Don’t Spend a Dime: The Path to Low-Cost Computing

ISBN-13 (pbk): 978-1-4302-1863-0

ISBN-13 (electronic): 1-4302-1864-9

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

Lead Editors: Joohn Choe, Jonathan Gennick
Technical Reviewer: Steve Nieman
Editorial Board: Clay Andres, Steve Anglin, Mark Beckner, Ewan Buckingham, Tony Campbell, Gary Cornell, Jonathan Gennick, Michelle Lowman, Matthew Moodie, Jeffrey Pepper, Frank Pohlmann, Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh
Project Manager: Richard Dal Porto
Copy Editor: Marilyn Smith
Associate Production Director: Kari Brooks-Copony
Production Editor: Ellie Fountain
Compositor: Linda Weidemann, Wolf Creek Publishing Services
Proofreader: Nancy Bell
Indexer: Broccoli Information Management
Artist: Kinetic Publishing Services, LLC
Cover Designer: Kurt Krames
Manufacturing Director: Tom Debolski

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax 201-348-4505, e-mail orders-ny@springer-sbm.com, or visit http://www.springeronline.com.

For information on translations, please contact Apress directly at 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705. Phone 510-549-5930, fax 510-549-5939, e-mail info@apress.com, or visit http://www.apress.com.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales–eBook Licensing web page at http://www.apress.com/info/bulksales.

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

CHAPTER 1

The Best Things in Life

As the song says, “The best things in life are free.” That statement can apply to numerous things: free coffee, free oil change, free Wi-Fi Internet, and even a free blood pressure check. But how many of us would expect to find computer software on that list? If you’re like me, you’ve discovered that software costs money. You might also be of the opinion that *good* software costs *more* money. And finally, you might have come to the matching conclusion that *free* software must not be that *good*.

Well, I’m here to tell you that I’ve never been so happy to be wrong. And by the end of this book, I think you, too, will be smiling when you discover that some of the best software in life is free.

Yes, the words *free* and *software* in the same sentence can sometimes still imply computers crashing, flaky installations, and cryptic controls. Unfortunately, the world of free software still has many hurdles to overcome. But rest assured that there are outstanding, five-star, free software applications out there that stand with the best and the brightest. And they’re growing in popularity and number every day. This is because they are getting easier to use and install, and are frequently on par in terms of quality (and sometimes 100% better) and features as those from the Big Boys (and you know who I mean). You’ll find free software that fills a void that pay-to-use software does not address, and free software written as a labor of love, with no strings attached to its use.

Free software raises many questions. Can this software really offer me the features I need? Can the software be trusted? Who do I contact if I have problems or questions? Are there hidden costs? How do I separate the wheat from the chaff? These are some good questions, and I’m sure you have more. I had my doubts when I started investigating free software, but again, I’ve never been so pleased to be so wrong.

You’re about to discover what I did: the words *free* and *software* do belong together. Let me explain how and why.

Software Wants to Be Free

Without getting too deep into the history of computing, let me summarize the early days of software in one sentence:

In the beginning, there was software, and it was free.

I’m not kidding. One of the biggest shocks to most people is finding out that software originally was free. Students and faculty at colleges and universities would create and share software with one another; no money exchanged hands. Most software at the time was written to fill a void—scheduling software for classes, simple text editing software for term papers, and even rudimentary e-mail software to communicate. Other software (games, for example) was written simply to show off programming skills or provide a nice break between classes and studying. Oftentimes, this software was modified by others to improve it and remove errors (called *software bugs*). But the Iron Age of software (some might even say the Golden Age) didn't last long. For software, as with all things, the *users* turned into *consumers*.

The public today has just come to assume that software is something you buy. Free software has always existed, but any software that is even remotely of interest to ten or more people is typically going to be boxed, priced, and stuck on a shelf.

But many modern-day programmers who create software have looked fondly back on those days and asked, "What would it take to return software to its origins?" What's required to have software shared with everyone, updated and repaired by colleagues, and 100% free to use? The answer is that it wouldn't take much.

Software Will Be Free

Today and any day, you can find all types of free software on the Internet. Just Google "free software," and you'll find hundreds of web sites that collect, summarize, and rate software, in addition to allowing you to download and install it. All right—free software!

But let's be careful here and examine what is meant by *free*. For me, free means I can download it without providing a credit card. But free also means that when I install the program, it shouldn't ask me for a credit card number or an installation key. But again, there are different definitions.

Some "free" software allows you to use it for a specified number of days (30 days, for example), and then requires you to either purchase it or uninstall it. This is often referred to as *trialware* or *shareware*. Other free software can be downloaded for free and installed on your computer, but consists of a slimmed-down version, often with many key features unavailable. The only way you can get the full feature set is to—you guessed it—purchase the full version.

There are many ways to define free, but for the purpose of this book, I want to share three varieties with you: open source, Software as a Service, and what I sometimes call the no-strings-attached freebie.

Open Source

Entire books have been written on the subject of open source software. Here, I'm going to spare you the essay, and simply give you what I feel is a simple, no-frills definition of what open source software is, how it works, and what it means to you.

An open source application is software that has the following characteristics:

- Created using a collaborative effort by individuals who share a common interest in producing an application of the highest quality
- Distributed for free (and kept free)
- Continually refined and improved by reducing bugs and adding features requested by its users

Open source software can include a variety of applications, such as word processors, spreadsheets, games, audio/video editors, and so on.

There are many benefits to using open source software, but the primary one for most people will be its cost: free. But if the software were low-quality, buggy, and not of any real use to people, the fact that it's free wouldn't matter much, because people wouldn't use it. So that's another benefit of open source software: quality.

Open source software is often being developed and tested by a group of knowledgeable programmers. This group might consist of two friends creating a game during their lunch hour. It could just as easily consist of 2,000 individuals scattered around the globe, who may never have met in person, contributing to and debugging one another's work around the clock. But one goal open source developers share is to create something that is of high quality.

And the "free" part of open source doesn't just apply to the application but also to its source code. The *source code* consists of the actual programming code that is used to create the application. If an application is defined as open source, this means that its source code is available for download so other programmers can inspect it and tinker with it. In fact, in some cases, programmers create a completely different version of an open source application by adding features or modifying the look and style. (But the variant source code must also be made available for free download and inspection or it loses the open source label.)

Finally, the group developing an open source application that is popular among users will, inevitably, receive requests for bug fixes and new features. Open source application developers typically are committed to continually improving an application to try to make it the best in class for a particular type of application (such as a word processor or graphics editor).

Open source isn't perfect. I want to make that absolutely clear. A lot of junky open source software is out there. But you can rest assured that the software I identify as open source in this book is some of the best you can find—it meets the high-quality criterion, as well as being free.

Software as a Service

Have you ever used a drive-through (or walk-up) ATM to get some cash? You probably used a computer screen and maybe a keypad to enter things such as your PIN and the amount of cash to withdraw. To get your money, you used software, right? Did you need to install the software on the ATM? Definitely not. You used the software that was already provided by the bank. The bank, on the other hand, had to create the software, test it, debug it, and install it on the ATM for your use. But you don't care how it came to be. Your only concern is that it works properly and lets you get your cash!

For purposes of this book, I'm going to use the term *Software as a Service* (SaaS for short, pronounced "sass") to describe a type of software that is growing in popularity. The primary difference between SaaS and open source software comes down to where the software resides. Is it installed on your computer or elsewhere?

■ **Note** Along with where the software resides, there are other differences between open source and SaaS. By the time you finish this book, you'll have a clearer understanding of how to differentiate the two. For now, just keep in mind that SaaS applications can easily be identified by the fact that you don't need to install any software on your computer to use them.

To make this more clear, let's consider online banking for a moment. Most banks allow their customers with Internet access to use a web browser to log in and access their account information. A customer doesn't need to install any special software (other than the web browser, of course), but she uses some very specialized software that's provided by the bank to view her balance, transfer funds, and maybe print out a monthly statement. When she closes the web browser or logs out of the bank web site, the software is no longer running on the customer's computer. That's the basic concept behind SaaS. Use it when you need it, and leave it when you don't. There is no need to install or uninstall the software.

These days, more and more companies are offering various services through their web sites. Some require you to pay a one-time fee (such as paying for two hours of Wi-Fi Internet access at a coffee shop). Others require a monthly subscription (such as an online newspaper subscription for *The New York Times*). These qualify as SaaS, even though they are not free, because the software you use to log in and perform tasks resides elsewhere; you're just accessing it from your web browser.

So what qualifies as free SaaS? Well, one of the biggest names in the industry offers free SaaS, and most people never even think about it. Imagine if you were charged a small fee every time you did an Internet search on Google (see Figure 1-1). You would probably never use it, right? Fortunately, Google makes its search engine completely free to use. It generates revenue from advertisers (yuck!), who place their ads on the right side of your web browser and hope that you'll see something interesting, click their ad, and buy, buy, buy.

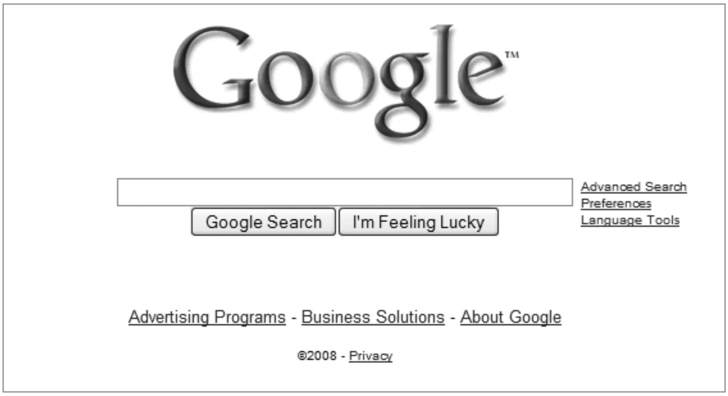


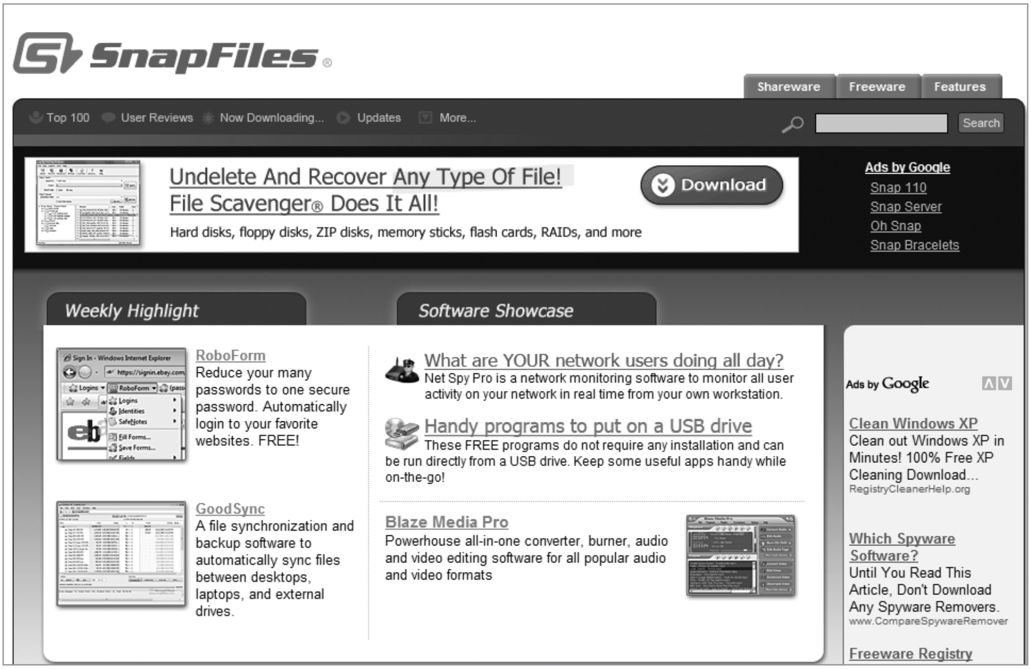
Figure 1-1. *Thank goodness that Google doesn't charge every time you click that search button.*

There are plenty more 100% free SaaS web sites. For example, `CNN.com` offers free news services, and `craigslist.org` lets you buy and sell items using posted ads. I'm going to introduce you to many free SaaS sites in upcoming chapters. By the time you finish this book, you'll have a collection of SaaS web sites that make your life much easier. You're going to be e-mailing me, asking when *Don't Spend a Dime, Volume 2* is coming out!

No-Strings-Attached Freebie

The last type of free software I'll talk about has been around forever. It's the 100% free application that someone created and put out there for the world to use. This software costs nothing, but also typically comes with nothing—no support, no web site, no phone number, and no updates. Usually, the people who create this type of free software do so because they need a particular functionality and cannot find any existing software that provides it. Typically, these folks share their software with the world, expecting no payment in return (just as in the early days of software!).

Hundreds of web sites offer access to free software. Just do a Google search for "free software" and be prepared to be overwhelmed. One of my favorite places to go when I'm looking for an application to do something very specific is `http://www.snapfiles.com`. Take a quick look at this web site. In the upper-right corner, you'll see a tab for Shareware and a tab for Freeware. Click the Freeware tab, and you'll see a categorized listing of everything SnapFiles offers that is free, as shown in Figure 1-2. (Be careful—you could spend hours browsing through this site's free software.)



Undelete And Recover Any Type Of File!
File Scavenger® Does It All!
Hard disks, floppy disks, ZIP disks, memory sticks, flash cards, RAIDs, and more

RoboForm
Reduce your many passwords to one secure password. Automatically login to your favorite websites. FREE!

GoodSync
A file synchronization and backup software to automatically sync files between desktops, laptops, and external drives.

What are YOUR network users doing all day?
Net Spy Pro is a network monitoring software to monitor all user activity on your network in real time from your own workstation.

Handy programs to put on a USB drive
These FREE programs do not require any installation and can be run directly from a USB drive. Keep some useful apps handy while on-the-go!

Blaze Media Pro
Powerhouse all-in-one converter, burner, audio and video editing software for all popular audio and video formats

Clean Windows XP
Clean out Windows XP in Minutes! 100% Free XP Cleaning Download...
RegistryCleanerHelp.org

Which Spyware Software?
Until You Read This Article, Don't Download Any Spyware Removers.
www.CompareSpywareRemover

Freeware Registry

Figure 1-2. *SnapFiles offers thousands of freeware applications for you to use.*

WHAT'S THE CATCH?

You may be thinking that there must be a catch. There's no such thing as free! You're right—sort of. High-quality software that is 100% free with no strings attached does exist. Open source software is the best example. But someone has to pay for those computers that run SaaS, and there are many ways this can occur.

Some companies (Google, for example) cover the costs of their free applications through advertising. Head over to `http://www.google.com` and do a search for "Star Wars." When you see the results page, look along the right edge of the screen. See the small blocks of advertisements? Those companies are paying Google to have their advertisements placed so you (possibly a Star Wars movie fan) can find and buy all kinds of Star Wars memorabilia. But being exposed to ads is not a big sacrifice for access to Google's free applications.

Some companies charge fees for technical support. For example, if you find you have a question about the software, a $25 phone call will get you the answer. Some companies charge fees for multiple users beyond the first user. Others charge a fee if you need customization of the application (such as adding your company name or logo to the application's opening screen).

The methods for generating income vary from company to company, but my best advice to you is to always read the fine print. Spend some time looking around the application's web site, and maybe perform a Google search of the application's name. You're sure to find information that will either verify the software's free cost or give you information about any hidden costs. That said, all of the software I cover in this book is free to download, install, and use. If there are any hidden costs or unusual methods for generating income, I will explain that in the discussion of that application.

But What About the PC?

Yes, the computer (the hardware) costs money. Even if you've been given a free computer as a hand-me-down, that PC required someone to spend money at some point in time. It's an unfortunate truth that there really is no such thing as an open source PC, with all its parts made and distributed for free.

For most of us, obtaining a computer usually requires buying one from a PC company (such as Dell or Acer). This isn't necessarily a bad thing, though. Purchasing a prebuilt PC provides you with a warranty, someone to call when things go wrong, and that new-computer smell (admit it—you like it, too).

If you're buying a new PC, you might want to find a company that allows you to order your new computer without any preinstalled software. Why have them install a $30 antivirus program when you can get some great antivirus software for free? (Chapter 9 tells you where.) Maybe the computer vendor offers to throw in a productivity suite (word processor, spreadsheet, and e-mail application) for only $150 (versus its retail price of $300 at the store). Tell them to forget it! Chapters 3 through 6 show you where to get the same features using reliable, easy-to-use, free software.

The best part of steering clear of pricey off-the-shelf software is that you can take the money you would have spent on that software and use it to get higher-performing hardware. You can put that cash toward a faster processor, more memory, and a larger hard drive. Not only will the free software run more efficiently (and take less hard drive space) on your new computer, but you will also have extended the usefulness of your computer by years.

If you've decided to purchase a new computer or build your own, you'll want to read through Chapter 2 next. There, I'll show you how to determine exactly what hardware you need. Not everyone will need the fastest PC with all the bells and whistles. In the world of computers, people frequently purchase more than they need. If you do this, the computer sellers will love you, but your bank account won't be so happy. Fortunately, open source software and SaaS typically have lower demands for computer hardware, meaning you can save even more money by pairing free software with the lower hardware costs. Now that's a win-win solution!

RECOMMENDED READING

If you have any interest in the history of computers and software, you owe it to yourself to hunt down a few books:

- *Hackers* by Steven Levy (Penguin, 2001) isn't about those troublesome, virus-writing, break-into-computers-and-steal-credit-card-numbers bad guys. Back in the day, the term *hacker* was a positive one (and it's coming full circle and regaining its status). *Hacker* referred to anyone who could take hardware or software and remake it, debug it, tweak it, and do other wondrous things. This book covers from the early days at MIT, when a bunch of model railroad fanatics began experimenting with controlling their trains with simple electronic circuits, to the humble beginnings of Microsoft and Apple. It's some good storytelling.
- *Dreaming in Code* by Scott Rosenberg (Three Rivers Press, 2008) is an entertaining story about a small group of individuals who decided to create an open source application. The book covers the highs and lows, and the successes and failures. It also explains the open source concept in detail.
- *On the Way to the Web* by Michael A. Banks (Apress, 2008) is a history of the Internet, including its origins, its successes, and its mistakes. This is a great read and helps you understand how the Internet got where it is today.

Summary

Many computer users are tired of buying software that is buggy, complicated, and overpriced. And since much of this software is updated every year or two, we can quickly find ourselves in a never-ending cycle of purchasing upgrades.

Enough is enough! With a minimum investment in hardware (possibly even $0), you have a wide selection of free, high-quality software that can save you from this dependency on big-name applications.

My advice is simple. Investigate the options and determine if the free software will meet your needs. If you do find that your needs can be met only by using the features found in an off-the-shelf application, then it's okay to buy that application.

In preparing this book, I made a list of the types of software that would most likely be found on users' PCs. I then picked the most commonly used name brand of each type of software and researched, found, installed, and tested what I believe to be the best, most realistic replacements that are 100% free. Read through the book and discover for yourself how easy it is to never pay for software again.