

contents

foreword xi
preface xiii
about the cover illustration xviii

PART I FOUNDATIONS..... 1

- 1** ***Data, data munging, and Perl 3***
- 1.1 What is data munging? 4
 - Data munging processes 4* ▪ *Data recognition 5*
 - Data parsing 6* ▪ *Data filtering 6* ▪ *Data transformation 6*
 - 1.2 Why is data munging important? 7
 - Accessing corporate data repositories 7* ▪ *Transferring data between multiple systems 7* ▪ *Real-world data munging examples 8*
 - 1.3 Where does data come from? Where does it go? 9
 - Data files 9* ▪ *Databases 10* ▪ *Data pipes 11*
 - Other sources/sinks 11*
 - 1.4 What forms does data take? 12
 - Unstructured data 12* ▪ *Record-oriented data 13*
 - Hierarchical data 13* ▪ *Binary data 13*
 - 1.5 What is Perl? 14
 - Getting Perl 15*

- 1.6 Why is Perl good for data munging? 16
- 1.7 Further information 17
- 1.8 Summary 17

2 General munging practices 18

- 2.1 Decouple input, munging, and output processes 19
- 2.2 Design data structures carefully 20
 - Example: the CD file revisited 20*
- 2.3 Encapsulate business rules 25
 - Reasons to encapsulate business rules 26* ▪ *Ways to encapsulate business rules 26* ▪ *Simple module 27*
 - Object class 28*
- 2.4 Use UNIX “filter” model 31
 - Overview of the filter model 31* ▪ *Advantages of the filter model 32*
- 2.5 Write audit trails 36
 - What to write to an audit trail 36* ▪ *Sample audit trail 37* ▪ *Using the UNIX system logs 37*
- 2.6 Further information 38
- 2.7 Summary 38

3 Useful Perl idioms 39

- 3.1 Sorting 40
 - Simple sorts 40* ▪ *Complex sorts 41* ▪ *The Orcish Manoeuvre 42* ▪ *Schwartzian transform 43*
 - The Guttman-Rosler transform 46* ▪ *Choosing a sort technique 46*
- 3.2 Database Interface (DBI) 47
 - Sample DBI program 47*
- 3.3 Data::Dumper 49
- 3.4 Benchmarking 51
- 3.5 Command line scripts 53

3.6 Further information 55

3.7 Summary 56

4 *Pattern matching* 57

4.1 String handling functions 58

Substrings 58 ▪ *Finding strings within strings (index and rindex)* 59 ▪ *Case transformations* 60

4.2 Regular expressions 60

What are regular expressions? 60 ▪ *Regular expression syntax* 61 ▪ *Using regular expressions* 65 ▪ *Example: translating from English to American* 70 ▪ *More examples: /etc/passwd* 73 ▪ *Taking it to extremes* 76

4.3 Further information 77

4.4 Summary 78

PART II DATA MUNGING 79

5 *Unstructured data* 81

5.1 ASCII text files 82

Reading the file 82 ▪ *Text transformations* 84
Text statistics 85

5.2 Data conversions 87

Converting the character set 87 ▪ *Converting line endings* 88 ▪ *Converting number formats* 90

5.3 Further information 94

5.4 Summary 95

6 *Record-oriented data* 96

6.1 Simple record-oriented data 97

Reading simple record-oriented data 97 ▪ *Processing simple record-oriented data* 100 ▪ *Writing simple record-oriented data* 102 ▪ *Caching data* 105

- 6.2 Comma-separated files 108
 - Anatomy of CSV data* 108 ▪ *Text::CSV_XS* 109
- 6.3 Complex records 110
 - Example: a different CD file* 111
 - Special values for \$/* 113
- 6.4 Special problems with date fields 114
 - Built-in Perl date functions* 114
 - Date::Calc* 120 ▪ *Date::Manip* 121
 - Choosing between date modules* 122
- 6.5 Extended example: web access logs 123
- 6.6 Further information 126
- 6.7 Summary 126

7 **Fixed-width and binary data** 127

- 7.1 Fixed-width data 128
 - Reading fixed-width data* 128 ▪ *Writing fixed-width data* 135
- 7.2 Binary data 139
 - Reading PNG files* 140 ▪ *Reading and writing MP3 files* 143
- 7.3 Further information 144
- 7.4 Summary 145

PART III SIMPLE DATA PARSING..... 147

8 **Complex data formats** 149

- 8.1 Complex data files 150
 - Example: metadata in the CD file* 150 ▪ *Example: reading the expanded CD file* 152
- 8.2 How not to parse HTML 154
 - Removing tags from HTML* 154 ▪ *Limitations of regular expressions* 157

- 8.3 Parsers 158
 - An introduction to parsers* 158 ▪ *Parsers in Perl* 161
- 8.4 Further information 162
- 8.5 Summary 162

9 HTML 163

- 9.1 Extracting HTML data from the World Wide Web 164
- 9.2 Parsing HTML 165
 - Example: simple HTML parsing* 165
- 9.3 Prebuilt HTML parsers 167
 - HTML::LinkExtor* 167 ▪ *HTML::TokeParser* 169
 - HTML::TreeBuilder* and *HTML::Element* 171
- 9.4 Extended example: getting weather forecasts 172
- 9.5 Further information 174
- 9.6 Summary 174

10 XML 175

- 10.1 XML overview 176
 - What's wrong with HTML?* 176 ▪ *What is XML?* 176
- 10.2 Parsing XML with XML::Parser 178
 - Example: parsing weather.xml* 178 ▪ *Using XML::Parser* 179 ▪ *Other XML::Parser styles* 181
 - XML::Parser handlers* 188
- 10.3 XML::DOM 191
 - Example: parsing XML using XML::DOM* 191
- 10.4 Specialized parsers—XML::RSS 193
 - What is RSS?* 193 ▪ *A sample RSS file* 193
 - Example: creating an RSS file with XML::RSS* 195
 - Example: parsing an RSS file with XML::RSS* 196
- 10.5 Producing different document formats 197
 - Sample XML input file* 197 ▪ *XML document transformation script* 198 ▪ *Using the XML document transformation script* 205

- 10.6 Further information 208
- 10.7 Summary 208

11 Building your own parsers 209

- 11.1 Introduction to Parse::RecDescent 210
 - Example: parsing simple English sentences* 210
- 11.2 Returning parsed data 212
 - Example: parsing a Windows INI file* 212
 - Understanding the INI file grammar* 213
 - Parser actions and the @item array* 214
 - Example: displaying the contents of @item* 214
 - Returning a data structure* 216
- 11.3 Another example: the CD data file 217
 - Understanding the CD grammar* 218 ▪ *Testing the CD file grammar* 219 ▪ *Adding parser actions* 220
- 11.4 Other features of Parse::RecDescent 223
- 11.5 Further information 224
- 11.6 Summary 224

PART IV THE BIG PICTURE 225

12 Looking back—and ahead 227

- 12.1 The usefulness of things 228
 - The usefulness of data munging* 228 ▪ *The usefulness of Perl* 228 ▪ *The usefulness of the Perl community* 229
- 12.2 Things to know 229
 - Know your data* 229 ▪ *Know your tools* 230
 - Know where to go for more information* 230

appendix A Modules reference 232
appendix B Essential Perl 254
index 273