

Take Control *of* Your Domain Names

by Glenn Fleishman

Table of Contents (Version 1.0)

Read Me First	2
Introduction	5
Quick Start	6
Master Domains and Domain Roles	7
Find and Register a Domain Name	21
Set Up Your DNS Host	36
Host Your Services	45
Redirect Your Domain's Web Sites.....	56
Use Dynamic DNS	60
Move Your Domain Name.....	68
Troubleshooting.....	80
Appendix A: Tools for DNS Lookups	88
Appendix B: Sell a Domain Name.....	91
Appendix C: Top-Level Domains	94
Glossary	96
About This Book	99

Help a Friend Take Control!

Click Here to Receive a Discount
Coupon for You and Your Friend

Check for Updates

Click Here to Look for
Updates to This Ebook

\$10

ISBN: 1-933671-21-1

TidBITS Electronic
Publishing

READ ME FIRST

Welcome to *Take Control of Your Domain Names*, version 1.0.

This book explains how to register and manage domain names to handle Web sites, email, and other kinds of services. This book was written by Glenn Fleishman, edited by Joe Kissell, and published by TidBITS Electronic Publishing.

To get in touch or learn more about the Take Control ebooks, you can:

- Contact us by sending email to tc-comments@tidbits.com.
- See [About This Book](#) to learn about the author and publisher.
- Read the fine print on the [copyright page](#).
- Find answers to general questions by reading the FAQ at <http://www.takecontrolbooks.com/faq.html>.
- Buy another ebook by checking out our [Featured Titles](#) or by visiting <http://www.takecontrolbooks.com/catalog.html>.

The price of this ebook is \$10. If you want to share it with a friend, please do so as you would with a physical book, meaning that if your friend uses it regularly, your friend should buy a copy. The Help a Friend Take Control button on the [cover](#) makes it easy for you to give your friend a discount coupon.

We may offer free minor updates to this ebook. Click the Check for Updates button on the [cover](#) to access a Web page that informs you of any available or upcoming updates. On that page, you can also sign up to be notified about updates via email.

Onscreen Reading Tips

We carefully designed the Take Control ebooks to be read onscreen, and although most of what you need to know is obvious, note the following for the best possible onscreen reading experience:

- Blue text indicates links. You can click any item in the Table of Contents to jump to that section. Cross-references are also links, as are URLs and email addresses.
- If nothing happens when you click a link or button, try picking a different “tool” for the mouse pointer. In Preview, look for a different tool button on the toolbar or look for a Tools menu on the menu bar. In Acrobat, make sure the Hand tool is selected.
- Work with the Bookmarks tab or drawer showing so that you can always jump to any main topic by clicking its bookmark.
- In Adobe Acrobat Pro version 6 or 7, set your preferences to view Web URLs in a Web browser: choose Acrobat > Preferences, switch to the Web Capture pane, and choose In Web Browser from the Open Web Links pop-up menu.
- The [Glossary](#) defines a number of television-related terms, which also appear in the text in blue, italic formatting. You can click blue, italic text to move to the glossary page that defines it; you can then return from the Glossary to where you were reading using a menu command or keyboard shortcut, as noted in **Table 1**.

Table 1: Navigating to the Glossary and Back		
Viewing Software	Menu Command	Keyboard Shortcut
Adobe Acrobat 6 and 7	View > Go To > Previous View	Command-Left arrow
Adobe Acrobat 5	Document > Go To > Previous View	Command-Left arrow
Preview	Go > Back	Command-[

- Clean your computer’s screen. It’s easy to forget how much that simple step can improve your viewing experience!
- Find more tips in the [Take Control FAQ](#) on the Web.

Printing Tips

Although our layout is aimed at making online reading an enjoyable experience, we've made sure that printing remains a reasonable option.

Want a high-quality, spiral-bound printout? *Instead of printing on your own printer, you can buy a printed copy using our print-on-demand service.*

The service prints on double-sided pages, scaled to 7" x 9" to reduce the font size to what's expected in print, and it binds the pages with a Wire-O binding so that they lie flat on your desk. You can buy the printed book in black-and-white or color.

Click the Check for Updates button on the [cover](#) to access the print-on-demand ordering link, or read more about the service at <http://www.takecontrolbooks.com/print-on-demand.html>.

Please review these tips before you print:

- Use the Check for Updates button on the [cover](#) to make sure you have the latest version of the ebook and to verify that we don't plan to release a new version shortly. If you want to commit this ebook to paper, it makes sense to print the latest possible version.
- Don't throw out your PDF after you print! You must click the Check for Updates button on the cover to get future updates. The link *must* be accessed from the cover of your PDF.
- For a tighter layout that uses fewer pages, check your printer options for a 2-up feature that prints two pages on one piece of paper. For instance, your Print dialog may have an unlabeled pop-up menu that offers a Layout option; choose Layout, and then choose 2 from the Pages per Sheet pop-up menu. You may also wish to choose Single Hairline from the Border menu.
- When printing on a color inkjet printer, to avoid using a lot of color ink (primarily on the yellow boxes we use for tips and figures), look for an option to print entirely in black-and-white.
- In the unlikely event that Adobe Acrobat or Adobe Reader cannot successfully print this PDF, try Preview; several readers have solved printing problems by using Preview.

INTRODUCTION

Imagine going to a meeting in the business district of a city you've never been to before. The office buildings have numbers, but the streets have no names. Entering one of the numbered buildings, you find that there's no building directory and no tenant signs on doors. Just floors full of enumerated suites. Within the offices, there are no receptionists. Each cubicle or office has just a number affixed.

Sounds like a nightmare, no? But it's a way to visualize the way the Internet would work without domain names, which translate the numeric addresses that identify connected computers into something that people can grasp and remember.

A *domain name* is part of what all visitors type in or click on to visit a Web site you operate, and it's the latter part of what they type to send you email. Setting up a domain name can be frustrating because so many discrete parties and pieces have to be put together. Even minute configuration errors can kill Web sites and cause email to bounce. Experience shows that it's often more irritating to register, configure, and manage a domain name than to operate a Web site.

This book helps you avoid domain name aggravation. It teaches you how to register and manage a new domain, how to work with hosting companies that handle each part of a domain name's operation, and how to use features you might not have thought of before. You will learn about the *domain name system (DNS)*, the set of technologies that allows Internet users to type in names and have them connected to Internet addresses by number.

I also show you how to migrate a domain's registration, hosting, and technical details from one or more firms to one or more others. Finally, I offer troubleshooting tips for common domain name problems.

NOTE What makes me such an expert? I registered my first domain name in 1994, and sold my first domain name in 1995 for a few hundred dollars. Over the last 12 years, I've dealt with every change in the commercialization of domain names. I've torn my hair out dealing with domain names so you don't have to.

QUICK START

If you're registering your first domain name, I recommend that you work through the book in order, paying attention in particular to the early sections that teach you how domain names work and how to register a domain name, modify the domain's settings, and launch a Web site under that name. After that, refer to other parts of the book as you need them.

Understand domain name basics:

- Get to know the parts of domains. See [Putting Domain Names Together](#) (page 8).
- Understand the roles for registering domains. See [Learning Domain Roles](#) (page 18).

Obtain a domain name:

- Decide on a domain name. See [Search for Names](#) (page 22).
- Register your domain name. See [Register a Domain Name](#) (page 28).

Set up hosting:

- Choose a DNS host and configure DNS settings for Web sites and email servers. See [Set Up Your DNS Host](#) (page 36).
- Choose a Web host and set up a Web server. See [Host Your Web Site](#) (page 45).
- Choose an email provider and set up accounts. See [Receive Email](#) (page 50).

Move your hosts around:

- Redirect access to Web sites via DNS, HTML, or JavaScript. See [Redirect Your Domain's Web Sites](#) (page 56).
- Have a subdomain follow a changing address. See [Use Dynamic DNS](#) (page 60).
- [Change Your Registrar](#) (page 70), [Change Your DNS Host](#) (page 73), [Change Your Web Host](#) (page 73), and [Change Your Email Host](#) (page 75).

Troubleshoot your domain problems:

- Solve common problems such as bouncing email, DNS server errors, and registration expiration. See [Troubleshooting](#) (page 80).

MASTER DOMAINS AND DOMAIN ROLES

In this section, I explain what domain names are and how the Internet keeps track of them, and give you important background information that will help the rest of this book make more sense.

Domain names hide the underlying complexity of the Internet, which uses numbers—called *Internet protocol (IP) addresses*—to identify computers and other Internet-connected devices rather than human-readable names. IP addresses can be entered by human beings, but they're difficult for most people to remember.

NOTE A human-unreadable name might be a long inventory number created by a system or a hated information technology manager, and look like “186-18-131-252” (an IP address turned into a legitimate part of a domain name); a human-readable name could be “bob-the-computer.”

IP addresses can change over time, as computers move from one place to another, as Web sites are migrated from one server computer to another, or when an *Internet service provider (ISP)* changes the temporary or *dynamic* IP address that's assigned for a period of time to a broadband subscribers. Domain names, by contrast, can remain constant over time. My Web site has been **www.glennf.com** for several years and my email **glenn@glennf.com** for the same period. The IP addresses assigned to the Web site's server and email server have changed at least ten times in that interval.

What makes this association work over time is that IP addresses may be assigned by an ISP or other network provider, but you can control your domain name directly. Domain names are set up to correspond to IP addresses that represents specific computers that you operate directly or that are operated by a hosting company that handles your Web site, email, DNS settings, and more. (See [Set Up Your DNS Host](#) and [Host Your Services](#).)

Domain names are unique worldwide, unlike people's names and business names. No two entities on the Internet can own and use the same domain name; there's only one **apple.com** and only one **bob-built-this-obscure-site.org**, too. This exclusivity requires

coordination, which is why you can't just invent and use a domain name, but must register it with an organization that ensures overlap doesn't occur.

Owners of domain names can create their own prefixes to domain names, too, resulting in unique local names for computers on their own networks that are also unique globally when the local name is combined with the domain name. For instance, networking firm Linksys owns `linksys.com`, and they have prefixed that with `support` to place technical support and firmware upgrades at `support.linksys.com`.

Before figuring out which domain name you might want or how to extend a domain name you own, let's look at what role domain names fill, how domain names are defined, and what level of control you have over them.

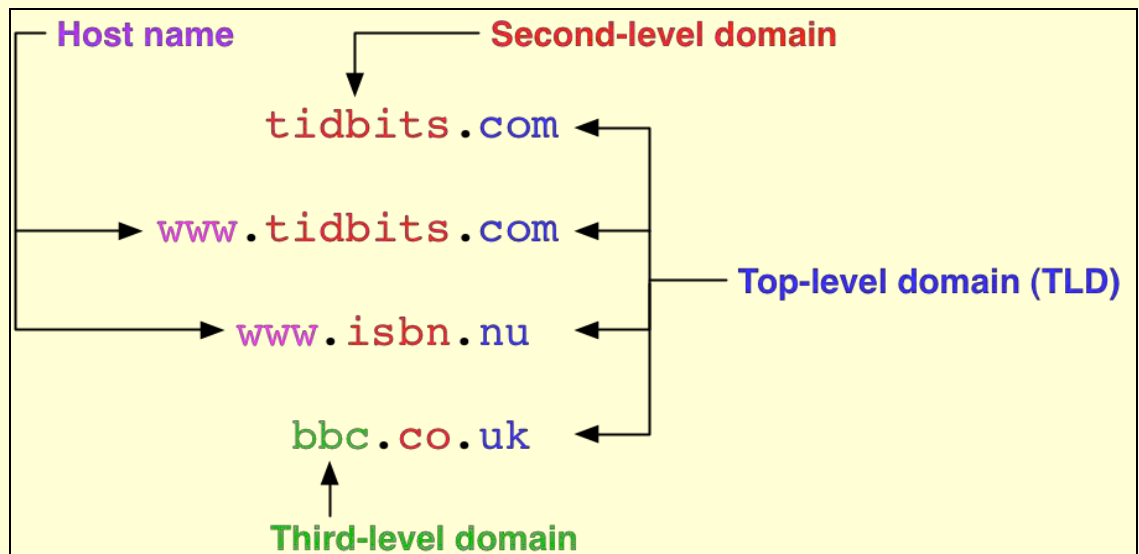
Putting Domain Names Together

What most people call a domain name typically comprises three distinct parts, only one of which meets the formal definition. The first part, a prefix, is the *host name*; the second part is the actual *domain name*; the third part is a *top-level domain (TLD)*, controlled by a global authority or a specific country.

Figure 1 (next page) identifies the parts of a domain name, which are controlled by the progressively lower levels of authority as you read it from right to left, divided by periods. For instance, `www.tidbits.com` is read as follows:

- **com:** A top-level domain (TLD). TLDs include familiar suffixes like `.com`, `.net`, and `.gov`; country codes such as `.cn` for China and `.nu` for the Pacific island-nation of Niue; and newer content-restricted generic TLDs such as `.aero` (related to airlines), `.museum` (museums), and `.coop` (cooperatives).
- **tidbits:** The unique domain name within the `.com` TLD. Names can contain only alphanumeric characters from Western alphabets, like English, and a handful of letters from non-Western alphabets. The only allowable punctuation is the hyphen. Because this name is one unit to the left of the TLD, it's called a *second-level domain*.
- **www:** The host name set by the domain name owner.

FIGURE 1



Domain names are read from right to left, starting with the element controlled by the highest authority, the TLD.

Some countries, such as the United Kingdom (.uk) and Australia (.au) don't allow the second-level domain to be assigned, instead further subdividing their TLD with their own second-level domains like "co" for company. In **Figure 1**, you can see that even the venerable "beeb" (the British Broadcasting Company) must be content with a *third-level domain*.

NOTE The **www** that identifies a Web site is just an early convention that has no structural meaning. You can give a Web site any host name you want, although people may type in **www** plus your domain name, assuming that's the Web site's location.

If you want the domain name by itself *and* preceded by **www** to point to the same Web site, you can use a shortcut. You define an IP address for the domain name, and then set up a canonical name (or CNAME) for the **www** host name. This shortcut returns the same IP address without having to specify it a second time. Many hosting companies automatically create that shortcut for you in DNS.

Traditionally, a host name was the name of a local computer, unique on the local network, while the domain name was unique globally. A host name might be something as straightforward as **www**, or as specific as **second-floor-server**. You or a network administrator

can control how hosts are named. A domain name is the global portion of how you reach a computer by name on the Internet. A domain name could be **glennf.com**.

Combine the host name and the domain name and you get a *subdomain*. A subdomain identifies a conceptual area, like **www.glennf.com**, where one or more services are operating, such as a Web site or email server; or just a specific computer, like **lab-powermac-03.brandeis.edu**.

The domain name by itself is actually a subdomain as well because the global name authority gives you control of the domain name coupled with any prefix. That means that **glennf.com** and **www.glennf.com** are equally subdomains within my control that I can set to point to any resource, such as a Web site, that I choose to. The domain name, **glennf.com**, is still globally unique in this context while under my local control.

TIP Subdomains are often referred to less precisely as domain names, and that's not exactly incorrect; the definition of a subdomain is a bit fluid. For the purposes of this book, however, I can't muddle the two together, or you'll never know what I'm talking about. Thus, in this book, a *subdomain* is always a host name plus a domain name, or, as in the paragraph just above, an implicit subdomain that comprises just the domain name. A *domain name* is always the globally registered part of the subdomain; subdomains are, by definition, unique because domain names and host names combined into a subdomain must be unique.

Large companies often use subdomains to distinguish offices or operations. For instance, a company like IBM might define **europa.ibm.com** and **australia.ibm.com**, and then *delegate* control of those subdomains to network administrators in each country. Those network administrators could further define subdomains, like **france.europa.ibm.com**, and pass those along to another network admin, and so on.

In order to get started with a domain name, an important step is to decide which name you want. However, since your domain name must be unique, your first choice might already be taken. I talk much more about this in [Register a Domain Name](#).

The Role of the Domain Name System (DNS)

Domain Name System (DNS) servers and related software are the glue that binds subdomains and IP addresses. A DNS server has two intertwined functions: hosting information about a given domain and its subdomains, and discovering information about a requested subdomain from another DNS server. Without DNS servers, people would have to enter numerical IP addresses whenever they wanted to read a Web page or send email.

A DNS server may be located within a company's network, on a server you operate, or, most frequently, on an ISP or network provider's network.

For DNS to work properly, every domain must have its technical details stored on at least one DNS server, and typically on two to four servers for redundancy and resiliency. So, when you set up a domain name, you don't just pick a name and register it, you must also ensure that it is hosted by a DNS server. (I talk about how to handle DNS hosting later in this section and in [Set Up Your DNS Host](#).)

DNS allows any Internet-capable software, such as a Web browser, an email program, a Web browser, or a streaming video player (say, QuickTime Player or RealPlayer), to accomplish a task like sending email, showing a Web page, or downloading video without knowing anything whatsoever about domain names. DNS accepts queries from these types of applications, performs magic known as resolution (described ahead in this section), and returns the necessary information so that the application can act upon it.

Let's first look at how subdomains relate to services, such as serving up Web pages or email messages, and then how resolution works its way through DNS.

Pointing subdomains at services

In everyday language, common examples of *services* are *mail servers* (which help to send and receive email for email addresses at a subdomain) and *Web servers* (which help to make Web pages available from a subdomain). If you are setting up a domain name, then presumably you want to run—or pay someone to run for you—one or more services from that domain name. Other well-known services include FTP (File Transfer Protocol), Samba (Windows-style file sharing), AppleShare (Mac-style file sharing), and remote access.

From a more technical perspective, services are pieces of server software that respond to queries, typically requests from a piece of client software for particular information. For example, a Web browser sends queries in the form of requests for pages or images from a Web server. For the Web browser to find the Web server, someone must give the browser a URL (Uniform Resource Locator) that includes the correct *scheme* (such as `http://` or `ftp://`) and a subdomain.

Because services run on computers that are located at specific IP addresses, a piece of Internet-enabled software trying to connect to a service relies on DNS to discover the IP address associated with the service's computer. For example, an obvious case is the way I've mapped `www.glennf.com` to my Web server, which lets any potential visitor reach `www.glennf.com` and view its home page without knowing the server's IP address.

Each common service has an assigned *port*, which is like a suite number in a building (where the IP address is like the street address). Just like a building manager's office might often be Suite 100, Web sites are almost always found at port 80. (Ports have been assigned until recently by the Internet Assigned Number Authority, IANA.)

TIP You can run services on ports to which they aren't normally assigned, but that requires coordination with users who want to find those services. The non-standard ports 8000 and 8080 are often used for Web servers built into home networking products, for instance. For someone to reach one of those sites, she would have to add *:port* to the URL. For instance, a special Spacely's Sprockets site might be at `http://www.sspacely.com:8080/`.

You can run many services on the same IP address. For instance, I have a few dozen Web sites on one Linux machine I operate that one server program handles. Each Web site's subdomain is mapped in DNS to the same IP address. But the browser-server communication language, *HTTP (Hypertext Transfer Protocol)*, includes a method for the browser to request a Web page from a particular subdomain. This enables the Web server software to act as many different Web sites; these are called *virtual servers* or *virtual hosts*. Large hosting companies might host tens or hundreds of thousands of Web sites on a single server computer.

NOTE SETTING UP ONE IP ADDRESS FOR MANY COMPUTERS

DNS can be set up so that many subdomains point to the same IP address—the many Web sites, one computer example given just previously—or the reverse: many IP addresses may connect to a single subdomain. Companies that need to distribute server load over many computers would seemingly need this feature; 1,000 computers might be available to respond to a request to eBay.com, for instance.

In practice, although DNS has the capability to *balance load*, or efficiently hand off requests to different IP addresses based on how busy the computers at those IP addresses are, it isn't truly designed for the job. Instead, companies that need this kind of balancing use special gateways that look like one IP address to the outside world and hand off traffic to many machines on their local network. This process is entirely transparent to the requester.

Because most Web server software lets you make different global choices for each software instance of a server, you can use subdomains to serve the same content in different ways. For instance, you might set up `www.example.com` as your main Web site, and `print.example.com` as a site that substitutes a different Cascading Style Sheet (CSS) file that causes browsers to format the same content in a way that can be easily printed.

NOTE FTP server software cannot differentiate among requests for files by subdomain when multiple subdomains point to the same IP address. While this has been a trivial feature of Web server software since about 1995, the FTP protocol was never extended to handle receiving the subdomain name along with an FTP command. Thus, the only way to operate unique FTP file repositories in which you require no login—allowing *anonymous users* to download software, for instance—is to assign each subdomain a unique IP address.

If you're setting up FTP for multiple subdomains in which each user must have an account, most FTP server software can be set up like a virtual Web site, using the account name at login to determine which directory of files to allow that user to see. That's how hosting companies use a single FTP address to handle many thousands of users.

Because DNS is just a pointer for IP addresses, you can locate services wherever you want, with more than one per computer, or different computers handling different services. You might run FTP on one computer, host your email at a paid provider, and use a high-end hosting firm to handle your Web site or sites. Each unique computer that runs your services requires at least one IP address, but that's the only real limitation. For instance, `ftp.tidbits.com`, `www.tidbits.com`, and `db.tidbits.com` point to entirely different computers, each of which has a slightly different function and a different IP number.

Resolving domains

DNS uses *resolution* to find the IP address associated with any domain name. Your computer (or handheld or other Internet-capable device) uses a piece of low-level software known as a *resolver* to let applications ask for the IP address that corresponds to a given subdomain. (It can also ask for the names of mail servers, and a few other more obscure pieces of less-used information.)

As a user, you interact only indirectly with a resolver; your software asks the resolver for the numeric IP address associated with the subdomains that you enter. The resolver needs to know the IP address of at least one DNS server to ask that server to look up the IP address for a human-readable subdomain.

Setting up DNS resolution for an individual's computer

A chicken-and-egg problem quickly emerges. If you are using DNS to turn a subdomain into an IP address, how do you find a DNS server? Something has to prime the pump. To bypass this problem, every computer or device that uses DNS must have the IP addresses of one or more DNS servers. These addresses can be entered manually, or they can be provided as part of dynamic address assignment:

- **Manually:** In the case of manual assignment, the person who operates your network will provide you with the IP addresses of two or more DNS servers that you then enter manually in the network configuration window of your computer or other device. My ISP, for instance, lists DNS server IP addresses on my account status page, so I can log in from any machine with Internet connectivity and look those details up if need be.

NOTE While your computer can be configured so it has just a single DNS server's IP address, that's not wise. Even reliable servers need to be rebooted or have maintenance performed at times.

- **Dynamically:** With DHCP (Dynamic Host Configuration Protocol), a computer typically receives an IP address that is assigned to the computer by a DHCP server that is located on the same local network. The DHCP server also provides the gateway address for moving data beyond the local network out to larger networks or the Internet, and it provides the IP address or addresses for DNS servers. This automatic handoff avoids requiring individual users to enter DNS server IP addresses.

How DNS resolution works behind the scenes

A DNS server has a preset list of *root nameservers*, which store information for reaching TLD nameservers. As you can see in **Figure 2** (next page), when you use software that references a domain name, your computer's resolver queries one of the DNS servers in its list. That server asks the root nameservers for the TLD's nameservers, which in turn provide information down the subdomain hierarchy from right to left until the DNS server finally arrives at the DNS server that has the requested address information.

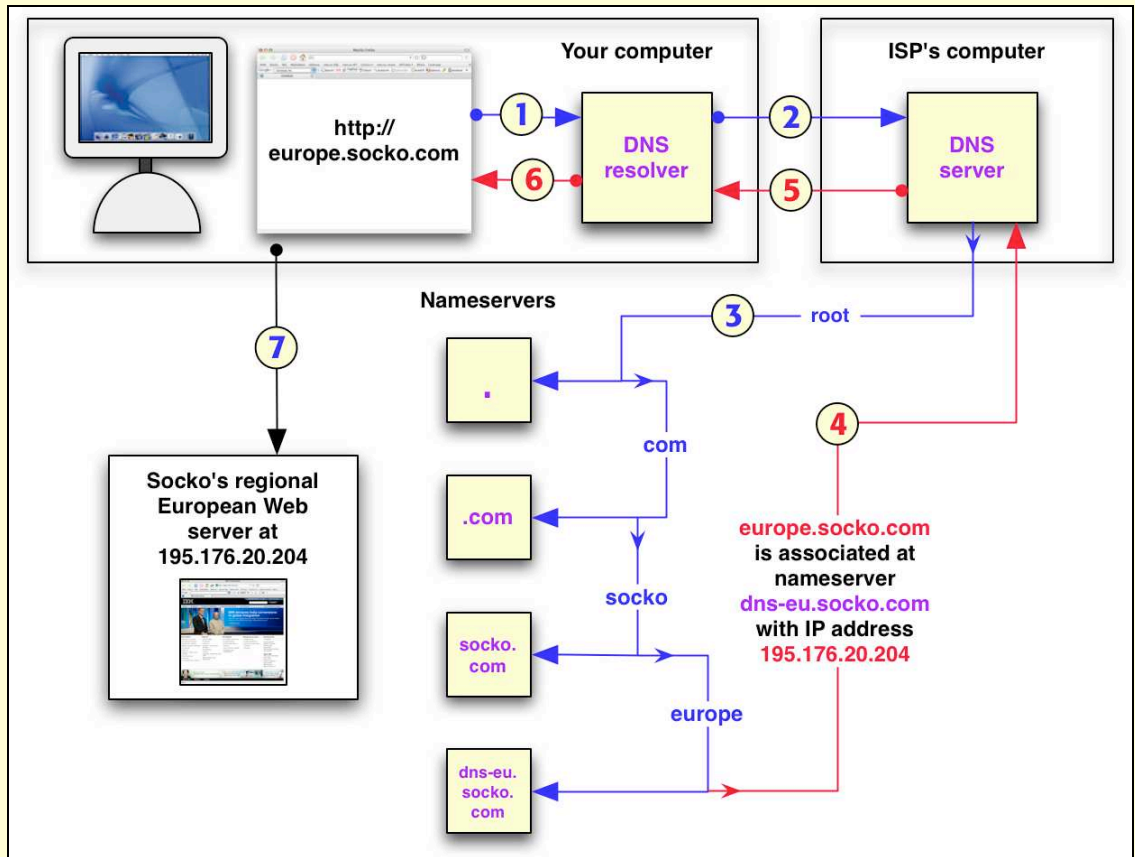
(Remember that domains names read hierarchically by parts separated by periods from right to left: **com** is at the top, then a domain name in the **.com** hierarchy, then a host name in that domain, if any, and so on until all the host names separated by periods are exhausted on the left.)

The last DNS server in the chain retrieves the requested address information and passes it back to your computer's resolver. Your application then uses that numeric address to create a direct Internet connection between itself and the server that has the service it's trying to work with. (That connection is technically known as a *socket*.)

DNS caching

DNS details are stored, or *cached*, in resolvers and DNS servers for a period of time that ranges from a few minutes to a week. This cached data means that particularly popular sites like YouTube don't have their DNS servers queried thousands of times a second from every

FIGURE 2



An example of DNS resolution:

1. Your browser queries your computer's DNS resolver for `europe.socko.com`'s IP address.
2. The resolver queries your ISP's DNS server.
3. The ISP's DNS server starts at the top of the domain hierarchy querying, in turn, the root nameserver (represented in DNS as just `.`, or a dot), `.com`'s nameserver, and `socko.com`'s main nameserver. Because Socko Corp. delegates authority for its divisions worldwide, your ISP's DNS server is told to check `dns-eu.socko.com` for European DNS information.
4. Your ISP's DNS server checks `dns-eu.socko.com` and receives back the IP address for `europe.socko.com`.
5. The ISP's server hands back the IP address to your DNS resolver.
6. Your DNS resolver hands that address to your browser.
7. Finally, your browser connects to the European Socko Web server by its IP address to retrieve the content.

ISP on the planet; rather, the cached data allows an ISP's own DNS servers to respond directly to DNS requests for minutes or days. On the scale of AOL or EarthLink or Comcast, that's possibly tens of billions of DNS requests that stay within those ISPs' networks each year. (For more on this caching period, see [Time to Live](#).)

It also means that DNS lookups take enormously less time for those visiting your site—a matter perhaps of milliseconds instead of tenths of or even whole seconds just to get an IP address and proceed.

SIDEBAR SMELLS LIKE DNS PROPAGATION

DNS values don't propagate over the Internet, although the process is invariably described this way. In propagation, there would be a central server that would distribute its information in an orderly fashion to a hierarchical network of servers beneath it. DNS is somewhat decentralized, relying on querying DNS servers finding out where to ask about a given domain name. Thus, with DNS, every domain specifies in its basic settings how long a time period other DNS servers should cache that domain's values before refreshing that information by directly consulting one of your domain's DNS servers. Think of it like buying milk. Most of us open the fridge and check the milk's expiration date or its smell. If the milk's past its prime (or we have no milk and need it), we go to the store and buy some fresh milk. That's a far cry from fresh milk appearing in our fridge all by itself when we need it. The same is true with DNS. Our DNS servers run out to the store for a gallon of milk as needed rather than receiving ongoing deliveries at the back door.

Freshness becomes critical if you've set up your domain values to be held for up to a week at other DNS servers, but you make a change that you want picked up right away. Perhaps you're migrating a service—moving a Web site from one computer to another—or you're using dynamic DNS, which allows a subdomain to be rapidly repointed to whichever IP address you're currently at (see [Use Dynamic DNS](#)). DNS lets you control freshness. (See [Time to Live](#) for further details, and how to change this setting.)

Learning Domain Roles

When you set up a new domain, you must figure out who will handle each of four jobs relating to the domain. Some of these jobs you can do yourself, but others you must pay someone else to handle. And, these jobs can be split among four or more companies, each of them potentially located on different continents (**Table 2**). It's all up you.

Table 2: Domain Roles			
Party	Metaphor	Party's Role	Cost
Registrar	Getting a business name license from a government office	<ul style="list-style-type: none"> Records your ownership of domain for a fee Provides pointer to global root nameservers Pays fees to global domain overseer (ICANN) 	<ul style="list-style-type: none"> Basic service can range from a few dollars a year to about \$35 per year
DNS server	A dynamic phone book that always has the current address and phone number of your office	<ul style="list-style-type: none"> Root nameservers point to servers that contain your domain's details Responds to queries from all Internet users about your domain 	<ul style="list-style-type: none"> Free to \$20 a year when bundled with other services If you are technically adept, you can run own at no cost, except your sanity
Web, email, other services host	A rented office with a mail slot and a street-level window for displaying signs	<ul style="list-style-type: none"> Web hosts serve up pages; email hosts receive email Other services include database servers, file servers, and chat hosting 	<ul style="list-style-type: none"> Web: Usually \$10 to \$40 per month, depending on site size, complexity Email: free to several dollars a month
You, the owner	Business owner who receives mail, has exclusive rights to business name in that area	<ul style="list-style-type: none"> Configure DNS server to point to Web site, email server, other resources Pay recurring fees to other parties Maintain current contact information 	<ul style="list-style-type: none"> Time Money Hair replacement (but not if you read the rest of this book)

Registrars Register

In most parts of the world, business names must be unique within some political unit, often a state or province. A registrar, like a business licensing office, accepts a fee to register a name as a unique entity you own, noting your address and other contact details. Fail to pay in a later year and your domain or business license goes defunct.

A better way to look at it is that you're not buying a domain name; you're leasing it. The fees you pay to the registrar—whether you pay one year or 10 years at a time—keep you in good stead as the current owner. A registrar manages the information shown in **Table 3**.

Table 3: Information Your Registrar Maintains		
Item	What it's used for	Specifics
Registrant	To identify the legal owner of the domain	Mailing address
Administrative contact	Contact for registration, ownership details	Mailing address, email, phone, fax
Technical contact	Contact for technical details, like name-server problems, spam attempts, or network attacks originating from the domain	Mailing address, email, phone, fax
Billing contact	Contact for billing matters	Mailing address, email, phone, fax
Last updated	Last change to the global registrar database that you or the registrar made	Date
Expiration date	When your registrar says the lease period your current fees have paid for runs out	Date
Domain servers	List of nameservers that feed out information about your domain	Nameserver subdomains and IP addresses

Hosting Your DNS

As discussed in [Master Domains and Domain Roles](#), a domain name's details must be available on at least one DNS server that operates 24 hours a day. This server may be one you or your company operates, or one that's operated by another firm. As noted earlier, a DNS server is a piece of software that replies to requests from anywhere on the

Internet for your domain's information; a *DNS host* is a firm operating DNS servers for hire.

DNS hosts enable you to enter the requisite details for pointing to servers at which you host Web sites, receive email, and handle other services (**Table 4**). The better hosts also let you tweak more obscure settings, which have become more important over time.

NOTE Each domain's information should be stored in at least two different nameservers to avoid problems if one nameserver breaks down or becomes unreachable. One nameserver acts as the *primary* or *master*, and all changes are made at that server; others are *secondary* or *slaves*, and pick up information on a regular basis from the primary server. (A domain listing a DNS server that lacks information about that domain can cause *lame delegation*. See [Lame Delegation](#), later.)

Table 4: The Details Your DNS Host Maintains

Record Type	What It's Used For	Specifics
Address (A)	The list of hosts associated with a domain and the IP addresses they correspond to	Subdomain paired with one or more IP addresses
Canonical name (CNAME)	A way to have another name point to an A record without repeating the IP address	Subdomain paired with another subdomain that has an A record defined
Start of authority (SOA)	Provides authoritative information for a domain	Primary nameserver, email contact, timing values for distributing information to other DNS servers, secondary DNS servers
Nameserver (NS)	A list of all nameservers that respond with information for this domain	Subdomain paired with a nameserver, plus an entry that explicitly includes the nameserver's IP address(es)
Mail exchange (MX)	A list of one or more mail servers by name	Subdomain for mail delivery, precedence number for attempting delivery, and mail server's subdomain
Reverse address (PTR)	A connection from an IP address back to a subdomain	IP address paired with an subdomain, one per IP address

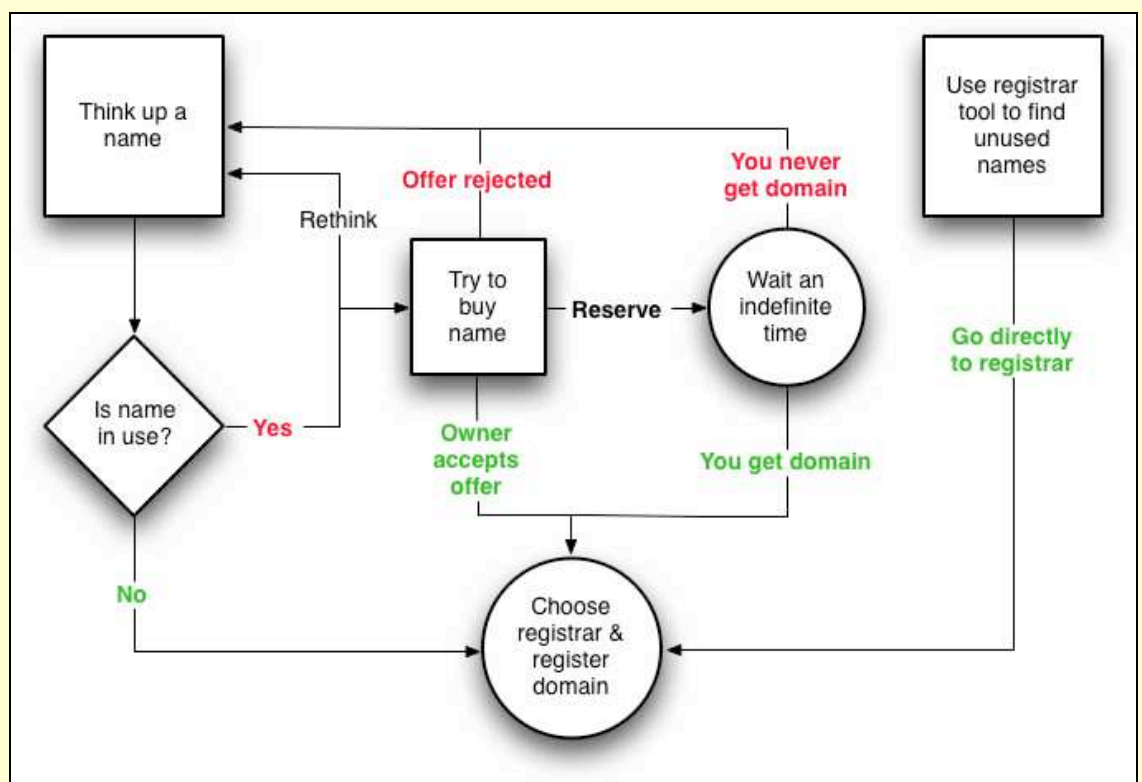
FIND AND REGISTER A DOMAIN NAME

Before you can register a domain name, you must choose a name to register. When you choose a domain name, you're generally making a long-term decision. Occasionally, you might pick a disposable name for a specific project (such as a high school reunion) and abandon the name after a year or two. However, most domain names persist.

Part of deciding on a name stems from the domain's intended use and immediate purpose. Will you start using the domain to serve Web sites immediately? Will you *park* it—reserving its use but displaying no content—while you figure out what to do with it? Will you redirect it to a server or hosted site you already operate? (I cover these options later in [Parking, Not Driving](#) and [Redirect Your Domain's Web Sites](#).)

Finding an available domain name can take effort, but **Figure 3** gives you an overview of the process and I look at the details next.

FIGURE 3



Choosing a domain name usually follows one of two tracks:

- Think up a name from scratch (upper left), see if it's in use, and, if not, register it. (If it is in use, you can try to buy or reserve it.)
- Use a registrar's tools to find unused domains (upper right).

Search for Names

Let's break down the process of searching for a domain name into two parts: thinking up a name and determining whether it's available.

What are you up to?

Start by carefully considering the domain name's purpose or organization. Here are some frequently used options:

- **Company name:** If you're trying to establish a company's identity on the Internet, the company name is a logical starting point. In the very early days of the Internet, I helped register some generic names, like **faucet.com**, for mail-order retailer Faucet Outlet. These days, it's nearly impossible to find something generic. And because there's no automatic protection for trademarks and business names, a company elsewhere in the world may have registered your business name already.

If your business is located outside the United States, consider choosing a domain name in your country code's TLD. However, many non-U.S. businesses still consider .com to be the only "real" commercial TLD. (For more details on TLDs, see [Appendix C: Top-Level Domains](#).)

TIP If you can't get the domain name you want and you're registering a business name, then taking your company name and adding "corp," "LLC," or "inc"—whichever the case may be—could help in finding a usable name. You might also precede a desired name with your location, like nyc (New York City), la (Los Angeles), or sf (San Francisco). Because most people find Web sites through search engines and links online, and direct mail and advertising offline, you can retain your identity without obtaining your precise name. The flip side is when several companies have similar names that aren't well differentiated. My colleague Jeff Carlson registered **necoffee.com** for his business "Never Enough Coffee Creations." The New England Coffee Company tried to acquire it from Jeff, but ultimately registered **necoffeeco.com**. Jeff routinely forwards email to the bean roasters, and has a link on his site. They've sent him some of their coffee as a thank you.

TIP Domain names may contain letters (including some non-English characters; see the next tip, below), numbers, and hyphens. But I recommend avoiding hyphens, even though they're the only legitimate punctuation mark allowed within a domain name. Having a hyphen means people have to remember to type it. Use a hyphen only if you expect the vast majority of your traffic to come from clicks, not typing, or you're so stymied that a piece of punctuation is your only solution to come close to your ideal domain name.

- **Your own name:** I missed registering both **glenn.com** and **fleishman.com** back in 1995 and 1996—blast you, Fleishman-Hillard International Communications!—but you might be luckier with a first name plus last name combination like my editor's **joekissell.com**. If you'd like a little more exclusivity, try the .name TLD. The TLD operator even offers a free trial, arranged through individual registrars licensed to offer .name addresses (<http://www.name/>; yes, that's really their address).
- **Information resource:** Many domains are set up just to offer a Web site with some specific information. This is where using registrars' tools to find good domain names can be a great help because so many English and other languages' nouns are taken.

TIP Most registrars can handle a limited set of non-English characters, such as letters with diacritical marks, ideograms, and non-Roman alphabets, as part of domain names. If your intended audience regularly uses symbols other than the 26 characters from A to Z that form the alphabet in written English, you might consider a domain name that includes language-specific characters or symbols. (Ideograms from non-Western languages are considered letters in this system, even though linguistically they're a full idea.)

I recommend avoiding the use of characters from languages that fall outside your readership. A lack of sensible, consistent standardization across browsers and operating systems could make you invisible to non-native speakers and non-local surfers.

- **Nonprofit or organizational purpose:** The .org domain is open to all comers, but I would suggest that any group primarily focused on service, charity, or religion consider .org for the

connotations it brings. Because .org isn't as highly desirable to businesses as .com, .net, and other TLDs, there's also a little more room in the universe of possible names—the *namespace*—to find good names. A nonprofit might also choose the same name in .com and .org, promoting the .org name in literature and redirecting the .com Web site and email to the .org domain.

- **Club or hobby:** Almost any obvious name associated with a sport or hobby has been registered, but you can use the research tools at registrars to get suggestions for unused domains. If mostly members will use the domain, you could choose something more elaborate, too.

WARNING! AVOID WELL-KNOWN NAMES

You might be tempted to register a name that includes a trademarked term that you don't own or license, which could be a product name or a business name. This is generally a bad idea. When you register a domain, you agree to a policy set by the global domain name authority (called ICANN, or Internet Corporation for Assigned Names and Numbers) that tries to prevent the unauthorized use of protected terms. This allows the trademark owners to request a hearing. If they can show that your domain name is similar to their trademark, that you have no relationship to the domain name, *and* that you're acting in bad faith, then they can be awarded the name.

Find a domain name

With those suggestions in mind, here's my strategy for picking names that you can then research:

- **Brainstorm:** Brainstorm with friends and colleagues. A friend of my wife's came up with the name Sudden Gardens for my wife's landscaping business—in which she rapidly preps and stages yards for houses going on the market—and **suddengardens.com** was available.
- **Search:** Search on Google and other search engines for associated words and concepts. If you're selling dahlias, type in "dahlias" and see what comes up in Google. Or enter "flowers" and see what kinds of words appear prominently on pages that match. Entering words that form part of an institution's name is particularly useful if you want to use an organization or business name in a domain name.

- **Synonymize:** Find an online thesaurus, like Merriam-Webster's (<http://www.m-w.com/>). Run through word combinations and riff on words that are associated with your starting point. An obscure but interesting fragment, word, or word phrase could stick in the mind. For instance, an ant specialist might choose the domain name **formicalliance.org** as a wink to those who study ants.
- **Make something up:** In a pinch, you can choose a set of letters and numbers that have no meaning in this life or the next until you breathe life into them, such as an abbreviation that's a sequence of letters with no pronunciation. But keep in mind that a newly coined domain name still needs to be memorable and easily typed—unless, of course, you're trying to hide your domain.

There is an example of someone registering a domain comprising 63 lowercase a's in a row, for instance. I'd hesitate to type that in, but **ajdsfldfjkad.com** doesn't leap trippingly off the fingers, either. My friend David Blatner sold the rights to the domain name **moo.com** in mid-2006, which I registered for him in the mid-90s, and replaced it with **63p.com**, a short, meaningless name that was still memorable.

TIP “Have a dirty mind,” my high-school journalism teacher used to say. Accidental double entendres are far too easy to make in headlines. With domain names, the lack of punctuation can turn Jim's Exchange into **jimsexchange.com**—read it again, and you'll see what I mean.

Research with "whois"

The simplest way to research whether a name is available is via the old “whois” service, an ancient but still useful method of pulling raw registration information from the central registration database. I prefer Geektools.com's whois site because they're not associated with a registrar and they present a quick, stripped-down set of results (<http://geektools.com/whois.php>). If your preferred domain name isn't listed in the whois directory, you can proceed right to [Register a Domain Name](#).

NOTE The whois directory explicitly disclaims being entirely up to date at every moment in time. If you try to register the domain and it's not available, you may need to return to this step for more attempts.

NOTE Some folks maintain that searching on a domain name at a registrar without immediately registering that domain name can result in the registrar reserving the name temporarily, perhaps in the interest of selling the name to you at a higher price than plain registration.

While this allegation has circulated for years, I've never been able to substantiate it. With the current set of tools and interest in domain names, I find it hard to believe that this kind of activity could be hidden from those registering names in large quantities.

Security journalist Larry Seltzer has recently documented what he thinks is an interception of some kinds of research requests via third-party operated whois search sites that led to reserved domains that had been queried about (<http://www.eweek.com/article2/0,1895,1991365,00.asp>).

Research with registrar suggestion tools

You've come up empty on your first choices. Despair not! Most registrars and domain name resellers are eager to take your money and thus offer tools that provide as many alternatives as possible.

Registrars offer dramatically different suggestions for what you might use as an alternative. As a test, I entered **soccerclub.com** at 1and1 (<http://1and1.com/>) and GoDaddy (<http://godaddy.com/>) and received extensive, useful lists that had little in common (**Figure 4**, next page), so if you're stuck, trying many registrars' suggestions should give you a variety of ideas. Or try several names at several registrars. Remember: you're not bound one bit to register a name you find with the registrar that suggested it!

With one or more domain names ready to be registered, let's proceed, next, to register your domain.

FIGURE 4

Domain Suggestions

- soccerclub.biz available
- soccerclub.ws available
- soccerclub.cc available
- soccerclubonline.tv available
- soccerclubpage.name available
- soccerclubshop.tv available
- my-soccerclub.ws available
- soccerclubweb.org available
- soccerclubweb.ws available

	.info	.net	.ws	.tv	.com	.biz	.org	.us	.cc	.name
mysoccerclub			<input type="checkbox"/>	<input type="checkbox"/>					<input type="checkbox"/>	<input type="checkbox"/>
soccerclub			<input type="checkbox"/>			<input type="checkbox"/>			<input type="checkbox"/>	
soccerclub-web	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
soccerclubweb	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
soccerclub-home	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
soccerclubpage	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
soccerclubshop	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
soccerclub-shop	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
soccerclub-site	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
soccerclub-page	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
soccerclubonline	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
soccerclub-online	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
my-soccerclub	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Also Available Smart Search Country Code Domains

[Select All](#)

<input type="checkbox"/> SOCCERCIRCLES.COM \$8.95*/yr	<input type="checkbox"/> DOSOCCERCLUB.COM \$8.95*/yr
<input type="checkbox"/> SOCCERBALLCLUB.COM \$8.95*/yr	<input type="checkbox"/> JUSTSOCCERCLUB.COM \$8.95*/yr
<input type="checkbox"/> SOCCERBARCLUB.COM \$8.95*/yr	<input type="checkbox"/> PROSOCCERCLUB.COM \$8.95*/yr
<input type="checkbox"/> SOCCERCLUBBAR.COM \$8.95*/yr	<input type="checkbox"/> GETSOCCERCLUB.COM \$8.95*/yr
<input type="checkbox"/> INFOSOCCERCLUB.COM \$8.95*/yr	<input type="checkbox"/> EASYSOCCERCLUB.COM \$8.95*/yr

1and1 (top) provided a list of ways to register my domain of choice in other TLDs (see the sidebar [Consider Another TLD](#)) as well as a matrix of potential alternatives with reasonable prefixes and suffixes at many TLDs, including .com.

GoDaddy.com (bottom) has three tabs for suggestions; Smart Search shows good alternatives in the same TLD. A list of dozens of names that I omitted from this screen shot—the full list scrolled on and on—shows more baroque but appropriate .com options like `floridasoccerassociation.com`.

SIDEBAR CONSIDER ANOTHER TLD

Most of the sites that provide some elaboration on a domain name you enter might suggest that you register a similar or identical second-level or third-level domain in different generic or country-code TLDs. For instance, registering **mybizco.com** might cause a registrar to suggest **mybizco.info**, **mybizco.net**, and **myzbizco.co.uk** (if your business is in the United Kingdom).

The .com TLD used to be a default choice for two main reasons:

- It seemed like the only “real” domain for serious companies.
- Many browsers would sandwich any word entered in the Location field with **www.** and **.com** when the Enter key was pressed.

In the last couple of years, however, some browsers—notably Firefox—have shifted to *searching* for a keyword entered in the Location field when that keyword is not a subdomain, and most Web traffic comes via search engines, links at other sites, and bookmarks, not through someone typing in a name. (Firefox sends you to the first match at Google, which could be the Web site matching the domain name if that page is highest ranked for its domain name by itself.)

If you can’t find a .com domain name you like, then perhaps a .info or other generic TLD could work for you. See [Appendix C: Top-Level Domains](#), for more details.

Register a Domain Name

It’s time to take action and turn your desire into reality by registering one or more domain names. Let’s look first at what you need from a registrar that will handle registration, and then get into the details of how to register at one of several major Web sites.

Choose a good registrar

A good registrar meets the following criteria:

- Has a simple-to-navigate Web site that allows you to log in and make changes directly to any of your registrar-specific information
- Offers a phone number for emergency support, in case of a serious problem that prevents your domain from being reachable
- Provides fast turnaround on email-based customer support for routine matters

A registrar should optionally, if important to you:

- Register domains at many different TLDs. This is important only if you are interested in domains beyond .com, .net, and so forth. Many country code TLDs can be registered only at a single site, while generic TLDs can be registered at many registrars.
- Provide *private registration* for contact information. You might feel that it's important to keep your mailing address and other contact information private—for instance, you might want to avoid receiving phone calls and letters from companies that are trying to sell you services. This conflicts with the requirement by the global domain authority that registrars obtain and list legitimate contact information for all domains.

Private registration works around that limitation by having another party register the domain name you want. That party assigns all legal rights to you. The other party's name and address appears as the registrant, while the registrar keeps your details entirely private. The registrar (or sometimes a third party to whom they contract this service) also monitors or forwards email, faxes, and snail mail, sometimes eliminating obvious spam, scams, and crud. Cloaking your contact information could send a message that you're not to be trusted, however, so use this feature with care.

- Offer a full range of services, including DNS hosting, Web hosting, and email handling. If you want one-stop shopping, make sure your registrar has the options you need for each of these services.

TIP Registrars for many TLDs are allowed to work with resellers. The reseller collects your info and fees, and often has a direct conduit to the registrar that makes the process seamless. Nonetheless, I'd prefer to avoid the middleman and reduce the potential for errors. And most resellers of any scale have become registrars over the last 3 years.

Registrars to consider

With hundreds of registrars, it's hard to offer a comprehensive survey of them all. But **Table 5** summarizes the basic details for two I've worked with (GoDaddy and easyDNS), and two that have registered

enormous numbers of domains and are worth a look (1and1 and Yahoo!). These suggestions aren't meant to exclude or denigrate any other company.

A primary consideration should be whether you are using a registrar for DNS, Web, or email hosting. Prices, services, and ease of use vary enormously among even these four registrars for comparable services. (Read [Set Up Your DNS Host](#) and [Host Your Services](#) before deciding on an all-in-one registrar and hosting service.)

Table 5: Comparing Four Suggested Registrars

Registrar	Yearly Rate	DNS Hosting	Private Registration	Longest Term
1and1	\$5.99	Included	Included	1 year
easyDNS	\$25.00*	+\$10	Not available	10 years
GoDaddy	\$9.20	Included	\$4.99/\$8.99†	10 years
Yahoo!	\$9.95	Included	\$9.00	5 years

* The [coupon](#) at the end of this book gives you a \$10 discount from this price for new customers.

† First year/subsequent years; free with 3+ domain registrations.

Purchase the domain name

You've figured out the name and decided on a registrar, so it's time to register the name! Here are the steps to follow:

1. Set up an account at your preferred registrar.

TIP Be wise about what you enter for your contact information when setting up a domain name. You most likely want to provide multiple email addresses, and appoint appropriate people to have authority as owner, billing, technical, and administrative contacts. For the full scoop on what to choose based on what can go wrong, see [Prevent an unexpected expiration](#).

Creating a special email address reserved for domain purposes, such as `domain@example.com`, would allow email to continue to be received even if a given individual leaves the company and her personal email account is disabled.

WARNING! Do not allow a registrar or DNS host to list itself as the owner/registrant of a domain. A host may serve as the technical contact, but you must be the owner. If you're not, it can be difficult to later prove ownership and move a domain off a given registrar or DNS host.

2. Enter the domain name or names to register.
3. Put them in your shopping cart or otherwise add them to your account. You may have to choose a duration of registration at this point. Some registrars offer year-at-a-time registration only; others offer prepaid terms of up to 10 years.
4. Choose other services. You will be offered 14,000 separate, add-on packages—GoDaddy is a harder sell than, say, Yahoo—but you can always figure out the other options you want and add them later.
5. Purchase the domain name and wait for confirmation.

If the DNS host for your domain is a separate firm, follow the instructions in [Set Up Your DNS Host](#) to get the correct nameserver settings, and then return to the registrar to enter those details.

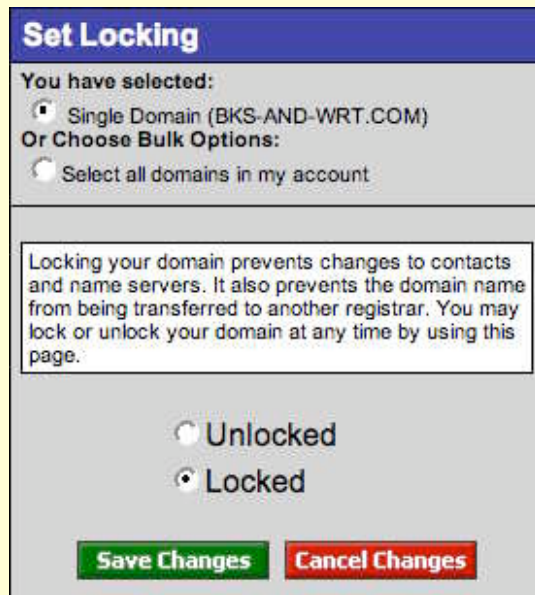
NOTE LOCK DOWN YOUR DOMAIN

To avoid *domain hijacking*—having your domain transferred against your will by a ne'er-do-well—the global domain authority ICANN requires registrars to provide Registrar Lock. When a domain is set to Registrar Lock status, no changes to contact information or nameservers are allowed and the registrar cannot allow the domain to be transferred to another registrar.

As the domain owner, you can temporarily disable Registrar Lock from within your registrar account in order to change those details (**Figure 5**). Most registrars enable locked status by default.

This lock prevents social engineering by telephone or fax in which someone claiming to own your domain uses fraudulent information to transfer the domain. Registrar Lock also forestalls typical means of domain theft, such as spoofed email or even a hack of another registrar.

FIGURE 5



GoDaddy's control for registrar lock, which they place under the Set Locking menu.

Get a Name That's Already Taken

Let's say you find that the perfect domain name is already registered. Is it possible to obtain that name in some legitimate fashion? Sure—for a price.

TIP If you have a legitimate interest in a domain name that's being used in bad faith by another party, you may be able to demand the name through a mandated ICANN arbitration and mediation process. See [Avoid Well-Known Names](#).

Not everyone wants to sell a domain name, but many owners are willing to go through some adjustment in losing a name if there's enough compensation involved. Earlier in the book, I pointed out that you *lease* a domain name, you don't *own* it. But the terms of most registrations include the ability to transfer it to another party.

WARNING! Not all TLDs allow domain transfers. The same kinds of requirements that govern initial registration in, say, .travel, would govern its sale. More restrictions may be in place for specialized domains, too.

Domains that are parked, lack a Web presence, or have little content on their Web site are more likely to be receptive to an offer. Those that have lots of content and are in active use might require a huge fee, if they'd accept it. A growing trend is to value Web sites based on

their traffic and advertising revenue; some domains that have sold recently for hundred of thousands to millions of dollars have high traffic based on their name and Google ranking, but lack real depth of content.

NOTE PARKING, NOT DRIVING

When you've parked a domain, you've registered that name but haven't yet put any real content on it—at least on a public Web site. The Internet seems full of parked domains at times. You mistype practically any domain name and wind up at a generic parking page full of ads or a “this domain for sale” banner.

Parking has become routine because a great deal of money is generated from certain domains that have no real content on them. Some sites, like Sedo (<http://www.sedo.com/>), can place content on your parked domains and split the revenue with you.

Buy a name

You have as many as four approaches to buying a domain name:

- Contact the registered owner, whose name and email might be on the Web site, or can be found via whois (see [Research with “whois”](#)).
- Use someone you know as a third-party agent. (I've acted in this role for a few friends selling their domains in exchange for a finder's fee of a few percentage points or a nice dinner.)
- Find a domain broker, like Sedo (<http://www.sedo.com/>). These firms specialize in estimating the value of a domain and then negotiating on your behalf. They can often provide escrow service for the funds and transfer the domain name to you. Fees can be 10 percent or higher for successfully obtained domains.
- Bid on the site directly. Some sites are already for sale, and you can enter a bid on the main page of the site or via a link on that site.

TIP Interested in selling a domain? See [Appendix B: Sell a Domain Name](#).

Make a bid

In any negotiation, it's essential not to tip your hand too soon. I have sold or brokered domains for amounts from \$250 to \$250,000, and in nearly every case, the buyers have provided far too much information early on about their resources and the price they want, making it much simpler for me to set the maximum reasonable price.

Simultaneously, you have to avoid lowballing a domain owner, who might then not give you the time of day. Because it costs so little for an owner to keep a domain parked, offering below \$200 could result in a lack of interest by anyone not desperate to unload the domain.

Make payment

Before you send payment, you should do two things:

- Be sure that the party selling the domain name has the rights to sell it. This can be tricky in and of itself. The best method to ensure this is that you can send email to the administrative contact listed for the domain in the registration records; visit any registrar to look up this information. Failing email, you may want to send a registered letter with return receipt requested to the mailing address listed for that domain.
- Obtain a written agreement—signed by the party who has the domain registered in his or her name—and obtain confirmation that the person signing is the person in question, such as a photocopy of a driver's license or passport. Some people may be chary of sending a stranger this information, and you might instead ask for a notarized statement that the person's provided name and mailing address are accurate.

If a third party, such as a domain broker, handles the sale, you're in much better shape than if you're buying direct. Just as with any eBay auction or craigslist purchase, it's risky to transfer funds before you're sure you can obtain the goods.

When buying a domain directly, consider suggesting Escrow.com (<http://www.escrow.com/>), the escrow service recommended by eBay for auctions; the company also handles domain transfers. You could offer to pay or split the fee, which varies based on the final sale price.

Reserve a name

Someone else's loss could be your gain. With millions of domains registered, some are inevitably abandoned and not renewed, either through oversight or by deliberate choice. In either case, you can snap up domain names as soon as they re-enter the pool. It's a bit like a combination of eBay and a police auction.

Many registrars now provide a reservation option that allows you to put down a deposit or simply state your intent to purchase a domain as soon as it expires. And, some services, like SnapNames.com (<http://www.snapnames.com/>), track expiration dates of domains and auction those in which someone has expressed interest. This firm charges \$60 per name that you successfully acquire, unless there's interest from multiple parties, in which case \$60 is the minimum bid.

Of course, with competition for the same domain names across many registrars and other parties, a popular name might have bids at more than one location, but only a single registrar can snag the domain name at a given time—so even if you have the winning bid at one firm, that firm might not be able to obtain the domain name.

My advice is to not spend too much time worrying about domains you can't buy easily, but rather to find a domain that suits your need.

SET UP YOUR DNS HOST

A DNS host occupies a critical role in the connection between those trying to reach you and the Internet's technical underpinning: it makes it possible for people who, for instance, use your domain name in a Web URL or email address, to communicate with the computers and services associated with that domain name. Don't choose your DNS hosting company arbitrarily. A firm that lacks the technical chops to deal with the modern exigencies—such as denial-of-service attacks designed to cause the host (and you) economic damage—isn't where your domain should be located.

NOTE Can I talk you out of hosting your own DNS? Good. It's quite involved and it's a 24-by-7 commitment. Not 24 hours a month, seven months a year, but every minute of every day you want your services available to the outside world. The Internet never sleeps, just like rust.

The same goes for running your own email or Web server; when the server stops functioning, you have to drop everything to make it work again. As someone who has spent a number of sleepless nights swearing at Linux boxes in frigid co-location centers, I advise against self-hosting unless you're truly committed (or unless you wish to be, somewhere with padded walls).

Select a DNS Host

A good DNS host handles these tasks:

- Offers an easy-to-use Web site to enter the IP addresses and mail server information that you get from your service hosts.
- Allows essentially unlimited entry of address records and mail server (more accurately, *mail exchange*) records. *Address records* map subdomains to IP addresses; *mail exchange records* provide a list of one or more mail servers for each subdomain or the entire domain.
- Has servers distributed around the globe for redundant performance, backup, and defense against attacks.

WARNING! Lest you think this kind of distribution is so much window dressing, DNS hosting companies are regularly attacked both by vandals and by criminals intent on extorting fees from hosted domains. These efforts are often *distributed denial-of-service* (DDoS) attacks, which can sometimes be blocked through distributed redundancy. One network or server might yield, but as long as other servers are still available, DNS lookups continue to resolve without missing a beat.

A DNS host can also provide these services as part of a basic package:

- Dynamic DNS as an option. (See [Use Dynamic DNS.](#))
- Email receipt and forwarding for an entire domain or a few specific addresses when you don't have an email server set up for the domain. This enables you to use a DNS host as a rudimentary mail host without setting up separate mailboxes or mail hosting for the domain. Messages can be forwarded to another email account or set of accounts, such as one at Google's Gmail or a mailbox included with your ISP's basic service.
- Email queuing when your mail server fails to accept mail in a timely fashion. (See [Backup mail service.](#))
- A backup copy of previous settings you've entered for your DNS stored so that you can retrieve older settings yourself without needing to ask for human technical support. (This may be unique to easyDNS, but, boy, I like that feature.)
- Access to all DNS resource records (see just ahead).

There are two additional advanced aspects of DNS hosting that you should be aware of when researching which company to use:

- **TTL granularity:** This is the level at which you can set the caching period, or time to live, for other DNS servers to cache information from your domain. This option can be critical if you change DNS information frequently or need to migrate Web, email, or other hosting services. GoDaddy allows you to set TTL separately for each record type—see the next bullet—but some hosts make you set TTL for an entire domain or not at all (see [Time to Live](#)).

- **DNS control:** DNS has several kinds of common records for different sorts of resources. These include Address (A) for IP address; Mail Exchange (MX) for mail servers; Canonical Name (CNAME) for aliases; and Text (TXT) for miscellaneous purposes, most quite obscure. (See <http://www.secondary.com/support/dnsprimer.html> for a primer on DNS that links to more advanced information, too.)

Suggested DNS Hosts

Tens of thousands of firms provide DNS hosting services—these include most registrars and almost all Internet service providers and Web hosting companies. To help you narrow your options, in **Table 6** I look at four hosts that I am comfortable recommending, plus two others suggest to me by colleagues.

Table 6: Comparing Suggested DNS Hosts			
DNS Host	Cost per Year	TTL Granularity	Editable DNS Record Types (Menu Name)
1and1	\$5.99*	No control	A, MX (via DNS Settings)
easyDNS	\$19.99	Each domain	A, MX, CNAME, TXT (via “dns” link)
GoDaddy	\$9.20*	Each record type	A, MX, CNAME, TXT (via Total DNS Control Manager)
Yahoo!	\$9.95*	No control	A, MX, CNAME (via Advanced DNS Settings)
EveryDNS.net	Free**	Per domain, with donation	A, MX, CNAME (via Manage DNS)
ZoneEdit	Usage††	Each domain	A, MX, CNAME (via Edit Zone)
* Includes domain registration fees. ** Yearly donation requested; enables some extra features. †† Free for up to five domains with some limits.			

Configure DNS

If your registrar handles domain registration plus DNS, Web, and email hosting, you don't need to configure DNS at all after registering your domain and setting up an account. Their internal tools will provide all the configuration you need.

However, if you use a separate DNS host that doesn't handle all these tasks, you first should sign up for any Web hosting and email hosting that you want a third-party service to provide, and then hook everything together. I've split the instructions for connecting services into three pieces; you should complete any of the instructions that apply to you:

- If your registrar and DNS host are not the same firm, follow the directions in *Point your registrar to your DNS host* (below).
- If your DNS host isn't the same firm as your Web host, follow the steps in [Point your DNS host to your Web host](#).
- If your DNS host isn't the same firm as your email host, follow the instructions in [Point your DNS host to your email servers](#).

Point your registrar to your DNS host

Remember, you need follow these steps only if your registrar is a different company from your DNS host.

First, obtain from your DNS host the nameserver details that you will use to point your registration information to the host's DNS server. This information comes in the form of subdomains paired with their corresponding IP addresses. For instance, GoDaddy's default nameservers that appeared in my test account are `ns3.secureserver.net.` and `ns4.secureserver.net.`, which correspond to IP addresses of `64.202.165.10` and `68.178.211.105`, respectively. (Some DNS hosts may require the trailing period; see [Trailing Period](#) for why.)

TIP Don't assume that a DNS host has just one set of nameservers. In fact, you and a colleague might register two domains at the same time and receive different nameserver details. Check your account at the DNS host rather than assuming that one nameserver set fits all.

Second, with the nameserver details in hand, visit the registrar's Web site and configure the information corresponding to your nameserver (**Figure 6**). Some registrars will accept just the subdomain and look up the IP address; others make you enter both.

FIGURE 6



The screenshot shows a web form for configuring DNS. It has three radio button options: 'Default Hosting Name Servers' (selected), 'Default Parked Name Servers', and 'Custom Name Servers'. Below these are three input fields for name servers. The first two are filled with 'ns3.secureserver.net' and 'ns4.secureserver.net' respectively, and each has a red asterisk to its right. The third field is empty.

At GoDaddy, you can choose among their hosting name servers (if they're your Web host), parked nameservers (which redirect Web queries to a page explaining there's no real site there yet), and custom nameservers (for nameservers you specify).

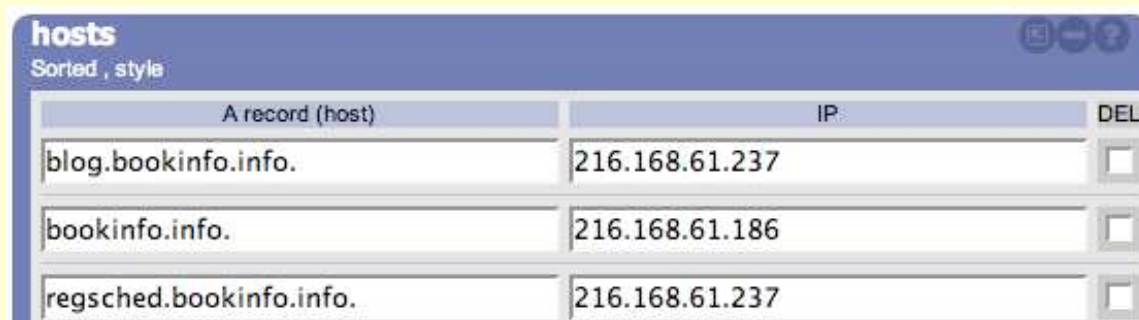
When you commit the changes at the registrar, note how long the registrar expects it to take for the changes to be available to the rest of the Internet. Nameserver changes are passed on to centralized global resources, and thus can take longer than other DNS changes.

Point your DNS host to your Web host

If your DNS host isn't also your Web host, follow these steps to make sure your DNS host knows how to find your Web host:

1. Obtain from your Web host the IP address (or addresses) they will use to serve your Web site.
2. Decide on the subdomains under which your Web site will be reachable. (See [Redirect Your Domain's Web Sites](#) for more on variations on this theme.)
3. Configure your DNS host to create those subdomains and connect the IP address or addresses to them (**Figure 7**).

FIGURE 7



The screenshot shows a web interface titled 'hosts' with a table of DNS records. The table has three columns: 'A record (host)', 'IP', and 'DEL'. There are three rows of data.

A record (host)	IP	DEL
blog.bookinfo.info.	216.168.61.237	<input type="checkbox"/>
bookinfo.info.	216.168.61.186	<input type="checkbox"/>
regsched.bookinfo.info.	216.168.61.237	<input type="checkbox"/>

Configuring subdomains at easyDNS.

NOTE It's rare that a Web host would choose to have several IP addresses for a single Web site. The point of this would be to share incoming traffic among multiple machines with the same content. In practice, DNS doesn't handle this task well, and hosting companies use a different approach. See [Pointing Subdomains at Services](#) for a little more detail.

4. Commit your changes at the DNS host.

You should note the time that the DNS host says it will take for their DNS servers to serve out this new information. Long ago, the *only* registrar restarted its nameservers just once or twice a day. Modern DNS hosts sometimes update new information within minutes.

Point your DNS host to your email servers

When you send email, you're actually uploading an email message to your ISP's or your own mail server, which must then figure out how to deliver it. A mail server uses DNS to retrieve a special kind of record, called the *mail exchange* or *MX record*. An MX record lists one or more mail servers by subdomain that are configured to accept email for an entire domain or some set of subdomains. You can define multiple mail servers so that if one is busy or crashed, another can accept incoming messages.

DNS also lets you choose the precedence, or order, in which a sending mail server attempts to deliver messages to your receiving mail servers. When a sending mail server tries to deliver a message to a given domain, the sending server proceeds through the list in order of precedence until it finds a receiving server that's ready to accept the message.

TIP The precedence is set numerically from lowest to highest, but there's no fixed importance to any particular number. Some system administrators and DNS hosts set the numbers to 10, 20, 30, 40, and so on. Others use 1, 2, 3, 4, etc. Some DNS hosts let you set these numbers.

If the sending mail server fails to connect with all the domain's receiving mail servers, it usually queues the message and tries again later. Some mail systems send the original sender a warning via email that the message wasn't delivered and will be tried again in a certain interval, often 4 hours.

TIP TEST YOUR MAIL SERVER'S CONFIGURATION

A mail server must be configured to accept email for users at a given domain, or that mail server will reject any incoming messages directed to users at that domain. For most mail servers, a system administrator need simply add the name of domains or subdomains for which that mail server should accept incoming mail and then force the server to reload its configuration files.

You'd be surprised how often this is done incorrectly, partly because of how domain names are decoupled from the services they point to. Because mail servers are configured by your host, you may have no control over whether this is done correctly.

I strongly recommend after you change an MX record in any way that you use an email account somewhere else—such as at Gmail or Yahoo—to send yourself an email message at that domain.

If you see bounces that say something like “server not configured for this domain,” “points to itself,” or “loops back to me,” first check the values you entered for the MX records at your DNS host. If they are correct, contact your mail host for help.

If the test message gets through, then your mail records and mail host may be configured correctly, though this quick method tests only the first mail server to respond to mail sent to your subdomain. If you want to test that each of the mail servers listed will accept email for your subdomain, see [Test your new email account](#), later, which shows you how to connect to each mail server in turn.

Enter mail exchange records

Here's what the mail exchange records look like in DNS configuration format for a typical email host:

```
glennf.com. IN MX 10 mail1.supermailhost.com.  
glennf.com. IN MX 20 mail2.supermailhost.com.
```

You see how both **glennf.com** and the mail server subdomain are set up as full subdomains. (The trailing period identifies that it's an entire subdomain name; see [Trailing Period](#) for more details.) The **IN MX** identifies each line as an Internet mail exchange record. The settings of **10** and **20** are arbitrary, as I noted earlier, and mean that host **mail1** should be the preferred email server to which messages are delivered, and **mail2** tried next if **mail1** is busy.

To configure email at a DNS host, you need the set of mail exchange records that your email host provides. Email hosts normally provide these details in a welcome email when you sign up or in an online support site.

Each DNS host enables you to configure mail servers somewhat differently, although the results are always the same (**Figure 8**).

FIGURE 8

Mailserv Hostname:	Priority:
Yahoo! Mail Server	20
Yahoo! Mail Server	30
	10
	10

Yahoo's form for entering mail server records is typical; their default mail servers are *not* in proper format, however! Yahoo makes an exception for its own mail servers, but other mail servers must be entered as subdomains.

Backup mail service

Many DNS hosts and Web hosts can provide backup mail queuing for a domain, which is helpful if you're running your own mail server or using a smaller firm to handle email and want a little extra insurance. If your main mail server or servers freeze or their network is unreachable from a sending email server's network for a period of time, the backup mail service accepts and then queues incoming mail for your domain. This queued mail is delivered whenever any of the mail servers that are intended to receive email directly for your domain name become available again. If it appears that your main server will be unavailable for an extended period of time, you can use the DNS host to set new mail server records, and then the new mail server will receive the queued mail from your DNS or Web host.

WARNING! Backup mail services have one big flaw. Because they don't know which email addresses are legitimate for the domain in question, they also accept and queue massive amounts of dictionary spam and directory spam, in which spammers use thousands or more email addresses for every domain they're trying to scam. Your primary mail server may filter mail sent through these backup servers with less scrutiny, thus leading to more spam.

Map Reverse DNS

As I've described so far in this book, subdomains in DNS configuration map to IP addresses, which is how Internet-aware software on a computer turns a subdomain into a reachable location. But what about the reverse? What turns an IP address into a subdomain? Reverse DNS lookups.

IP addresses are assigned in numeric ranges to network operators, which in turn hand off blocks to companies and smaller ISPs. Associated with those ranges is a set of domain records that provide the glue to turn an IP address back into a subdomain.

Reverse mapping used to be crucial. And sometimes it still is when you're trying to run your own services from your own network with ISP-provided IP addresses. For instance, many large ISPs and hosting companies block email from ISP-based IP addresses unless there's a reverse-mapped entry. Some ISPs refuse to offer this reverse mapping, and that may prevent you from deploying those services. Some mail servers also block any mail server that doesn't have a forward lookup (subdomain to IP) and reverse lookup (IP to subdomain) that match.

Generally speaking, though, with so many people using hosting services that map millions of domains to a handful of IP addresses, reverse mapping is of minimal importance for almost all of us. And apart from mail, it has no importance these days.

HOST YOUR SERVICES

Upon reading this section's heading, you may yell, "At last!" Yes, you've reached the point in the book at which I tell you about the real reason for all this infrastructure: putting servers at domain names.

Among the most popular reasons to have your own domain name are publishing a Web site and receiving email. These two services have particular relationships with DNS and domain names that need some explanation.

Host Your Web Site

Earlier in the book, I explained that the Web browser/Web server communication language, HTTP (Hypertext Transfer Protocol), lets one computer running one piece of Web server software handle the responses back to many Web browsers.

Some people prefer to have a unique domain name for each Web site they operate. Others, to save domain registration fees or avoid conflicts with other domain names, use subdomains. I use a mix of both.

For instance, I run six wireless data industry news blogs. The flagship, Wi-Fi Networking News, has been at **wifinetnews.com** for several years. When I added other blogs, I positioned four of them as subdomains of **wifinetnews.com**, and registered a new domain name for a site that covers WiMax—what I think will become a separate and equally important technology—as **wimaxnetnews.com**. I handle all six Web sites through one Web server, with some of the image files and style sheets located in shared directories.

Require certain options of any host

Finding a host requires matching what you need with what they offer. I suggest looking for a baseline of features that every Web site will likely need. A good Web host offers:

- A robust amount of storage for files—at a minimum, 50 megabytes (MB). If you're hosting audio or video files, 1 gigabyte (GB) is a bare minimum. Most popular Web hosts now offers 5 GB or more.
- A reasonable amount of included monthly data transfer, which is a measure of incoming requests, downloads and Web visits by users; in some cases, this also includes uploads by you or your users, if

you allow them to upload images or other files to your Web site or other repositories. Don't accept under 1 GB per month for basic Web sites; 50 GB is a bare minimum for a site with audio and video files.

TIP If you host a lot of media files, you should shop around for a provider that offers generous storage and data transfer limits. For instance, DreamHost's hosting service includes 200 GB of storage and 2 terabytes (TB)—that's 2,000 GB—of monthly data transfer for \$10 per month. You can compare plans from many hosting companies at Hosting Review (<http://www.hosting-review.com/>), which uses reviews to rank hosts.

Mac users in particular might consider Apple's .Mac service, once considered pricey and underfeatured (<http://mac.com/>). Now, Apple provides 4 GB of storage and 250 GB of monthly transfer for \$200 per year with their highest account upgrade, or about \$17 per month.

My editor uses an interesting network called the Coral Content Distribution Network (<http://www.coralcdn.org/>), which allows free use of their mirroring service, and involves practically no effort to use. This enables anyone to serve large media files without running afoul of an ISP's data transfer limits. (The catch? No catch; they're a research project partly funded by the National Science Foundation.)

- FTP access for uploading files to your Web site.
- Basic statistics on visitors.
- Regularly updated data transfer figures to track whether you're within your inbound/outbound allotments.
- Web traffic analysis, regularly updated, to give you a sense of number of visitors, popular pages, and referring URLs.

A Web host could also offer:

- WebDAV support. WebDAV is an alternative to FTP that some Web publishing programs use for collaborative site management.
- Secure file transfer when uploading or downloading files from your site—via secured WebDAV (often called WebDAV SSL or WebDAV HTTPS) or secured FTP (see the [warning](#) a few pages ahead).
- Blog hosting with common blog server software.

- Detailed Web traffic statistics.
- Access to database services, such as Microsoft SQL Server or MySQL.
- Support for scripting languages for Web use, such as perl, PHP, ASP, python, and ColdFusion. Most hosts specialize in just a subset of these.
- Secured Web sites with shopping carts and ecommerce options.

Suggested Web hosts

As with DNS hosts, there are over one bazillion firms that “specialize” in Web hosting. Almost always, if you have a broadband Internet connection, you’re provided with some kind of Web hosting by your ISP. I’ve listed a handful that are run by registrars or that colleagues have worked with.

But ISPs tend to offer scanty hosting, even when sold as a standalone package. For instance, compare EarthLink’s included hosting plan and standalone plan against four hosting-only firms in **Table 7**.

Table 7: Comparing Suggested Web Hosts (Basic Plans)				
DNS Host	Monthly Hosting Fee	Storage	Monthly Transfer Limit	Number of Email Accounts
1and1	\$2.99*	5 GB	250 GB	500
DreamHost	\$9.95*†	200 GB	2000 GB	3000
EarthLink (high-speed)	Included with broadband	80 MB	Not specified	8
EarthLink (standalone)	\$9.98 1st 6 mo./ \$19.95 thereafter	2 GB	20 GB	30
GoDaddy	\$3.99 (+\$1.99 for domain reg.)	5 GB	250 GB	500**
iPowerWeb	\$9.95†	10 GB	250 GB	2500
Yahoo!	\$11.95*†	5 GB	200 GB	200
* Includes one domain name registration. † Setup fee of \$10 to \$50. Discounts for nonprofits. ** Each account limited to 250 outbound emails per day.				

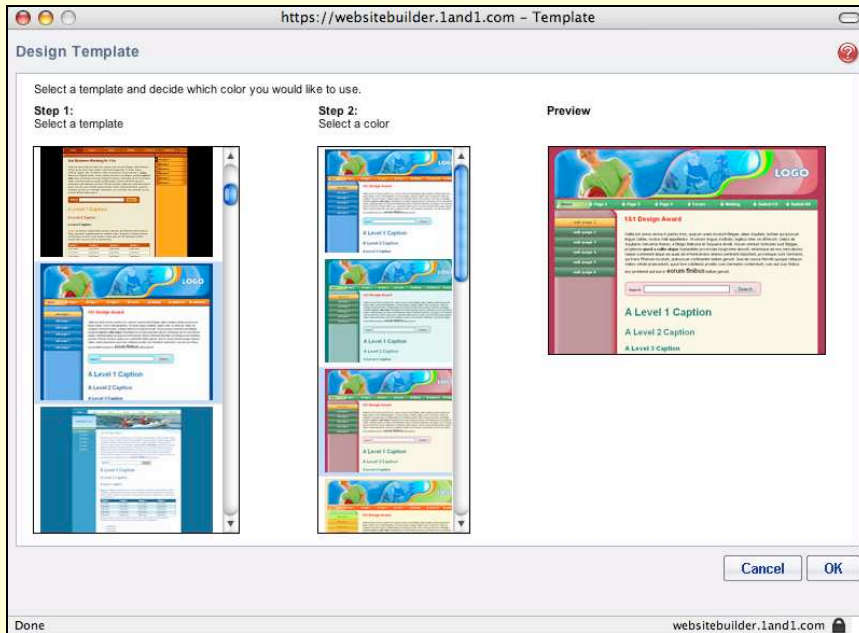
Set up a Web site at a hosting service

Here's how to configure and start your Web site at a Web host:

1. Set up a hosting account.
2. Create a subdomain for your Web site at your DNS host, whether your DNS host is the same company as your Web host or not. (See [Point your DNS host to your Web host.](#))
3. Activate a Web server with that domain name at your Web host. Directions vary completely from server to server; follow their tutorials on creating the Web site storage area.
4. Create the Web site.
 - You may be able to use templates or wizards provided by the host if you have no design in mind or design tools. These let you use a preset design and fill in the blanks (**Figure 9**, next page).
 - If you're using a Web design program like GoLive or Dreamweaver, you can use its built-in file transfer client that can synchronize files in your local copy with the Web server.
 - If you're using an FTP program to transfer files, you need to find out from the Web host the name of the FTP server and whether it uses your same user name and password. When you connect via FTP to most Web hosts, you'll often see a set of folders, one of which may be labeled `web_site` or `public_html`. Check the documentation at your Web host, because this is the folder that acts as the root of your Web site rather than the folder you view when first logging in via FTP.
5. Test the site by visiting your URL.

TIP If your Web host provides the flexibility, set up multiple sites, including a staging site that lets you test changes to a site before they're publicly available. You could conceivably password-protect that site using tools provided by your Web host.

FIGURE 9



1and1.com's template selection interface.

WARNING! FTP is an *insecure protocol*, which means that it's not designed to protect the data that it passes between client and server. This is an issue if you're using FTP in a public place or on a shared network, such as on an Internet café's network, at a public Wi-Fi hotspot, or on a college campus.

Instead of using plain FTP, use Secure (SFTP). It's a common encrypted alternative to plain FTP handled by most FTP client software and many Web hosts. Ask a host if they support SFTP. To use SFTP with DreamHost, for instance, you have to check a box to turn on SSH (Secure Shell), the encryption part of SFTP, in the hosting company's control panel under account management.

Prevent Web host failures

Because of the way HTTP works, you can't reliably specify a backup Web server if your main server is down. While DNS lets you set multiple IP addresses for a given subdomain, few DNS programs and no popular Web browsers have a mechanism for failover. If the Web server at one IP address is dead, popular browsers don't try another server at another IP address, but rather report a page retrieval failure to a visitor.

Here are a few scenarios for dealing with disaster.

If you're hosting your own Web site or using a smaller firm

It's not uncommon for a Web host to have a lacuna in operating Web sites. Even the best have occasional downtime, and average Web hosts and ISPs may have more frequent errors. Thus, keep your DNS hosting separate from Web hosting. If your DNS server is on the same network or machine as your Web site, then a failure in the machine or network could take you off the air without much recourse. This does create two separate points of failure, but it also allows you to fix one problem independent of the other.

I recommend keeping a full backup of your Web site that's constantly refreshed. Even better, copy the files to another Web server on a regular basis, so that you could take that other server live in a pinch. You should also keep a record of your DNS settings, as these may be unreachable in case of a DNS host failure.

With a separate DNS host and a backup Web server, you could enter a new IP address for your Web site if your primary Web host fails. When a Web host suddenly dies due to hurricane, hardware failure, or terrorist attack—it can happen—you could shift your Web site in a matter of minutes to hours, depending on how prepared you are.

If you or a smaller Web host handles your DNS, too

In this scenario, your registrar must be a separate firm. In the event of a total collapse of the Web and DNS host, you could go to your registrar and change the nameserver information to point to a new DNS host, which could then be set to point to a new Web host. Consider the possibility of this happening and make plans about what you'd do ahead of time.

Receive Email

While virtually all ISPs and Web hosting firms also offer email, you might consider separating your email service from the rest of what the firm hosts for you. For starters, it introduces a separate point of failure—normally a bad thing, but when your Web host fails and your email still works, you might consider that a positive commodity.

Second, a Web host that provides everything you might want for a Web site might have poor email capabilities. With prices incredibly cheap for combined Web/email hosting, it's not a huge expense to use two companies, one specializing in Web hosting, the other specializing in email.

Questions to ask about an email host

No two people or companies need the same features from email. I suggest asking these questions of yourself and of the potential email hosting firm. Finding these answers at an email host's Web site or via their technical support can be difficult, but many hosts offer trial subscriptions or a trial period during which you can receive a refund. Test the services thoroughly using forwarding from existing accounts or via test email you send from other accounts before you decide.

- **How many unique email accounts can you have, each with its own login and mailboxes?** Many email hosts offer from 1 to 3,000 email accounts with basic service plans—I have no idea who needs that many email addresses outside of major organizations. If you need many separate email accounts for your own purposes—such as orders, comments, and feedback—or for employees or family members, make sure the host you're using doesn't skimp.
- **How many aliases?** Aliases allow multiple addresses to flow into the same mailbox. The aliases address is included in the mail headers, which enables your email client to filter mail based on the alias. Aliases can alleviate the need for many mailboxes, too.
- **How many outgoing messages per day?** Many hosts put a limit on the number of outgoing messages to discourage spammers. 1and1 limits new accounts to 100 outbound messages an hour, for instance; I had to ask customer support for that number, which is not found on their site. GoDaddy limits all email accounts to 250 messages per day each, unless extra fees are paid.
- **What are the limits on storage?** Mail hosts often put a limit on how much storage you can use to retain email, most important with heavy incoming mail accounts, when you use IMAP and leave mail on the server, or with long absences between mail checks.
- **What are the limits on data transfer?** As with Web hosting, some email hosts measure the amount of bandwidth used for incoming and outgoing email along with your bandwidth used to retrieve email, and restrict that. If you send many large attachments or handle massive amounts of email, this may trip you up. Web hosts that offer email usually include email transfer as part of the overall account transfer limit.

- **How big may incoming and outgoing attachments be?** Attachment size is often a big sticking point because it limits what you can receive and what you can send. Some services bounce incoming and outgoing messages containing attachments larger than 2 to 10 MB (either individually or as a total per message). It can be quite irritating to find these limits at Web hosts that also offer email. I was unable to find an answer at 1and1.com without logging in to my account (10 MB of attachments per message), and had to search on a support wiki at DreamHost to get their answer (40 MB per attachment).

TIP One easy way around file attachment size limits is to bypass email for sending enclosures altogether. Several services now allow you to use either a Web site or simple client software (Mac and Windows) to perform transfers. Check out free software packages Civil Netizen (unlimited size, <http://www.civilnetizen.com/>) and Pando (1 GB while in beta, <http://www.pando.com/>), and free Web sites YouSendIt (100 MB, <http://www.yousendit.com/>) and DropLoad (100 MB, <http://dropload.com/>).

- **How easy is it to set up forwarding addresses?** For any incoming address or alias, is it a simple matter to forward mail for that address to another account? Can you store a copy at the email host and also forward a copy?
- **Can you configure the spam filters?** Some email hosts have strong or weak anti-spam policies that may conflict with your ability to receive email or to block messages you don't want. The best email hosts allow you to set the severity and some of the policies around filtering. An important option is a whitelist, which lets you pre-approve certain addresses or domains.
- **Finally, can you accept email that is addressed to a user at a subdomain?** I may be able to receive email to `glenn@glennf.com`, but what if I want to receive email for `admin@web.glennf.com` or `admin@freestuff.glennf.com`?

Access types and security with an email host

In addition to account limits and spam options, you should figure out which email protocol you need to use and how you can safely access email:

- **POP, IMAP, and SMTP access:** That's Post Office Protocol, the most commonly supported way to retrieve inbound mail; Internet Message Access Protocol, which is designed to allow inbound email to remain on a server; and Simple Mail Transfer Protocol, a way to send outgoing mail. While you may exclusively use only POP or only IMAP, flexibility is best, as you may want to switch from one to another. SMTP is universally available.

TIP IMAP has the distinct advantage to people who use many computers of offering them access to the same pool of current email and email archives via whatever device or webmail interface they use at any given time. IMAP can reduce the need to retrieve email to one computer or synchronize archives among many computers. (This also means that the IMAP server should be centrally backed up—or you might lose your stored mail.)

If the mail server will be used by people while they travel, these three items are key:

- **Secure access:** Secure access can be achieved via a SSL/TLS (Secure Socket Layers/Transport Layer Security) connection to POP, IMAP, and SMTP servers. This enables users to retrieve email anywhere without risk of interception.
- **Authenticated SMTP:** This form of SMTP requires a login, and became widespread as ISPs were forced to lock down their outbound SMTP servers against spammers who would misuse them.
- **Secure webmail access:** Secure webmail should have a login from a secure entry page. Travelers may need webmail on the road if something goes wrong with the computer they travel with, or to avoid bringing a computer. The secure page is as important as the secure connection. If someone enters a user name and password on an unsecured page, certain Wi-Fi vulnerabilities could allow them to be intercepted. For more on this security weakness and how to solve it, see *Take Control of Your Wi-Fi Security* (<http://www.takecontrolbooks.com/wifi-security.html>).

Suggested email hosts

All the Web hosts listed earlier in [Suggested Web hosts](#) also offer an extensive set of email features. I used to have a dedicated email provider that I recommended, but my account was recently down for four full days, which turned them into a *former* provider.

NOTE FREE EMAIL HOSTING FOR DOMAINS? NOT QUITE

Can you just go free and use a service like Google's Gmail? Yes, but you can't yet set up a free service to accept email for an entire subdomain; some testing is in place at Gmail, AOL, and other services to act as full subdomain mailers, but it's unclear whether the final product will be free.

A DNS host with the right features could make a free address work, however. Some DNS hosts will let you forward all email sent to a domain name, ignoring the specific delivery address (*catchall* forwarding); some will also or instead let you forward specific addresses to other email addresses. With either, you could forward email to a single address (a *catchall* address) or a set of addresses.

WARNING! Catchall email addresses sound like a good idea. Any email sent to any address at a given subdomain is delivered or forwarded to a single address—email to many addresses winds up in one mailbox. Unfortunately, spammers will sometimes send messages to hundreds of thousands of spurious account names at any given domain to see if anything gets through.

With a catchall address, because every possible user name at your domain name can potentially receive email, every spam email would have a legitimate recipient. For instance, if I had catchall addressing turned on for `glennf.com`, `asdfadfd@glennf.com` and `adf793r_dfds@glennf.com` would be as legitimate as `glenn@glennf.com`. This means that your mail host receives all those emails, and then has to filter them for you; or they're just passed through directly to you if you don't filter email.

Set up email at an email host

Turning on an account at an email host requires these steps:

1. Set up an account at the new email host.
2. Obtain their mail exchange record information.
3. Follow the steps in [Set Up Your DNS Host](#) to set up your DNS host to point to this email host.
4. At your email host, enter the subdomains at which you want to receive email. This may be just a single subdomain that represents your domain name. (If you want to receive email for one or more host names that precede your domain name, you normally set each of these up separately at the email host.)
5. Create a list of addresses for which you want to set up separate email accounts.
6. Create a list of aliases that you want to redirect email from, mapping them to email accounts you've set up, or to email addresses at other services (if supported).

Email should start arriving as soon as the DNS host's information is updated and the email host has restarted its mail server or servers with your new account details; this could happen immediately. Of course, someone *does* need to send you email for you to receive it.

REDIRECT YOUR DOMAIN'S WEB SITES

Registering more than one domain name to point to a single Web site isn't unusual. Nor is it strange to host your Web site at an ISP or online community site and want a subdomain to bring people directly to what's often a long and hard-to-remember URL.

Redirection is the answer. With redirection, browsers can be pointed from a subdomain to a Web site. You use redirection when you want something you don't control to be pointed at by something you do.

For instance, if you have a community page at a social or group site that has a URL like `http://www.yahogroups.com/groups/interests/railroads@78734@234,2343.html`, you might prefer that `http://comm.rr_fans.com` act as the public URL.

I explain how to create a redirection in three ways in turn later in this section:

- With HTML or JavaScript on a Web page
- Using stealth redirection
- Directly via DNS settings

NOTE If you run your own Web server or have access to the configuration settings on your Web host, you can also use tools for that server that redirect from within the Web server software itself.

Web Page Redirects

To set up a redirect, you can use a special feature in HTML or a command in JavaScript. Redirects can take a little longer, must be applied on every affected Web page, and require the browser's involvement. While these seem like disadvantages, Web page redirection is entirely within your control, and doesn't require special Web server software configuration that your Web hosting firm may not provide.

HTML tags

The HTML markup language offers the META tag, which provides information about a page. The tag takes attributes that control browser behavior. The attributes we're interested in are HTTP-EQUIV, which tells most browsers released since 1998 to reload the

page using the URL specified next, and CONTENT, which provides the parameters. The parameters are the time to refresh measured in seconds and a URL. The tag on a real Web page might look like this:

```
<meta http-equiv="refresh"
content="0;url=http://glennf.com/new_page.html">
```

META tags must be in the HEAD portion of the HTML for a Web page. For redirection, I suggest creating the shortest possible page to make reloads as fast as possible, as shown below; note that I've formatted the redirection URL in red italics; you'll use your own URL, not mine.

```
<html>
  <head>
    <title>Redirecting</title>
    <meta http-equiv="refresh"
      content="0;url=http://glennf.com/new_page.html">
  </head>
  <body></body>
</html>
```

Setting the refresh to 0 seconds can apparently disable the Back button on older browsers, but I don't think that's a modern concern.

JavaScript

A simple JavaScript, such as the one shown below, works more efficiently than the META tag, because the script redirects to the new URL (red italics) as soon as the browser receives it. However, some users may disable JavaScript, rendering the script useless for them. For best results, pair the script with the META tag redirect above, with the JavaScript in the HEAD portion of the page, followed by the META tag. The BODY can be empty.

```
<script type="text/javascript">
  <!--
  window.location = "http://www.glennf.com/"
  //-->
</script>
```

Stealth Redirection

Stealth redirection uses Web frames to make the browser display the originating URL but the destination Web page. Most modern browsers support frames. (Stealth redirection can be a bad idea, since it can negatively affect the way search engines index your site, and some tools to fight fraud might flag your site as problematic.)

Most DNS hosts offer some form of stealth redirection in addition to regular redirection. Here's how easyDNS handles it, which is similar to many other DNS hosts. To turn on either form of redirection, follow these steps:

1. Log in to your account at easyDNS.
2. Find the domain name for which you want to add stealth redirection, and click the DNS link to the right of its name.
3. Scroll down to URL Forwarders. (If the contents aren't displayed, click the plus (+) sign on the right side of the box.)
4. Enter the subdomain you want to forward and the URL to which it should forward.
5. Check Stealth to use stealth redirection; leave the box unchecked for normal redirection (**Figure 10**).
6. Click Next to confirm the changes and activate them.

FIGURE 10



The screenshot shows a web interface titled "url forwarders" with a subtitle "Sorted, expanded style". Below the title is a section titled "Add a URL Forward". This section contains two input fields: "A record (host):" with the value "stealth.glennf.com" and "Forwards to URL:" with the value "http://isphost.com/glennf/~webfiles/". To the right of the second input field is a checkbox labeled "Stealth:" which is checked.

Configure easyDNS to provide a stealth redirection.

Map Many to One with DNS

Using multiple subdomains at one domain name allows you to have many addresses that resolve to the same Web site. For instance, when users type in `www.fresh-eggs.com` or `www.cage-free-chickens.com`, you might want them to land at the same place.

To point multiple subdomains to one set of files, follow these steps:

1. Contact your Web host and inform them of the subdomain names you want to point at a Web site. Some Web hosts may let you enter this information directly via your account on their Web site.
2. Obtain the IP address settings from your Web host for where to point the subdomains.
3. Visit your DNS host and enter the subdomains you're defining and the associated IP addresses (or just the IP addresses, if you're moving the pointer for existing subdomains).

RANT My editor Joe Kissell and I are sick and tired of Web pages that say, "You will be redirected to the new page in 10 seconds." Why wait? In the blurry olden days, redirects were perhaps strange and new, like fire. ("Fire...bad! RRRRRR.") But now, they are just a slight inconvenience. By default, certain servers, including older versions of Microsoft IIS, would display a redirection message. Just give us the new page, already. We'll fix our bookmark if we care enough.

USE DYNAMIC DNS

Dynamic DNS (or DDNS) answers the question: How can I attach a fixed domain name to my computer when it gets a different IP address every time I connect to the Internet? DDNS requires some software, a compatible broadband gateway, or manual intervention, but it's an effective solution to a frustrating problem. DDNS works in two scenarios:

- When you have an ISP that temporarily assigns you a *dynamic IP address*, one drawn from a pool of addresses they manage, but doesn't guarantee you'll keep that address for any period of time.
- When you're a mobile user with a laptop or handheld that you connect to many different networks.

NOTE The opposite of a dynamic address is a *static* address: It's assigned to a given computer and remains the same over time. Regular DNS maps subdomains to static addresses.

Companies that offer DDNS—which includes many DNS hosts—let you replace the IP address for a given subdomain with whatever address you're currently at when you fire up a computer or gateway.

Address Translation Limits Dynamic DNS

One roadblock may stand between you and DDNS: *Network Address Translation (NAT)*, a technology that's embedded in most broadband and wireless gateways. In its most common form, NAT takes a single IP address assigned by your ISP to your Internet connection and performs network magic so that more than one computer using your Internet connection can access the Internet via that one IP address.

This magic involves creating internal *private* IP addresses that can't be reached via unsolicited connections from outside the local network; that is, all Internet connections must be initiated by a local computer on the internal network, which then receives only responses, as occurs when a Web browser receives a Web page. What happens is that NAT listens on the local network to outbound requests, like "get me a Web page." It then routes the request to a port on its public side and mediates traffic between the private and public networks.

Without setting up something special, a computer that has only a private IP address cannot act as a server on the Internet, because it lacks a *public* or *routable* IP address and cannot receive requests from outside the local network.

Because DDNS requires a public address to work, NAT is a bar to its use, but you can lift that bar if your gateway offers *port forwarding* or *port triggering*. These two features make possible a connection between a privately addressed local computer and a specific port on the public side of the gateway, each in its own way.

Port forwarding allows you to specify a port on your public network gateway to connect directly to a port on a local computer. This is a fixed mapping of a public port to a private port. In contrast, port triggering, used commonly with games, lets an outbound request from a computer on the network *trigger* your gateway to open up and forward one or more ports to that particular computer.

Thus, DDNS can map a dynamically assigned public address, provided by your ISP to your gateway, and a dynamic or static address provided to a computer by NAT in the gateway. Forwarding is often used to operate a public Web server within a private network, and triggering typically punches through NAT to enable you to play multi-player games.

NOTE Every single gateway—often even two apparently identical devices from the same manufacturer—has a different way of labeling and configuring port forwarding and triggering. It’s one case in which reading the manual is your only recourse.

Unfortunately, it can get worse. It’s bad enough if the IP address dynamically assigned to your gateway can change, but ISPs increasingly use another layer of NAT on their broadband services, too, which means that the IP address you’re assigned isn’t just dynamic, it’s also private to the ISP. This produces an untenable situation: If the ISP uses NAT, there’s no way to use DDNS because the ISP won’t let you connect via a public, routable IP address that’s specifically mapped to your broadband gateway. With many ISPs, you can pay more to obtain a static address, obviating the need for DDNS.

TIP I discovered recently that Skype, the Internet telephony program, can create computer-to-computer tunnels no matter how many NAT gateways stand between the two computers. Skype first opens a connection outward to a routable network address for each party, and then connects the two. Skype has an interface that allows developers to piggyback on its connection. Netopia's Timbuktu Pro, for instance, can use Skype for its remote-control and file-transfer programs.

Am I reachable?

You're now asking the 64-bit question: How do I tell if my ISP is using NAT? And do I have a routable (static or dynamic) IP address? I have a couple of answers. First, you could ask your ISP, and they could give you an accurate answer. Yeah, right.

Second, you can check the address dynamically assigned to your gateway by using its Web interface or another tool (such as AirPort Admin Utility for Apple's AirPort base stations). If that assigned address begins with 10, 192.168, or 172.16.0 through 172.16.31, you know that your ISP is using NAT, because addresses that begin with those numbers are reserved for private networks by the global numbering authority.

Failing that, try a third method, using a couple of Web utilities. Visit AuditMyPC.com's exposed IP check at <http://www.auditmypc.com/whats-my-ip.asp>. This shows you the IP address at which the rest of the Internet sees your computer. The site will read: "Your current IP address is" followed by an IP address. If it lists a second address, your ISP is likely using NAT. If that second address starts with the numbers noted above, it's definitely NAT.

If there's just a single address listed, copy it and visit a Ping Test site at <http://www.tellurian.com/scripts/tools/ping.asp>, where you enter the address and click Ping. (A *ping* is a tailored data request that checks to see if anything responds at a given IP address.) This checks to see if that IP address is reachable from the rest of the world. If the Web page shows "Reply from:" followed by the address you entered, then the address is reachable. Some operating systems, gateways, ISPs, and network operators offer to block all ping requests for security reasons, which would also make this test fail, unfortunately.

If your address appears to be a private one, you'll need to talk to your ISP about how to get a routable address. They may not offer one, and if that's true, you might need to switch ISPs or use hosted services to run your Web server or other publicly reachable server.

Set Up Dynamic DNS

Assuming you can use DDNS, you have two basic choices: using a DNS host that supports it so that you can use a subdomain of a domain name you own, or using a DDNS service that lets you “borrow” a subdomain of theirs. (In the latter case, they create a subdomain for you using a domain name they have registered and own; they pair that subdomain with your ever-changing IP address.)

Of the DNS hosts that I suggested earlier, only easyDNS supports DDNS. Later in this section, I also cover two services that offer free DDNS support using generic domains they operate. These services also register domains and provide DNS hosting, and could thus be a better choice if you use dynamic DNS extensively.

For any of the methods of using DDNS that I describe, you must use software or a manual process to update the DDNS settings on a regular basis (or whenever your dynamic IP address changes). Don't forget that after following these instructions for setting up DDNS.

NOTE If you'd like more options, consult the Google Directory listing of services at http://directory.google.com/Top/Computers/Software/Internet/Servers/Address_Management/Dynamic_DNS_Services/.

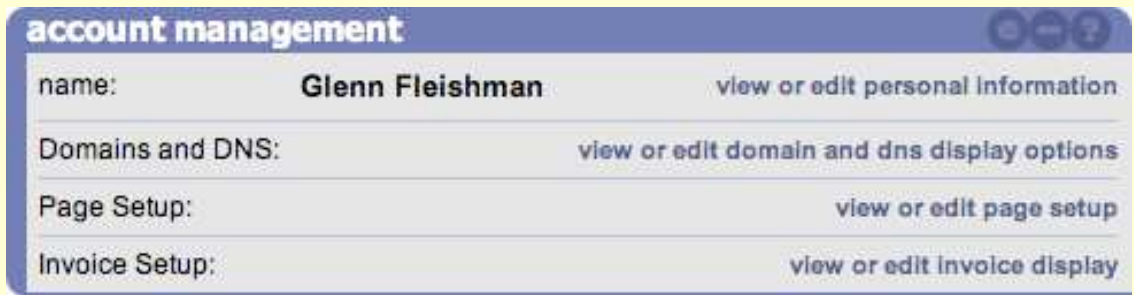
Configure domains for DDNS with easyDNS

You can use dynamic DNS with any host at any domain at easyDNS as long as your service level for the domain isn't Free Parking.

You have to enable DDNS for your account and then for particular domains. Follow these steps:

1. Log in to your easyDNS account.
2. Click “view or edit domain and dns display options” in the Account Management section (**Figure 11**).

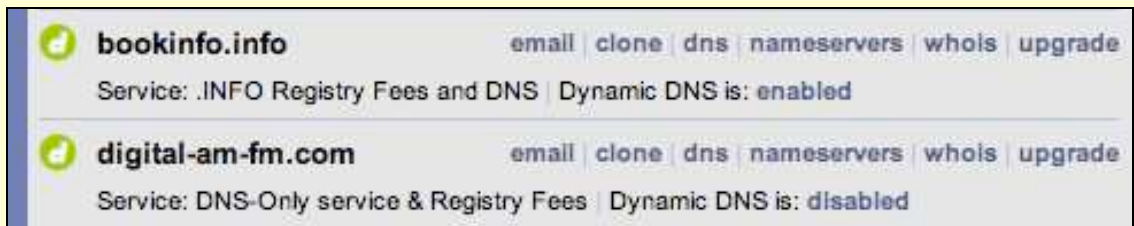
FIGURE 11



Click the link to the right of Domains and DNS.

3. From the Dynamic DNS menu, select On; then click Done at the very bottom of the page.
4. For each domain that qualifies for Dynamic DNS support, you'll see an item that says "Dynamic DNS is:" followed by enabled or disabled (**Figure 12**). Click Enabled or Disabled to change status.

FIGURE 12



The links *enabled* and *disabled* change the domain's DDNS status.

Now all the subdomains (host name plus domain name) in any domain you enabled with DDNS can have their IP addresses set using the software described ahead in [Update with software](#).

Configure free services

I know of two firms that offer free DDNS accounts using subdomains within a variety of generic-sounding domain names. DynDNS offers names like [gotdns.com](#), while No-IP.com offers gamer-oriented domains like [servehalflife.com](#). (You can DynDNS's full list at <https://www.dyndns.com/services/dns/dyndns/domains.html> and No-IP.com's at http://www.no-ip.com/services/managed_dns/free_dynamic_dns.html.)

To use either of these services, follow these steps:

1. Sign up for a free account.

WARNING! Don't use a user name and password that's used for another, more private account, because DDNS info can be sent in the clear, and you're often invoking it on public networks, such as Wi-Fi hot spots. For more about passwords, see *Take Control of Passwords in Mac OS X* at <http://www.takecontrolbooks.com/passwords-macosx.html>.

2. Activate the account via an email they send you, and log in.
3. Configure a host in one of their domains:
 - DynDNS:
 - a. Under the My Hosts menu, click Add Host Services.
 - b. Click Add Dynamic DNS Host.
 - c. Enter a hostname and select a domain. The IP address field will show your current public IP address.
 - d. Click Add Host.
 - No-IP.com:
 - a. Under the Hosts/Redirects menu, click Add.
 - b. Enter a hostname you choose, and select a domain. The IP address field will show your current public IP address.
 - c. Click Create Host.

Update Dynamic Subdomains

Obviously, the most critical part of DDNS is the *dynamic* aspect: When your IP address changes, you want the new address to be rapidly assigned to the subdomain you've chosen. With every DDNS service I'm aware of, a user name and password is required to make DDNS changes. Otherwise, malicious individuals could vandalize domain records. You have two options for updating DDNS values: via a downloadable application that runs on your computer or using a gateway with DDNS support.

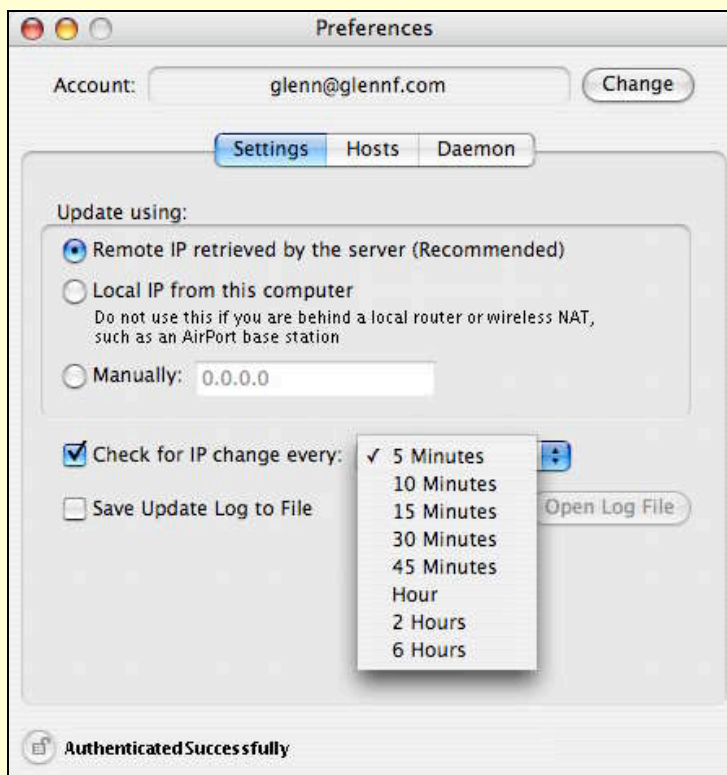
NOTE Dynamic DNS changes are signaled through a de facto standard that uses a Web page request that passes certain specific parameters. Not all DDNS services support all parameters or use them in the same way. You can read about these parameters at DynDNS (<http://www.dyndns.com/developers/specs/syntax.html>).

Update with software

To keep your dynamic IP address mapped automatically to your domain names with software, you must install a DDNS client on the computer that determines the dynamic IP address assigned to that computer. The software then sends a message to the DDNS server whenever your IP address changes; it can make a change periodically or whenever you request it. Numerous software packages perform this service. No-IP.com (<http://www.no-ip.com/downloads.php>) and DynDNS (<https://www.dyndns.com/support/clients/>) have custom software designed for their own services. DynDNS also provides a fantastic annotated list of third-party DDNS client software (<https://www.dyndns.com/support/clients/third-party.html>). Most of this software also works with easyDNS.

Generally, you set up the software on your local computer with the login information for your DDNS account and the domains you want to have updated on that computer. The software provided by DynDNS and No-IP.com for their own systems retrieves a list of hosts you've set up (**Figure 13**).

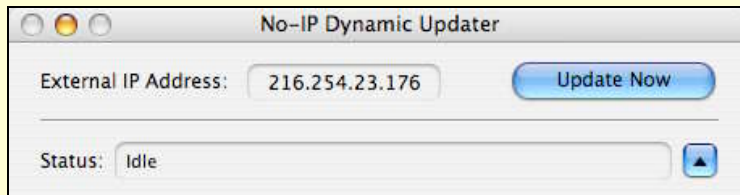
FIGURE 13



When configuring No-IP's DDNS software, you can set how often the software checks for an IP change.

When you configure the software, you will see some indication of current settings and whether an update has occurred. In **Figure 14**, the current IP address is displayed, and the “Idle” status message means that no update has happened recently.

FIGURE 14



The No-IP software shows the external IP address it's currently using. Click Update Now to force an immediate change.

Update with a gateway

A number of broadband and Wi-Fi gateways used to share Internet connections or to enable Wi-Fi-equipped computers include DDNS support. If your gateway supports DDNS, then you can use it, instead of software as I described just earlier, to communicate with a DDNS host about a dynamic IP address.

To set up a gateway to provide DDNS for itself, you typically need the user name and password for the DDNS service, along with the host name for the subdomain you're setting; the host name is the part before the domain name, like **www** in **www.tidbits.com**. The gateway does the rest.

While not every gateway supports every DDNS service, you may achieve the best results with DynDNS, which offers a long list of gateways that support its particular service (<http://www.dyndns.com/support/clients/hardware/>).

MOVE YOUR DOMAIN NAME

Not every relationship ends beautifully. Companies underperform, lie outright, or just annoy you. Your needs change and a company lacks what you want. You're running your own servers and are tired of being your own technical support. Whatever the reason, moving a domain name is typically harder and more frustrating than registering a domain and setting it up. Why? Because you have active services you don't want to interrupt, and because you have companies that want to keep your business.

Moving a domain name can involve one or more of these parts:

- **Change your registrar:** Some registrars are cheaper than others; some provide much better self-service Web sites than others. Whatever the reason, you are entitled to pay whichever registrar you want to handle your registration information.
- **Change your DNS host:** You might find that whoever handles your DNS—whether it's the registrar or another party—doesn't give you the flexibility you need, or has poorly run DNS servers. It's generally simple to move your DNS host by changing information at the registrar. If your registrar and DNS host are the same party, there's another approach to take. Changing a DNS host is easier than changing a registrar.
- **Change your Web host or email host:** The simplest yet paradoxically most difficult task of the three, moving a service means changing a DNS value that points to the service. If you have a good DNS host, making the DNS changes are easy, but the transition still requires planning at your new Web host or email host in order to avoid interruptions.

The general principle for each of the changes is to *set up the new stuff first*. No matter what kind of move you're planning, you want an account, settings, and content to be in hand or transferred before you make the move.

WARNING! Don't try to perform a move in the last few days of a subscription to your existing host. With only a few days, you may encounter problems that won't let you revert back while you resolve difficulties.

This is a bit like lifting a heavy tray off a table while a friend pulls the first table away and puts another one beneath what you're holding. You want that friend in place and the second table prepped before you're left holding the tray.

Time to Live

One step that helps to ensure a smooth transfer of DNS hosting and the hosting of other services is reducing the *time to live (TTL)* value of your DNS settings. The TTL value tells other DNS servers how long to cache information about your domain. TTL is typically set to something like 86,400 seconds (one day) or 604,800 seconds (one week).

When you're about to make a significant change to your DNS, first change the TTL to a few minutes. Many DNS hosts allow this change through their regular or advanced interface.

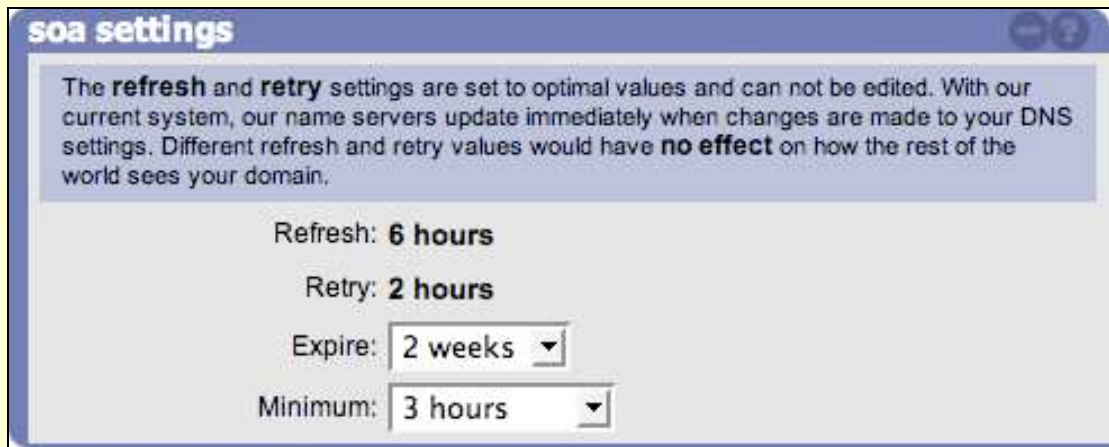
Because of how DNS works, plan to make changes after the current TTL has passed. If your TTL was set to a day and you change it to a minute, wait until a day has passed before making changes so that any nameserver that cached your DNS information up until a second before you lowered the TTL value removes that information on the next request.

Some DNS hosts already keep TTLs short, like a few hours, to facilitate this. Others set TTLs to a default value and then don't let you change that value, like Yahoo!'s small business hosting service, mentioned previously.

The flip side of very short TTL, however, is that if there's any interruption or overload on the DNS servers that provide information about your domain, your hosted sites will be unreachable even if they're perfectly fine because the DNS won't resolve, and the values will have expired from the caches of other DNS servers.

With most DNS hosts, you set the TTL using the part of their Web-based configuration that lets you set raw DNS values. For instance, at easyDNS, raw DNS configuration is found in the basic DNS configuration option for a domain (**Figure 15**, next page); at GoDaddy, it's located in Manage Domains when you click on the link for a domain name, and it is labeled Total DNS Control.

FIGURE 15



easyDNS labels TTL as Minimum in the SOA (start of authority) section of DNS configuration. The first three values control how secondary servers query a primary server for information, and they have no effect on caching at unrelated DNS servers.

WARNING! Many DNS servers ignore the TTL value for their own purposes. A Slashdot posting in 2005 that included testing by a Slashdot administrator showed that large ISPs override TTL values in the interest, apparently, of sending fewer queries despite the fact that queries are tiny. This, of course, breaks the central tenet of DNS: that the DNS configuration files on the authoritative nameservers contain information that other DNS servers accept as correct. Feh.

Change Your Registrar

I can't tell you how much aggravation I have experienced over the last decade with domain registration migration. It's definitely improved dramatically, but there was a point when we were all so beholden to Network Solutions—originally the *sole* registrar—that it required patience and deep-tissue massage.

Registrars maintain their own databases of domain ownership, but coordinate this information with a centralized database that all registrars interact with. This centralized resource contains the current pointers that allow the root nameservers that handle all the top-level domains to redirect DNS queries to the right DNS servers.

When you change registrars, you're moving not just information like your name and address, but the IP addresses of your DNS host's

nameservers. Any discontinuity during a transition means that your domain's resources, such as a Web site, would be unreachable.

NOTE Many registrars credit your remaining time purchased at another registrar (often up to one year) when you transfer your domain name well in advance of a domain's expiration date. Some charge a transfer fee, however, which may negate that credit.

Changing a registrar involves verification steps by a third party to prevent someone from stealing a domain by transferring it. Each registrar has a slightly different process for initiating and approving transfers, but the transfer requires these actions:

1. **Disable the Registrar Lock (if enabled):** If you've enabled Registrar Lock on your domain—always a good idea and sometimes turned on by default by your registrar—you need to disable this lock before attempting the transfer. (For more details, read the note [Lock Down Your Domain](#), much earlier.)
2. **Set up a new account:** You initiate a transfer from your new registrar, and so you first must set up an account and pay any domain fees for the transfer.
3. **Set up your DNS:**
 - If your new registrar is *not* your DNS host:
 - a. Visit your DNS host and obtain their nameserver information.
 - b. Enter this information at your new registrar.

TIP Some registrars and DNS hosts walk you through the process of setting up nameserver information when you start a domain transfer, or offer to copy existing information from your current registrar.

- If your new registrar and DNS host are the same firm:

Because the registrar and DNS host are the same, you need not enter nameserver information at the new registrar. But you should enter the new nameserver information at the soon-to-be-old registrar/DNS host, so that your DNS is under the control of the new host even before the transfer is complete.

- a. Visit your soon-to-be-old registrar/DNS host and write down all DNS information.
 - b. At your new registrar/DNS host, enter all the DNS information.
 - c. At your new registrar/DNS host, obtain their nameserver information.
 - d. Go to your soon-to-be-old registrar/DNS host and enter that nameserver information.
 - e. Verify that the new entries work with a DNS lookup tool ([Appendix A: Tools for DNS Lookups](#)).
4. **Start the transfer:** Follow your new registrar's procedure for a domain name transfer, which can vary by registrar. Read each registrar's instructions carefully. This action is sometimes part of Step 3.
 5. **Confirm the transfer:** Respond to email sent to the address in your domain's registration, using the method the email describes for validating your desire to transfer the domain.

NOTE If the email address in the domain registration no longer works, you may be able to make the change via the account you have at your new registrar, but it could take a few days for that change to be recorded. Some registrars don't allow email changes via their Web account management tools, and you must contact them directly, sometimes primitively printing out and faxing them a signed form or a note on letterhead, and even a fax of your driver's license. I have more than once had to create fake letterheads for domains I have registered in order to complete a legitimate transfer.

TIP Expect delays when transferring domains—they are normal and seem to occur when you least want them. Don't try to transfer a domain from one registrar to another within 3 weeks of the domain's expiration. You don't want an error to render the domain expired or in a strange limbo.

Change Your DNS Host

It's fairly easy to change your DNS host, because if you control the domain's registration, you can simply move the pointer that directs DNS queries from one DNS host to another. Follow these steps:

1. Set up an account at the new DNS host.
2. Copy the information from your current host, typically by taking a screen shot of the appropriate information or, because there are just a few salient numbers, typing that information into a note or writing it down by hand.
3. Enter this information at your new DNS host.
4. Obtain the nameserver details from your new host.
5. Enter the nameserver information at your registrar. (If your registrar and DNS host are the same, see [Change Your Registrar.](#))
6. Wait for the registrar's nameservers to reload new values. Most registrars tell you how long that will take.

Change Your Web Host

The steps you follow to change a Web host depend on whether your DNS host or registrar/DNS host serves as your Web host, or whether DNS and Web hosting are separate.

TIP You can retrieve DNS settings directly from any DNS server using command-line tools like `dig` and `nslookup`, and through graphical software that performs DNS lookups and presents the values nicely. This can help you when making any changes to your Web host as you can look up whether the new values are in place locally (at the DNS server your computer uses) or globally (at other DNS servers, including your DNS host's own nameservers). And it's a good double check when you have problems after making changes. See [Appendix A: Tools for DNS Lookups](#).

When your DNS host *isn't* your Web host

If you're moving only your Web host, follow these steps:

1. Set up a Web-hosting account at a new firm.

2. Upload all your Web content, and test it under a temporary subdomain or by using the IP address directly in place of your normal subdomain. Make sure everything works before proceeding. (Obviously, if you used absolute URLs that include your subdomain in your HTML files, those links will break, but should work again once your subdomain is in place again.)

TIP You or a Web designer might have used the HTML tag called BASE, which allows you to define a URL that's the starting point for any relative URLs in the HTML file. If the URL used as this starting point includes a subdomain that's in transition, links and image display might fail even if all the references within the Web pages are relative. Be sure to look for BASE tags to avoid this problem; you could comment out the BASE tags during a transition, for instance, and then re-enable them when the DNS change has taken full hold.

3. Obtain from the Web host the IP address information that you need to enter at your DNS host.
4. Visit your DNS host and enter the values for all the Web sites that you are now serving from your new host. After you save the changes, the DNS host will typically tell you how quickly those changes will start to appear.

You will absolutely want to keep your old Web host account alive until your old DNS values expire at DNS servers around the world that have retrieved IP addresses for your Web site or for users who have recently sent you email. Your DNS host may tell you how long that will take—possibly as short a period as a few hours. For more on how DNS values are spread and cached across the Internet, see the sidebar, much earlier, [Smells Like DNS Propagation](#).

When your DNS host *is* your Web host

If you're planning to split DNS hosting from Web hosting for an existing site, follow the steps above, with the addition of a step 5:

5. Go back to your DNS host and cancel the Web-hosting service at that site. If you forget to change this—as I once did—you'll continue to pay unnecessary fees.

Change Your Email Host

Moving your email host is one of the most traumatic experiences you can go through, especially when *many* people are using email accounts at the domain name for which you're making the change. The problem is often fourfold:

- It takes time for changes to DNS values to time out around the Internet, and email could go to the old account if it's sent during the period between when you change the mail exchange record and when the older values cached by other DNS servers expire. (See [Time to Live](#) for how to adjust this duration.)
- It can be hard to test whether your new email host is perfectly set up to receive email for your domain. (See [Troubleshooting](#).)
- You may be unable to access your old email account once you make DNS changes, and thus email at your old account that accumulates could be difficult to retrieve.
- If you use IMAP for email, you may have messages that are stored on your old email account; you need to retrieve those entirely and perhaps delete them from that server before starting a transition.

These problems crop up whenever you change DNS records for email no matter which companies host your registration, DNS, Web site, and email.

Let's start at the beginning.

Set up new account at the email host

First create an account at the new email host. Then follow these steps.

1. Add the subdomains for which you want to receive email.
2. Create the user accounts and aliases on the new server. You might want to display your old accounts in a separate browser window to make copying settings simpler. (Sadly, there's no standard for defining mailboxes, so you can't export and import settings.)
3. Test the new account or accounts, as I explain next.

Test your new email account

Before you change DNS records, there's no simple way to test if your new email server has been correctly configured to accept messages for your subdomains. But there's a way to use Internet protocols to test, if you have the stomach for it.

First, obtain the mail server IP addresses for your new email host from wherever the new host provides this configuration information. (You'll need the mail server IP addresses when you change DNS settings, too, in just a moment.)

In this example, the new mail server is `mail.mynewhost.com.`, and your current email address is `foo@example.com`.

Now, bring up the right software for testing:

- On a Mac, open `/Applications/Utilities/Terminal`. At the prompt, type `telnet mail.mynewhost.com 25` and press Return.
- Under Windows 95 and later, choose Start > Run, enter `telnet`, and click OK. The Telnet application launches. Choose Connect > Remote System. In the Host Name field enter the subdomain of the mail server followed by a period: `mail.mynewhost.com`. In the Port field, enter `25`. Click Connect.

These steps connect you directly to the remote mail server. The server responds with some kind of welcome, such as:

```
220 mail.nynewhost.com ESMTTP.
```

Now, follow these steps to run the test, pressing Enter wherever you see a paragraph mark (¶):

1. Type: `EHLO anything.com` ¶

The mail server replies with a list of parameters, which isn't important.

2. Type: `MAIL FROM:<foo@example.com>` ¶

The server replies `250 Ok` or something similar.

3. Type: `RCPT TO:<foo@example.com>` ¶

The server replies `250 Ok` again.

4. Type the following:

DATA ¶

Subject: Testing ¶

¶ *(Just press Enter on this line)*

Testing to see if my new email account works. ¶

. ¶ *(That's a period on a line by itself)*

The server should reply **250 Ok** and note that the message was accepted or queued.

5. Type: **QUIT** ¶

The email server closes the connection.

6. Now, using a webmail interface or a newly set-up account in your email program, check that the mail was received at your new email host. If not, contact the email host, explain that you don't believe the subdomain was correctly added as a legitimate incoming address, and send them the text of everything that the server responded with above.

Set up transition account(s)

There's no way to avoid a period of time after you change your DNS settings for receiving email and before all the email servers that have delivered to your domain in the recent past catch up. This period could be a few minutes or up to a full week. And you don't want to miss even one legitimate email message. You have two strategies that can help you over the hump.

Keep checking email at the old email host

If your old email service allows you to check your account even after the DNS records have changed, that's the simplest method. However, this can get complicated, depending on how the old and new email hosts require you to set up your DNS mail server records. For instance, some email services require you to set up a subdomain that's called **mail.yourdomain.com**, and set the IP address to the email services' mail servers. If both your old and new email hosts have this requirement, you may have problems with reaching your old email hosting company's mail server while the transition is underway.

Here are three suggestions for how to get around that limitation:

- **Use the old mail server IP address:** At your DNS host, look at the IP address for the mailserver(s) you set up for your old mail host. In your email program, replace your mail subdomain's name with the IP address you found. You may have to re-enter your password.
- **Use webmail:** You may be able to access webmail at your old email host temporarily through a domain they operate.
- **Ask your new host if you can use a different hostname:** Some email hosts might require **mail** as the hostname, as in **mail.glennf.com**; most shouldn't care and might let you set any name. Then you could leave your old mail server's hostname as **mail** and set your new hostname to **messages**.

Forward email from the old server to another address

The tricky part in forwarding email from your previous email host is that your old email-hosting firm can't forward email received after you've changed DNS settings to your new email host's mail servers if the old email-hosting firm continues to receive email to addresses at your subdomain.

That is, if the old email host receives an email message addressed to **bill@tidbits.com**, it can't turn around and forward that message to **bill@tidbits.com**, because the old hosting company's mail servers think they're authoritative for that subdomain until you tell it otherwise, and they will accept that email as if they should receive it. You don't want to turn off email receipt for your domain at your old host until you're sure that the DNS values have timed out everywhere.

Many email hosts allow you to forward incoming mail for a given account or all accounts to another address. This could be a solution for any incoming mail that would arrive at your old email host after you change DNS records: Set up forwarding for all your accounts to a temporary address or addresses before changing DNS. Then, when your new mail server addresses have taken hold in DNS, you could set those temporary addresses to forward incoming email to your real email address or addresses. These temporary addresses could be free email account(s) at Yahoo, for instance. Hosts may also provide you an address that's at a domain they control, which you can use to make

the transition. Some providers don't offer a choice of a local address and automatically set up an account under the name of a domain they control—like `foobar@dreamhost.com`—for handling administrative matters.

Change DNS to point to the new mail servers

Your new email host will tell you whether to create a special mail subdomain or to point directly to their servers:

- If you need to create a new subdomain, visit your DNS host to create a new address record, typically for the host `mail` at your domain, and use the IP address that the new hosting firm provides for that mail server. Then, change the mail exchange record to point to the `mail` host in your domain.
- If you need to point to their servers, go to your DNS host and change just the mail exchange records. Your email host should provide you with a list of subdomains for their mail servers and a priority setting for each. (See [Enter mail exchange records](#) for more on mail exchange records.) As soon you apply the changes at your DNS host, and the host restarts their DNS servers, you should start receiving email at your new host.

Clean up

If your old email host is a large firm, like EarthLink or Qwest, you may regularly receive email from people on that same server. And if the email records aren't updated properly, the ISP will continue to deliver email to your old, no-longer-used account—or bounce it.

If you tell your old email host to stop accepting email for your subdomain before the DNS values are available at all potential senders' DNS servers, your old email host will bounce messages. To clear out all those DNS servers I suggest waiting a week before assuming that all the email intended for you is no longer delivered to your old email host.

Once a week has passed, you can remove the subdomain or subdomains from your old mail server hosting account and cancel that hosting account if it's just handling email. I suggest removing the subdomains first and then testing with the method I describe earlier—see [Test your new email account](#)—to make sure your old email host doesn't still think it's supposed to accept email addressed to you.

TROUBLESHOOTING

Whatever can go wrong, will, of course. In this section, I guide you through common problems related to DNS errors and failures. Some of this troubleshooting is worth reading to predict problems and prevent them before you have to decode errors later.

Registration Expired

If I had a dollar for every time a domain name registration expired without the owner knowing about it, I'd be a rich man. Unfortunately, many registrars still don't have their act together on better informing domain owners when expirations draw near.

Some send a lot of email, which often looks somewhat like spam and gets discarded. Most use email exclusively, which means that any email problem—such as your failure to update a contact email address—means you miss receiving the notification.

WARNING! Because many registrars now offer domain reservations for expiring domains, your expired domain could be swept up rather quickly by another party. Registrars offer grace periods of varying duration after expiration.

Prevent an unexpected expiration

You can prevent your domain from expiring without your knowing that expiration was imminent through several techniques:

- **Add the registrar's domains to your spam filter whitelists:** You should receive several pieces of initial email from a new registrar, and marking them as *ham* (that is, legitimate email) by adding them to a *whitelist* of legitimate email senders can ensure that future email makes it through.
- **Set up different email addresses for admin, tech, and billing contacts:** Most of us have many email addresses, some of which are rarely used, such as a free Yahoo! Mail or Google Gmail account. It's worth creating such an account if you lack one when registering a domain, because you could use one of these alternate addresses for one of the domain registration roles (admin, tech, or billing). There's this recursive problem that if you use an email

address at the domain for which you have entered contact information, and that domain has a problem, you may be unable to receive necessary email that must be responded to in order to migrate the domain or to make other administrative changes.

For a company, you should use a generic incoming address like **dnsadmin@company-name.com** to avoid registration and renewal messages bouncing if a particular individual leaves the company and his or her email address no longer accepts messages.

TIP What happens if your email is dead and you can't use an online account? You'll have to contact the registrar for their procedures. A few years ago, the only registrar, Network Solutions, required a letter faxed to a special number they provided you after waiting in sometimes hour-long phone queues. This letter had to be on letterhead. Of course, many domains weren't and aren't registered by businesses or use made-up business names. My editor and I have both had to create fake, but convincing, letterhead to force domain changes in domains that we owned more times than we would like to admit.

Because of the risk of social engineering, most domain registrars use more elaborate procedures and may charge an administrative fee. You might need to send a fax of a photocopy of your driver's license, too, along with other proof of your mailing address and identity.

- **Use different snail mail addresses for admin, tech, and billing contacts:** For the best luck in receiving snail mail reminders, if your registrar sends those out—though few do—listing different addresses would increase the chances of receiving a reminder.

NOTE Many registrars follow an ICANN recommendation and remind their domain customers regularly to update their contact information. But these reminders only work if your email address or addresses are up to date in the registration.

- **Use calendar reminders:** It's a very basic idea, and often forgotten. Just plug in the domain's expiration date and give yourself an alarm a few weeks ahead.

- **Sign up for 10 years:** Most registrars allow multi-year registrations at reduced cost, with 10 years typically being the maximum. Of course, you'd better have a really amazing reminder that goes off in your cybernetic implant in 10 years. GoDaddy charges \$8.75 per year for a 10-year registration (including an ICANN fee), although they frequently have sales and bundles that reduce that cost even further. easyDNS costs about \$21 per year (\$25 for the first year and an 18 percent discount for the 9-year renewal period). Register.com charges \$15 per year for a 10-year term.

TIP Series editor Tonya Engst, an adherent of the “Getting Things Done” philosophy of life hacking, suggests having a yearly trigger that reminds you to review your domain name registrations. The trigger would link to some paper or electronic record that listed all your domains and their expiration dates. If there were a domain that needed action that year, you'd then schedule that action.

Undo an expiration

If your domain name has expired, immediately visit your registrar's Web site and see how rapidly you can renew it. The governing authority for domain names obliges registrars to send a notice after expiration, but there's no uniform requirement yet for a grace period. Some registrars hold the name for longer than others.

I recommend making a phone call—yes, the phone, not email—if you have any confusion about how rapidly your domain will be reinstated.

An officemate recently had her domain name expire in the middle of working with a graphic designer who was revising and migrating her Web site. Email stopped working and the Web site was dead. The designer made a phone call to the registrar, and within a few minutes, the domain was back up and working.

Bouncing mail

The most common cause of mail that you should be receiving being bounced back to the sender is your mail server not being properly set up to receive email for your domain name or subdomains within that domain name.

The easiest way to test whether this is the case is to use my instructions in [Test your new email account](#). If the server doesn't deliver

your test email, contact your email host. If those tests don't reveal a problem, then it's likely that the bounced email was caused by one of two problems: server overload or spam filtering—I cover each of these in turn, next.

Server overload

Floods of spam, or even legitimate outpourings of email (from senders that use a mail server as their sending mail server or from anyone sending email that the server handles as a receiving mail server) can overwhelm the mail server. If someone sends you email that routes through a too-busy mail server that receives messages for you, that receiving server will reject the message with a status code telling the sender that it is too busy.

The sending server will then try other mail servers listed in your recipient DNS records, but if it runs out of options, the sending server will send an email notification message to the sender. This notification can look like a bounce, but it typically warns of a delay, indicating that the sending server will keep trying. However, mail servers that are having trouble keeping up with incoming email might send inaccurate error messages, bouncing email by stating that the recipient doesn't exist, or accurate ones, stating that a recipient's quota is exceeded or his email box is full.

If you're getting calls or email messages to alternate accounts telling you about these problems, check with your email host immediately.

Spam filtering

We all have a love-hate relationship with even the best spam filters: They keep our mailboxes from being clogged, but they also prevent us from receiving some email we need. Particularly aggressive filters installed on mail servers bounce some messages back to the apparent sender with an error message inserted at the start of the bounced email. (Most spam filters delete the most apparent spam or file or tag questionable messages.)

For instance, phone giant Verizon has been plagued with an email problem that has caused, at times, email from most senders to Verizon employees and Verizon DSL customers to be rejected wholesale. The bounce message appears to talk about a problem in delivering the email for a technical reason, until you read some of the fine print, which reveals that no technical reason is involved at all.

There's usually contact information in such bounce messages that tells a sender how to attempt to get his or her email delivered in the future. If those trying to reach you see a message like that, you can contact your email host or try to adjust settings for spam filtering in your account. In the worst case, turn off your spam filtering until you can figure out what's going on; it's better to receive more spam than it is to reject all legitimate mail.

Trailing Period

This problem will just kill you, as it's the kind of thing that only a programmer could have thought up. In the raw DNS records that map a domain name to an IP address, subdomains require a period at the end. If they lack that period, the configuration file will break DNS resolution for that subdomain and perhaps the entire domain name's set of information.

Here's why. These raw DNS files are set up with a root domain, and everything derives from that; the trailing period means "to the root of the DNS hierarchy." Subdomains listed this way are technically known as *fully qualified domain names (FQDN)*.

This needs an example, so let's look at my domain **glennf.com**. The raw file for **glennf.com** says, "Hey, this file defines hosts and other DNS details for **glennf.com**." That trailing period means that it's **glennf** in the top-level **.com** hierarchy, period—there's no higher level above that. And that domain is always appended to any entry in the file that lacks a trailing period.

Within that file, I can define, say, the Web server as **www**, which implies **www** plus **.** plus **glennf.com**.; or, I can define it explicitly as **www.glennf.com**.—with that all-important trailing period. (I can also define the root of my domain name as just **@**, the at-sign, which means, "configure for the domain name by itself without a host name at all.")

What if I omit the trailing period? Then I've accidentally created **www.glennf.com.glennf.com**.—which isn't the desired goal. This is a typical error when editing raw DNS files, but—in an ideal world—it shouldn't affect you when working with a DNS host that has a Web-based interface, or when you work by phone or email with a company that sets up the details for you. Either one should get the trailing period right, no matter what you do.

Unfortunately, the setup doesn't always go right. Many DNS hosting services assume that anyone entering IP addresses also knows about the trailing period and don't warn you about a potential problem if you enter the domain name without a period following it. Instead, they unthinkingly stick the incorrect information in your raw file and mess up your settings.

If you have just changed DNS settings and suddenly are suffering from unreachable Web servers or bouncing email, try checking for the trailing period first. You may need to call your DNS host if it's not obvious from their Web interface how they handle the trailing period.

Non-Resolving Addresses

What should you do if you get an error that one of your subdomains doesn't have an IP address associated with it? This often shows up as a "Web site unreachable" message in a browser, or an email address may bounce to someone who has tried to send you email with an embedded error message explaining that the specified mail server doesn't exist.

A common cause of this sort of problem is if you've failed to pay your Web-hosting bill, or if the Web hosting firm fails to credit your account even though you did pay. In either case, they then automatically disable your Web site. Check on this first.

Next, check your hosting company's Web site. If you can't reach it, there's either a problem with your local network or theirs. If you can reach their site, see if they have a system status page and see if any errors are listed that they're working on. If no errors are listed, contact them for help.

To aid in diagnosis, use one of the tools described in [Appendix A: Tools for DNS Lookups](#) to look up the IP address for the subdomain. If there's no address associated, check your DNS host's settings. If the settings look correct to you and your DNS hosting firm is reachable and has no system-wide errors noted, time to get in touch with tech support.

The problem could also have to do with routing, which occasionally affects parts of the Internet. The Internet is a whole composed of many paths among its parts, but not all paths are redundant. That is, you might be able to get from Point A to B to C to Z, but if Point C goes dead, you could find yourself without a route from A to Z.

The traceroute tool is one way to determine whether your Web site or other needed subdomains are reachable from outside the network on which they reside at the hosting company, and whether the network from which you're trying to reach your subdomain has a problem of its own.

Traceroute shows a response from each router between the first and last points in the trace, typically from computer you're testing with to the destination you enter. From a command-line prompt or from a site like Multiple Traceroute (<http://www.tracert.com/cgi-bin/trace.pl>), enter either the subdomain name or the IP address that's assigned to the Web server software that handles your Web site.

Whatever result you get from traceroute should help your various hosting companies troubleshoot what's wrong. Check the Web host's Web site for status updates (if they offer these), and then contact technical support if it doesn't show a routing problem or server issue. (Traceroutes use a special kind of data packet that some ISPs and network providers block for security concerns.)

Lame Delegation

While it sounds like an insult, *lame delegation* refers to domain servers that are listed for a given domain name—in either its registration or at a delegated DNS server—and yet fail to provide any information for that domain name. In short, the delegation is faulty.

Lame delegation results from a nameserver that, according to the registrar's records for a domain, is responsible for providing DNS information on request, but which is not configured to provide that information. This can happen when a nameserver hasn't been updated to add your domain or when the wrong name server is listed at the registrar, as can happen if your DNS hosting company changes the IP addresses of their nameservers and they fail to update the corresponding records at their registrar.

I recently helped my dad troubleshoot a problem with a domain of a nonprofit public library foundation for which he was redesigning a Web site. My dad's ISP, a large telecom, returned incorrect information from its DNS server. But the DNS host handling the nonprofit's domain returned the correct information. What was going on?

With the help of <http://www.dnsreport.com/>, I determined that the DNS host for the nonprofit's domain had set up only their primary nameserver to respond to requests for information about the domain. The secondary nameserver knew nothing about the domain. Coupled with that problem was that my dad's ISP's DNS server consulted only the secondary nameserver, for reasons that are unclear. My dad told the DNS host, which fixed the information in their secondary nameserver, and I was a minor hero.

TIP A domain's DNS records should contain not just a list of nameservers that have information about the domain, but also each nameserver's IP address as a separate entry. While including nameserver IP numbers is optional, if you don't include them, you burden every visitor with an additional DNS query by their resolver, which slows down DNS resolution and increases the delay in them reaching you. This extra IP address entry is called *glue*. A related problem is that if the DNS servers change their IP addresses, all DNS records with glue must be individually updated as well, unless your DNS-hosting company offers that as an automatic feature.

APPENDIX A: TOOLS FOR DNS LOOKUPS

While applications can easily consult the DNS resolver on a computer to turn a subdomain into an IP address, it's a little harder for us human beings. But there are tools that allow us to check the values that DNS servers store for various DNS record types.

When retrieving DNS information as part of testing your configuration, you need to think about which server you're asking for details. Depending on when changes to DNS values were made and the duration of the time to live (TTL), you could get an older or a newer value from four different categories of DNS server:

- **Stub resolver:** The DNS software built into an operating system that queries a full-fledged DNS server.
- **Local DNS server:** The DNS server operated by your ISP or that's on your local network.
- **DNS host's server:** The DNS server operated by the host that maintains your DNS information.
- **Any other DNS server:** Any random DNS server on the Internet might have a different cached value from the first two.

It's also possible with misconfigured DNS servers to get a variety of answers from any DNS server. See [Lame Delegation](#).

Web-Based Lookups

For a comprehensive report about your domain name's DNS settings, try DNSreport.com (<http://www.dnsreport.com/>). At no cost, the site performs an extensive evaluation, providing feedback on how to fix problems, and shows you the values it can retrieve "blind."

For specific lookups, such as checking that a given subdomain produces the right address or mail record, try DNSReport.com's sister site, DNSstuff.com (<http://www.dnsstuff.com/>). Enter the subdomain into the DNS Lookup box in the upper right. Extensive results are provided.

Command-Line Tools

If you want to get your hands dirty, try `dig`, a Unix program that's built into almost all Linux, Unix, and BSD flavors, including Mac OS X. There's a version you can install for Windows, too (<http://pigtail.net/LRP/dig/>).

Open a terminal window—under Mac OS X, open `/Applications/Utilities/Terminal`—and then you can take `dig` out for a spin. `dig`'s general syntax is

```
dig @server subdomain record_type
```

where *server* is the nameserver you're testing, *subdomain* is any subdomain you want to test, and *record_type* is one of the standard DNS record types, such as **A** (address), **MX** (mail exchange), or **SOA** (start of authority). If you omit *server*, `dig` uses the default DNS server set up for the computer you're using. Omit *record_type* to query just address records.

A practical run-through:

Let's say I was moving my mail hosting from `wwwwidgets.com` to `sssprockets.com`. My DNS host is `jetsondns.com`. (Yes, all names are invented.) I first query my DNS records to find the DNS host's nameserver, like this:

```
dig glennf.com ns
```

(This request doesn't need a server name specified as part of it, because I should get the same results from the DNS servers my computer consults as from my DNS host's DNS servers, unless I'd made a recent change.)

The results, which include a bunch of annotation and comments, include these two important lines:

```
glennf.com. 798 IN NS ns1.jetsondns.com.  
glennf.com. 798 IN NS ns2.jetsondns.com.
```

(The 798 indicates how many seconds are left in the cache for this information.)

I can now query my DNS host to see the current mail exchange record setting:

```
dig @ns1.jetsondns.com glennf.com MX
```

and be told:

```
glennf.com. 789 IN MX 10 mail1.wwwidgets.com.  
glennf.com. 789 IN MX 20 mail2.wwwidgets.com.
```

Now I change the mail server record to link to my new mail-hosting domain—**sssprockets.com** via my DNS host, and query again. I see this in response:

```
glennf.com. 670 IN MX 10 mail1.wwwidgets.com.  
glennf.com. 670 IN MX 20 mail2.wwwidgets.com.
```

Strange, right? I made the change, but I'm not seeing the new values at my DNS host. Ah, yes, I forgot to check the time it takes for my DNS host to load new values into their DNS servers—15 minutes. So I wait 15 minutes, and try my query again. The DNS host replies:

```
glennf.com. 3599 IN MX 10 us.ssprockets.com.  
glennf.com. 3599 IN MX 20 eu.ssprockets.com.
```

I'm all set. The change has been made and my DNS shows the value. If I want to find out whether an arbitrary ISP or company has picked up my new MX values, I can make a simple query. For instance, what's one of Yale's nameservers? (Some DNS servers block third-party queries like this.)

```
dig yale.edu ns
```

One of their DNS servers is **serv1.net.yale.edu**. I check with it to see if it has the right settings:

```
dig @serv1.net.yale.edu glennf.com MX
```

and it says:

```
glennf.com. 3600 IN MX 10 us.ssprockets.com.  
glennf.com. 3600 IN MX 20 eu.ssprockets.com.
```

Now I know that other DNS servers besides my local ones are retrieving the correct mail server values from my DNS host.

APPENDIX B: SELL A DOMAIN NAME

This book discusses registering and purchasing domain names, whether from a registrar or, in some cases, from a current domain owner. But selling domain names is quite common, and I'd be remiss if I didn't offer you advice on how to carry out that end of a deal.

If you own a domain that you want to sell, you might work with a company like Sedo (<http://www.sedo.com/>), a domain broker that can estimate of the domain's value, accept bids, and handle payment and escrow. If you go it alone, however, make sure you don't transfer the domain until you're absolutely sure that you have received legitimate payment that can't be reclaimed or found fraudulent. Fraud is, unfortunately, quite difficult to detect these days because of more sophisticated forgery and social engineering tactics.

NOTE Here's how some scams work. A buyer, often from Nigeria, offers you a large amount of money for a domain, but wants to send a cashier's check for a large amount above the offer because the buyer needs to get money to a U.S. relative or friend. You agree, receive and deposit the check, transfer the domain, and send off the excess funds. A few days or weeks later, your bank tells you the cashier's check was forged and refused, and now you're out the domain and the extra money. These scams are all too common for physical goods and are getting more complicated. Someone just tried to scam my father out of gravesites and excess fees for plots my grandparents didn't use (they chose to be cremated), and that he had listed for sale.

TIP If you can meet face to face with a buyer, you could accompany him to the bank to complete a transaction. However, remember that banks must report to the U.S. government any large or suspicious cash withdrawals. (Remember the movie *Say Anything?*) A business with a solid reputation could cut you a company check, but you should still wait for the check to clear. Company checks can be forged or stolen, so you should call the firm named on a company check (using a listed phone number) to confirm they wrote it.

Here are my tips for best practices when selling a domain name:

- **Get a written agreement:** You might consult an attorney—always advisable if large sums of money are involved—but a simple letter of understanding could simply state the required method of payment, the amount, the date it's due by or the deal is off, the method by which payment must be sent, the time for clearing the payment, and a waiting period (which I recommend) before the full transfer.
- **Require payment through limited methods:** Cashiers' checks and money orders are now frequently forged and used to defraud craigslist and eBay sellers. Even wire transfers can be reversed if an account contained fraudulent funds.

I recommend a U.S. Postal Money Order for amounts up to the limit of \$1,000 in U.S. currency. While they can be forged, real money orders have specific security features much like those in U.S. currency, such as embedded threads and watermarks (<http://www.usps.com/cpim/ftp/notices/not299/welcome.htm>). You can also take a U.S. Postal Money Order to a post office and receive cash. *Do not accept other money orders.* Money orders can be issued by a wide array of institutions, including Wal-Mart, and typically are very difficult to validate.

Cashiers' checks are an alternative method, and best for larger amounts. Because cashiers' checks represent actual cash receipts by a bank, as long as you can be assured that the check has not been forged, there's no way of reversing payment after a waiting period I describe next. (Calling the issuing bank is a good precaution, too.)

- **Require a waiting period:** I recommend a waiting period of 10 business days after receipt of payment before concluding the deal. Almost all forms of remuneration that are forged or based on fraudulent deposits elsewhere are discovered relatively quickly. After 10 working days, it's very unlikely that your bank would reclaim funds in your account.

For large transactions, retain the domain registration for 30 days after payment has cleared before initiating the actual transfer. This is a tricky step, but one I have to recommend strongly. You can point nameservers to whatever DNS host the purchaser uses and

they will immediately have full control of DNS values. But by retaining the registration, you are ensuring that any problems that could crop up within a month, however unlikely, are covered.

TIP As I write this, a friend sold a domain for six figures in which the bulk of the money will be paid *years* later. If the buyer doesn't pay at that time, the domain will revert to my friend. I recommended that my friend retain registration of the domain for the entire period in order to avoid a lawsuit to recover the domain should the purchaser ultimately default.

- **Consider an escrow service:** As an alternative, you could use an escrow service that handles domain transactions. eBay recommends Escrow.com for its own transactions, and Escrow.com has a domain name transfer service that charges reasonable fees (<http://www.escrow.com/>). They accept the money from a buyer, alert you that they've received it, and release funds to you when the transfer is complete. With an escrow service, you must transfer the domain, but the escrow service is responsible for assuring and releasing valid payment to you.

WARNING! A common scam involves a fake escrow service that's set up by the person who is trying to con you or by a confederate of that person. The buyer suggests using a certain service, you visit the site and it seems to work, but after you send them the money, the site disappears. Stick to well-known escrow services, preferably those who have references from major online traders like eBay.

APPENDIX C: TOP-LEVEL DOMAINS

Domains are hierarchical. Read right-to-left and separated by periods, they start with high-level authorities down to machines under your local control. Top-level domains (TLDs) identify the overarching authority that controls which registrars may handle domains in that particular hierarchy. TLDs include such varied entries as .com, .cn, and .info.

Second-level domains are typically the part that you register uniquely in a given TLDs. For instance, **glennf.com** is a second-level domain, comprising the TLD and a second part to its left.

Some countries have decided to allow users to register only *third-level domains* in their TLDs, providing a short list of second-level domains that are valid in that country. The United Kingdom and Australia, for instance, decided early on to partition their country-level TLDs in a manner similar to how the United States had divided .com, .net, and .org. Except for limited special cases, only names such as **bbc.co.uk**—where **bbc** is the unique, third-level component—may be registered.

TLDs include the original seven generic endings, country codes, and new entries for specialized business. (You can find a description and links to all TLDs at <http://www.iana.org/>.)

Original TLDs

The original seven TLDs were .com, .edu, .gov, .int, .mil, .net, and .org. They are now lumped together as *Generic TLDs*, abbreviated gTLDs. (There's a special eighth case, too, .arpa, that has a single purpose left for reverse mapping: IP addresses mapped back to sub-domains.) Only .com, .net, and .org are open for generic use. There are no restrictions on registering or using .org domain names for any purpose, commercial or otherwise, despite the widely held perception that the TLD is restricted to nonprofit organizations.

The other original domains have restricted purposes: .edu (accredited educational institutions), .gov (U.S. government decides), .int (NATO and other international treaty organizations), and .mil (U.S. military).

Country codes (ccTLDs)

Every country has an internationally established two-letter country code designated by the international standards organization ISO. Country code TLDs, horribly abbreviated as *ccTLDs*, mostly but not entirely, follow this ISO standard.

There are nearly 250 country codes, and they cover nations like the United States (.us), the United Kingdom (.uk), and the island nation of Niue (.nu).

Countries vary as to whether they permit foreign entities to register domains within those TLDs, and you'll need to check the registrar for each country code to find out. ICANN maintains a list of contacts for every country code's registrar at <http://www.iana.org/cctld/cctld-whois.htm>. Some country codes are handled by numerous other registrars. For instance, .uk domains are all registered by Nominet (<http://www.nic.uk/>), but you can purchase them through other registrars and hosting companies who handle the process.

Generally, smaller countries have more liberal policies, because their country codes have become the equivalent of the fancy postage stamps that used to be produced by tiny nations, like San Marino. San Marino offers domain names, too!

<http://www.telecomitalia.sm/default.asp?id=1048>

Some tiny nations have even sold their TLD rights to commercial firms to reap a financial reward, like .cc and .tv. The .to country code is frequently used for redirects, because Tonga's two-letter code is the same as the English word "to"—easy to spell, to boot.

Generic TLDs (gTLDs)

Starting in 2001, new generic TLDs began to be offered following years of discussions. These gTLDs are subject-specific categories.

These TLDs have specific, verifiable requirements you or your firm must meet to register within that TLD. They include .aero (air transportation industry), .cat (Catalonian language and culture), .museum (museums only), and .travel (travel industry related). The only exception is the .info TLD, which has no restrictions. The .biz (businesses), .name (personal names), and .pro (professionals) TLDs have very low bars for qualification, too.

GLOSSARY

caching: Temporary storage of information meant to reduce the amount of network traffic that produces the same result over short periods of time. Caching for DNS values is controlled via the *time to live* (TTL) setting in a DNS record.

DNS: Domain name system or domain name service, the technical part of how domain names connect to IP addresses.

domain name: A unique name that's registered within a given gTLD or ccTLD. The uniqueness allows the domain name to be used by itself or as part of a longer name to point to a specific resource anywhere in the world with a human-readable name.

dynamic IP address: An IP address assigned out of a pool of available IP addresses to a computer. The address may change over time, even as the computer is connected to the network, or may be assigned randomly any time a computer is connected.

FQDN (fully qualified domain name): A subdomain with a trailing period, as in `www.tidbits.com.`, where the final period indicates that the subdomain is globally defined.

glue: An entry for a nameserver in a DNS record that includes the nameserver's IP address. This entry is optional, but it can speed up domain resolution.

host name: The locally controlled portion of a subdomain. A host names a computer or server. A host name could be `www`.

HTTP (Hypertext Transfer Protocol): The language that Web browsers and Web servers use to communicate with one another.

IP address: The numeric, computer-oriented address that each device connected to the Internet uses to make connections to transfer data or pass queries.

Internet service provider (ISP): A company that provides a connection to the Internet to consumers or businesses.

lame delegation: A case in which domain registration states that a given nameserver contains information about that domain, while the nameserver professes ignorance. It says, "Lame!"

namespace: The universe of possible names within a given set of rules. Domain names must be no longer than a certain length and contain only letters (including some non-English characters), numbers, and hyphens, which restricts the namespace.

nameserver: An alternative name for a DNS server.

Network Address Translation (NAT): NAT servers assign private, non-routable addresses for computers on a local network in order to share a network connection (from an ISP or network provider) that has just a single IP address assigned to it.

port: A number that's paired with an IP address to correspond to a particular service, like a Web server. An IP address is separated from a port by a colon when used in a URL. The address 72.21.206.5:80, for instance, is Amazon.com's Web server IP address and port. For outgoing connections, port numbers are chosen randomly. A Web connection from your browser might originate from port 4644 on your computer but would have a destination port of 80 to reach Amazon.com's Web server.

port forwarding/triggering: A way to bypass NAT by connecting a broadband gateway's public IP address with a port on an internal computer that has a private address.

registrar: A party authorized by the global domain authority to register domain names on behalf of others. Registrars coordinate efforts through the global authority to keep domain names unique and handle technical details associated with operating the domain name system.

resolution: The process of taking a human-readable subdomain, like `www.takecontrolbooks.com`, and using the hierarchy of DNS servers from root to TLD to second-level domain and beyond to discover the IP address or mail exchange records, among other details, for that subdomain.

root nameserver: A DNS server that contains information about TLDs.

scheme: The part of a URL that defines the kind of resource being address. For instance, "http" defines resources on Web sites and "ftp" on FTP servers. In most cases, the scheme is followed by "://", as in "http://".

second-level domain: A domain name with two parts, right to left, such as `filbertfarms.com`. All generic TLDs (see [Appendix C: Top-Level Domains](#)) offer second-level domains for registration.

service: A kind of activity that's performed by a piece of server or peer-to-peer software. Web, FTP, and email are all services, as are AppleShare file sharing, BitTorrent, and instant messaging.

static IP address: An IP address that is persistent for a given computer or device.

subdomain: A host name plus a domain name is a subdomain. A special case is that a domain name by itself is an implicit subdomain as if “nothing” plus the domain name were used.

third-level domain: A domain name with three parts, right to left, like `bbc.co.uk`. Some countries reserve second-level domains to further subdivide their ccTLD, making only third-level domains available for registration.

time to live (TTL): The duration, defined in a domain name's settings, that defines the period that another DNS server may cache those settings. At the end of the TTL period, any cached data is considered stale, and any request to a DNS server for that domain's information results in a fresh query from an authoritative server.

TLD: Top-level domain, including `.com`, `.net`, and `.org`. TLDs define the top-level of the domain system from which addresses are assigned by registrars or country authorities. There are generic TLDs (gTLDs) and country-code TLDs (ccTLDs).

virtual server or virtual host: A virtual server or host appears to be a separate service to client software that's interacting with it, while it's run as part of a single service on a server computer. A Web server might be configured with 100,000 virtual hosts, each of which responds uniquely with requests to the subdomains to which they correspond.

ABOUT THIS BOOK

Thank you for purchasing this Take Control book. We hope you find it both useful and enjoyable to read. We welcome your comments at tc-comments@tidbits.com. Keep reading in this section to learn more about the author, the Take Control series, and the publisher.

About the Author

Glenn Fleishman is a freelance journalist based in Seattle, Wash., where he lives with his wife and son. He contributes regularly to the *Economist*, the *New York Times*, *Popular Science*, and the *Seattle Times*. He's a contributing editor for *TidBITS*. Glenn writes daily about wireless data and Wi-Fi at a set of six blogs (<http://wifinetnews.com/>). In his spare time (ha!), he develops ideas about book information through his price comparison service, isbn.nu (<http://isbn.nu/>). Glenn tries to bike to his office 3½ miles away every day, but usually only rides three times a week.



Author's Acknowledgements

I appreciate the patience of Tonya and Adam Engst and my editor Joe Kissell in the delays in my writing what appeared to be a very straightforward book. As with most issues surrounding domain names and DNS, scratch the surface and you have a long story to tell.

Huge thanks also to the crew of Take Control authors and TidBITS Irregulars among others who reviewed the manuscript and made literally thousands of helpful comments, notably John Baxter, Keith Dawson, Chris Pepper, and William Porter.

About the Publisher

TidBITS Electronic Publishing has been publishing online since 1990 when publishers Adam and Tonya Engst first created their online newsletter, *TidBITS*, about Macintosh- and Internet-related topics. *TidBITS* has been in continuous, weekly production since then.



At the TidBITS Web site you can subscribe to *TidBITS* for free, join in TidBITS Talk discussions, or search many years of news, reviews, and editorial analysis (<http://www.tidbits.com/>).



Adam and Tonya are known in the Macintosh world as writers, editors, and speakers. They are also parents to Tristan, who thinks ebooks about trains, clipper ships, and castles would be cool.

Production Credits

- Cover: Jeff Carlson, <http://www.necoffee.com/>
- Take Control logo: Jeff Tolbert, <http://jefftolbert.com/>
- Editor: Joe Kissell, <http://alt.cc/jk/>
- Editor in Chief: Tonya Engst, <http://www.tidbits.com/tonya/>
- Publisher: Adam Engst, <http://www.tidbits.com/adam/>

Thank you to Glenn and Joe for their patience with my many questions about the terminology and procedures in this book.

Thank you to John Nemerovski for keeping us going with *Philadelphia Chickens* and other treats.

And, thank you to Tristan for not requiring any childcare and even cleaning the kitchen on his own.

Take Control of Your Domain Names

ISBN: 1-933671-21-1

December 2006, Version 1.0

Copyright © 2006, Glenn Fleishman. All rights reserved.

TidBITS Electronic Publishing

50 Hickory Road

Ithaca, NY 14850 USA

<http://www.takecontrolbooks.com/>

TAKE CONTROL books help readers regain a measure of control in an oftentimes out-of-control universe. Take Control books also streamline the publication process so that information about quickly changing technical topics can be published while it's still relevant and accurate. Please send any comments to tc-comments@tidbits.com.

This ebook does not use copy protection because copy protection makes life harder for everyone. So we ask a favor of our readers. If you want to share your copy of this ebook with a friend, please do so as you would a physical book, meaning that if your friend uses it regularly, he or she should buy a copy. (Use the “Help a Friend Take Control!” button on the [cover](#) of this ebook to give your friend a discount!) Your support makes it possible for future Take Control ebooks to hit the Internet long before you'd find the same info in a printed book. Plus, if you buy the ebook, you're entitled to any free updates that become available.

Although the author and TidBITS Electronic Publishing have made a reasonable effort to ensure the accuracy of the information herein, they assume no responsibility for errors or omissions. The information in this ebook is distributed “As Is,” without warranty of any kind. Neither TidBITS Electronic Publishing nor the author shall be liable to any person or entity for any special, indirect, incidental, or consequential damages, including without limitation lost revenues or lost profits, that may result (or that are alleged to result) from the use of these materials. In other words, use this information at your own risk.

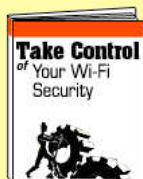
Many of the designations used to distinguish products and services are claimed as trademarks or service marks. Any trademarks, service marks, product names, or named features that appear in this title are assumed to be the property of their respective owners. All product names and services are used in an editorial fashion only, with no intention of infringement of the trademark. No such use, or the use of any trade name, is meant to convey endorsement or other affiliation with this title.

FEATURED TITLES

Now that you've seen this book, you know the Take Control books have an easy-to-read layout, clickable links if you read online, and real-world info that puts you in control. Click any book below or visit our [Web catalog](#) to add to your Take Control collection!

Take Control of Your Wi-Fi Security

by Engst & Fleishman



Learn how to keep intruders out of your wireless network and protect your sensitive communications!

\$10

Take Control of Your AirPort Network

by Glenn Fleishman



Make your AirPort network fly—get help with buying the best gear, set up, security, and more.

\$10

Take Control of Passwords in Mac OS X

by Joe Kissell

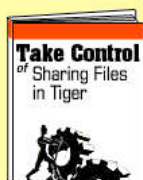


Create and manage strong passwords that keep your data safe without taxing your memory!

\$10

Take Control of Sharing Files in Tiger

by Glenn Fleishman



This detail-packed title explains how to set up Tiger to share files with Macs, Windows, and Unix machines.

\$10

Take Control of Permissions in Mac OS X

by Brian Tanaka



Solve quirky problems, increase privacy, and share files better by managing Mac OS X permissions.

\$10

More Titles!

Delve into even more topics, including:

- Running your Mac—upgrading the OS, understanding accounts, syncing, backups, fonts, and more.
- Buying gear—Macs, cameras, and digital TVs.
- More topics—.Mac, Dreamweaver, iWeb, spam, podcasting, GarageBand, Microsoft Office, and more.

now anyone can have their own domain name.



www.sallythewriter.com

(pocket-protector not necessary.)

Now you don't have to be a geek – or a big company – just to have your own Internet Domain Name. And you don't need a Web site to have a professional-sounding email address. With easyDNS, you can register your own domain name in minutes, and point it to any location on the net. Just log into our ultra-secure website and easily add e-mail addresses, or change where it points to – the control is in your hands.

From Domain Names to web forwarding to email to Dynamic DNS, all of our secure bullet-proof services come with the best customer support in the industry. Our friendly, knowledgeable and responsive customer support staff you are always available to help you along the way. Stop by easyDNS.com today and take control of your domain.

Click here & save
\$10 when you register
or transfer a
domain to easyDNS
as a new member*

easy DNS 
the way things *should* work

* or visit tidbits.easyDNS.com to redeem this coupon. Limit one coupon per customer. Offer expires Jan 1, 2008.

www.easyDNS.com

support@easyDNS.com

call toll-free: 1-888-677-4741 (North America)

0-8000-321943 (UK)