Sample Chapters and
Other Information from

The First Book of Hazel:
A Quick Guide to the Hazel Internet Merchandising System

by Roger MacBride Allen

***The* Guide to the Internet's Most Powerful and Flexible Small-Business E-Commerce, Shopping Cart and Sales-Management System!**

FoxAcre Press

What's In This Sample File

This file contains sample text from THE FIRST BOOK OF HAZEL: A QUICK GUIDE TO THE HAZEL INTERNET MERCHANDISING SYSTEM. It includes the Introduction, and the first two chapters, with instructions on how to install Hazel, and advice on what you'll want an online webstore to do—and not do.

The file also includes the table of contents and the index for the complete book. If you already own the non-printing PDF version of the book, you might want to print out the Table of Contents and Index to provide you with a quick and easy means of locating information in the non-printing PDF. Note: there are minor formatting adjustments made between the sampler version and the full version of the book, and some PDF search and bookmarking are disabled because the text is truncated.

More information about this book, including pricing and availability for the full book, is available at www.hazelbook.com. Visit www.foxacre.com to learn more about other titles available from FoxAcre Press. See the end of this PDF file for more information.

More information about Hazel, including information about how to order the program, is available at hazel.netsville.com.

Sample Chapters and

Other Information from

The First Book of Hazel: A Quick Guide to the Hazel Internet Merchandising System

Hazel

The Internet Merchandising System

Revision Date: February, 2007

FoxAcre Press

FoxAcre Press
www.foxacre.com
www.hazelbook.com
hazel-fox@foxacre.com
401 Ethan Allen Avenue
Takoma Park, Maryland 20912
USA

Sample Chapters and

Other Information from

The First Book of Hazel: A Quick Guide to the Hazel Internet Merchandising System

by Roger MacBride Allen

Hazel

The Internet Merchandising System

Table of Contents

Tables and Diagrams

Introduction

For a business or organization to have an online presence has gone from being an exciting novelty to being a standard convenience that customers assume will be there. It used to be people were pleasantly surprised that the XYZ MegaCorp had a website. Now they are surprised and annoyed when Sarah's Shoe Shack doesn't.

But even today, many commercial websites for small businesses are not much more than high-tech bulletin boards, with a map showing where the place is, a brief description of the store, and maybe a badly scanned photo of the owner's cat.

If you already have a website for your store, or mail order business, or non-profit organization, or other entity that wants to sell online, there is very good news. If your webserver meets a few fairly basic technical requirements, and if you (or someone who works for you) knows how to start and maintain a web page, you are, right now, about two hours and $200 away from having a basic but functional online store set up using a mainly automatic installation process. You are probably only a few weeks away from a slick, customized, and sophisticated online sales operation.

Nothing is ever quite as easy as it sounds, and there are lots of ways to spend a lot more than $2,000, let alone $200, and it will take some study and work to launch your web store—but it can be done, with Hazel.

The Hazel Internet Merchadising System is an extremely powerful and flexible way to create and maintain an online shopping site. With it, you can create just about any sort of online shopping setup you might want. Hazel can do it all. Hazel can manage your product list, manage your orders, present your merchandise in any number of different ways, and handle discounts, shipping rates, surcharges, sales tax, products that come in different sizes and colors, and a lot more besides.

The problem is in figuring out how to *tell* her what to do. The available documentation is, shall we say, sometimes a bit hard to work with. Hazel's standard documentation consists mainly of a set of HTML pages that are available at http://hazel.netsville.com/docs/. The most current version of

the same pages can be downloaded, in one compressed file, to your computer, as discussed at the top of that web page.[1] We'll refer to these pages as the *online documentation* throughout this manual. There are other useful Hazel information sources, such as a searchable *Knowledge Base*, at http://hazel.netsville.com/support/kb/ consisting of past questions that have been emailed to Hazel's developers, along with the answers. You can subscribe to the mailing list of which the Knowledge Base is the archive by going to http://hazel.netsville.com/support/hazeltalk.html and following the instructions there. However, fair warning: the present version of the Knowledge Base can be hard to work with, and includes a lot of dated information about past versions of the program—for example bug fixes and work-arounds for problems that been resolved in later versions of the software, or requests for features that have since been added to the program. It can be hard to clear through the underbrush.

Hazel's FTP site, ftp://ftp.netsville.com/pub/hazel has the latest versions of all the files needed to run Hazel, along with various other goodies.[2]

There is a lot of good information in all those sources, but here's the bad news: nearly all the explanations for Hazel are written by and for people who *already* know Hazel backwards and forwards (for example, Quinn, the guy who wrote much of Hazel in the first place), with the result that the documentation includes so many assumptions about the user's knowledge that non-computer programmers (such as myself) can't make heads or tails of most of what the documentation says.

When I first got started with Hazel, it seemed to me that most of her documentation read as if the first three paragraphs of each section, containing all the basic assumptions for what followed, had been left out. In fact, it was only in the process of writing this manual that I came to understand some of the underlying assumptions that were never clearly expressed in the program's online documentation.

The gaps in Hazel's documentation create a real lose-lose situation, in that they make the system seem far more challenging to use than it really is. What should be a relatively smooth learning curve is left looking an awful lot like a brick wall.

I have no doubt that, as a result, more than a few online businesses that could benefit from Hazel's power and flexibility—and the relatively low cost of the product—have failed to take the plunge.

---

1. Note that certain elements of the online docs have special features that won't work if the pages are simply copied to your home computer.

2. I have set up a bulletin-board-style discussion section for Hazel, assessable via www.hazelbook.com, but it'll be up to the users of Hazel and the readers of this book to make it a worthwhile resource.

That means fewer customers for Hazel's owners, fewer web designers using a system that could make for a smooth and slick experience for their customers, fewer sales for those websites, and, more than likely, a goodly number of web designers and shoppers who feel frustrated by the cookie-cutter, pre-fabricated template limitations of shopping system products that compete with Hazel.

This manual is intended to solve at least part of that problem. It *not* intended as a complete guide to Hazel. In a sense, no such guide is possible for Hazel, or for any other programming language. (And Hazel is, among other things, a programming language, except perhaps by some rigid and technical definition that wouldn't mean much to the non-specialist.) A full guide to Hazel would show you *everything* Hazel could do—which would be roughly analogous, on a much smaller scale, to a single book that showed everything you could do with the *English* language. Good luck on that one.

Instead, this manual is intended as a detailed *introduction* to Hazel, and to her grammar and structure. It will *not* cover every single command, action, loop, and setting—but it *will* demonstrate how the most common and powerful commands, actions, loops and settings work. With that grounding, you will be far better prepared to contend with the more esoteric and obscure elements of the program. While we won't cover 100 percent of what Hazel can do, we will cover about 95 to 99 percent of what most users will *want* to do with Hazel.

Most (but by no means all) of Hazel's features *are* discussed somewhere in the online documentation, or in the knowledge base, but boy, do you have to do some digging to find some of them. More frustrating, you often have to do some real legwork just to find some basic and essential information. I have often had to go to three or four spots in the knowledge base and the online documentation to find what should have been all in one place. And some pieces of information are just not there at all. Other information is seriously dated.

(And by, the way, fair warning: you are now reading a static copy of a guide meant to cover a product that is frequently updated. For the most part, what this means is that new features and capabilities are added, but it can also mean that something that works in version 4.012 doesn't work the same way in 4.091.)

In this manual, I have tried to assemble and arrange the basic information in a clear and logical way, and in a way that complements the existing documentation. To put it all in broad strokes, this manual will tell you how to use the most common examples of a given class of tool. You will then be able to apply that knowledge to using other examples of the same class of tool, as discussed in the online documentation. In other words, with this manual in hand, and the online documentation in front of you on screen, you ought to be all right.

The online documentation is almost entirely nonlinear. There is no real beginning, middle, or end to it. Nor is there supposed to be. It is intended much

more as a reference guide, a place to look things up, and not as a tutorial. And as a reference guide, it works quite well—well enough that I haven't tried to duplicate its coverage.

This guide isn't going to be 100 percent linear either, but it will come close, presenting the topics one after another, each building on what has come before, to the greatest extent possible.

I have tried to present the material in a logical, step-by-step way, but there are limits to this approach. Many coding examples will include elements that have not been discussed up to that point in the text. For example, Hazel uses Loops a lot, and several early coding examples in this guide include Loops without explaining them in detail. I have tried to keep this sort of thing to a minimum, but the elements of Hazel are intended to work with each other, and it was all but impossible to explain every element the first time it appears. Making a virtue out of necessity, I have tried to use these unexplained early appearances to prime the pump, so to speak, giving the reader a bit of a heads-up on what's coming next.

Hang in there. All will be explained in time, in as orderly a basis as I could manage.

The Audience(s) for, and Purpose(s) of, This Book

It can be dangerous to try and please everybody, as the result can be a book that pleases nobody. Even so, I have tried to write a book that will be suitable both for Hazel beginners and experienced Hazelistas. There is so little information on the program out there for either group that it seemed likely that not just absolute beginners, but even programmers who know Hazel a lot better than I do, might have some holes in their educations—and it's not like there are forty-seven other Hazel books out there. This might be the only available manual for a good long while. Therefore, this one book does cover topics from the very basic to the fairly advanced.

Experienced users of Hazel might be tempted to skip over the early chapters, as they concern themselves with some topics that might seem pretty basic. However, it would still be at least worth skimming through these chapters. The odds are pretty good that there are a few tidbits of information that even grizzled old Hazel veterans will not have stumbled across before.

Those who are new to Hazel, or are merely considering the purchase of Hazel, should definitely start at the beginning and stick with it in order to get a good solid grounding in both the concepts of the program and the nuts-and-bolts stuff.

Both groups of users should find the material in the remainder of the book quite useful. Even the most experienced user is likely to find tricks, tips and techniques that ought to prove of interest.

The book will also serve as a reference regarding Hazel's Actions, Tags, Tokens, Rules, and what have you. A quick glance through the table of contents ought to direct you to what you need. The **Bits and Bobs** chapter, full of interesting items that didn't fit anywhere else, almost certainly has some item of info that will be useful to someone.

Don't skip the non-technical chapters, which discuss the theory and practice of online store management, customer relations, and other topics. The rest of the book tells you *how* to do things. These chapters will tell you *what* you should want to do, and why—and also discuss a few things you *don't* want to do.

Making Assumptions

Having muttered about the assumptions implicit in the online docs, let me explicitly state a few assumptions *I'm* making. Most of this is stuff you'll have to know anyway if you are running a website, and/or running a site with an online shopping system.

I assume you know enough about HTML coding to do some of it for yourself: you can at least write a simple web page with a text editor. If not, bail out now and learn. Being able to do point-and-click work on Frontpage or a similar web editor isn't likely to cut it. You need to be able to sit there and type HTML code without hyperventilating.

If you are totally new to HTML, I can recommend Elizabeth Castro's *HTML for the World Wide Web*. It's a very clear and concise book. I have used my copy of the fourth edition for all sorts of purposes. There is now a fifth edition, officially entitled *HTML for the World Wide Web, Fifth Edition with XHTML and CSS: Visual QuickStart Guide* (jeez, what a mouthful). I still use the fourth edition, but the fifth is of course going to be more up to date. Visit www.cookwood.com for more info. And there are any number of other books and sources on HTML. Use them. Learn.

That being said, I used Hazel for years before I wrote a single Hazel file completely from scratch. Instead, I simply edited and re-edited the files that shipped with Hazel—though I edited some of them almost out of all recognition. The point is you can go an awfully long way simply by cutting and pasting bits of code that work from wherever you get them, thus keeping the amount of typing in fresh code to a minimum.

Many web page editing software packages will likely be baffled by Hazel, mainly because she uses a lot of her own tags that don't exist in standard HTML, and also because some web page editors like to tidy up pages in ways that can screw up Hazel—in effect, these programs see Hazel's special code as improperly written HTML and try to "fix" it. Some web editors can be told to ignore whatever new tags you tell them about. You will at least need a web editor that *can* be told to bypass new tags, so that it won't freak out over a Hazel tag or token.

However, instead of a web editor, probably it will be easier just to use a standard ASCII text editor to write and modify your Hazel files. (I like *Notetab* from Fookes software at www.notetab.com. The Windows utility *tidy.exe*, available at tidy.sourceforge.net cleans up HTML, catching various typos, and can easily be told about non-standard tags, such as those used by Hazel. Tidy works well with Notetab.) *The CoffeeCup HTML Editor* (www.coffeecup.com) also takes Hazel in stride, and simply ignores the Hazel tags without trying to correct them—unless you turn on certain of Coffeecup's "advanced" features. Namo's WebEditor (www.namo.com) also works well. It's often handy to have two or three programs that can edit HTML, as each will have its own strengths and weaknesses.

There's another reason a web editor with a hotsy-totsy what-you-see-is-what-you-get interface isn't necessarily the best choice when writing Hazel files: you *won't* see what you get. Hazel files are *not* HTML pages. They are instruction sets, computer programs if you will, that Hazel *uses* to write the HTML code of the actual web pages. Therefore, you *can't* see what your final page will look like in the web editor, because the underlying HTML code for that page will be written later on, on the fly, by Hazel.

I'm further assuming you want to run an online store, or maybe set up an e-commerce site for a client, and are either considering Hazel or have already purchased a license for Hazel. (You can try before you buy: You can get a free demo of Hazel by entering your contact information at hazel.netsville.com/trial/ and then experiment to your heart's content—as long as your heart is content with a thirty-day trial period of a catalog limited to 100 items. Converting that trial version to a full version is as simple as entering a license key number into the Hazel.config file and uploading that file to your website.)

I'm assuming you know what an online shopping basket *is*, and that you are more or less aware that you'll have to have some way to manage such issues as discounts, shipping, charging tax, and so on.

I am assuming that you know how to use an FTP (file transfer protocol) program to upload files to a website, or else are willing to learn it (and it's not that hard) and that you can at least bluff your way through such esoterica as setting file permissions on a website. (This isn't as tough as it sounds, or as baffling as it once was: if you shop around just a bit, you can find lots of FTP programs that will make setting permissions pretty painless.) We'll cover the FTP stuff in enough depth to get a neophyte going a bit later on.

It's also up to you to do your homework and make sure that your website hosting service will allow you to run Hazel. Any host that *won't* run Hazel has likely been deliberately configured so it won't run any *other* shopping-cart program either (except for, maybe, the cheesy one that comes pre-installed from the hosting company). If so, that's probably not the host you want for your site,

whether or not you go with Hazel. A hosting setup that won't let you run Hazel is likely to tie your hands in lots of other ways, and won't let you do anything else interesting.

Hazel is available in two versions: Hazel 3.x and Hazel 4.x, with the x being replaced with the current sub-version number as the program is tweaked, fine-tuned and upgraded. Think of them as Hazel-Light and Hazel-Pro. Hazel 4.x has various features that Hazel 3.x does not, but pretty much anything that works in Hazel 3.x will work in 4.x, and most of the additional features in 4.x consist either of elements that were options in 3.x that have become standard in 4.x, or else quite esoteric features that are way out of my league. Hazel 4.x is certainly worth getting if you can afford it. This manual is, for the main part, a guide to 3.x, plus the main features that are options in 3.x and standard in 4.x. However, that being said, just about everything in this manual will apply to 4.x as well as 3.x. See **Appendix C** for a summary of the differences between the two programs.

Be sure to have the latest version of whichever Hazel you have. More and more features work, and work better, in later and later versions, and it's not worth my time or yours to label which features became available in version 3.234, or 3.370. Assume the latest version is the best.

We will discuss installing Hazel in some detail. For now, suffice to say that Hazel's installation is not that hairy, can be done more or less automatically, and at least *that* part of the on-line docs are clear and thorough. (See the Walkthrough section of the online docs.) For now, the basics are: you have to enter some values in a few places, you must install Hazel's files to specific directories on your website's host computer, yo'll have to set some files permissions, and you *won't* install Hazel on your home computer. You'll either use an automatic installation process that ships with Hazel, or else use your FTP program to do the job. You'll get a lot of guidance on the subject of installation. I am therefore further assuming that you have—or will have—the installation of Hazel more or less under control.

I am assuming that you're *not* going to read this whole manual at one go and then never look at it again. It is far more likely that you'll thumb through this book for the bit of data you need, and then put it to one side until the next time you are confronted with a Hazelian mystery. Therefore, this guide often says the same thing more than once and in more than one place, and in more than one way, so that the information is available in the context you need it. A bit of redundancy should also help make things clear. If I explain the same thing three different ways, the odds that one of the three ways will make sense to you increases dramatically. Also, frankly, it was easier to leave in the repetition than to edit it all out.

Finally, I am assuming that you have spent a bit too much time digging through Hazel's online documentation, looking for things that aren't there or are far too well hidden, and have gotten as frustrated as I have trying to find very simple information.

Conventions

For the most part, I will be using two imaginary products, blivvets and cheesebenders, in this manual. In Hazel, each product has its own SKUID (**S**tock **K**eeping **U**nit **ID**entity, or part number). I have assigned blivvets the SKUID of "bliv," and cheesebenders the SKUID of "cheese." In real life, of course, you'd have more than two products, and you'd have more sensible SKUIDs, based on some sort of set of standard rules or numbering pattern. I will use the imaginary domain name http://www.domain-name.com, and, when appropriate, its secure-server equivalent, https://secureserver-domain-name.com.

I will use this typeface to set off examples, Hazel elements, web addresses, and so on from the regular text. No doubt to the horror of purists in the art of book layout, when it has come down to a choice between, for example, maintaining exactly even top and bottom margins and keeping a section of code all one page, I have at least tried to favor readabilty over page design.

A slight but useful digression: Many webservers will flip like a cheese omelet (translation: they won't show the file) if you use the wrong letter case when calling a file. In other words, they might think

www.domain-name.com\index.html
www.domain-name.com\INDEX.HTML

and

www.domain-name.com\index.HTML

referred to three different files. It's easier on everyone if you just use lower-case on all your web files names.

Also, different browsers are fussy about different HTML rules. I was driving myself crazy with one picture element that showed up in Miscreant Exploder (sorry, Microsoft Explorer) but did not appear in Mozilla. Finally I realized that I had a dash (sometimes known as a hyphen) in the picture name. As soon as I changed it from test-candidat.jpg to test_candidat.jpg all was well. The moral of the story is to stick as closely as possible to the standards in order to make your website work for as many browsers—and users—as possible.

In order to set a good example, all file names used in this guide (except for a few likely typos) are in lower.case. Hazel herself doesn't much care about upper and lower case, and in the code examples I have used Mixed Case and UPPER CASE and lower case as seemed appropriate.

Because Hazel is always referred to as "she" and "her," I will assume all customers are male. That way, if I mention "her" you know I mean Hazel, while if I mention "him" it's the imaginary Steve Shopper or his friends who are under discussion.

Disclaimers Are Us

I should note a few other things, right at the outset: I am not a paid cheerleader. This is *not* an authorized or commissioned manual. In fact, the people who own Hazel didn't even know that I was working on this manual until I had the first draft done. When I sat down to write it, I had no financial or business relationship with Hazel, other than as a customer who had paid the full price for the product, and who had asked more than his share of questions via the Knowledge Base. However, once the owners of Hazel saw the first draft, I was given a upgrade from a 100-product 3.x license to a full 4.x license in order to let me experiment with various 4.x features and write about them more clearly.

It is likely that I will come to a business arrangement with Hazel's owners as to selling or distributing this manual. While I will eagerly and gratefully accept corrections of whatever technical errors I have made, from them, or from any and all others sources, the views and opinions in this manual are my own. No one but myself has any control over the content of this manual.

Going just a bit further, unless I note otherwise, I have no relationship with the makers of any other software or hardware products I mention in the text, other than as a paying customer. I don't make a dime more or less if you do or don't buy one of these products. (It never hurts to be clear about these matters.)

Tracking Some Details

Here are a few housekeeping notes regarding the text.

All the code examples, other than those that are only a line or two long, are posted at www.hazelbook.com. Corrections and clarifications to the code, and to the text, will be posted there as well.

If you spot an error in the book, of whatever size or importance, please drop me a line at hazel-fox@foxacre.com. Please don't post that address anywhere spammers might lurk.

It should be noted that this book is trying to chase not one, not two, not three, but *four* moving targets: the current version of the Hazel program itself; the content and design of Hazel's templates and other files; Hazel's online documentation, FTP site, Knowledge Base, and other resources; and, to a far less significant extent, the configuration of my Hazel installation on my business website, www.foxacre.com.

There have already been reasons for me to alter code on foxacre.com even though I have used that same code as the basis for examples here in the book,

and there will doubtless be other cases in future when I do it again. I'm not going to leave an improvement out of the website just because a coding example is in the book—and I'm not going to delete a valid coding example just because my website doesn't use it anymore.

It bears mentioning that Hazel is a mature program, but still undergoes periodic updates. I discuss various bugs in the program and in the distribution files in this book, and I report quite a number of omissions, and some errors, in the online docs. If the good people at Netsville correct a problem I mention in the text, or add a new feature, then this book will be out of date—but in a good way.

Updates and corrections to this text, and/or notifications of updated versions, will also be posted to www.hazelbook.com.

The Electronic Version of This Book

This book will be sold in various markets and in various ways. Some versions of it will ship with a CD-ROM. As of this writing, that CD-ROM will *not* provide any content to help you install Hazel. It will, however, include the entire text of this book as an Acrobat PDF file, compatible with Acrobat Reader 5.0 or higher. This PDF version has various security features enabled that are intended to prevent the user from printing it out, or cutting and pasting text from the file. Probably there is some determined jerk out there who can find ways around these features, but these security features are intended to keep less-determined jerks from printing out free copies of this book and stealing the years of effort that went into creating it.

Even with the printing and cut-and-paste feature disabled, the PDF version should come in awfully handy. It has a full set of bookmarks keyed to the Table of Contents, so that you can just point and click to jump to any chapter or topic in the book. It is fully searchable, so if all you can remember about what you are looking for is that I used the word "gibberish" to describe it, you can simply search on that word.

The disk also contains a plain ASCII text file listing all the non-trivial code examples in the book, keyed by page number. That file you *can* copy, cut-and-paste, edit, or what-have-you, and that should save you a lot of effort. The CD-ROM may or may not contain other features, and its contents will vary from one edition of the book to the next. To some degree at least, Netsville will have a say as what code I can and cannot include.

If your physical copy of the book did not ship with the CD-ROM, or if you'd like to get a CD-ROM of the latest version of this book, you may order it from www.hazelbook.com or by writing to FoxAcre Press. If you want your disk to match your version of the print book, please be sure to tell us the revision date shown on the copyright page of your copy.

Getting It Out The Door

Inventing the greatest product in the world can't make you rich if you keep tinkering with it, and it never gets out of the workshop because you want it absolutely perfect first. The same argument applies to how you *sell* your products. When it comes to web design in general, or online store design in particular, better is the enemy of good enough. If you don't have an online store yet, you are currently losing 100% of your potential online sales. Don't wait until everything is perfect, perfect, perfect to unveil your store. Get a basic, reliable system set up as soon as you can—and *then* start improving it. You will learn more from your mistakes, and from your customers' reactions to those mistakes, than you will ever learn from a store that never gets used. Even if the first and very basic version of your store isn't ideal, it's *got* to catch something better than zero percent of your potential customers.

With all that having been said, if you have any sense at all, you're now eager to get started right now. So turn the page, and let's go for it.

Chapter One

High-Speed Hazel, or E-commerce in an Hour

Before we dive into the following two hundred and forty or so pages on how to use some of Hazel's many sophisticated features, let's get you started on a very basic e-commerce site that you can have up and running in a couple of hours. And, not so incidentally, you'll get at least a quick and glancing introduction to many of the features we'll discuss at length later on.

In writing this chapter on getting started fast, I have tried to strike a balance between providing lots of useful detail and being a Relentlessly Repetitive Pendant. It's possible I've erred in the direction of RRP-dom. If so, bear with me, and keep in mind that reading something dull takes less time than doing something wrong.

Read this chapter from start to finish. Once you're done, you'll know what you need to know to set up a functioning webstore, and, with just a little luck and planning, you'll be able to do it in something like the time it takes you to wade through the chapter for the second time. Even if you get stuck somewhere along the line, the odds are good that you ought to have a basic system up and running in little more than an evening.

One other thing: it might take only an hour or two to get things set up, but you might have to work on collecting some information before you start the clock. See the section on **Gathering Information** below, and have your ducks in a row before you start the job.

Even if you're the one poor schlub in a thousand who can't get things to work after days of trying, be of good cheer: once you do get the durn thing working, you can test-drive Hazel for free. You can get a thirty-day trial of the software—that's a month's worth of a fully functioning e-commerce site for nothing. Of course, after that, you have to pay for the software—and if you've just spent a month becoming dependent on Hazel's great features, and have started to get used to selling on line, you really ought to be prepared to pay that license fee.

A few other things to bear in mind:

Once we're done here, *you will have a functioning e-commerce site.* Customers

will be able to enter your store, select merchandise, go to your checkout page, enter payment information, *and place real orders*. So you'd better be ready with real merchandise, real shipping materials, real postage, real payment processing, and so on.

Second is that what Hazel's actual final output typically consists of is a set of e-mail messages. There will be a shopper's invoice letter to the customer, which will list his billing and shipping address, the items he ordered, the costs of shipping and/or taxes, and the order total. It will also report how the customer has paid. However, if the customer has used a credit card to pay, the e-mail sent to him will *not* list his credit card number or expiration date.

Hazel will also generate a second store invoice e-mail and send it to you, the merchant, which will include all the same data as the e-mail message to the customer. If it is a credit card transaction, *it will also include the credit card number and expiration date.*

So long as you set up Hazel so that the e-mail sent to you is sent to an address on the same domain, this should be highly secure. In other words, if your website was called www.domain-name.com, and you instructed the order e-mail to be sent to hazelorder@domain-name.com, that should be quite secure because the message in question would never leave the server it was on until you told the e-mail program on your local computer to go get mail for that account.

The only remotely plausible way a bad guy could get that information would be to intercept your e-mail in the zillionth of a second as it was moving from the domain-name.com server to your home computer—or else by stealing your e-mail password and having the order emails downloaded to him. Suffice to say the first stunt (intercepting e-mail in transit) is very hard to do, and a bad guy savvy enough to do it would likely be able to come up with about fifty other crimes that would be easier, would yield a higher income, and would be easier to get away with than snagging a credit card number in transit over the Internet.

The second stunt (plain old stealing a password and downloading your e-mail before you can get to it) is your problem: you have to make sure your process for handling other people's money is secure.

However, I have cobbled together a way to protect credit card numbers from both dangers, though it does require moderately advanced knowledge of Hazel. See **Appendix D** for more on this point.

But with those points in mind, we can move along to getting your web store fired up.

IMPORTANT. The following instructions apply to websites that run a version of Linux or Unix compatible with Hazel—which is just about all versions of Linux or Unix. See the online docs page http://hazel.netsville.com/docs/walkthrough.html for more information.

Nearly all web servers use a version of Linux or Unix, or else one version or

another of Windows Server. Insofar as running Hazel is concerned, Linux/Unix is the preferred system for your website. Websites run by Windows won't be able to use the installation process discussed below, and you should read through the installation docs at the above address to learn how to get up and running. However, it will still be worth your while to read through this chapter, even if you run Windows and/or elect to use some other means of installing Hazel. A lot of what will be discussed below with either be immediately helpful no matter how you install, or else will be of use once you're up and running.

EVEN MORE IMPORTANT. The operating system that your website runs under has nothing at all to do with what operating system you have on your own computer. Whether you are running Windows or Linux, whether you use a PC or a Mac or whatever doesn't matter one little bit.

Not Quite As Important. The following discussion of how to set up Hazel in a hurry assumes, among other things, that you are running a small business and want to ship physical objects to people, and that your primary market is in the United States. If you are selling downloadable computer files exclusively in the Canadian market, some of what follows won't apply—but you'll have the common sense to work that out. The key thing to bear in mind is that Hazel can do it all. She can be used to sell tangible and non-tangible objects, and can compute just about any form of shipping charge or local tax or discount or surcharge you can imagine, and she can be adapted for use in just about any country. For starters, however, we're going to keep things very basic, and trust in your ability to judge what applies to your own circumstances, and what you'll need to alter.

A large part of the idea in this chapter is that any webstore at all is better than none whatsoever. We're going to get a quick and basic store up fast, and not worry about the bells and whistles. You will be able to change just about anything and everything about your store later, and change it in lots of different ways. It would probably be smart to limit the number of products you sell online, just at first, so as to limit the number of things you have to change if you later decide to do things a different way But first, let's get it up and running.

Fun with FTP

FTP stands for File Transfer Protocol. Among other things, it's a way to move files back and forth across the Internet. You'll need a FTP program on your local computer if you're going to run a Hazel store.

For right now, we're going to have to upload exactly one file to your website, which will then, with luck, do all the work needed to install all the *other* files to your website. However, a bit later in our high-speed Hazel process, we are going to have to transfer some other small files as well. Once you have Hazel up and running, you will likely use FTP to do some, or even most or all, of your site maintenance.

If you have run any sort of website wherein you create and/or edit content on your local computer and then upload it to the website proper, you are almost certainly using some sort of FTP program already. That being said, in my experience, at least, most of the FTP programs built into web editor software packages are extremely limited and hard to use, and/or don't do some of the things you'll need done. The good news is that there are a lot of free and cheap FTP programs out there that will provide the features you need. Find one you like. Here are a few of the issues you'll have to deal with regarding FTP.

- **ASCII vs. Binary transfer**. ASCII stands for American Standard Code for Information Interchange, and it is of course the standard way to encode text characters, and therefore, text documents, in a computer file. It can display numbers and letters. HTML and text documents are (or at least should be) pure ASCII text. Computer programs, such as those that run on your local computer, and also such as Hazel, are binary files, and as such will have all sorts of characters that will look weird and screwy if viewed as text. Hazel herself *must* be uploaded as a binary program or she just won't work. Most other programs with a name that ends in .cgi or .exe or a few other extensions should be transferred in binary mode. HTML and text files should be transferred as ASCII. (However, be aware that many web scripts, including some with the extension .cgi are actually text files, and must be transferred as ASCII in order to run. The hello-hazel.cgi script we'll be using later on is an ASCII web script. Rule of thumb: if you can open the file in a text editor and see legible text and/or HTML formatting codes with no funny characters, it is ASCII and should be treated as such. But be aware that some text editors, such as Windows *Notepad*, might, for example, show carriage return characters as black boxes.) The products file and the rules files and the template files that Hazel uses are all ASCII files.
- **File Permissions**. Files on most websites can be tagged with various levels of *permission* for three classes of user—the owner of the file, member of the group that includes the owner, and everybody else. You're going to run across references to CHMOD, which is the command in Linux/Unix boxes to control these values. For reasons I am not going to go into here, these permissions are often expressed as a three-digit number, one digit for each category of person. Most FTP programs will simply let you enter that number. Others might have you check the appropriate boxes, or manage permissions by some other means. You want Hazel, and most other programs, to have their permissions set to 755, which means the owner can read, write, and execute the file (7), whereas members of the group (5) and everyone else (the other 5) can only read and execute the file.
- **Directories**. In order for Hazel to work, all the various files must wind up

in the correct directories. We are going to use an automated system to install her, which should result in everything winding up just where it should be. However, that *first* file needs to be in the right place if we are going to pull this off. This file needs to be in whatever directory your Internet Service Provider (ISP—the people who run the server your website lives on) has designated as being for CGI programs. Usually this will be a directory called something like cgi-bin or the like. It will likely be a subdirectory of your site's www directory. From a browser it will be something like http://www.domain-name.com/cgi-bin. If you try to view this directory, you ought to get an error message that says something like Forbidden. You don't have permission to access /cgi-bin/ on this server. This is a good thing, as you *shouldn't* be allowed to view this directory. From your FTP program this directory will be something like /home/www/cgi-bin. Directories can have permissions set as well, and this directory should normally have them set to 755.

Information Gathering

The first hunk of information you'll need is a Hazel license. If you've already purchased a license for a full version of Hazel, you'll use that license number. See **Appendix B** for more information on Hazel licensing.) If you're looking to try before you buy, go to http://hazel.netsville.com/trial/ and enter your name, address, phone, and email info. Be sure to check the box that signs you up for the Hazel-talk message board. There's lots of good info there. Complete the form, click on the box that shows you agree with the license agreement, and then sit back and wait for the email message with your trial license to arrive. While you're waiting, you might as well scoop up all the other info you'll need.

In order to install Hazel, you're going to have to know a few things besides your license number. Mostly these will be file locations and the like, and mostly they will be things your ISP ought to be able to tell you.

- You'll need to know if you are allowed to create directories outside the WWW directory. Hazel's default is to create a directory called hazel-cat on the same level as your WWW directory, such that if you have a directory called /home/domain-name/www with your regular website in it, you would also have /home/domain-name/hazel-cat. You don't need to have hazel-cat in this location, but you do need it outside your WWW directory, so that anyone using a web browser won't be able to access it directly. So long as you can tell Hazel where this directory is, she'll manage everything on her own.
- You'll need to confirm that you can install third-party CGI programs (like Hazel) on your website, and you'll need to know what directory such programs should go in. Typically, this will be the www.domain-name/cgi-bin/ directory, or, in FTP-speak, something like /home/domain-name/www/cgi-bin/.

- You'll need to confirm that you have access to a *secure server*, and the http address for it. You might have to ask to have this service switched on, and it might cost you extra. If you're going to accept credit cards online, you're going to need this service. A secure server provides encrypted access to the website, so as to prevent anyone snooping on the transaction from obtaining personal info. In some cases, the secure server might be a physically different server from the rest of your website. If so, that will be a real nuisance, as it means you'll have to upload everything twice—once to the regular server, and once to the secure server. A far more common (and far more convenient) arrangement is for your secure server simply to provide a secure and encrypted routing to the same location on the same server. Your secure server address might be very similar to your regular address: something like https://www.secure-domain-name.com instead of http://www.domain-name.com. However, your secure server address could be entirely unrelated to your standard server. Some ISPs offer matched-named secure servers or brand-name secure-servers for an additional fee. You'll have to decide what makes sense for you.

If most of the foregoing sounds like barely coherent gibberish, don't worry about it. Just crib the following text into an e-mail message to your Internet Service Provider (ISP).

To Whom It Concerns:

I wish to install a third-party program called Hazel (hazel.netsville.com) into my website, www.yoursitename.com, which is hosted on your service. I need to know the following things in order to complete installation.

(1) There are versions of Hazel for both Windows Server and for various Linux versions. Please advise which operating system, and precisely which version of the particular operating system, is used on my website.

(2) I need to confirm that your service permits users to install this type of program.

(3) Hazel is a CGI program. Please advise as to in what directory I should install the base program file, Hazel.cgi. (Hazel.exe if you're installing on a Windows server.) Please confirm that permissions have been set properly on this directory.

(4) Hazel requires that various data files be placed in a directory called hazel-cat (which will contain various sub-directories). The hazel-cat directory must be inaccessible to web browsers. It should be outside the WWW directory, or else otherwise made "invisible" to a browser. The default setting would be to place hazel-cat on the same directory level as WWW. Is your service configured in such a way as to allow me to create this directory in that location? If not, please advise as to your preferred

(and secure) location for this directory.

(5) Please advise as to the secure server that corresponds to my main, non-secure server. Please advise whether this is actually a physically different server, or whether it simply provides secure access to the files accessible via my non-secure server. The second would be my preference.

(6) Please advise as to any additional charges or optional services associated with the above queries.

Sincerely,

Your Name Here

If your ISP can't answer those questions, get another ISP. But assuming it is a marginally competent shop, they will be able to provide the information, and you'll have the directory and other info so you can plug it all into the right spots when you get to that point in the installation process. As long as you get the right answers, it doesn't really matter if you know what the answer means. Don't worry if you don't understand it all. Follow the recipe, and do what the cookbook says.

The above is all stuff you should only have to deal with once, while installing Hazel, and then, with luck, never again. (But keep a reliable record of all the information for when something crashes!)

So, now you know more or less what to do with FTP, and ought to have a handle on directories and servers and so on.

You will also need to be ready with your online store's web address, name, mailing address, phone numbers (if any), and, perhaps most important, a live and functional e-mail address to which orders can be sent—i.e., order@domain-name.com. Be sure to set up and test an appropriate address to be used for that purpose and no other before you install Hazel. You should also probably have a general-purpose e-mail address—i.e., info@domain-name.com, ready for other purposes.

And of course, you need to be ready with the product information that you will need to set up in the proper format. Each product will need its own SKUID, or Stock Keeping Unit IDentity. Be sure your SKUIDs are useful and meaningful. You'll need a description of each item, a price, and, if at all possible, a picture of it in digital format.

With all that mind, fire up your web browser and tell it go to ftp://ftp.netsville.com/pub/hazel/misc/ and locate the file named hello-hazel.cgi. Create a directory on your own local computer where you'll store Hazel stuff, and save the hello-hazel.cgi file there. Next, fire up your FTP program, log onto your website, go to your cgi-bin (or equivalent) site, and use the *ASCII* mode of the FTP program to upload hello-hazel.cgi. (ASCII because it's a script, not a binary

program file like ham.cgi or hazel.cgi.) Make sure that the permissions for hello-hazel.cgi are set to 755.

Now, turn to your web browser again and use it to start up hello-hazel.cgi. If your website was www.domain-name.com, with your cgi-bin directory in the normal place, you would enter www.domain-name.com/cgi-bin/hello-hazel.cgi.

With luck, and the wind behind you, you will then be presented with information about your server, and a prompt urging you to click on it so as to continue to install Hazel. If hello-hazel.cgi doesn't run properly, make sure you uploaded it as an ASCII file, and that you have set its permissions properly. If it does run properly, and you see that link, click on it and do what it says.

You will either get another program called ham.cgi loaded into your cgi-bin directory automatically, or else you will get prompted for where and how to download ham.cgi to your home computer.

If the latter, you'll then have to upload ham.cgi to your website's cgi-bin directory and run it by entering your website's equivalent of the address www.domain-name.com/cgi-bin/ham.cgi. (Remember -- ham.cgi and hazel.cgi are binary files and must be uploaded as binary or raw, not ASCII or text.) Either way, ham.cgi should run, and walk you through the installation of Hazel—with a brief time-out or two for entering the appropriate licensing info.[1] You'll be taken through the process of entering the proper directories, mailing address data, phone numbers, e-mail addresses, and so on.

When the smoke clears, Hazel should be installed. More or less. (If you try and fail to install Hazel using this procedure, go to hazel.netsville.com/docs/walkthrough.html and follow the procedures discussed there.)

The "more or less" part is that Hazel will be installed with dummy products and (extremely) default choices as to means of payment, shipping rates and tax rates. We'll adjust all these files to your needs in a bit, but there are two steps to take first.

Log onto your new webstore by entering Hazel's URL. For www.domain-name.com her URL would be

www.domain-name.com/cgi-bin/hazel.cgi

Try "ordering" a few of the imaginary items on sale. If you want to try a credit-card purchase, use the dummy card number 4444333322221111. It should pass the checksum tests used by Hazel to see if the card number follows the standard formatting rules. Don't be shy about placing orders. All that will happen is that you will receive two e-mail messages from Hazel for each order—one for the

1. Certain versions of ham.cgi might be smart enough to walk you through the process of getting a license on the fly, but I have not been able to confirm that—and it could be a real hassle to get nine-tenths of the way through the process and find out it doesn't work. It's almost certainly smarter and easier to get yourself a trial license before firing up ham.cgi, as discussed earlier in this chapter.

customer sent to whatever e-mail address you entered when you placed the order, and one to the e-mail address you earlier gave Hazel for reporting orders to your office. The customer e-mail should contain full details of the order, except for the credit card info. The store e-mail message should contain full info—including the credit-card data. The only thing that will happen is that the emails will be sent. The sky won't fall, and the credit card number won't be charged.

The above only applies to credit card orders. Under normal default settings, if you do a test order using the "printed order" option, Hazel will *not* send any e-mail messages. (It is assumed that the customer has the printout and doesn't need the email message, and since there is no way to be sure the customer will actually mail in the order until he does, there is no point in Hazel alerting you that an order simply *might* happen.)

Don't feel shy about abandoning an order or two part way through. Leave Hazel. Go somewhere else on the web. Using the same browser on the same local computer, go back to Hazel, and she should have remembered your order as you last left it.

Once you are satisfied that Hazel is installed properly, and once you have the basic feel for how the shopping-and-ordering process takes place, it is time to create a backup of your default Hazel installation before you start fooling around and modifying things.

Fire up your FTP program again and log onto your site. The details of how you do it are up to you, but you'll need to backup three directories, and all their various sub-directories, to your local hard drive. It's going to be up to you to locate the three directories, and up to you to copy them to a sensible place on your local computer's hard drive. The three are your cgi-bin directory (which might have some other name, and is probably under your WWW directory) your hazel-doc directory (which ought to be under your WWW directory) and your hazel-cat directory. The default location for hazel-cat is the same directory level as your WWW directory, but of the three, it is the one that your ISP is most likely to require to be in some other funny place. On the bright side, if it is in some non-standard location, you probably just put it there yourself, so you ought to be able to find it.

It would be an outstandingly good idea if you copied your backup of your Hazel installation from your local hard drive to a CD-ROM.

Helpful Hint. Most FTP programs allow to save locations you go to a lot, and allow you to associate directories on your local drive with a particular directory on your web server. Create one such saved-location (whatever your FTP program calls it) for your hazel-doc directory, associating it with the hazel-doc directory on your hard drive. Do the same with your hazel-cat directory. If you get started with themes and product images and other bells and whistles, your FTP program is likely to start sprouting lots of these saved startups.

Once you have tested Hazel, and once you have a backup of your initial

installation, it is time to start editing a few files so Hazel is selling your products instead of imaginary ones, and charging what you tell her to charge for shipping, taxes, and so on.

The products.txt file (which might simply be called products depending on your version of Hazel) should have wound up in your /hazel-cat directory, and, not surprisingly, it is a list of products for sale. We need to replace the dummy products in this list with real ones.

The other items we need to tweak right away are controlled by the files payment_methods.txt, ship_method.txt and tax_regions.txt, all to be found in the /hazel-cat/loops sub-directory; by the sales_tax.rules file in the /hazel-cat/rules sub-directory; and by the various shipping rules files found in the /hazel-cat/rules/shipping sub-directory. (Obviously, your hazel-cat directory is going to be a sub-directory of something else, but what the something else could be will vary widely from one website to another, so you're on your own.)

We'll go into all this in endless detail later (otherwise, why have a two-hundred-sixty-page book?) but the short form is that, while you can do madly complex things with all of these files, the versions that ship with the installation version are all pretty simple.

All of these files, except the products file, are lists of items, listed one per line. The products file is a special case that we will look at separately in a minute. All the other files share one same basic format. Each line lists one item per line, but there will be two names of each item. On the left will be the computerese abbreviation that Hazel will use to look up the item. A semicolon serves as a divider. On the right is the human-friendly description of the item that your customers will be shown. The files will be structured like this:

```
computerese_name;The human-friendly name
more_gibberish;Another bit of gibberish
```

payment_methods.txt. This is a list of all the ways that your site accepts payment. When your customer reaches checkout in the default version of Hazel, he will be presented with a drop-down list of these possibilities. Hazel will pull up the appropriate form for your customer to use. If you don't plan to offer one of these methods (say, COD orders) simply erase that line. Adding a payment method is more complex, and we're not going to worry about it now. The one you are most likely to want to add is PayPal, and that is discussed in detail later in the text. Get the file the way you want it, then save the file to your local computer's hard drive and close the file.

ship_methods.txt. This is a list of all the ways that you are ready to ship an order. On the left is the prefix of the file name. Following the semicolon is the human-friendly description. The filename prefix corresponds to files found in the /hazel-cat/rules/shipping/ directory. Thus, the lines

```
standard;Standard Shipping
express;Express Shipping
region;Region-Based Shipping Rates
```

would correspond to files named standard.rules, express.rules and region.rules, all to be found in the /hazel-cat/rules/shipping/ directory. You will find a variety of basic shipping rule files in the aforementioned directory. Open the ship_methods.txt file in your text editor, and then add and remove references to shipping methods until you have the list of choices the way you want it.

Even if you delete the reference to a given shipping method from the ship_methods.txt file, there is no need to go monkeying with the checkout template. If the reference to it is deleted from the ship_methods.txt file, the customer will have no way to select the shipping method in question. You can delete the corresponding file from the /hazel-cat/rules/shipping/ directory if you like, or you can leave it there on the off chance you'll use it later. It won't do any harm to leave it. Save the ship_methods.txt file to your local drive and close it.

Alternate: if you only plan to have use one type of shipping, there is no point in offering the customer a choice, and the currently shipping checkout template takes that into account. If you leave only one shipping method in ship_methods.txt, Hazel will select that method in the background and not offer the user any other options.

Shipping Rules. Calculating shipping charges in such a way as to be equitable to all customers in all places at all times ordering any possible item is, to use the technical terminology, a frigging nightmare. We'll talk about the shipping rules files and all the strange things you can do with them later on. My own choice has been for a region-based system that computes charges based on a selection by the customer, plus the number of items ordered. It contains well over a hundred lines of code—and doesn't yet do everything I want it to do. But we're not getting that fancy in the midst of our High-Speed Hazel install. We'll keep it simple. The syntax for the various shipping files is different from the files we have just seen. We'll learn by example.

Just for starting out, my advice would be to use a very simple system based on the version of quantity.rules that ship with the distribution version of Hazel. Set the per-item shipping charges so they will be appropriate to the majority of your customers—preferably the vast majority of them. If you get orders from customers wherein the computed shipping charge isn't enough, you'll either have to absorb the excess shipping charges, or else send an apologetic note to the customer with a corrected shipping charges. Later you can invent whatever horribly complex shipping rules you want—and, more than likely, you'll *still* have to absorb some shipping costs now and then. No one set of shipping rules

can cover all the bases all the time.

A few high-speed notes on the quantity.rules file that ships with Hazel at present. Here is the basic code. (Your distribution of Hazel might have somewhat different contents for all of the files we are discussing here, but the following should be pretty close to what you have.)

```
[properties]
NAME: Quantity

[quantity] # A base of $4 ($6.95 for first minus $2.95 for successive).
1-:_4

[quantity] # Plus $2.95 for each item.
1-:x2.95  @$6.95 shipping for first item, $2.95 each additional.
```

We won't worry about the deep inner meaning of the underscore just yet. (And yes, it does have deep inner meaning) For the moment, remember that the idea is that you're going to charge X for every item, plus the additional amount Y for the first item. In the example, X for every item is 2.95, and Y for the first item is 4. Plug in whatever values seem about right to you for X and Y. *Don't* enter dollar signs—just the numbers. Adjust the text after the #s and the @ to reflect whatever changes you have made, (and it's okay to use dollar signs after a # or @). Save and close the file.

One quick piece of advice on shipping charges: don't try to make shipping into a profit center. Doing so will annoy your customers. Instead, try and come up with break-even charges, and make your money selling merchandise. If you usually break even or come out five or ten cents ahead on shipping, that will compensate for the rare occasion where you find you have undercharged for shipping. If you discover that you have grossly understated the charges, contact the customer and explain—and then rewrite your shipping rate rules. But don't tempt a customer to cancel a $50 order by asking him to make up the difference when the shipping rate you set up is under by 47 cents.

sales_tax.rules. The demonstration sales tax rules that come with the distro version of Hazel at present looks like the code below:

```
# sales_tax.rules

[%HZE_BILL_STATE;STR]
PA:+0.06_ @ 6% tax for orders billed to Pennsylvania residents.

[%HZE_SHIP_STATE;STR]
PA:+0.06_ @ 6% tax for orders shipped to Pennsylvania residents.
```

Don't worry. You won't be nailing some poor sod in Pennsylvania for twelve percent tax. Without going into fiddly explanations, (which we'll get to later) the code in this file will check to see if the shipping and/or billing address shows the state abbreviation PA. If either and/or both do, it will charge six percent. It won't charge him twice.

Change the state name, the state abbreviations, the tax rates and the descriptive text after the @s to reflect your home state. Be sure not to mess with any of the strange punctuation. Save and close the file.

There are ways to contend with circumstances wherein a city or county charges its own sales tax. We'll cover them in depth in the chapters on rules, but for now let's not worry too hard about such matters. (But if you're in that situation, be prepared to pay any such taxes out of your side of the deal until you sort it out. How's that for an incentive to learn fast?)

products.txt. Now we come to the big dog, the one you'll really have to do some work on: the products file.

Here again, we'll go into greater detail about this later, but for the moment, let's do a high-speed sprint through the file format. Note that the bold-faced fields are required. The other can be omitted altogether, or simply left blank for items to which they don't apply. You can add just about any other fields you want. See the comment lines at the top of the sample products.txt file for more information about the format.

SKUID:
NAME:
PRICE:
DESC:
IMAGE:
GROUP:

The SKUID (Stock Keeping Unit IDentity) is the part or item number—the index entry. It should be assigned based on some logical organizing principle that makes sense to your situation. Bear in mind that if you change your SKUID naming scheme later on, you'll have to change the SKUIDs wherever else they appear, and probably you'll need to do some sort of cross-reference between the old and new SKUIDs in order to be able to track queries or deal with orders from people using old printed order forms or whatever. Plan ahead.

The online docs say that all SKUIDs must be no longer than 64 characters, must begin with a letter, and should not contain anything but letters and numerals and underscores. (No hyphens). However, current versions of Hazel accept SKUID values that start with a numeral. Many of my SKUIDs are ISBNs (International Standard Book Numbers) with the hyphens stripped out. They

are nothing but numbers, and that works just fine.

The NAME is just the name of the product.

The PRICE should be entered with two decimal places and with a leading zero if need be, for example: 4.00 and not 4; 0.75 and not .75). Do *not* include a dollar sign or other currency marker—just the number part of the price.

DESC is the description of the item, and it's going to be a big part of what sells the item. This is your ad copy. Make it good.

IMAGE is optional, but having a picture of your product is a darned good idea. It tells Hazel where the file with the image is, relative to your hazel-doc directory (unless you really start messing with the defaults.)

If you keep the images in your hazel-doc directory, all you need here is the file name itself—but keeping the images there can get things cluttered up pretty fast. Better to keep them off by themselves in another directory. Getting and keeping things straight when it comes to directories and relative and absolute addressing can get incredibly confusing, but here's a nice, straightforward example that should get you started: If, as seen through your FTP program, your hazel-doc is in /home/domain-name/www/hazel-doc (a fairly typical location) and you keep your images in the directory /home/domain-name/www/images, then you would enter ../images/image_name.jpg in the IMAGE field.

Those who understood the foregoing, but who tend to be easily confused, should not read this next bit, as it might be Too Much Information.

Just for the record,

/home/domain-name/www/hazel-doc

as seen by a web browser would be

http://www.domain-name.com/hazel-doc

and

/home/domain-name/www/images/image_name.jpg

would be

http://www.domain-name.com/images/image_name.jpg.

If you have other files that you want Hazel to pull up—a thumbnail view (that is, a smaller-size version of the image file) or a PDF file, say—you would be well advised to use similar directories and addressing rules.

As with SKUIDs, it would be smart to have a logical naming scheme for

your images, and/or for thumbnails or other related products. It might be smart to name them based on the related SKUID. If the product's SKUID is bk001, (book number one) then the image file could be img_bk001.jpg, the thumbnail image could be thumb_bk001.jpg, and the PDF with the detailed information brochure could be bro_bk001.pdf. Having standardized named will make it easy to search for, say, all the files related to a given product, all the images, all the thumbnails, etc.

GROUP is sort of an odd duck, and I would just as soon it wasn't in the demo file for products, as it likely to be a source of needless confusion (as opposed to the bit about images, which you *do* need to deal with, even if it is confusing.) Not all users will use GROUP. We'll discuss it later. For present purposes of getting you up and running (and we're nearly there!), just use GROUP:ALL (that is, enter the value ALL in each product's GROUP field) for all your products. If GROUP is not in the products.txt file of your distribution, don't add it.

If you have an extensive product list, I would strongly advise that you *not* try to put everything into Hazel's product list all at once just as you are starting out. Put your most popular items (and/or the ones that ought to sell best over the web) into the products list first and see how it goes. You'll likely discover you want to add more fields to the file, or that you want to adjust the format of the DESC file, or that you want to put the images in *this* directory instead of *that*. None of these changes are a big deal if you start out with twenty items—but any of them could be a major headache to change on, say, two thousand items. If you have your products in any sort of database, you ought to be able to generate an ASCII text file properly formatted for use as your products.txt file—but even so, start small, just in case you've accidentally mis-formatted all three thousand entries. Get things under control first—*then* do the full-scale launch.

In any event, now you are just about done with your initial install. Save and close the products.txt file. Use your FTP program to copy all the files you have edited on your local computer back over to the appropriate directories on your web server. If you have images for your products (and it's a very good idea!) or any other additional files, make sure they are uploaded to the proper directories as well. Do another backup to CD of your modified base install of Hazel.

Now comes the big moment. Start up Hazel again with your equivalent of www.domain-name.com/cgi-bin/hazel.cgi.

With any luck at all, you should be looking at a list of your products, ready for real live customers to click on and order. You might have some debugging of one sort or another ahead of you, and there are a zillion ways you can modify your site—or even rebuild it from scratch—but, in something like the time it will have taken you to read this chapter through, you will have created a brand-new online store.

Nice work. Now let's make that online store even better.

The first step in doing that is to take a look at the things an online store is supposed to *do*, and get a customer's-eye view of how Hazel does them.

Chapter Two

Online Sales: Some Basic Concepts

Before we dive into the nuts and bolts of writing code and so on, it would be worthwhile to discuss what it is we're trying to accomplish.

Online shopping systems in general, and Hazel in particular, follow a fairly close parallel with shopping in a bricks and mortar store—or at least Hazel does her best to simulate the more pleasant elements of the bricks-and-mortar experience.

When Steve Shopper goes to a physical store, he walks in the entrance, and then starts looking at merchandise. The store owner has done her best to arrange her products in a logical way that (a) makes things easy to find, and (b) also makes them attractive to the customer.

The management has also done several things that are not only for the convenience of the customer, but that are also efforts to encourage the customer to shop more. For example, the store provides shopping carts so as to make it easier for the shopper to buy more than he can carry in his hands. The cart also to make it easy to know what has been selected and whose selections are whose. If an item is in Steve's cart, we know it isn't (or at least shouldn't be) something Marvin has selected. Steve could even park his cart off to one side, leave the store, and if he doesn't leave the store for too long, with a little luck he will find his cart still where he left it, with all his items still in it, the next time he comes in. That way, he doesn't have to re-select everything all over again when he returns.

When Steve is finished shopping, he proceeds to the checkout stand, where he is told how much his purchases cost in total, along with what discounts he might have earned, how much he must pay in tax, and maybe even (to stretch our analogy a bit) how much it would cost to ship his purchase to his home.

If, let's say, Steve discovers at the checkout stand that Brand X Blivvets are on sale, he might be tempted to return his Z-Brand Blivvets to the shelves and go with Brand X. Or he might find out that he can't afford to spend as much as his order will cost, or he might discover that if he got eight instead of four

Blivvet-brighteners, he'd get an extra ten percent off.

In real life, aside from Steve asking the clerk not to ring up an item that he has just realized he really can't afford, there are social pressures that prevent people at the checkout from redoing their orders much. If Steve did in fact want trade in his Z-Brand Blivvets, and/or make other changes, he would have to get everything in his order off the checkout stand and back into his cart, cancel the transaction, back out of the checkout line by shoving all the other customers out of the way, and then go around the whole store again, removing items and replacing them—and then face the dirty looks from his fellow shoppers and the clerk when he returned to the checkout stand. No one does that if they even plan to be in the store again. So Steve Shopper stays meekly in line, and watches the clerk process his order.

Once he has his purchases toted up, Steve Shopper must, of course, pay, by whatever means the store accepts, and following the store's rules as to spending limits, identity checks, and so on. And, of course, no merchandise leaves the store until it is paid for. Once Steve has paid for his order, he leaves the store with his merchandise.

The online shopping experience is meant to echo that process, while leaving out driving to the store in the rain, parking in scary parking lots, contending with surly clerks, and so on. The downside is that online shopping also leaves out lots of *positive* elements—the chance for Steve to touch and feel the merchandise, to browse more or less randomly and find something he wasn't planning to get, to pop into the store next door as long as he's in town, to run into a friendly neighbor, to ask a non-surly clerk a question and get a helpful answer, or just the chance to get out of the house and enjoy the occasion of shopping.

And there are other downsides to online shopping. Except for cases where you're downloading "soft" merchandise, like a computer program, online shopping also means that Steve must wait days or even weeks for his merchandise to arrive. And, in many cases, the actual total order price of an online order will be higher, (mainly due to shipping costs), than the order price of the same items purchased locally.

In effect, so far as a rational shopper is concerned, shipping costs for online orders and mail-order purchases compete on price against local sales tax plus other costs, such as transportation to and from the store, parking fees, etc. In most cases, Steve will pay shipping but no tax if he orders online from outside the taxing jurisdiction, and tax but no shipping if he buys in a bricks-and-mortar store. In other words, it couldn't hurt if your shipping charges were more or less comparable to a typical sales tax charge, plus, maybe, the cost of the gasoline burned to get to the store, parking fees, etc.

In another sense, the hassles of shopping online compete negatively with the hassles of shopping in person. Is it more of a drag to schlep downtown, or

more frustrating to wait for the website to load, and, later, to be forced to stay home in case the delivery guy shows up?

There are also gains and losses when it comes to privacy. Steve can pay cash and remain nameless when he buys face to face—but the clerk *has* seen Steve's face, and so, potentially, has everyone else in the store. If he's buying something he doesn't want his neighbors to know about, for whatever reason, he's taking a fair size chance that he'll be spotted. Contrariwise, when Steve buys online, he remains completely faceless, but far from nameless. He must provide all sorts of information, such as his name, address, credit card, and so on—and then, of course, the letter carrier or the UPS guy at the very least knows that Steve is getting *something* delivered from the XYZ company.

It's up to the online merchant to understand the parallels between online shopping and bricks-and-mortar shopping, and likewise the differences between them. The online merchant then needs to accentuate the positive and eliminate the negative, insofar as possible. A good online merchandising system like Hazel is a vital tool in that effort.

Part of what Hazel does is to provide a structure similar to bricks-and-mortar shopping. Let's see how. We'll delve deep into how it's all done later, but from the customer's point of view, the typical Hazel store will follow familiar patterns from square one.

Let's follow Steve through a Hazel-run online store, and see the process through his eyes. Bear in mind that from the technical point of view, what's happening is that Hazel is using inputs from Steve Shopper, along with her other data sources, to create web pages on the fly. To do this, she uses a series of templates, which are, in effect, blank forms that Hazel fills out and uses to generate web pages. To emphasize the point: the templates listed below are *not* the web pages the shopper will see. They are, in effect, short computer programs that Hazel uses to *generate* web pages. But Steve Shopper doesn't know that, or care.

Somehow or another, Steve finds the store. He types in a web address, or does an online search, or sees an advertisement, or hears a friend mention the name. He clicks on a link that will bring him to the online store's "entrance" (presented by Hazel's start.html template). This web page can be whatever the merchant desires—a simple welcome page, a billboard for a featured product, a notice board explaining how the store is organized, a list of links to store categories, or even just a simple listing of everything that is for sale. Most online stores will have some or all of these elements combined together in their entrance or welcome or start pages.

In some form or another, on the entrance page or in sub-pages in links from it, Steve Shopper will, the merchant hopes, click on an item that interests him, or perhaps he will perform a search that brings up a list of item that might be

what he wants. If his browsing or his searching are successful, sooner or later he'll click on a link, which will probably be the name of the item, or even a picture of it.

The link will bring up a page (presented by the `detail.html` template) that describes the item more thoroughly, perhaps with a longer description, or a larger picture. This, obviously, is the equivalent of browsing the merchandise.

A sensible online merchant will work hard to make it easy for Steve to find things, and provide as much information in as attractive and clear a form as possible. The idea here is to compensate as much as possible for the fact that Steve *can't* take the item off the shelf and hold it in his hands. Good navigation aids, a good search system, and lots of pictures and information help.

A sensible online merchant will make also sure there are well-placed, easy-to-understand, and non-obnoxious add-item-to-cart links on *every* page that lists items, and on every page that presents detailed information.

The merchant might wish the customer to be shown the complete contents of his cart every time he adds something to it, or the merchant might wish to let the add-item process happen in the background, so as not to disturb the shopping process. Either way, Steve needs lots of links to allow him to view the cart contents whenever he wants, and reassurances that he can add and remove items without being committed to buying them.

Steve shouldn't have to finish shopping on his first visit. Even if he leaves the merchant's online store and returns at a later time, perhaps even days or weeks later, he can (or at least should be able to) view the unchanged contents of his cart.

Here the merchant has to strike a balance: she wants to allow the customer to return later and resume shopping, but, of course, she doesn't want to encourage him to leave the store before buying. She would much rather get the order today, rather than be left hoping to get it tomorrow. Mostly merchandising system have this sort of remember-cart feature (Hazel does) but usually the customer isn't told about it. The balance is struck by allowing but not encouraging the behavior.

This is a small but significant example of social engineering. You are not forcing Steve to what you want, but you are trying to steer him toward what you want him to do. If you tell him he can leave his cart and come back, you are increasing the odds that he will leave and *not* come back. If you don't telll him, you are discouraging him from leaving—but making life easier for him if he does leave and come back.

Let's put Steve back in our imaginary bricks-and-mortar store for a moment, but make some changes to it. Let's make him the *only* customer in the place, and remove even the possibility that other customers might come in. Let's make

sure the checkout line is *always* open, and staffed by clerks that Steve knows to have been heavily medicated in such a way that they can never lose their tempers, and can never make a mistake. In fact, because Steve is all alone in a online store, let's make the checkout clerk in our imaginary bricks-and-mortar store a completely emotionless robot that does addition perfectly but doesn't talk at all.

Now Steve has no inhibitions about going in and out of the checkout line as often as he wishes, changing his order each time. He can get his current order total, see what discounts have kicked in, check the shipping charges, and so on as many times as he wants.

That's pretty much how Hazel's checkout page (presented by the checkout.html page) functions. It presents the current order with all charges and totals, but it does *not* require the shopper to complete the order. Here again, there is a balance that must be struck. The shopper shouldn't feel committed just because he's checking his order total, but at the same time he needs to be encouraged to go ahead and complete the order.

The checkout page is also where Steve has to fill out a few boxes, with his name, address, his choices for shipping, and the means by which he will pay. Yet another balance that must be struck: the merchant needs to retain information without making the customer worried that the information is going to be abused. If the customer has filled out name and address information, your online merchandising system should remember the data—but should also make sure that the customer knows the information will not be used inappropriately. Your store will need a fair and clear privacy policy, which should be posted prominently. At the same time, the process of filling in the data should be as clear, easy and inuitive as it can be, and be as hassle-free as possible.

Once Steve has everything the way he wants it, he swallows hard, and clicks the SUBMIT ORDER button.

If he has made some mistake that Hazel has been programmed to test for—such as leaving out his postal code, or entering a malformed email address, he will see an error page (presented by the error.html template) telling him what's wrong and what he needs to fix. If everything is filled out properly, he will see whichever finish-order pages (presented by the FINISH action, which calls the template that applies to his payment type: credit_card.html, printout.html, etc.).

The payment page will present Steve with all the name and address information he has filled out already, giving him a chance to spot any errors. The page provides a link he can click on to go back and make any needed corrections.

The various payment types work a bit differently from each other, but

most orders are going to be by credit card. We'll follow that example here. The credit card payment page will request that Steve enter his credit card number and expiration date. Steve fills in the information, and hits that final submit payment button. He'll get another error message from error.html if the credit card number doesn't pass a very preliminary check. If the number seems all right, Steve will see a page generated by message.html telling him the order has been accepted.

He will also receive an email message, generated by the shopper_invoice.txt template, providing him with full details of his order. With that, his order is complete, and all Steve has to do is wait for delivery.

A lot of what is going on in all this can once again be thought of as benign social engineering: you are making it as easy as you can for Steve to do what you want him to do: buy something. Steve, on one level or another, knows he is being guided (or, to put it more bluntly, manipulated), and is cooperating, because he is getting what *he* wants: information, and, perhaps, your merchandise.

Information comes at him directly and indirectly from all sorts of cues on the web pages he sees, in all sorts of ways. He absorbs it consciously and at a gut level. Those gut feelings will help him decide if he likes the merchandise and the offered services enough to pay for them; if he trusts the merchant; if the product prices and service charges are no more than he is willing to pay; if the shopping process is easy enough and pleasant enough for him to continue; and if the shopping system is not so intimidating or so seemingly untrustworthy that he is chased away.

In other words, Steve is there in your store deciding if he wants to buy, and he is willing to be convinced—but, to return one last time to the analogy of the bricks-and-mortar store, Steve is in there *all alone*. You can't go up there and talk to him. There are no clerks to chat with him, no other customers clustered around the popular items, no manager there to listen to his questions. In that sense, an online store is a very private place, and maybe even a lonely one. It's just Steve alone with your store. And therefore, it's your store, *all by itself*, that has to convince him to buy.

With Hazel, you can build a store whose ways of doing business Steve will understand intuitively, on the first try, a store that provides him with lots of flexibility, and yet prods him, very gently but firmly, toward that checkout stand.

As we have seen in the **High-Speed Hazel** chapter, it doesn't take long to set up a basic Hazel store. If we assume that you already have a functioning website on a properly configured server, you are only a few hours works away from having a very basic store, based directly on Hazel's distribution files, online

and up and running. With that much accomplished, and with this manual in hand and Hazel's online documentation on screen, you can then move on to build your own customized store, designed exactly and precisely the way *you* want it.

Read on to see how it's done.

A Closing Note

That pretty much wraps up this Quick Guide to Hazel.

I can safely say that this guide not only tells you everything I know about Hazel—it tells you *more* than I know about Hazel. I found myself looking things up in this guide even as I was writing it. I know, with absolute certainty, that there are a lot of Hazelistas who know more than I do, and who monitor the Hazel Knowledge Base. If it's not in this guide, or the online documentation, try asking a question in the Knowledge Base next.

However, if you're feeling optimistic about your chances of getting useful information, or (far more likely) if you've spotted an error in this guide, drop me a line at hazel-fox@foxacre.com. And do what you can not to share that address with spammers. Thanks. And stop by www.hazelbook.com to see if there's anything of interest on the web board.

As I noted earlier, no single guide can explain everything that a system like Hazel can do. But, by now, you should have a least a good strong grounding in how Hazel is structured, the syntax of her various elements, and the power and flexibility of the Hazel Internet Merchandising System.

However, knowing all that doesn't do you much good by itself. Even the most powerful tool is not of much help if you don't put it to use. Hazel can't do you any good until you get her up and running. So dive in there. Get an initial, basic version of your shopping system up and running—and then start inventing your very own, customized, unique, built-just-for-you web-based merchandising system.

It'll be fun.

Roger MacBride Allen
January, 2007
Takoma Park, Maryland

Index

About the Author

Roger MacBride Allen was born in September, 1957. In his day job, he writes science fiction novels. His wife, Eleanore Fox, is a diplomat in the United States Foreign Service, which accounts for Roger's twenty-two months of commuting from Washington to London while they were dating, their living in Brasilia, Brazil for two years, the posting to Washington, and their three-year stint in Leipzig, Germany. They returned to the Washington area in summer 2005, and live there with their sons, Matthew and James. They are all constantly harassed by two cats—one three-legged and Brazilian, the other American and paranoid. You can learn more about Roger (or what's left of him) at www.rmallen.net, and learn more about FoxAcre Press at www.foxacre.com. Visit www.hazelbook.com for more on Hazel.

Books by Roger MacBride Allen

Novels
The Torch of Honor
Rogue Powers
(the above two novels were reissued in one volume as *Allies & Aliens*)

Orphan of Creation

The Modular Man

The War Machine (with David Drake)

Supernova (with Eric Kotani)

In the Universe of the Hunted Earth:
Farside Cannon
The Ring of Charon
The Shattered Sphere

A Trilogy of Asimovian Robot Novels:
Caliban
Inferno
Utopia

The Corellian Trilogy of Star Wars Novels:
Ambush at Corellia
Assault at Selonia
Showdown at Centerpoint

The Game of Worlds

The Chronicles of Solace:
The Depths of Time
The Ocean of Years
The Shores of Tomorrow

BSI Starside: The Cause of Death

forthcoming:
BSI Starside: Death Sentence

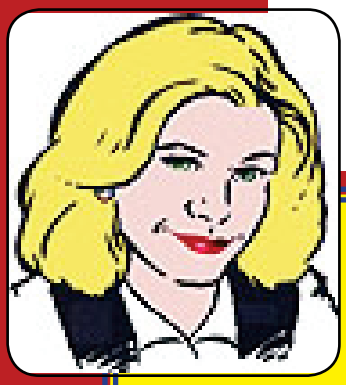
Nonfiction:
A Quick Guide to Book-on-Demand Printing
The First Book of Hazel: A Quick Guide to the Hazel Internet Merchandising System

More Information

The First Book of Hazel is chockful of Hazel tips, tricks, code examples, syntax examples, rule explanations, and so much more. You can order your copy, in one of a variety of formats, from www.hazelbook.com. We are also working to set up an online discussion board for Hazel at that website, but it's going to be up to the users to log on and contribute. Hope to see you there!

Another great place to learn more about Hazel is at hazel.netsville.com. The online documentation there contains lots more information—and of course, that's the only place you can BUY Hazel—and you can try before you buy. Go to hazel.netsville.com/trial/ to apply for a free thirty-day license. Fair warning though—at the end of the thirty days, you're going to want Hazel—so be prepared to buy!

Also visit the Hazel Knowledge Base at hazel.netsville.com/support/kb/. There are hundreds, if not thousands, of questions and answers about Hazel from years gone by—and plenty of current queries as well. Search on a topic, and see what the combined received wisdom of the Hazel community might be. Fair warning, however, that a question and answer from six or seven years ago, based on old versions of Hazel, might lead you in the wrong direction. (For example, you might see a long involved work-around that makes up for a feature that wasn't in the 1999 version of Hazel—but is there now.) Tip: pay attention to the dates on your search results. You can even search on a year. Search on search 2007 to get the latest comments on Hazel search—without getting comments from 1997.



Sample Chapters and Other Information from

The First Book of Hazel:

A Quick Guide to the Hazel Internet Merchandising System

by Roger MacBride Allen

Learn how to use the most flexible and powerful Internet Sales System available for small business. Run your shopping cart, track shipping charges and sales tax. Handle credit cards, PayPal transactions, checks and money orders. Use graphic images, frames, Java—and do so much more besides.

Don't be hamstrung by the pre-fab, cookie-cutter solutions other so-called e-commerce systems force you to use. Design your website your way, with total flexibility.

If it's about selling from your website—Hazel can do it all! This manual will take you through the basics of installing and integrating Hazel into your website, and show you all the ins and outs of this powerful online program.

FoxAcre Press
www.foxacre.com

Hazel
The Internet Merchandising System