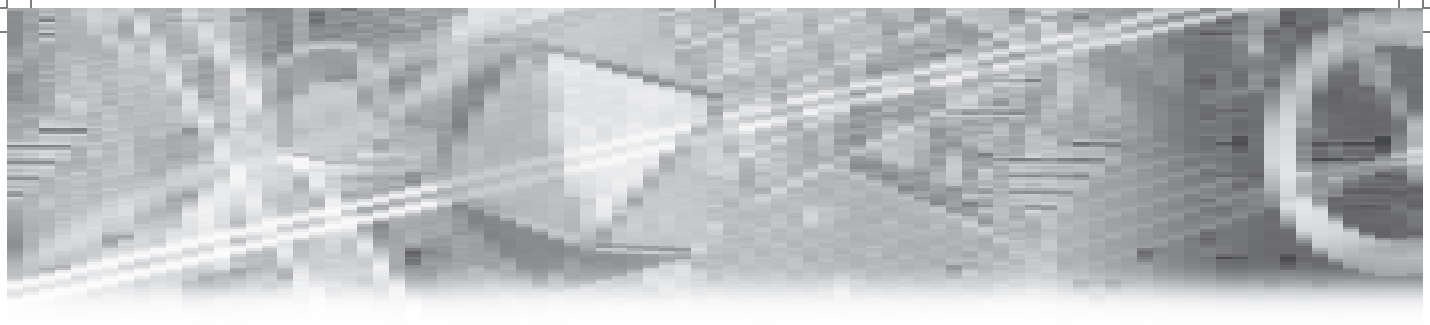

Open Source Development with CVS

3RD EDITION

Moshe Bar
Karl Fogel

PARAGLYPH™ PRESS

Open Source Development with CVS, 3RD EDITION

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

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

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

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





Contents

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:

```
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**.