of testing, it will become \LaTeXX, and support for older versions will not be provided (at least not by the \LaTeX developers). \LaTeXX_2 includes a compatibility mode for old \LaTeXX documents so the transition should be relatively painless.

\footnote{Of course, that’s not strictly true. You can change the page layout of \LaTeX (and most other packages) to be almost anything, but it does require learning a lot about the macro package. If you are already familiar with Plain \TeX (or some other Plain \TeX-derived package), you probably have a set of macros that produce documents in the style you like. Why reinvent the wheel?}

\footnote{Extended Plain \TeX is available in the macros/eplain directory in the CTAN archives.}
The central theme of \LaTeX X is ‘structured document preparation.’ An ideal \LaTeX X document is described entirely in terms of its structure: chapters, sections, paragraphs, numbered lists, bulleted items, tables, figures, and all the other elements of a document are identified descriptively. For example, you enclose figures in a figure environment identified by the control sequences \begin{figure} and \end{figure}.

When you are ready to print your document, select an appropriate document class, and \LaTeX X formats your document according to the rules of the selected style. In the case of the ideal document, it might first be printed in a magazine or newsletter using the article class. Later, when it is incorporated into a book, selecting the book class is all that is required to produce appropriate output; the document itself is unchanged.

\LaTeX X is written on top of Plain \TeX. This means that almost any control sequence or macro that you learn about in Plain \TeX can also be used in \LaTeX X. Of course, \LaTeX X insulates you from many Plain \TeX commands by wrapping a much more user-friendly interface around them.

The \LaTeX X\_e input that produces the sample page in Figure 1 is shown in Example 2. The only difference between this document and an old \LaTeX X document is the use of the \documentclass declaration instead of the \documentstyle declaration.\footnote{\LaTeX X\_e will process documents that use \documentstyle in \LaTeX 2.09 compatibility mode.} For more complex documents, other minor changes may also be necessary.

**Example 2: \LaTeX X\_e Input**

```latex
\documentclass{report}
\setlength{\parindent}{0pt}
\setlength{\parskip}{\baselineskip}
\begin{document}
\chapter{Unsolved Problems}
\begin{section}{Odd Perfect Numbers}
A number is said to be \textbf{perfect} if it is the sum of its divisors. For example, 66$ is perfect because 6 is perfect because $1(1+2+3 = 6)$, and $1, 2, 1$, and $3$ are the only numbers that divide evenly into 66$ (apart from 6 itself). It has been shown that all even perfect numbers have the form $2^{(p-1)}(2^{(p)}-1)$ where $p$ and $2^{(p)}-1$ are both prime. The existence of odd perfect numbers is an open question.
\end{section}
\end{document}
```

**Building the \LaTeX X\_e format** The \LaTeX X\_e distribution is available from the directory macros/latex2e/core in the CTAN archives.

The following steps will build the \LaTeX X\_e format. For more complete installation instructions, read the file install.12e in the \LaTeX X\_e distribution.

1. Place the \LaTeX X\_e distribution in a temporary directory and make that directory the current directory. After the installation is complete, you will need to move only selected files into the standard places.

2. Restrict to only the current directory the directories that \TeX X searches for input files

   This can usually be accomplished by setting the environment variable TEXINPUTS\footnote{Under em\TeX X, this variable is called TEXINPUT. On the Macintosh, file searching is frequently controlled by a configuration file or dialog box.} to a single period or the absolute path name of the current directory.

3. Copy the `hyphen.tex` file from the Plain \TeX X distribution into the current directory.

4. Issue the command:

   `\$ initex unpack2e.ins`

   This will unpack all of the distribution files.

5. Build the format file by issuing the command:

   `\$ initex latex2e.ltx`

   Move the resulting files latex2e.fmt and latex2e.log into the \TeX X formats directory.

6. In addition to the files needed to build the format, unpacking the \LaTeX X\_e distribution creates many files that are needed for formatting documents. These files must be placed in a location where \TeX X will find them. However, in order to maintain a functioning \LaTeX 2.09 system, you must not place the new files in the same input directory as the existing files.\footnote{Strictly speaking, this is only true for files that have the extension .sty because the old version of \LaTeX X will not attempt to use the other files.}

   Create a new directory (or folder) for the new files. On the UNIX system that I use, where existing input files are stored in a directory called `usr/local/lib/\texinputs`, \LaTeX 2.09 created `\$usr/local/lib/\texinputs/latex2e` to store the new files. You will have to add the new directory to the front of the list of directories that \TeX X searches for input files whenever you format a document with \LaTeX X\_e.

   Move the files that the installation script produces into the new directory. Move the files `docstrip.tex`, `latexbug.tex`, `sfontdef.ltx`, `slides.ltx`, `testpage.tex`, and all of the files that end in .cfg, .cls, .clo, .def, .fd, and .sty. You should also move the files `g glo.ist` and `gind.ist` someplace where MakeIndex can find them. (MakeIndex is described in Chapter 12 Bibliographies, Indexes, and Glossaries.)

   One of the aspects of the test releases that continues to change is the exact list of files that must be moved. Consult the install.12e file in the distribution for the exact list. The list above is from the test version of January 28, 1994.

\subsection{3.2.5 \LaTeX X}

This section briefly covers \LaTeX X version 2.09. This version of \LaTeX X is still very widely used but it is being phased out.
The \LaTeX{} input to produce the sample page in Figure 1 is shown in Example 3.

Example 3: \LaTeX{} Input File

```latex
\documentstyle{report}
\setlength{\parindent}{0pt}
\begin{document}
\chapter{Unsolved Problems}
\section{Odd Perfect Numbers}
A number is said to be \textit{perfect} if it is the sum of its divisors. For example, $6$ is perfect because $1 + 2 + 3 = 6$, and $1, 2, \text{ and } 3$ are the only numbers that divide evenly into $6$ (apart from 6 itself).

It has been shown that all even perfect numbers have the form $2^{(p-1)}(2^p-1)$ where $p$ and $2^p-1$ are both prime.

The existence of \textit{odd} perfect numbers is an open question.
\end{document}
```

Support for the NFSS in \LaTeX{} 2.09 has been withdrawn. If you need to build a format with support for NFSS, consult the \LaTeX{}2\text{n22}' section of this chapter.

Building the \LaTeX{} format with the OFSS The \LaTeX{} distribution includes three subdirectories, \texttt{sty}, \texttt{doc}, and \texttt{general}. All of the \LaTeX{} files required to build the format file are in the \texttt{general} subdirectory. You will also need the \texttt{hyphen.tex} file required to build the Plain format. In the \texttt{general} subdirectory, the command:

```
$ initex lplain
```

will create the \LaTeX{} format. Move the resulting files, \texttt{lplain.fmt} and \texttt{lplain.log}, to the \TeX{} formats directory. In order to complete the installation, copy the files from the \texttt{sty} directory in the \LaTeX{} distribution into a directory where \TeX{} searches for input files.

3.2.6 AMSTeX

When the American Mathematical Society selected \TeX{} as a document preparation system, they decided to extend it in a number of ways to make the creation of papers and journals easier. They had two goals: to make it easier for authors to write mathematical papers in \TeX{} and to make the resulting papers conform to a particular set of style specifications. AMSTeX is described completely in \textit{The Joy of \TeX{}}.

AMSTeX provides many commands that resemble \LaTeX{} environments. These have the form `\begin{environment}'...

\begin{environment}. In addition, AMSTeX provides the notion of a document style to control style-related formatting issues.

Another important contribution made by the American Mathematical Society when creating AMSTeX was the construction of a large number of new fonts. The American Mathematical Society provides fonts with many more mathematical symbols than the fonts that come with \TeX{}. These fonts are available as a separate package and can be used with any \TeX{} macro package, not just AMSTeX.

The AMSTeX input required to produce the document in Figure 1 is shown in Example 4. The result of formatting this document does not appear exactly like Figure 1 because AMSTeX uses the style conventions of the American Mathematical Society.

\begin{figure}[h]
\centering
\includegraphics[width=0.8\textwidth]{ams-sample-page.png}
\caption{AMS sample page}
\end{figure}

Example 4: AMSTeX Input File

```
\documentstyle{amspppt}
\begin{document}
\section{Odd Perfect Numbers}
A number is said to be \textit{perfect} if it is the sum of its divisors. For example, $6$ is perfect because $1 + 2 + 3 = 6$, and $1, 2, \text{ and } 3$ are the only numbers that divide evenly into $6$ (apart from 6 itself).

It has been shown that all even perfect numbers have the form $2^{(p-1)}(2^p-1)$ where $p$ and $2^p-1$ are both prime.

The existence of \textit{odd} perfect numbers is an open question.
\end{document}
```
1.1 Odd Perfect Numbers

A number is said to be \textit{perfect} if it is the sum of its divisors. For example, $6$ is perfect because $1 + 2 + 3 = 6$, and $1$, $2$, and $3$ are the only numbers that divide evenly into $6$ (apart from $6$ itself).

It has been shown that all even perfect numbers have the form $2^{p-1}(2^p-1)$ where $p$ and $2^p-1$ are both prime.

The existence of \textit{(\textit{it odd\textit{}})} perfect numbers is an open question.

Building the AMSTeX format In order to build the AMSTeX format, you need the \texttt{plain.tex} and \texttt{hyphen.tex} files from Plain TeX and the \texttt{amstex.ini} and \texttt{amstex.tex} files from the AMSTeX distribution.

The command:

\begin{verbatim}
$ initex amstex.ini
\end{verbatim}

will create the AMSTeX format. Move the resulting files, \texttt{amstex.fmt} and \texttt{amstex.log}, into your TeX formats directory.

3.2.7 AMSLaTeX

AMSLaTeX, like AMSTeX, provides many features to make typesetting mathematics convenient while meeting the standards of the American Mathematical Society for publication. However, AMSLaTeX lacks many of the features that are present in \LaTeX{}, like automatically numbered sections and tools for creating tables of contents and indexes.

When \LaTeX{} gained popularity, many authors requested permission to submit articles to the American Mathematical Society in \LaTeX{}. In 1987, the American Mathematical Society began a project to combine the features of AMSTeX with the features of \LaTeX{}. The result is AMSLaTeX.

AMSLaTeX provides all of the functionality of \LaTeX{} because it is an extension of \LaTeX{}. It also provides the functionality of AMSTeX in \LaTeX{} syntax and access to additional mathematical constructs and math symbols not present in \LaTeX{}.

The input required to produce Figure 1 is not shown because they do not differ significantly from the \LaTeX{} sample. Because the sample document doesn’t use any of AMSLaTeX’s additional features, it is exactly the same as the \LaTeX{} document.

Building the AMSTeX format The AMSTeX macros are embodied entirely in style files for \LaTeX{}. It is not necessary to build a special format file. However, the AMSLaTeX macros require the New Font Selection Scheme (NFSS). Consult the section on \LaTeX{}, above, for instructions on building the \LaTeX{} format with NFSS.

3.2.8 Lollipop

It can be argued that \LaTeX{} has the following deficiency: although there are many different style options available, it is not easy for a novice user to change a style option. Changing the internals of most \LaTeX{} style options requires a deep understanding of \TeX{}.

The Lollipop format is very different from the other formats. The central thrust of Lollipop is that it should be easy to change and customize document styles. All Lollipop documents are built from five different generic constructs: headings, lists, text blocks, page grids, and external items.

The Lollipop input required to produce a document like the sample page in Figure 1 is shown in Example 5.

Example 5: Lollipop Input File

\begin{verbatim}
$ Format: Lollipop
\DefineHeading:Chapter
  breakbefore:yes whitesafter:12pt
  line:start PointSize:20 Style:bold
  literal:Chapter Spaces:1 ChapterCounter
  line:stop
  vwhite:36pt
  line:start PointSize:24 Style:bold title
  line:stop
  vwhite:24pt
  Stop
\DefineHeading:Section
  whitebefore:{20pt plus 2pt} whitesafter:14pt
  line:start PointSize:14 Style:bold
  ChapterCounter . SectionCounter
  Spaces:1 title line:stop
  label:start ChapterCounter . SectionCounter
  label:stop
\GoverningCounter:Section=Chapter
\AlwaysIndent:no
\Distance:parskip=12pt
\Distance:hoffset=.75in
\Distance:voffset=.5in
\Start
Chapter Unsolved Problems
Section Odd Perfect Numbers
\Stop
\Section Odd Perfect Numbers

A number is said to be \textit{(\textit{it perfect\textit{}})} if it is the sum of its divisors. For example, \$6\$ is perfect because \$1+2+3=6\$, and \$1\$, \$2\$, and \$3\$ are the only numbers that divide evenly into \$6\$ (apart from \$6\$ itself).

It has been shown that all even perfect numbers have the form \$2^{p-1}(2^p-1)\$ where \$p\$ and \$2^p-1\$ are both prime.

The existence of \textit{(\textit{it odd\textit{}})} perfect numbers is an open question.
\Stop
\end{verbatim}

Building the Lollipop format To make the Lollipop format, you need the Lollipop distribution and the file \texttt{hyphen.tex} from the Plain TeX distribution.

This command will create the Lollipop format:

\begin{verbatim}
$ initex lollipop \dump
\end{verbatim}

It should be performed in the directory where you installed the Lollipop distribution so that all of the Lollipop files can be located. Move the resulting files \texttt{lollipop.fmt} and \texttt{lollipop.log}, into your TeX formats directory.
### 3.2.9 \TeXinfo

The \TeXinfo format is a general-purpose format, but it was designed to support a particular application: to produce both online documentation and professional quality typeset documentation from the same source file. It is discussed in more detail in Chapter 10 *Online Documentation*.

The input file shown in Example 6 produces the typeset output shown in Figure 3. The input file for this example is complicated by the fact that it contains mathematics. None of \TeX’s sophisticated mechanisms for handling mathematics are applicable to plain ASCII online documentation. The online documentation produced by the example in Example 6 is shown in Figure 4.

The \TeXinfo format is the official documentation format of the Free Software Foundation (FSF). Although less commonly used, a \LaTeX variant called \LaTeXinfo is also available.

---

### Unsolved Problems

#### 1 Odd Perfect Numbers

A number is said to be odd if it is the sum of its divisors. For example, 6 is perfect because \(1 + 2 + 3 = 6\), and 1, 2, and 3 are the only numbers that divide evenly into 6 (apart from 6 itself).

It has been shown that all even perfect numbers have the form

\[
2^{p-1}(2^p - 1)
\]

where \(p\) and \(2^p - 1\) are both prime.

The existence of odd perfect numbers is an open question.

---

### Figure 3: \TeXinfo sample page

**Example 6: \TeXinfo Input**

```latex
\input texinfo \ escapes -\ TeXinfo -\ escapes
@setfilename perf-inf.inf
@ifinfo
 @paragraphindent 0
@end ifinfo
@iftex
 @defaultparindent=0pt @parindent=0pt
@end iftex
@node Top, , (dir), (dir)
@chapter Unsolved Problems
@section Odd Perfect Numbers

A number is said to be \textit{perfect} if it is the sum of its divisors. For example, 6 is perfect because

\[1 + 2 + 3 = 6\]

and 1, 2, and 3 are the only numbers that divide evenly into 6 (apart from 6 itself).

It has been shown that all even perfect numbers have the form

\[2^{p-1}(2^p - 1)\]

where \(p\) and \(2^p - 1\) are both prime.

The existence of odd perfect numbers is an open question.
```

---

### Figure 4: Online documentation produced by MakeInfo

**3.2.10 Other Formats**

There are a number of other macro packages available for \TeX. Some of them are summarized below. The fact that they are not discussed more fully here (or listed below, for that matter) is not intended to reflect on the quality of the format. The formats discussed above are examples of the ways in which \TeX can be extended. All of the formats below extend \TeX in a way similar to one of the formats already mentioned. For any particular application, one of these macro packages might be a better choice than the formats discussed above.

**EDMAC** Provides support for typesetting critical editions of texts in a format similar to the Oxford Classical Texts with marginal line numbers and multiple series of footnotes and endnotes keyed by line number. EDMAC is available from the CTAN archives in the directory macros/plain/contrib/edmac.
INRST\TeX\ Provides support for multilingual documents in French and English. INRST\TeX\ is available from the CTAN archives in the directory `macros/inrstex`.

LamST\TeX\ Extends AMST\TeX\ with LaTeX-like features and improved support for commutative diagrams. LamST\TeX\ is available from the CTAN archives in `macros/lamstex`.

REV\TeX\ Extends LaTeX to provide support for typesetting articles for journals of the American Physical Society, the Optical Society of America, and the American Institute for Physics. REV\TeX\ is available in the directory `macros/latex/contrib/revtex` of the CTAN archives.

\TeXsis\ Provides facilities for typesetting articles, papers, and theses. It is particularly tuned for physics papers. \TeXsis\ also provides support for other kinds of documents, such as letters and memos. It is based upon Plain \TeX. \TeXsis\ is available from the CTAN archives in the directory `macros/texsis`.

In addition to REV\TeX\ and \TeXsis, there are several other packages in the `macros` directory on CTAN that were designed for typesetting documents about physics: PHYSYE, PHYZZX, and PSIZZLE.

\TeX/Mathematica\ Supports interactive use of Mathematica on UNIX workstations running GNU emacs. Mathematica explorations can be annotated with \TeX/LaTeX\, and Mathematica graphics can be incorporated into documents. \TeX/Mathematica\ can be obtained from the CTAN archives in the directory `macros/mathematica`.

Script\TeX\ Supports typesetting screenplays in \TeX. Script\TeX\ macro packages/Script\TeX\ is available from the CTAN archives in the directory `macros/scripttex`.

Ver\TeX\ Supports typesetting articles for economic journals. Ver\TeX\ is available from the CTAN archives in the directory `macros/plain/contrib/vertex`.

3.3 Special-purpose Formats

In addition to the general-purpose packages discussed above, there are dozens, if not hundreds, of extensions to \TeX\ that are designed for very specific tasks. Many of the extensions are \LaTeX\ style files; they provide styles for many academic journals, university theses, resumés, diagrams of various sorts, PostScript interfaces, linguistics, multinational language support, UNIX `man' pages, program listings, and countless other tasks.

To give you a feel for the range of tasks that \TeX\ can perform, I've selected a few packages to highlight the latitude of customization that is possible. Figure 5 shows the chemical structure of caffeine (a molecule dear to my heart) rendered with the Chem\TeX\ package. The source is shown in Example 7. Another chemistry package, ChemStruct, was used to draw Figure 6. Its source is shown in Example 8. Taking \TeX\ in another direction, the Music\TeX\ package was used to typeset the first two bars of Mozart’s K545 sonata in C-major in Figure 7. The Music\TeX\ input is shown in Example 9. Several more examples are presented in Chapter 7 where formats for typesetting non-English documents are described.

**Figure 5: Caffeine by Chem\TeX**

```
\begin{verbatim}
\initial
\len=4
\def\H{\hbox{\rm H}}
\def\C{\hbox{\rm C}}
\def\O{\hbox{\rm O}}
\purine{$\H_3\C$}{$\O$}{$\C\H_3$}
{Q}{$\O$}{Q}{Q}{D}{$\C\H_3$} \\
\end{verbatim}
```

**Figure 6: A lithium cation rendered by ChemStruct**

```
\begin{verbatim}
\structure{atom("Li")}{
\side{nwbellow}{atom(O)}
\side{nsingle}{atom(H)}
\side{nsingle}{atom(H)}
\side{nsingle}{atom(H)}
\side{nsingle}{atom(H)}
\side{nsingle}{atom(H)}
\side{nebelow}{atom(O)}
\side{nebelow}{atom(O)}
\side{nebelow}{atom(O)}
\side{esingle}{atom(H)}
\side{esingle}{atom(H)}
\side{esingle}{atom(H)}
\side{esingle}{atom(H)}
\end{verbatim}
```
Another popular special-purpose application of \TeX{} is the production of transparencies, also called foils or slides. There are a few different options for this application.

### \section{3.3.1 Sli\TeX{}}

Sli\TeX{} is part of the standard \LaTeX{} distribution. Input to Sli\TeX{} consists of a main file and a slides file. Individual slides are composed in a slide environment.

Sli\TeX{} has provisions for black-and-white slides, color slides, and overlays. Unlike the other slide-making formats, which rely on special printer commands\footnote{The \texttt{\textbackslash{special}} mechanism is a way of passing arbitrary information through \TeX{} to the DVI driver that will ultimately print the document.} to incorporate color, Sli\TeX{} produces separate output pages for each color. For example, if you use red to highlight words on an otherwise black-and-white slide, Sli\TeX{} will produce two output pages for the slide: one with all the black text (excluding the words in red) and one with just the red words. Both of these pages will be printed in black.

You must construct the colored slide by copying the pages onto colored transparencies and overlaying them. Producing slides with overlays in the same color is accomplished with a special ‘invisible’ color. This method of producing colored transparencies has been made obsolete by modern color printers. Of course, nothing prevents you from using the \texttt{\textbackslash{special}} extensions of your DVI driver in Sli\TeX{} to produce colored transparencies directly on a color printer.

Sli\TeX{} has always been a separate macro package, distinct from and not 100\% compatible with \LaTeX{}. With the introduction of \LaTeX{}2ε, Sli\TeX{} is simply an extension package to the \LaTeX{}2ε core format. Support for a separate Sli\TeX{} format is being phased out.

### \section{3.3.2 Foil\TeX{}}

Foil\TeX{} is an extension of \LaTeX{} for producing slides. The primary advantage of Foil\TeX{} over Sli\TeX{} is that it is completely compatible with \LaTeX{}.\footnote{With the caveat that it still uses the Old Font Selection Scheme.} Note, however, that the defaults in many cases are not precisely the same as \LaTeX{} because of the radically different goal of Foil\TeX{}.

Foil\TeX{} provides support for running headers and footers, modified theorem environments for better mathematics in slides, support for AMS and PostScript fonts, and colors.

Color slides in Foil\TeX{} are handled by DVI special commands (most commonly \texttt{dvips} \texttt{\textbackslash{specials}}). However, Foil\TeX{} includes a number of new and extended features for better handling of color. See the section ‘\TeX{} in Color’ later in this chapter for a detailed discussion of using color in \TeX{}.

### \section{3.3.3 Seminar}

The Seminar style is another alternative for producing slides and notes. Like Foil\TeX{}, the Seminar style is designed to work on top of \LaTeX{}\footnote{Seminar also works with AMS\LaTeX{}.} (Seminar also works with AMS\LaTeX{}.) The Seminar style is designed to produce output for a PostScript printer. It isn’t strictly necessary to produce PostScript output, but if you do not, many of Seminar’s features will be unavailable to you.

Seminar provides for a mixture of portrait and landscape styles and can support color using either a color-separation technique (similar to Sli\TeX{}’s method) or direct use of PostScript color. In either case, the PSTricks macro package is required. (Consult the section ‘PSTricks’ in Chapter 6, \textit{Pictures and Figures}, for more information.)

The Seminar style has support for a number of interesting options (including two-up printing of slides), automatic resizing by changing a magnification parameter, instructions for converting your Sli\TeX{} slides, and several extensive demonstration files. Also included are explicit instructions for placing Encapsulated PostScript drawings in your slides.
3.4 \TeX in Color

With color printers and copiers becoming more common, the application of color, especially in transparencies, is more important than ever. Unfortunately, \TeX knows nothing about color.

A little reflection about the design of \TeX will make it clear why this is the case. \TeX produces device-independent output. Even when color printers are as common as \textquote{regular} printers, if that ever becomes the case, it will always be true that color is an inherently device-dependent attribute. It does not make sense for \TeX to understand color. However, this does not prevent \TeX from using color.

At the lowest level, all that is required to use color in \TeX is some way of telling the printer \textquote{start printing in color here.} This is easily accomplished with a special command. In the discussion that follows, the \texttt{dvips} special commands are used as concrete examples, but conceptually, any color printer can be used in this way.

3.4.1 Setting Up Color

Color support at the DVI driver level is provided by special commands, but these are not typically convenient to enter directly into your document. Frequently, these commands are specified in terms of percentages of red, green, and blue (RGB color) or cyan, magenta, yellow, and black (CMYK color).

Higher-level support is provided by a collection of color control sequences. These sequences are loaded either by inputting the file \texttt{colordvi.tex} (in Plain \TeX, for example) or using the \texttt{colorart} style file (in LATEX). \texttt{dvips} defines the colors in terms of \textquote{crayon names.} If you need very precise control of the colors, you can adjust the precise mix of CMYK values in the file \texttt{colordvi.tex} after comparing the output of your printer with a standard scale (typically, the PANTONE scale). The following color names are standard in \texttt{dvips}.

<table>
<thead>
<tr>
<th>Color</th>
<th>Color Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apricot</td>
<td>Emerald</td>
</tr>
<tr>
<td>Aquamarine</td>
<td>ForestGreen</td>
</tr>
<tr>
<td>Bittersweat</td>
<td>Fuchsia</td>
</tr>
<tr>
<td>Black</td>
<td>Goldenrod</td>
</tr>
<tr>
<td>Blue</td>
<td>Gray</td>
</tr>
<tr>
<td>BlueGreen</td>
<td>Green</td>
</tr>
<tr>
<td>BlueViolet</td>
<td>GreenYellow</td>
</tr>
<tr>
<td>BrickRed</td>
<td>JungleGreen</td>
</tr>
<tr>
<td>Brown</td>
<td>Lavender</td>
</tr>
<tr>
<td>BurntOrange</td>
<td>LimeGreen</td>
</tr>
<tr>
<td>CadetBlue</td>
<td>Magenta</td>
</tr>
<tr>
<td>CarnationPink</td>
<td>Mahogany</td>
</tr>
<tr>
<td>Cerulean</td>
<td>Maroon</td>
</tr>
<tr>
<td>CornflowerBlue</td>
<td>Melon</td>
</tr>
<tr>
<td>Cyan</td>
<td>MidnightBlue</td>
</tr>
<tr>
<td>Dandelion</td>
<td>Mulberry</td>
</tr>
<tr>
<td>DarkOrchid</td>
<td>NavyBlue</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Color</th>
<th>Color Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>OliveGreen</td>
<td>Orange</td>
</tr>
<tr>
<td>OrangeRed</td>
<td>SeaGreen</td>
</tr>
<tr>
<td>Orange</td>
<td>Peach</td>
</tr>
<tr>
<td>Tan</td>
<td>TealBlue</td>
</tr>
<tr>
<td>OliveRed</td>
<td>ProcessBlue</td>
</tr>
<tr>
<td>Purple</td>
<td>Thistle</td>
</tr>
<tr>
<td>Purple</td>
<td>Turquoise</td>
</tr>
<tr>
<td>RawSienna</td>
<td>Violet</td>
</tr>
<tr>
<td>Red</td>
<td>VioletRed</td>
</tr>
<tr>
<td>RedOrange</td>
<td>White</td>
</tr>
<tr>
<td>RedViolet</td>
<td>WildStrawberry</td>
</tr>
<tr>
<td>Yellow</td>
<td>Rhodamine</td>
</tr>
<tr>
<td>YellowGreen</td>
<td>RoyalBlue</td>
</tr>
<tr>
<td>YellowOrange</td>
<td>RoyalPurple</td>
</tr>
</tbody>
</table>

For example, to typeset something in red, use the \texttt{\textcolor{Red}{something in red}} control sequence in your document, like this:

\texttt{\textcolor{Red}{something in red}}

Alternatively, you can change the default text color with the \texttt{\textcolor{Blue}{
\textcolor{Yellow}{\textcolor{Purple}{\textcolor{Red}{\textcolor{Green}{\textcolor{Cyan}{\textcolor{Magenta}{\textcolor{Maroon}{\textcolor{Melon}{\textcolor{MidnightBlue}{\textcolor{Mulberry}{\textcolor{NavyBlue}{\textcolor{RoyalPurple}{\textcolor{YellowGreen}{\textcolor{YellowOrange}{\texttt{\textcolor{Red}{something in red}}}}}}}}}}}}}}}}}}

To change the background color, use the \texttt{\background{Yellow}} macro. For example, to make the current and all future pages yellow, enter:

\texttt{\background{Yellow}}

You can enter a precise color by specifying it in terms of its CMYK components. The \texttt{\Color{.25 .35 .4 .1}{some text}} and \texttt{\textcolor{Blue}{\textcolor{Yellow}{\textcolor{Purple}{\textcolor{Red}{\textcolor{Green}{\textcolor{Cyan}{\textcolor{Magenta}{\textcolor{Maroon}{\textcolor{Melon}{\textcolor{MidnightBlue}{\textcolor{Mulberry}{\textcolor{NavyBlue}{\textcolor{RoyalPurple}{\textcolor{YellowGreen}{\textcolor{YellowOrange}{\texttt{\textcolor{Red}{something in red}}}}}}}}}}}}}}}}

3.4.3 Now I've Got Color, but I Need Black and White!

If you have reason to print a colored \TeX document on a black and white printer, you don’t have to tear out all of the color commands. \texttt{dvips} includes a \texttt{blackdvi} file (analogous to \texttt{colordvi}—an input file or style file depending on your macro package), which translates all color commands into black and white.

Alternatively, \textquote{good} implementations of PostScript in a black and white printer should translate all colors into shades of grey. This can be an inexpensive way to preview a color document. Most screen previewers simply ignore color commands so they print in black and white even if the document is colored.

3.4.4 Color Under LATEX

At the time of this writing, the LATEX team has not officially adopted a standard for using color. However, it is likely to follow a slightly different model than the one described above. The final design should provide color selection commands that are device-independent at the DVI driver level (in other words, the color commands will not insert device-specific commands, like snippets of PostScript, into the DVI files).

3.4.2 Using Color

After \texttt{dvips} has loaded \texttt{colordvi}, typesetting text in color is simply a matter of using the appropriate color control

\texttt{I made these numbers up. I take no responsibility for the artistic merits (or lack thereof) of the resulting color.}
3.4.5 Color Is Subtle
Color commands implemented as `\special commands` may introduce occasional problems. For example, if \TeX introduces a page break in a paragraph that you have typeset in yellow (`\Yellow{This is a long paragraph...}`), the resulting output may print the page footer (and even the header) in yellow, although that was not intended.

Circumventing these problems may require careful use of color commands in front of text that you want to appear black. For example, in Plain \TeX the difficulty described above can be avoided by specifying that the page number should be printed this way:

```
\footline{\Black{\hss\tenrm\folio\hss}}
```

This definition guarantees that the page number will be set in black, and because it is a local color change, colored text can flow across the page around it.

You may want to make sure that other typographic elements are printed in the current global color (which may vary). \dvips provides a local color macro called `\globalColor` for that purpose. Every time the text color is changed globally (with a `\textColor` command), `\globalColor` is redefined to print text in that color.

3.4.6 Further Reading
Read the documentation for your DVI driver carefully with respect to color. Because it is device-dependent, there is a lot of room for interpretation, and it may not always be obvious why some things appear the way they do. And DVI drivers are free to implement color in any way they choose.