

Open Source Development with CVS

3RD EDITION

Moshe Bar
Karl Fogel





Open Source Development with CVS, 3RD EDITION

Copyright © 2003 Karl Fogel and Paraglyph Press.

You can redistribute and/or modify this book under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This book is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this book; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA.

Paraglyph Press, Inc.
4015 N. 78th Street, #115
Scottsdale, Arizona 85251
Phone: 602-749-8787
www.paraglyphpress.com

Paraglyph Press ISBN: 1-932111-81-6

President

Keith Weiskamp

Editor-at-Large

Jeff Duntemann

**Vice President, Sales,
Marketing, and
Distribution**

Steve Sayre

**Vice President, International
Sales and Marketing**

Cynthia Caldwell

Production Manager

Kim Eoff

Cover Designer

Kris Sotelo

Printed in the United States of America
10 9 8 7 6 5 4 3 2 1





The Paraglyph Mission

This book you've purchased is a collaborative creation involving the work of many hands, from authors to editors to designers and to technical reviewers. At Paraglyph Press, we like to think that everything we create, develop, and publish is the result of one form creating another. And as this cycle continues on, we believe that your suggestions, ideas, feedback, and comments on how you've used our books is an important part of the process for us and our authors.

We've created Paraglyph Press with the sole mission of producing and publishing books that make a difference. The last thing we all need is yet another tech book on the same tired, old topic. So we ask our authors and all of the many creative hands who touch our publications to do a little extra, dig a little deeper, think a little harder, and create a better book. The founders of Paraglyph are dedicated to finding the best authors, developing the best books, and helping you find the solutions you need.

As you use this book, please take a moment to drop us a line at feedback@paraglyphpress.com and let us know how we are doing—and how we can keep producing and publishing the kinds of books that you can't live without.

Sincerely,

Keith Weiskamp & Jeff Duntemann
Paraglyph Press Founders
4015 N. 78th Street, #115
Scottsdale, Arizona 85251
email: feedback@paraglyphpress.com
Web: www.paraglyphpress.com
Phone: 602-749-8787



Recently Published by Paraglyph Press:

Jeff Duntemann's Drive-By Wi-Fi Guide

By Jeff Duntemann

Visual Basic .NET Black Book

By Steven Holzner

C++ Black Book

By Steven Holzner

C# Core Language Little Black Book

By Bill Wagner

The SQL Server 2000 Book

*By Anthony Sequeira
And Brian Alderman*

The Mac OS X.2 Power User's Book

By Gene Steinberg and Pieter Paulson

Mac OS X v.2 Jaguar Little Black Book

By Gene Steinberg

The Mac OS X.2 Jaguar Book

By Mark R. Bell

Game Coding Complete

By Mike McShaffry

Monster Gaming

By Ben Sawyer

Looking Good in Print, 5th Edition

By Roger C. Parker





To Yisrael—The Land, the People, and its Torah
—Moshe Bar



This book is dedicated with love to my parents, Frances and Henry, for everything. Literally.

—Karl Fogel





About the Authors

Moshe Bar, has an M.Sc. and Ph.D. in computer science and teaches advanced operating systems courses at Tel Aviv University and some European universities. Over the last ten years he has contributed to several open source projects, such as the Linux kernel, the JFS file system for Linux, and most prominent, openMosix. He has authored books on the Linux kernel and its file systems. Moshe is also Chief Technology Officer and co-founder of Qlusters, Inc., a clustering software company in the Silicon Valley.

Next to programming, Moshe also works as senior editor for *BYTE Magazine* as well as for several other computer journals. Whenever he is not working, Moshe can be spotted on one of his custom motorcycles. Currently, he enjoys his brand-new Harley-Davidson Road King, next to his Yamaha RoadStar Classic 1100.

Karl Fogel was born in 1971 and managed to make it all the way through the '80s personal computer and BBS craze without learning a thing about computers, networks, or email. In this state of technological ignorance—which he has been trying ever since to regain—he headed off to Oberlin College/Conservatory of Music in 1991 to study the piano, but ended up with a degree in Chinese and an accidental education in computer programming.

In 1995 he and Jim Blandy started Cyclic Software, to provide maintenance and commercial support for CVS. After they sold Cyclic, he headed to southwest China and taught English and Unix/C programming for a year. He now lives in Chicago, working as a free software programmer for CollabNet on the Subversion project, a new revision control system intended to succeed CVS.

In his copious spare time, he is careful to avoid any contact with computers; instead, he interacts with live human beings and plays the piano.



Acknowledgments

The writing of this book, as for every book written, took a toll on social and family life. Avivit always showed patience when the book took first priority on many weekends and evenings. Thank you.

Finally, I need to thank the people who made me learn how to use CVS for my daily development and sysadmin work: the good folks at SAP Portals, Baan Development, and last but not least, the fantastic world of open source where I learned—and still continue to learn—the dynamics of contribution and open source project management.

I feel I am living in a very special time and I am very glad to be one of OpenSource's participants. Next to the obvious stars like Linus Torvalds, Jordan Hubbard, and others, a great deal of other, lesser known, but equally important programmers make OpenSource the economic power that it is today. My appreciation goes to these lesser known contributors in the same measure as for the well-known stars.


—*Moshe Bar*





Contents at a Glance

Chapter 1	Why Open Source Development and CVS Go Together	1
Chapter 2	An Overview of CVS	17
Chapter 3	CVS Repository Administration	87
Chapter 4	Advanced CVS	125
Chapter 5	Tips and Troubleshooting	171
Chapter 6	The Development Process	187
Chapter 7	The Open Source Process	203
Chapter 8	Designing for Decentralized Development	225
Chapter 9	Third-Party Tools that Work with CVS	239
Chapter 10	Complete CVS Reference	255
Chapter 11	CVS versus BitKeeper—A Comparison	307
Appendix A	GNU General Public License	315
Appendix B	GNU Free Documentation License	323
Appendix C	Bibliography	331







Contents

Introduction	xvii
Chapter 1 Why Open Source Development and CVS Go Together	1
What Is Free Software? 1	
Open Source Software 2	
Open Source Licenses 3	
Open Source Business Models 4	
How It All Started 5	
Stallman's Idea 5	
The Two Types of Development 6	
What Does CVS Have to Do with It? 7	
diff and patch 8	
RCS 9	
The Winner: CVS 9	
Principles of Open Source Development and How CVS Helps 10	
What Makes It All Tick? 12	
Necessity 12	
Community 12	
Glory 13	
Money 13	
Factionalism as a Sign of Strength 14	
Chapter 2 An Overview of CVS	17
CVS Basics 17	
What CVS Is Not: The Lock-Modify-Unlock Model 18	
What CVS Is: The Copy-Modify-Merge Model 18	
Other Revision Control Systems 21	
BitKeeper 21	
BitKeeper License 22	
Microsoft VSS 23	
RCS and GNU/RCS 24	
SCCS 24	

- A Tour of CVS 25
 - Invoking CVS 27
 - Repository Access and the Working Environment 27
 - Starting a New Project 30
 - Checking Out a Working Copy 32
 - Making a Change 35
 - Finding Out What You (and Others) Did: **update** and **diff** 35
 - CVS and Implied Arguments 40
 - Committing 43
 - Finding Out Who Did What (Browsing Log Messages) 51
 - Examining and Reverting Changes 54
- Other Useful CVS Commands 58
 - Adding Files 58
 - Adding Directories 59
 - Removing Files 59
 - Removing Directories 61
 - Renaming Files and Directories 61
 - Avoiding Option Fatigue 63
- Getting Snapshots (Dates and Tagging) 63
 - Acceptable Date Formats 67
 - Marking a Moment in Time (Tags) 67
- Branches 73
 - Merging Changes from Branch to Trunk 80
 - Multiple Merges 82
 - Creating a Tag or Branch without a Working Copy 85
- Chapter 3 CVS Repository Administration..... 87
 - The Administrator's Role 87
 - Getting and Installing CVS 87
 - Building CVS from Source 88
 - Getting and Installing CVS under Windows 91
 - Getting and Installing CVS on a Macintosh 92
 - Limitations of the Windows and Macintosh Versions 92
 - Anatomy of a CVS Distribution 92
 - Informational Files 92
 - Subdirectories 94
 - Other Sources of Information 96
 - Starting a Repository 97
 - The Password-Authenticating Server 99
 - Repository Structure Explained in Detail 104
 - RCS Format Always Quotes @ Signs 110
 - What Happens When You Remove a File 112

	The CVSROOT/ Administrative Directory	113
	Finding Out More	124
Chapter 4	Advanced CVS	125
	Beyond the Basics	125
	CVS as a Communication Device	125
	Watches: Knowing Who's Working on What, When	125
	Log Messages and Commit Emails	139
	Getting Rid of a Working Copy	141
	A Bird's-Eye View of Project History	142
	Bird's-Eye View, with Telescope: The annotate Command	145
	Using Keyword Expansion	150
	Going out on a Limb: How to Work with Branches and Survive	152
	Merging Repeatedly into the Trunk	153
	The Dovetail Approach: Merging in and out of the Trunk	160
	The Flying Fish Approach: A Simpler Way	162
	Tracking Third-Party Sources: Vendor Branches	164
	New CVS Features	168
	You Are Now a Guru!	169
Chapter 5	Tips and Troubleshooting	171
	What to Do When Things Go Wrong	171
	The Usual Suspects	172
	The Working Copy Administrative Area	172
	Repository Permissions	174
	Common Problems and How to Solve Them	175
	Some Real-Life Problems, with Solutions	176
	Things Change	186
Chapter 6	The Development Process	187
	What Good Are Releases?	187
	Starting the Release Process	188
	Avoiding the "Code Cram" Effect	189
	Freezing	190
	Development vs. Stable Branches	191
	Testing	192
	Recruiting and Retaining Testers	193
	Automated Testing	193
	Building, Installing, and Packaging	194
	Building and Installing: make and autoconf	194
	Let CVS Help You with Packaging	197
	Releasing	199
	Telling the World about Changes	200

Recording the Release in CVS: Tags and Revision Numbers 200
Finding Out More 201

Chapter 7 The Open Source Process 203

Failure and Success 203
Starting a Project 204
 Release Something Useful 206
 Packaging 209
 Announcing the Program 212
Running a Project 212
 Cultivating Technical Judgment 215
 So, Who Is the Maintainer, Really? 217
 Rule by Committee 218
 How to Do a Fork, if You Absolutely Must 220
 Changing Maintainers 222
 Stasis 223
Knowing What We Don't Know 223

Chapter 8 Designing for Decentralized Development 225

The Importance of Software Design 225
Proprietary Software Design vs.
 Free Software Design 226
 Cost Issues 227
Design Invariants 228
Code Design 229
 The Design Document 229
 Dividing Code into Files and Directories 230
 Dividing Code into Modules 231
Evolution-Centered Design 233
Principles of Free Software Design 234
 Don't Limit Input 235
 Use a Consistent Interface 235
 Document Data Structures 236
 Make It Portable 237
When in Doubt, Abstain 238

Chapter 9 Third-Party Tools that Work with CVS 239

What Are Third-Party Tools? 239

pcl-cvs: An Emacs Interface to CVS	239
Installing pcl-cvs	240
Using pcl-cvs	242
Error Handling in pcl-cvs	243
cvsutils: General Utilities for Use with CVS	243
Cervisia	244
cvsu	244
cvsdo	245
cvschroot	246
cvsrcadm	246
cvsprurge	246
cvsdiscard	246
cvsco	247
cvs2cl.pl: Generate GNU-Style ChangeLogs from CVS Logs	247
-h, --help	248
-r, --revisions	248
-t, --tags	248
-b, --branches	248
-g OPTS, --global-opts OPTS	248
-l OPTS, --log-opts OPTS	249
-d, --distributed	249
cvslock: Lock Repositories for Atomicity	249
Other Packages	251
Jalindi Igloo	251
CVSup (Part of the FreeBSD Project)	252
CVSWeb: A Web Interface to CVS Repositories	252
The CVS contrib/ Directory	252
Writing Your Own Tools	252

Chapter 10 Complete CVS Reference 255

Organization and Conventions	255
Commands	255
General Patterns in CVS Commands	256
Global Options	257
List of Commands	261
Keyword Substitution (RCS Keywords)	291
Controlling Keyword Expansion	291
List of Keywords	292
Repository Administrative Files	294
Shared Syntax	294

List of Repository Administrative Files	295
Run Control Files	301
Working Copy Files	302
Environment Variables	304
Chapter 11 CVS versus BitKeeper—A Comparison	307
A Sample BitKeeper Session	308
A Comparison of CVS and BitKeeper	309
Comparing Commands and Syntax	310
Appendix A GNU General Public License	315
Appendix B GNU Free Documentation License	323
Appendix C Bibliography	331
Index	333
CVS Quick Commands	343



Introduction

Hardly a day goes by that you don't make use of open source software, even though sometimes you're unaware of it. Each time you receive an email from your spouse, friend or colleague, there's an almost 80 percent chance that it got to you through a classic piece of open source software: Sendmail.

If you look at a Web page, about 65 percent of the time, that page is being served by an open source Web server. In fact, most if not all open source applications are usually written with the help of open source tools like emacs (the venerable user environment and program editor), gcc, the official GNU C compiler, and debugged with gdb, the GNU debugger. Best of all, the source code of those applications and many others are maintained by one utility dutifully storing it all and keeping care of the ever-changing versions: CVS.

Open source software, in other words, has become a power player in the market and in some areas (like those mentioned above) even dominates it. And CVS is the very foundation of the open source movement, serving as the repository for the developers and for the end users. Often, these end users are no different at all from the developers, because in the open source world, the quality assurance is done by the them and then they contribute bug fixes back to the community. Therefore, a source code repository and version control system like CVS has to be quite a flexible tool, providing a stable and reliable front end to the open source community at large.

This book has two goals, one cultural, the other technical. The cultural goal is to document this open source culture to a certain extent and provide practical advice for people managing or participating in open source projects. The technical goal is to tell you how to use CVS effectively, with an eye toward using it on open source projects.

As to the first goal, we want to stress the word “advice.” In fact no one, maybe not even Richard Stallman, can speak about or document authoritatively the open source phenomenon. The field is simply too vast and it affects too many aspects of economic, cultural, social, and political sciences to be fully grasped by one individual, and certainly not by the authors of this book.

And as far as CVS is concerned, note that although it will be taught in the context of open source projects, you will learn CVS well enough to use it anywhere. It’s not just for managing program source code; people also use it to version—yes, that’s a verb now—Web sites, text documents, configuration files, and so on.

We assume that you know something about programming or working with online documents, but previous familiarity with CVS is not required. At least some familiarity with Unix and the `sh` or `bash` shells will prove handy, because the CVS examples are given in a Unix environment.

Why a Third Edition?

Books go into second and later editions when the earlier editions sold well. There is no better proof for the success of a book than it being republished in another edition.

Open Source Development with CVS is undoubtedly a highly successful book. The challenge in writing a third edition lies in not destroying what made this a successful book, while at the same time enhancing it to keep up with new developments.

From the time the first edition came out, the open source world has changed considerably. Certainly, the open source world changed more than CVS itself changed or the way in which CVS is used.

Open source grew quickly as Linux grew in popularity and as the Nasdaq made open source “in” and “sexy.” Many companies, such as VA Linux, LinuxCare, Red Hat, and thousands more, embraced open source and hacker ideals. Therefore, it was suddenly justifiable—even desirable—for investors to release all software and all specifications back to the community. Instead of making money from selling software, the investors then made money from the added value of thorough understanding.

Open source was so popular that many big IT users, such as banks, insurance agencies, and government agencies, decided to have an “open source strategy” for their IT departments. Coauthor Moshe Bar is an “open source consultant” to many such companies and agencies.

Hardly any software companies were able to afford *not* to have an open source strategy of some sorts, but some companies made big announcements about the availability of their software in open source without ever delivering on that promise. Then, abruptly, with the bursting of the New Economy Bubble in early 2001, open source suddenly

became “out” again. The investors demanded that high-tech companies finally deliver a profit. So, Web sites started asking money for their services, service companies asked for more money, and software companies started again to sell their software. Or at least they tried.

What Has Changed?

In the first edition, the aim was to intersperse purely CVS-related chapters with those dealing with open source and development organization. For the second edition, the approach was changed to separate the two issues so that the reader would not be confused unnecessarily.

Thus, this book first covers all aspects of the CVS system (Chapters 1 through 7) and only then addresses open source aspects (Chapters 8 through 11).

The CVS chapters now cover also the intricacies of working with CVS in big projects with many developers spanning several time zones. Also, aspects of the administration of CVS for professional environments will be explored more in depth, covering aspects of tuning, backups, storage, and clustering.

Finally, the open source chapters have adapted to the changes in the industry. They mention lessons to be learned from some of the exceptionally difficult challenges in open source (for instance, the Mozilla browser project) and from some of the failures in open source.

A Word About Terminology

Today, *free software* means the freedom to modify and redistribute the source. It is this freedom, not the software’s low cost, that has been the key to free software’s success.

Is it *open source* or *free software*? One of its earliest proponents, Richard Stallman, insists the proper term is *free software* (with *free* as in “free speech,” not as in “free beer”). The debate about this term has been going on for decades and will probably never end. Essentially, the two terms are synonymous, and they will be used interchangeably in this book. See Richard Stallman’s essay “Why ‘Free Software’ is better than ‘Open Source’” at www.gnu.org/philosophy/free-software-for-freedom.html for a well-written presentation of the case that the terms are *not* interchangeable. Increasingly, the term *free software* is used for software of the GNU project, such as gcc, emacs, make, and many more. In the ever-growing Linux world, however, software fitting the free software description is nowadays called open source or OpenSource. You will find the general press and the trade press often using only the term open source, even for GNU software.

Conventions Used in this Book

Throughout the book, you’ll find command-line examples interspersed with explanatory text. The primary example user’s name is ahauzer, and she works on a machine named yarkon.moelabs.com, so the command prompt looks like this:

xx Introduction

```
yarkon$
```

with output (if any) shown in the same font immediately below the prompt:

```
yarkon$ whoami
ahauzer
yarkon$
```

Occasionally, the command itself is so long that it occupies two or more lines of a standard Unix terminal. In that case, a backslash at the end of a line indicates that the next line is to be considered a continuation, although it will be indented by the length of the prompt for readability. For example:

```
yarkon$ cvs diff -c -r prerelease-beta-2_09-19990315 -r \
    postrelease-3_0-19990325 fudgewinkle.c
```

(Don't worry; by the end of the book, you will know what that command means!)

Sometimes we need to show commands run from other locations (when demonstrating concurrent development by two different people, for example). In those cases, the other user's name is mbar, and he works on a machine named paste:

```
paste$ whoami
mbar
paste$
```

All commands take place in a Unix standard shell (either sh or bash) environment unless otherwise specified. If you have even a basic familiarity with Unix, you won't encounter anything unusual in this book. However, you may notice that the `ls` command sometimes behaves a little oddly:

```
yarkon$ ls
foo.txt  bar.c  myproj/
```

The trailing "/" in **myproj/** is not part of the name—it just indicates that myproj is a directory. The reason the slash is displayed is that, in ahauzer's environment, the `ls` command is aliased to run `ls -CF`—that is, to show files arranged in columns and displaying their type ("/" for directories, "*" for executable files, "@" for symbolic links, and so on).

This format was chosen for many of the examples because it's often very helpful to be able to distinguish files from directories when reading the output. So even if you don't see the `-CF` options passed to the `ls` command, the output may behave as though they're there.



Practicing What We Preach

The CVS-specific chapters of this book—2, 3, 4, 5, 10, and 11—are copyrighted under the GNU General Public License and can be browsed or downloaded from <http://cvsbook.red-bean.com>. If you find bugs in either the online or the treeware version, please report them to bug-cvsbook@red-bean.com.



