DNS on Windows 2000, 2nd Edition

Printed in the United States of America.

Published by O'Reilly & Associates, Inc., 101 Morris Street,
Sebastopol, CA 95472.

While every precaution has been taken in the preparation of this book, the publisher assumes no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

While every precaution has been taken in the preparation of this book, the publisher assumes no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

DNS on Windows 2000, 2nd Edition

Preface

You may not know much about the Domain Name System—yet—but whenever you use the Internet, you use DNS. Every time you send electronic mail or surf the Web, you rely on the Domain Name System.

You see, while you, as a human being, prefer to remember the *names* of computers, computers like to address each other by number. On an internet, that number is 32 bits long, or between zero and four billion or so.[1] That's easy for a computer to remember because computers have lots of memory ideal for storing numbers, but it isn't nearly as easy for us humans. Pick 10 phone numbers out of the phone book at random, and then try to recall them. Not easy? Now flip to the front of the book and attach random area codes to the phone numbers. That's about how difficult it would be to remember 10 arbitrary internet addresses.

This is part of the reason we need the Domain Name System. DNS handles mapping between hostnames, which we humans find convenient, and internet addresses, which computers deal with. In fact, DNS is the standard mechanism on the Internet for advertising and accessing all kinds of information about hosts, not just addresses. And DNS is used by virtually all internetworking software, including electronic mail, remote terminal programs such as *telnet*, file transfer programs such as *ftp*, and web browsers such as Netscape Navigator and Microsoft Internet Explorer.

Another important feature of DNS is that it makes host information available *all over* the Internet. Keeping information about hosts in a formatted file on a single computer helps only users on that computer. DNS provides a means of retrieving information remotely from anywhere on the network.

More than that, DNS lets you distribute the management of host information among many sites and organizations. You don't need to submit your data to some central site or periodically retrieve copies of the "master" database. You simply make sure your section, called a *zone*, is up to date on your name servers. Your name servers make your zone's data available to all the other name servers on the network.

Because the database is distributed, the system also needs to be able to locate the data you're looking for by searching a number of possible locations. The Domain Name System gives name servers the intelligence to navigate through the database and find data in any zone.

Of course, DNS does have a few problems. For example, the system allows more than one name server to store data about a zone for redundancy's sake, but inconsistencies can crop up between copies of the zone data.

The worst problem with DNS is that despite its widespread use on the Internet, there's really very little documentation about managing and maintaining it. Most administrators on the Internet make do with the documentation their vendors see fit to

[1] And, with IP Version 6, it's soon to be a whopping 128 bits long, or between zero and a 39-digit decimal number.

provide and with whatever they can glean from following the Internet mailing lists and Usenet newsgroups on the subject.

This lack of documentation means that the understanding of an enormously important internet service—one of the linchpins of today's Internet—is either handed down from administrator to administrator like a closely guarded family recipe or relearned repeatedly by isolated programmers and engineers. New zone administrators suffer through the same mistakes made by countless others.

Our aim with this book is to help remedy this situation. We realize that not all of you have the time or the desire to become DNS experts. Most of you, after all, have plenty to do besides managing your zones and name servers: system administration, network engineering, or software development. It takes an awfully big institution to devote a whole person to DNS. We'll try to give you enough information to allow you to do what you need to do, whether that's running a small zone or managing a multinational monstrosity, tending a single name server or shepherding a hundred of them. Read as much as you need to know now, and come back later if you need to know more.

DNS is a big topic—big enough to require two authors, anyway—but we've tried to present it as sensibly and understandably as possible. The first two chapters give you a good theoretical overview and enough practical information to get by, and later chapters fill in the nitty-gritty details. We provide a roadmap up front to suggest a path through the book appropriate for your job or interest.

When we talk about actual DNS software, we'll concentrate on the Microsoft DNS Server, which is a popular implementation of the DNS specs included in Windows 2000 Server (and Windows NT Server 4.0 before it). We've tried to distill our experience in managing and maintaining zones into this book (One of our zones, incidentally, was once one of the largest on the Internet, but that was a long time ago.)

We hope that this book will help you get acquainted with DNS on Windows 2000 if you're just starting out, refine your understanding if you're already familiar with it, and provide valuable insight and experience even if you know it like the back of your hand.

Versions

This book deals with name servers that run on Windows 2000 Server, particularly the Microsoft DNS Server. We will also occasionally mention other name servers that run on Windows 2000, especially ports of BIND, a popular implementation of the DNS specifications. However, if you need a book on BIND, we suggest this book's sister edition, *DNS and BIND* by Paul Albitz and Cricket Liu (O'Reilly). This book is essentially a Windows 2000 edition of *DNS and BIND.*

We use *nslookup*, a name server utility program, a great deal in our examples. The version of *nslookup* we use is the one shipped with Windows 2000 Server. Other versions of *nslookup* provide similar functionality to that in the Windows *nslookup*. We have tried to use commands common to most *nslookup*s in our examples; when this was not possible, we tried to note it.

What's New in This Edition

The first edition of this book was called *DNS on Windows NT* and dealt with Microsoft's DNS implementation for that operating system. This new edition has been comprehensively updated to document the many changes to DNS, large and small, found in Windows 2000. The most significant new feature in Windows 2000 is Active Directory, and this edition describes how Active Directory depends on DNS, including the extra DNS resource records required for a domain controller to function properly. Other new DNS features explained are dynamic update, incremental zone transfer, and storing DNS zone information in Active Directory itself rather than in a text file on disk. The new material appears throughout the book, but many features are described in a new chapter for this edition, Chapter 11. The resolver, or client side of DNS, has also changed in Windows 2000, and Chapter 6 has been updated to document the behavior of the Windows 2000 and Windows 98 resolvers.

Organization

This book is organized, more or less, to follow the evolution of a zone and its administrator. Chapter 1 and Chapter 2 discuss Domain Name System theory. Chapter 3 through Chapter 6 help you to decide whether to set up your own zones, then describe how to go about it, should you choose to. The middle chapters, Chapter 7 through Chapter 11, describe how to maintain your zones, configure hosts to use your name servers, plan for the growth of your zones, create subdomains, secure your name servers, and integrate DNS with Active Directory. The last chapters, Chapter 12 through Chapter 14, deal with common problems and troubleshooting tools.

Here's a more detailed, chapter-by-chapter breakdown:

- Chapter 1 provides a little historical perspective and discusses the problems that motivated the development of DNS, then presents an overview of DNS theory.
- Chapter 2 goes over DNS theory in more detail, including the DNS namespace, domains, and name servers. We also introduce important concepts such as name resolution and caching.
- Chapter 3 covers how to choose and acquire your DNS software if you don't already have it and what to do with it once you've got it; that is, how to figure out what your domain name should be and how to contact the organization that can delegate your domain to you.
- Chapter 4 details how to set up your first two name servers, including creating your name server database, starting up your name servers, and checking their operation.
- Chapter 5 deals with DNS's MX record, which allows administrators to specify alternate hosts to handle a given destination's mail. The chapter covers mail-routing strategies for a variety of networks and hosts, including networks with security firewalls and hosts without direct Internet connectivity.
- Chapter 6 explains how to configure a Windows resolver.

- Chapter 7 describes the periodic maintenance administrators must perform to keep their domains running smoothly, such as checking name server health and authority.
- Chapter 8 covers how to plan for the growth and evolution of your domain, including how to get big and how to plan for moves and outages.
- Chapter 9 explores the joys of becoming a parent domain. We explain when to become a parent (i.e., create subdomains), what to call your children, how to create them (!), and how to watch over them.
- Chapter 10 goes over less common name server configuration options that can help you tune your name server's operation, secure your name server, and ease administration.
- Chapter 11 describes the new bells and whistles in Microsoft's DNS implementation for Windows 2000 that weren't present in Windows NT.
- Chapter 12 shows the ins and outs of the most popular tool for doing DNS debugging, including techniques for digging obscure information out of remote name servers.
- Chapter 13 covers many common DNS problems and their solutions and then describes a number of less common, harder-to-diagnose scenarios.
- Chapter 14 ties up all the loose ends. We cover DNS wildcarding; special configurations for networks that connect to the Internet through firewalls; hosts and networks with intermittent Internet connectivity via dial-up; network name encoding; and new, experimental record types.
- Appendix A contains a byte-by-byte breakdown of the formats used in DNS queries and responses as well as a comprehensive list of the currently defined resource record types.
- Appendix B describes how to load the Microsoft DNS Server from the Windows 2000 Server CD-ROM.
- Appendix C covers migrating from an existing BIND 4 name server to the Microsoft DNS Server.
- Appendix D lists the current top-level domains in the Internet domain namespace.

Audience

This book is intended primarily for Windows 2000 system administrators who manage zones and one or more name servers, but it also includes material for network engineers, postmasters, and others. Not all the book's chapters will be equally interesting to a diverse audience, though, and you don't want to wade through 14 chapters to find the information pertinent to your job. We hope this road map will help you plot your way through the book.

System administrators setting up their first zones should read Chapter 1 and Chapter 2 for DNS theory, Chapter 3 for information on getting started and selecting a good domain name, then Chapter 4 and Chapter 5 to learn how to set up a zone for the first time. Chapter 6 explains how to configure hosts to use the new name servers. Soon after, they should read Chapter 7, which explains how to "flesh out" their implementation by setting up additional name servers and adding additional zone data. Chapter 12 and Chapter 13 describe useful troubleshooting tools and techniques.

Experienced administrators may benefit from reading Chapter 6 to learn how to configure DNS resolvers on different hosts and Chapter 7 for information on maintaining their zones. Chapter 8 contains instructions on how to plan for a zone's growth and evolution, which should be especially valuable to administrators of large zones. Chapter 9 explains parenting—creating subdomains—which is essential reading for those considering the big move. Chapter 10 covers security features of the Microsoft DNS Server, many of which may be useful for experienced administrators. The new-to-Windows 2000 features covered in Chapter 11 will be helpful to experienced administrators making the jump from Windows NT. Chapter 12 and Chapter 13 describe tools and techniques for troubleshooting, which even advanced administrators may find worth reading.

System administrators on networks without full Internet connectivity should read Chapter 5 to learn how to configure mail on such networks and Chapter 14 to learn how to set up an independent DNS infrastructure.

Network administrators not directly responsible for any zones should still read Chapter 1 and Chapter 2 for DNS theory, then Chapter 12 to learn how to use *nslookup*, plus Chapter 13 for troubleshooting tactics.

Postmasters should read Chapter 1 and Chapter 2 for DNS theory, then Chapter 5 to find out how DNS and electronic mail coexist. Chapter 12, which describes *nslookup*, will also help postmasters dig mail routing information out of the domain namespace.

Interested users can read Chapter 1 and Chapter 2 for DNS theory, and then whatever else they like!

Note that we assume you're familiar with basic Windows 2000 system administration and TCP/IP networking. We don't assume you have any other specialized knowledge, though. When we introduce a new term or concept, we'll do our best to define or explain it. Whenever possible, we'll use analogies from Windows (and from the real world) to help you understand.

Obtaining the Example Programs

The example programs in this book are available from this URL:

http://www.oreilly.com/catalog/dnswin2/

Extract the files from the archive using WinZip by typing:

```
C:\temp>
winzip dns.zip
```

If WinZip is not available on your system, get a copy from http://www.winzip.com/.

Conventions Used in This Book

We use the following font and format conventions:

Italic

Used for new terms where first defined, Registry values, domain names, filenames, and command lines when they appear in the body of a paragraph exactly as a user would type them (for example: run *dir* to list the files in a directory). *Italic* is also used for Windows commands when they are mentioned in passing and not as part of a command line (for example: to find more information on *nslookup*, a user could consult the Windows help system).

Bold

Used for menu names and for text appearing in windows and dialog boxes, such as names of fields, buttons, and menu options. For example: enter a domain name in the **Server name** field and then click the **OK** button.

`Constant width`

Used for excerpts from scripts or configuration files. For example, a snippet of Perl:

```
if ( -x /winnt/system32/dns.exe )
{
    system( /winnt/system32/dns.exe );
}
```

Sample interactive sessions showing command-line input and corresponding output are also shown in a `constant width` font, with user-supplied input in **`constant width bold`**:

```
C\>
more <\winnt\system32\drivers\etc\hosts
# Copyright (c) 1993-1999 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for
Windows.
#
```

Indicates a tip, suggestion, or general note.

Indicates a warning or caution.

How to Contact Us

Please address comments and questions concerning this book to the publisher:

O'Reilly & Associates, Inc.
101 Morris Street
Sebastopol, CA 95472
(800) 998-9938 (in the United States or Canada)
(707) 829-0515 (international/local)
(707) 829-0104 (fax)

There is a web page for this book, which lists errata, examples, and any additional information. You can access this page at:

http://www.oreilly.com/catalog/dnswin2/

To comment or ask technical questions about this book, send email to:

bookquestions@oreilly.com

For more information about books, conferences, software, Resource Centers, and the O'Reilly Network, see the O'Reilly web site at:

http://www.oreilly.com/

Quotations

The Lewis Carroll quotations that begin each chapter are from the Millennium Fulcrum Edition 2.9 of the Project Gutenberg electronic text of *Alice's Adventures in Wonderland* and *Through the Looking-Glass*. Quotations in Chapter 1, Chapter 2, Chapter 5, Chapter 6, Chapter 8, Chapter 11, and Chapter 14 come from *Alice's Adventures in Wonderland*, and those in Chapter 3, Chapter 4, Chapter 7, Chapter 9, Chapter 10, Chapter 12, and Chapter 13 come from *Through the Looking-Glass*.

Acknowledgments

The authors would like to thank their technical reviewer for this edition, Levon Esibov, as well as Jon Forrest and David Blank-Edelman, technical reviewers for *DNS on Windows NT*, for their invaluable contributions to this book. Paul Robichaux provided assistance from his wealth of Exchange knowledge for Chapter 5, and John Peterson offered helpful suggestions based on his production Windows 2000 environment.

Matt would like to thank his wife, Sonja, for her support and unflagging patience, and Cricket for asking him to help with this book. He'd also like to thank his manager at VeriSign Global Registry Services, Aristotle Balogh, for his support.

Cricket would like to thank his wife, Paige, for her support during the writing of this book. Thanks also to Walter B and Dakota and Annie, for providing occasional but much-needed relief from writing.

We would also like to thank the folks at O'Reilly & Associates for their hard work and patience. Credit is especially due to our editors, Mike Loukides and Deb Cameron.

Chapter 1. Background

> *The White Rabbit put on his spectacles. "Where shall I begin, please your Majesty?" he asked.*
>
> *"Begin at the beginning," the King said, very gravely, "and go on till you come to the end: then stop."*

It's important to know a little ARPANET history to understand the Domain Name System (DNS). DNS was developed to address particular problems on the ARPANET, and the Internet—a descendant of the ARPANET—remains its main user.

If you've been using the Internet for years, you can probably skip this chapter. If you haven't, we hope it'll give you enough background to understand what motivated the development of DNS.

1.1 A (Very) Brief History of the Internet

In the late 1960s, the U.S. Department of Defense's Advanced Research Projects Agency, ARPA (later DARPA), began funding an experimental wide area computer network that connected important research organizations in the U.S., called the *ARPANET*. The original goal of the ARPANET was to allow government contractors to share expensive or scarce computing resources. From the beginning, however, users of the ARPANET also used the network for collaboration. This collaboration ranged from sharing files and software and exchanging electronic mail—now commonplace—to joint development and research using shared remote computers.

The *TCP/IP* (Transmission Control Protocol/Internet Protocol) protocol suite was developed in the early 1980s and quickly became the standard host-networking protocol on the ARPANET. The inclusion of the protocol suite in the University of California at Berkeley's popular *BSD Unix* operating system was instrumental in democratizing internetworking. BSD Unix was virtually free to universities. This meant that internetworking—and ARPANET connectivity—were suddenly available cheaply to many more organizations than were previously attached to the ARPANET. Many of the computers being connected to the ARPANET were being connected to local area networks (LANs), too, and very shortly the other computers on the LANs were communicating via the ARPANET as well.

The network grew from a handful of hosts to tens of thousands of hosts. The original ARPANET became the backbone of a confederation of local and regional networks based on TCP/IP, called the *Internet*.

In 1988, however, DARPA decided the experiment was over. The Department of Defense began dismantling the ARPANET. Another network, funded by the National Science Foundation and called the *NSFNET*, replaced the ARPANET as the backbone of the Internet.

Even more recently, in the spring of 1995, the Internet made a transition from using the publicly-funded NSFNET as a backbone to using multiple commercial backbones, run by long-distance carriers such as MCI and Sprint, and long-time commercial internetworking players such as PSINet and UUNET.

Today, the Internet connects millions of hosts around the world. In fact, a significant proportion of the non-PC computers in the world are connected to the Internet. Some of the new commercial backbones can carry a volume of several gigabits per second, tens of thousands of times the bandwidth of the original ARPANET. Tens of millions of people use the network for communication and collaboration daily.

1.2 On the Internet and Internets

A word on "the Internet," and on "internets" in general, is in order. In print, the difference between the two seems slight: one is always capitalized, one isn't. The distinction between their meanings, however, *is* significant. The Internet, with a capital "I," refers to the network that began its life as the ARPANET and continues today as, roughly, the confederation of all TCP/IP networks directly or indirectly connected to commercial U.S. backbones. Seen up close, it's actually quite a few different networks—commercial TCP/IP backbones, corporate and U.S. government TCP/IP networks, and TCP/IP networks in other countries—interconnected by high-speed digital circuits.

A lowercase internet, on the other hand, is simply any network made up of multiple smaller networks using the same internetworking protocols. An internet (little "i") isn't necessarily connected to the Internet (big "I"), nor does it necessarily use TCP/IP as its internetworking protocol. There are isolated corporate internets, and there are Xerox XNS-based internets and DECnet-based internets.

The new term "intranet" is really just a marketing term for a TCP/IP-based "little i" internet, used to emphasize the use of technologies developed and introduced on the Internet within a company's internal corporate network. An "extranet," on the other hand, is a TCP/IP-based internet that connects partner companies, or a company to its distributors, suppliers, and customers.

1.2.1 The History of the Domain Name System

Through the 1970s, the ARPANET was a small, friendly community of a few hundred hosts. A single file, *HOSTS.TXT*, contained a name-to-address mapping for every host connected to the ARPANET. The familiar Unix host table, */etc/hosts*, was compiled from *HOSTS.TXT* (mostly by deleting fields Unix didn't use).

HOSTS.TXT was maintained by SRI's *Network Information Center* (dubbed "the NIC") and distributed from a single host, *SRI-NIC*.[2] ARPANET administrators typically emailed their changes to the NIC and periodically *ftp*ed to *SRI-NIC* and grabbed the current *HOSTS.TXT* file. Their changes were compiled into a new

[2] SRI is the former Stanford Research Institute in Menlo Park, California. SRI conducts research into many different areas, including computer networking.

HOSTS.TXT file once or twice a week. As the ARPANET grew, however, this scheme became unworkable. The size of *HOSTS.TXT* grew in proportion to the growth in the number of ARPANET hosts. Moreover, the traffic generated by the update process increased even faster: every additional host meant not only another line in *HOSTS.TXT*, but potentially another host updating from *SRI-NIC*.

When the ARPANET moved to the TCP/IP protocols, the population of the network exploded. Now there was a host of problems with *HOSTS.TXT*:

Traffic and load

> The toll on *SRI-NIC*, in terms of the network traffic and processor load involved in distributing the file, was becoming unbearable.

Name collisions

> No two hosts in *HOSTS.TXT* could have the same name. However, while the NIC could assign addresses in a way that guaranteed uniqueness, it had no authority over hostnames. There was nothing to prevent someone from adding a host with a conflicting name and breaking the whole scheme. Adding a host with the same name as a major mail hub, for example, could disrupt mail service to much of the ARPANET.

Consistency

> Maintaining consistency of the file across an expanding network became harder and harder. By the time a new *HOSTS.TXT* file could reach the farthest shores of the enlarged ARPANET, a host across the network may have changed addresses or a new host may have sprung up.

The essential problem was that the *HOSTS.TXT* mechanism didn't scale well. Ironically, the success of the ARPANET as an experiment led to the failure and obsolescence of *HOSTS.TXT*.

The ARPANET's governing bodies chartered an investigation into a successor for *HOSTS.TXT*. Their goal was to create a system that solved the problems inherent in a unified host table system. The new system should allow local administration of data, yet make that data globally available. The decentralization of administration would eliminate the single-host bottleneck and relieve the traffic problem. And local management would make the task of keeping data up-to-date much easier. It should use a hierarchical namespace to name hosts. This would ensure the uniqueness of names.

Paul Mockapetris, then of USC's Information Sciences Institute, was responsible for designing the architecture of the new system. In 1984, he released RFCs 882 and 883, which describe the Domain Name System. These RFCs were superseded by RFCs 1034 and 1035, the current specifications of the Domain Name System.[3] RFCs 1034

[3] RFCs are Request for Comments documents, part of the relatively informal procedure for introducing new technology on the Internet. RFCs are usually freely distributed and contain fairly technical descriptions of the technology, often intended for implementers.

and 1035 have since been augmented by many other RFCs, which describe potential DNS security problems, implementation problems, administrative gotchas, mechanisms for dynamically updating name servers and for securing zone data, and more.

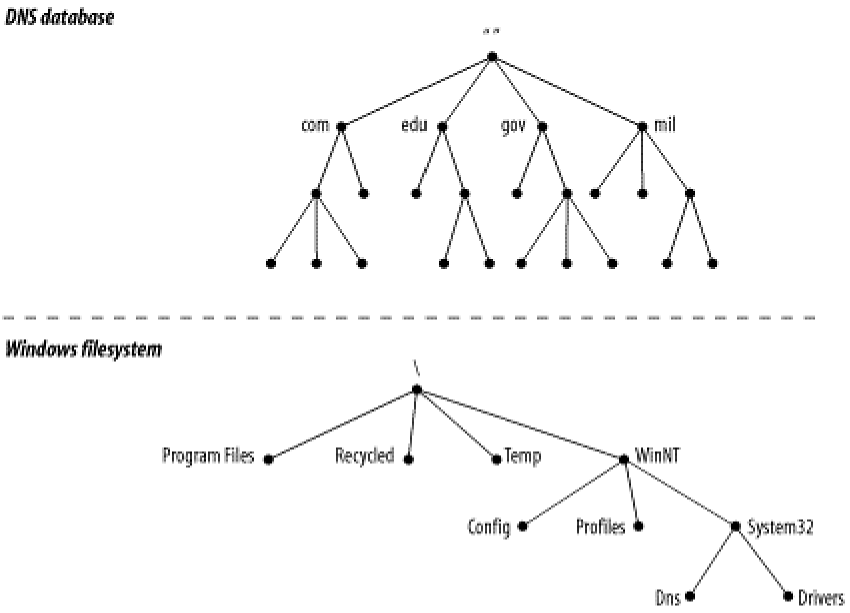
1.3 The Domain Name System, in a Nutshell

The Domain Name System is a distributed database. This structure allows local control of the segments of the overall database, yet data in each segment is available across the entire network through a client/server scheme. Robustness and adequate performance are achieved through replication and caching.

Programs called *name servers* constitute the server half of DNS's client/server mechanism. Name servers contain information about some segments of the database and make that information available to clients, called *resolvers*. Resolvers are often just library routines that create queries and send them across a network to a name server.

The structure of the DNS database, shown in Figure 1-1, is similar to the structure of the Windows filesystem. The whole database (or filesystem) is pictured as an inverted tree, with the root node at the top. Each node in the tree has a text label, which identifies the node relative to its parent. This is roughly analogous to a "relative pathname" in a filesystem, like *bin*. One label—the null label, or ""—is reserved for the root node. In text, the root node is written as a single dot (.). In the Windows filesystem, the root is written as a backslash (\ ).

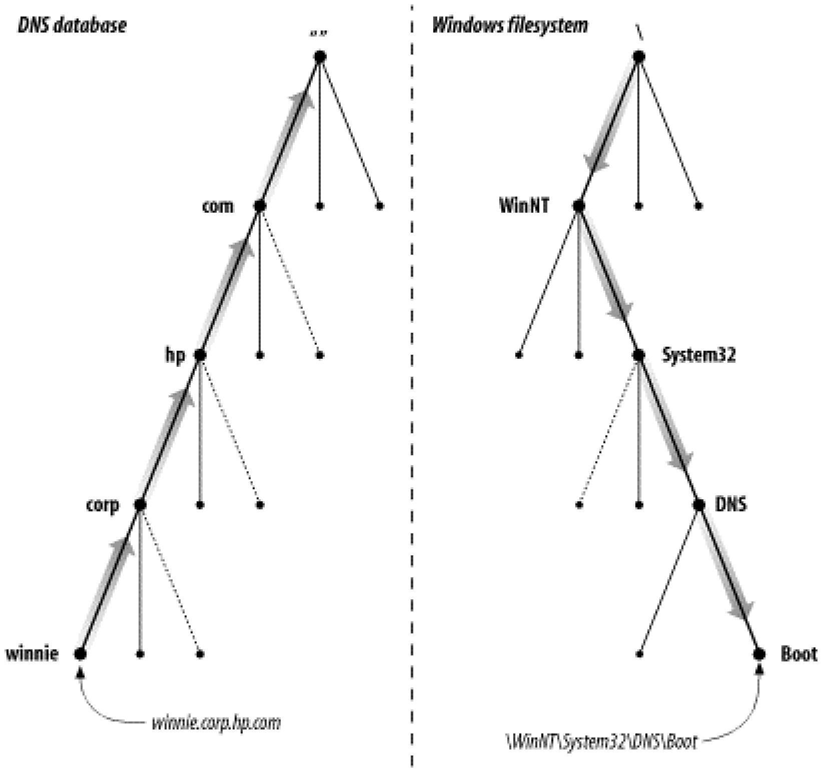
Figure 1-1. The DNS database versus a Windows filesystem



Each node is also the root of a new subtree of the overall tree. Each of these subtrees represents a partition of the overall database—a "directory" in the Windows filesystem, or a *domain* in the Domain Name System. Each domain or directory can be further divided into additional partitions, called *subdomains* in DNS, like a filesystem's "subdirectories." Subdomains, like subdirectories, are drawn as children of their parent domains.

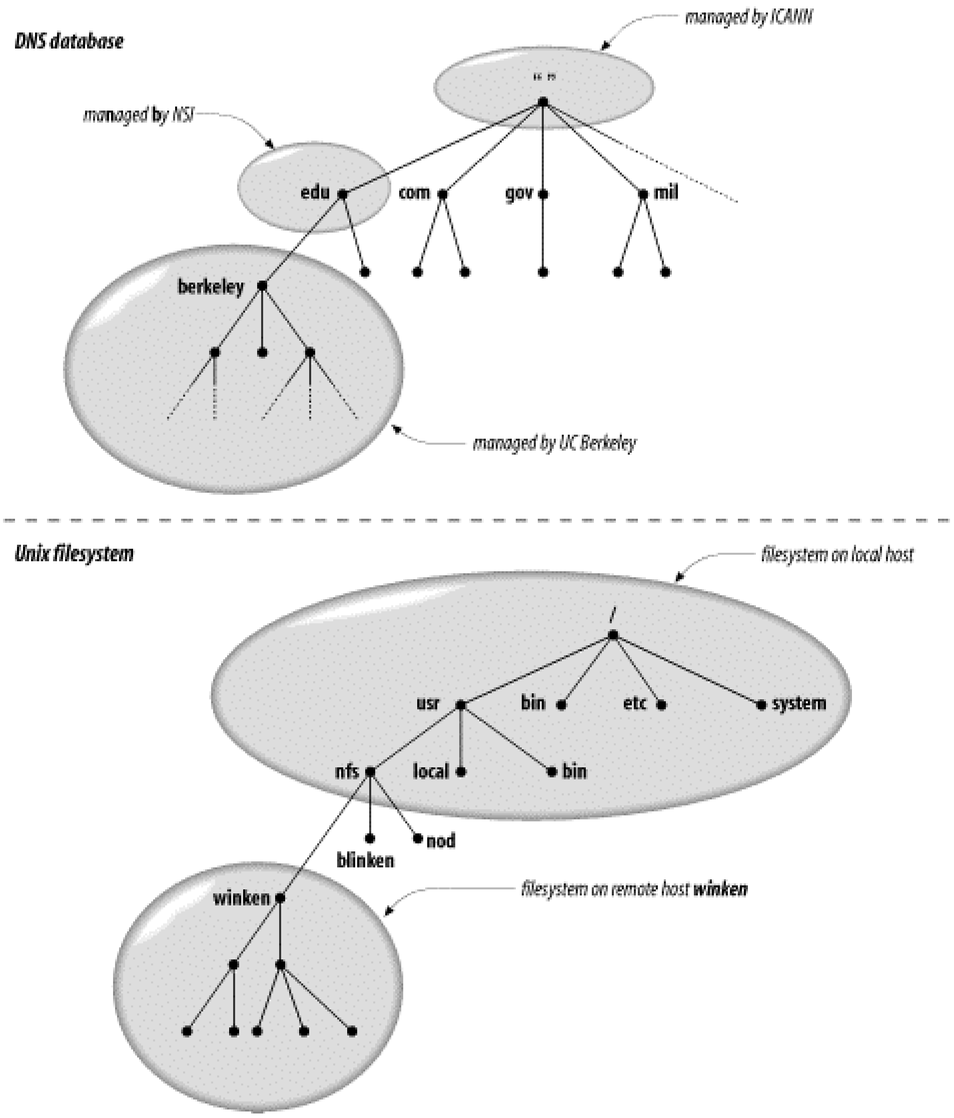
Every domain has a unique name, like every directory. A domain's *domain name* identifies its position in the database, much as a directory's "absolute pathname" specifies its place in the filesystem. In DNS, the domain name is the sequence of labels from the node at the root of the domain to the root of the whole tree, with dots (.) separating the labels. In the Windows filesystem, a directory's absolute pathname is the list of relative names read from root to leaf (the opposite direction from DNS, as shown in Figure 1-2), using a slash to separate the names.

Figure 1-2. Reading names in DNS and in a Windows filesystem



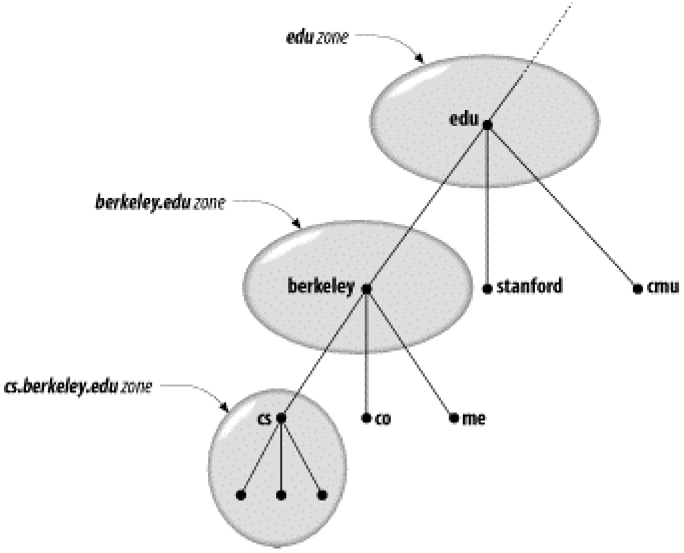
In DNS, each domain can be broken into a number of subdomains, and responsibility for those subdomains can be doled out to different organizations. For example, the InterNIC runs the *edu* (educational) domain, but delegates responsibility for the *berkeley.edu* subdomain to U.C. Berkeley (Figure 1-3). This is similar to remotely mounting a filesystem: certain directories in a filesystem may actually be filesystems on other hosts, mounted from remote hosts. The administrator on host *winken*, for example (again, Figure 1-3), is responsible for the filesystem that appears on the local host as the directory */usr/nfs/winken*.

Figure 1-3. Remote management of subdomains and of filesystems



Delegating authority for *berkeley.edu* to U.C. Berkeley creates a new *zone*, an autonomously administered piece of the namespace. The zone *berkeley.edu* is now independent from *edu*, and contains all domain names that end in *berkeley.edu*. The zone *edu*, on the other hand, contains only domain names that end in *edu* but aren't in delegated zones like *berkeley.edu*. *berkeley.edu* may be further divided into subdomains, like *cs.berkeley.edu*, and some of these subdomains may themselves be separate zones, if the *berkeley.edu* administrators delegate responsibility for them to other organizations. If *cs.berkeley.edu* is a separate zone, the *berkeley.edu* zone doesn't contain domain names that end in *cs.berkeley.edu* (Figure 1-4).

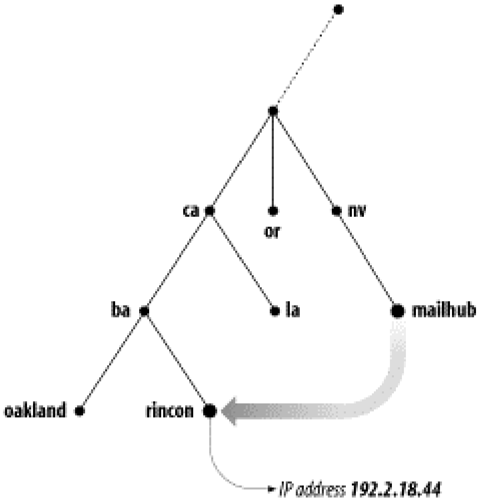
Figure 1-4. The edu, berkeley.edu, and cs.berkeley.edu zones



Domain names are used as indexes into the DNS database. You might think of data in DNS as "attached" to a domain name. In a filesystem, directories contain files and subdirectories. Likewise, domains can contain both hosts and subdomains. A domain contains those hosts and subdomains whose domain names are within the domain.

Each host on a network has a domain name, which points to information about the host (see Figure 1-5). This information may include IP addresses, information about mail routing, etc. Hosts may also have one or more *domain name aliases*, which are simply pointers from one domain name (the alias) to another (the official or *canonical* domain name). In Figure 1-5, *mailhub.nv...* is an alias for the canonical name *rincon.ba.ca....*

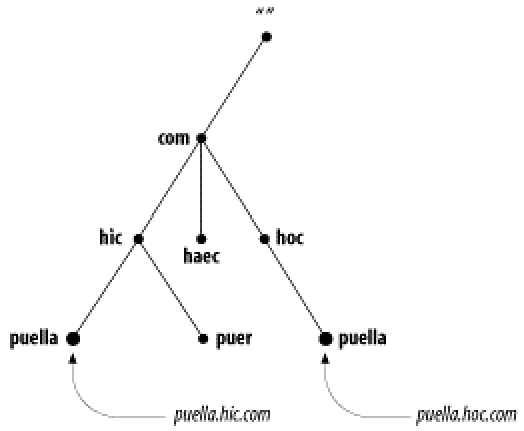
Figure 1-5. An alias in DNS pointing to a canonical name



Why all the complicated structure? To solve the problems that *HOSTS.TXT* had. For example, making domain names hierarchical eliminates the pitfall of name collisions.

Each domain has a unique domain name, so the organization that runs the domain is free to name hosts and subdomains within its domain. Whatever name they choose for a host or subdomain won't conflict with other organizations' domain names, since it will end in their unique domain name. For example, the organization that runs *hic.com* can name a host *puella* (as shown in Figure 1-6), since it knows that the host's domain name will end in *hic.com*, a unique domain name.

Figure 1-6. Solving the name collision problem



1.4 The History of the Microsoft DNS Server

The first implementation of the Domain Name System was called *JEEVES*, written by Paul Mockapetris himself. A later implementation was *BIND*, an acronym for *Berkeley Internet Name Domain*, written for Berkeley's 4.3BSD Unix operating system by Kevin Dunlap. BIND is now maintained by the Internet Software Consortium.[4]

Although the Microsoft DNS Server can read BIND's configuration and data files, it is not BIND. Microsoft wrote its server from scratch, according to the DNS specifications. The first version of the Microsoft DNS Server was a beta version that ran on NT 3.51. Microsoft made it available for some time from one of its FTP servers. The first product version of the DNS server was shipped with Microsoft Windows NT Server 4.0 (but not with NT Workstation 4.0). The server was updated in several NT Service Packs, including the latest (as of this writing), Service Pack 6a. The DNS server shipped with Windows 2000 Server comes from the same code base as the NT DNS server—it's really just a later version.

There are other name servers that run on Windows. For example, the Internet Software Consortium provides a free port of BIND 8.2.4, which runs on Windows NT and Windows 2000. Check Point offers a commercial version of the BIND 8.2.3 server. It also runs on both Windows NT and Windows 2000.

[4] For more information on the Internet Software Consortium and its work on BIND, see http://www.isc.org/bind.html.