CONTENTS IN DETAIL

FOREWORD FOR THE FIRST EDITION by Ralf Wildenhues **xix**

FOREWORD FOR THE SECOND EDITION by Eric Blake **xxi**

PREFACE **xxiii**

Why Use the Autotools? xxiv
Acknowledgments for the First Edition xxvi
Acknowledgments for the Second Edition xxvi
I Wish You the Very Best xxvi

INTRODUCTION **xxvii**

Who Should Read This Book xxviii
How This Book Is Organized xxviii
Conventions Used in This Book xxx
Autotools Versions Used in This Book xxxi

1
AN END USER'S PERSPECTIVE ON THE GNU AUTOTOOLS **1**

Software Source Archives 2
Unpacking a Source Archive 3
Building the Software 4
Testing the Build 7
Installing the Built Software 9
Summary 11

2
A BRIEF INTRODUCTION TO THE GNU AUTOTOOLS **13**

Who Should Use the Autotools? 14
When Should You Not Use the Autotools? 14
Apple Platforms and Mac OS X 15
The Choice of Language 16
Generating Your Package Build System 17
Configuration 18
Autoconf 18
    autoreconf 19
    autoheader 20
    autoscan 20
    autoupdate 20
    ifnames 21
    autom4te 21
    Working Together 21

Automake 22
automake 23
aclocal 23
Libtool 24
libtool 25
libtoolize 25
ltdl, the Libtool C API 25
Building Your Package 26
Running configure 26
Running make 29
Installing the Most Up-to-Date Autotools 30
Summary 33

3
UNDERSTANDING THE GNU CODING STANDARDS 35

Creating a New Project Directory Structure 36
Project Structure 37
Makefile Basics 39
Rules 39
Variables 41
A Separate Shell for Each Command 42
Variable Binding 43
Rules in Detail 44
Resources for Makefile Authors 50
Creating a Source Distribution Archive 50
Forcing a Rule to Run 52
Leading Control Characters 53
Automatically Testing a Distribution 54
Unit Testing, Anyone? 55
Installing Products 56
Installation Choices 58
Uninstalling a Package 60
Testing Install and Uninstall 61
The Filesystem Hierarchy Standard 63
Supporting Standard Targets and Variables 64
Standard Targets 64
Standard Variables 65
Adding Location Variables to Jupiter 66
Getting Your Project into a Linux Distro 67
Build vs. Installation Prefix Overrides 69
User Variables 71
Nonrecursive Build Systems 73
Configuring Your Package 77
Summary 78

4
CONFIGURING YOUR PROJECT WITH AUTOCONF 79

Autoconf Configuration Scripts 80
The Shortest configure.ac File 82
Comparing M4 to the C Preprocessor 82

The Nature of M4 Macros . . . 83
Executing autoconf . . . 84
Executing configure . . . 85
Executing config.status . . . 86
Adding Some Real Functionality . . . 87
Generating Files from Templates . . . 90
Adding VPATH Build Functionality . . . 91
Let's Take a Breather . . . 94
An Even Quicker Start with autoscan . . . 95
    The Proverbial bootstrap.sh Script . . . 97
    Updating Makefile.in . . . 99
Initialization and Package Information . . . 100
    AC_PREREQ . . . 100
    AC_INIT . . . 100
    AC_CONFIG_SRCDIR . . . 101
The Instantiating Macros . . . 102
    Generating Header Files from Templates . . . 107
    Using autoheader to Generate an Include File Template . . . 108
Back to Remote Builds for a Moment . . . 111
Summary . . . 112

5
MORE FUN WITH AUTOCONF: CONFIGURING USER OPTIONS 113

Substitutions and Definitions . . . 114
    AC_SUBST . . . 114
    AC_DEFINE . . . 115
Checking for Compilers . . . 116
Checking for Other Programs . . . 117
A Common Problem with Autoconf . . . 119
Checks for Libraries and Header Files . . . 123
    Is It Right or Just Good Enough? . . . 126
    Printing Messages . . . 131
Supporting Optional Features and Packages . . . 132
    Coding Up the Feature Option . . . 134
    Formatting Help Strings . . . 137
Checks for Type and Structure Definitions . . . 138
The AC_OUTPUT Macro . . . 141
Summary . . . 143

6
AUTOMATIC MAKEFILES WITH AUTOMAKE 145

Getting Down to Business . . . 146
    Enabling Automake in configure.ac . . . 147
    A Hidden Benefit: Automatic Dependency Tracking . . . 151
What's Actually in a Makefile.am File? . . . 153
Analyzing Our New Build System . . . 154
    Product List Variables . . . 155
    Product Source Variables . . . 160
    PLV and PSV Modifiers . . . 161

Unit Tests: Supporting make check . . . 162
Reducing Complexity with Convenience Libraries . . . 164
    Product Option Variables . . . 167
    Per-Makefile Option Variables . . . 169
Building the New Library . . . 169
What Goes into a Distribution? . . . 171
Maintainer Mode . . . 172
Cutting Through the Noise . . . 173
Nonrecursive Automake . . . 175
Summary . . . 177

7
BUILDING LIBRARIES WITH LIBTOOL 179

The Benefits of Shared Libraries . . . 180
How Shared Libraries Work . . . 181
    Dynamic Linking at Load Time . . . 181
Using Libtool . . . 185
    Abstracting the Build Process . . . 185
    Abstraction at Runtime . . . 186
Installing Libtool . . . 187
Adding Shared Libraries to Jupiter . . . 188
    Using the LTLIBRARIES Primary . . . 188
    Public Include Directories . . . 189
    Customizing Libtool with LT_INIT Options . . . 192
    Reconfigure and Build . . . 197
    So What Is PIC, Anyway? . . . 200
    Fixing the Jupiter PIC Problem . . . 203
Summary . . . 207

8
LIBRARY INTERFACE VERSIONING AND RUNTIME DYNAMIC LINKING 209

System-Specific Versioning . . . 210
    Linux and Solaris Library Versioning . . . 210
    IBM AIX Library Versioning . . . 212
    Microsoft DLL Versioning . . . 214
    HP-UX/AT&T SVR4 Library Versioning . . . 215
The Libtool Library Versioning Scheme . . . 216
    Library Versioning Is Interface Versioning . . . 216
    When Library Versioning Just Isn't Enough . . . 220
Using libltdl . . . 221
    Necessary Infrastructure . . . 221
    Adding a Plug-In Interface . . . 222
    Doing It the Old-Fashioned Way . . . 223
Converting to Libtool's ltdl Library . . . 228
    Preloading Multiple Modules . . . 232
    Checking It All Out . . . 233
Summary . . . 234

9
UNIT AND INTEGRATION TESTING WITH AUTOTEST 235

Autotest Overview 238
Wiring Up Autotest 241
Adding a Test 248
Defining Tests with AT_CHECK 250
Defining Test Groups with AT_SETUP and AT_CLEANUP 251
So What Happened?. . . . 255
Unit Testing vs. Integration Testing 257
Administrative Details 261
Distributing Test Files 261
Checking Installed Products 262
Cleaning Up. . . . 266
Niceties 267
A Minimal Approach 268
Summary 268

10
FINDING BUILD DEPENDENCIES WITH PKG-CONFIG 271

A pkg-config Overview 272
Diving In 274
Writing pkg-config Metadata Files 276
Informational Fields 278
Functional Fields 279
Generating .pc Files with Autoconf. . . . 282
Generating pc Files from pc.in Templates 282
Generating .pc Files with make. . . . 283
Uninstalled .pc Files 285
Using pkg-config in configure.ac 287
pkg-config Autoconf Macros 290
Summary 292

11
INTERNATIONALIZATION 293

Obligatory Disclaimer. . . . 294
Internationalization (I18n) 295
Instrumenting Source Code for Dynamic Messages 296
Instrumenting Source Code for Static Messages 325
Summary 329

12
LOCALIZATION 331

Getting Started 331
Language Selection 332
Building Message Catalogs 334

Integrating gettext with the Autotools 339
    What Should Be Committed? 348
    Adding a Language 349
    Installing Language Files 350
    Manual make Targets 351
Summary 352

13
MAXIMUM PORTABILITY WITH GNULIB 353

License Caveat. 354
Getting Started 354
Adding Gnulib Modules to a Project. 355
Summary 365

14
FLAIM: AN AUTOTOOLS EXAMPLE 367

What Is FLAIM? 368
Why FLAIM? 368
Logistics 369
An Initial Look 370
Getting Started 372
    Adding the configure.ac Files 372
    The Top-Level Makefile.am File 376
The FLAIM Subprojects 378
    The FLAIM Toolkit configure.ac File 379
    The FLAIM Toolkit Makefile.am File 388
    Designing the ftk/src/Makefile.am File 391
    Moving On to the ftk/util Directory 393
Designing the XFLAIM Build System 394
    The XFLAIM configure.ac File 394
    Creating the xflaim/src/Makefile.am File. 398
    Turning to the xflaim/util Directory 399
Summary 405

15
FLAIM PART II:
PUSHING THE ENVELOPE 407

Building Java Sources Using the Autotools. 408
    Autotools Java Support 408
    Using ac-archive Macros 411
    Canonical System Information. 412
    The xflaim/java Directory Structure 413
    The xflaim/src/Makefile.am File 414
    Building the JNI C++ Sources 415
    The Java Wrapper Classes and JNI Headers 416
    A Caveat About Using the JAVA Primary 418

Building the C# Sources 418
    Manual Installation 421
    Cleaning Up Again 422
Configuring Compiler Options 422
Hooking Doxygen into the Build Process 424
Adding Nonstandard Targets 426
Summary 429

16
USING THE M4 MACRO PROCESSOR WITH AUTOCONF 431

M4 Text Processing 432
    Defining Macros 433
    Macros with Arguments 435
The Recursive Nature of M4 436
    Infinite Recursion 438
    Quoting Rules 438
Autoconf and M4 439
Writing Autoconf Macros 441
    Simple Text Replacement 441
    Documenting Your Macros 444
    M4 Conditionals 445
Diagnosing Problems 449
Summary 450

17
USING THE AUTOTOOLS WITH WINDOWS 451

Environment Options 452
Tool Chain Options 452
Getting Started 453
Cross-Compiling for Windows on Linux. 454
    Installing a Windows Cross Tool Chain 454
    Testing the Build 455
Windows Subsystem for Linux 460
Cygwin. 462
    Installing Cygwin 464
    Opening the Cygwin Terminal 470
    Testing the Build 471
    Building True Native Windows Software 474
    Analyzing the Software 476
MinGW: Minimalist GNU for Windows 477
    Installing MinGW 478
    Testing the Build 485
Msys2 487
    What's Msys? 488
    Installing Msys2 488
    Installing Tools 493
    Testing the Build 495
Summary 497

18
A CATALOG OF TIPS AND REUSABLE SOLUTIONS FOR CREATING GREAT PROJECTS 499

Item 1: Keeping Private Details out of Public Interfaces 499
Solutions in C 501
Solutions in C++ 502
Item 2: Implementing Recursive Extension Targets. 505
Item 3: Using a Repository Revision Number in a Package Version 508
Item 4: Ensuring Your Distribution Packages Are Clean 510
Item 5: Hacking Autoconf Macros 511
Providing Library-Specific Autoconf Macros 516
Item 6: Cross-Compiling 517
Item 7: Emulating Autoconf Text Replacement Techniques 523
Item 8: Using the Autoconf Archive Project 528
Item 9: Using Incremental Installation Techniques 529
Item 10: Using Generated Source Code 529
Using the BUILT_SOURCES Variable 529
Dependency Management 530
Built Sources Done Right 533
Item 11: Disabling Undesirable Targets 536
Item 12: Watch Those Tab Characters!. 537
Item 13: Packaging Choices 539
Wrapping Up 540

INDEX 541