WebMaster

Macintosh®

Bob LeVitus
Bestselling Author of
New and Improved Stupid Mac Tricks
Jeff Evans

How To Build Your Own World Wide Web Server Without Really Trying

AP PROFESSIONAL

CD-ROM INCLUDED

Contains all the shareware you need to set up your Web site, including WebSTAR™PS server software, HTML editors, graphics tools, CGIs galore, scripting additions, and other essential goodies.

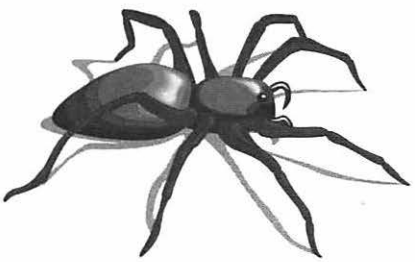
"Bob and Jeff have tamed the complexity of setting up a Mac Web server. This book and toolset will unleash a flood of cool Web sites in no time flat." -Danny Goodman, bestselling author of *Danny Goodman's AppleScript Handbook, 2/e*

WebMaster

How To Build Your Own World Wide Web Server Without Really Trying

WebMaster

Macintosh®



How To Build Your Own World Wide Web Server Without Really Trying

Bob LeVitus
Jeff Evans

AP PROFESSIONAL

Boston San Diego New York
London Sydney Tokyo Toronto

This book is printed on acid-free paper ♾

Book interior design and typesetting by Bob LeVitus,
using FrameMaker (Mac), Adobe Stone font family, Adobe Photoshop,
Mainstay Capture, Ambrosia Snapz, Microsoft Word 6.0 and 6.0.1, Thunder 7,
Adobe Illustrator, and the clip art collection with the cool EPS spider.

AP PROFESSIONAL
1300 Boylston Street, Chestnut Hill, MA 02167

An imprint of ACADEMIC PRESS, INC.
A Division of HARCOURT BRACE & COMPANY

United Kingdom Edition published by
ACADEMIC PRESS LIMITED
24–28 Oval Road, London NW1 7DX

Library of Congress Cataloging-in-Publication Data
LeVitus Bob.
WebMaster Macintosh : how to build your own world-wide web server
without really trying / Bob LeVitus and Jeff Evans.
p. cm.
Includes index.
ISBN 0-12-445574-3 (acid-free paper). --ISBN 0-12-445575-1 (CD-ROM)
1. World Wide Web (Information retrieval system). 2. Client/server
computing. I. Evans, Jeff, 1955– . II. Title.
TK5105.888.L48 1995
004.6'7--dc20 95-20137
CIP

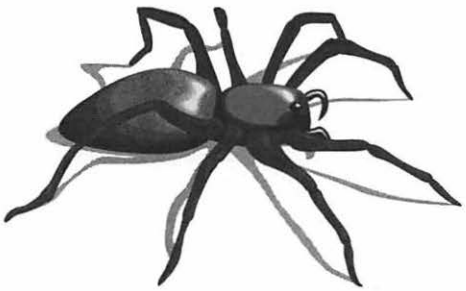
Printed in the United States of America
95 96 97 98 IP 9 8 7 6 5 4 3 2

For Allison and Jacob, my wonderful little WebSurfers-in-training

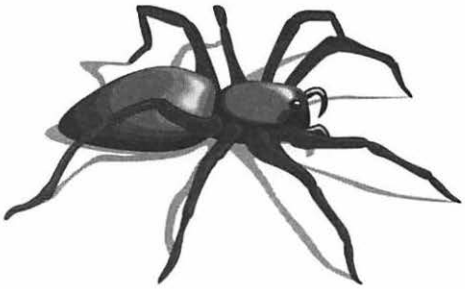
— Bob LeVitus

For my parents and sisters—for Kathy, Mikey, and our one on the way

— Jeff Evans



CONTENTS



FOREWORD

Carl de Cordova

"Even if you have a great graphic arts department, you still need some help [figuring out] what works online," says Jordan Gold, publisher of online services at Macmillan Publishing USA's digital division in Indianapolis. "Too many people treat the Net like people treated desktop publishing 10 years ago, churning out a lot of fonts and making it pretty. If your Web page looks completely different from your corporate image, it defeats the purpose."

Macmillan chose Free Range Media in Seattle to help design its ambitious home page, which runs on a Sun Microsystems SparcServer 20. Macmillan's "Information SuperLibrary" offers Net users the ability to browse Macmillan's titles, select books they wish to order by moving them into an onscreen "bag," search a database of all the bookstores in the U.S. and even listen to audio clips of *Politically Correct Bedtime Stories*.

Like most corporate "Webmasters," Gold won't disclose what Macmillan spent to go online. But he concedes, "You need to spend at least $100,000, including the hardware, if you do it on any scale at all."

This is an incredible time to be alive. The growth now happening on the Web is the birth of something monumental. As a species we only get to connect all the computers on the planet together once. The decisions we make today will impact future generations. Its crazy to think that we are going to build a new super "information highway." What's going to happen is that we are going to use what is here, now, today: we're going to use the Internet.

If you have bought this book, then it can be assumed that you are a Macintosh user. (Either that or you're a Windows guy who's reading this while standing in the book store.) As a Mac-savvy person you are in a very unique position because your skills are about to be in very great demand.

The Macintosh is the easiest to use computer in the world. The Internet is primarily based on UNIX, which is the hardest to use computer operating system in the world. It only makes sense that when you put a Mac on the Internet, you make the Internet easier to use. For the past couple of years, Macs have been by far the best client machines on the Net.

But if you're looking at this book you're not concerned about using your Mac as a client—you want to use it as a server. The nice thing about being a Mac user today is that all the pieces are falling into place to allow the Macintosh to become the dominant player in the Internet server game. You, at whatever level of Mac guru-hood, are about to be in big demand because there is a real rebellion brewing in the UNIX community.

At this writing we are starting to see the first murmuring of a backlash against the Internet. Cliff Stoll, the author of *The Cuckoo's Egg*, has just finished a book called *Silicon Snake Oil,* in which he disparages the current hype about the Internet as a great commercial resource. There have also been articles in *Forbes, Time* and *BusinessWeek* "exposing" the sham of universal wealth and prosperity that will come from the commercialization of the Internet.

It is important to keep the growth and change on the Internet in perspective. When the great, life-changing inventions have been unleashed on mankind, the development and deployment of those inventions did not proceed in a logical and linear flow.

Take the steamboat, for example. Everyone learned in school that Robert Fulton invented the steamboat in 1805. But all of the technologies needed to create a steamboat were in existence by 1750. In fact, John Fitch operated a commercial steamboat on the Hudson River for the entire summer of 1790. But then for 15 years, no commercial steamboat operated anywhere in the world.

My point here is that just because technology exists and just because it's a great idea to use the technology in a particular way, and even if the technology makes it into a real product, acceptance and general use by the public of a new idea can take a very long time.

I think the backlash we are seeing comes from two different sources: 1) the wildly optimistic and overblown claims about the immediate gains to be made from the commercialization of the Net, and 2) the great quantities of expertise and money involved in setting up a successful UNIX WWW server.

For an example of the first source, pick up any magazine on the rack. The overhype of the Net will be one of the big stories of 1995. Much of this is just the media yacking on about the next big thing.

Let's talk for a minute about the second cause of the backlash. Before New Year's Day 1995, when Chuck Shotton introduced the 2.0.2 version of MacHTTP, precursor of WebSTAR 1.0, if you wanted to put up a WWW server you had to do it on a computer running the UNIX operating system.

If you had an underutilized UNIX box laying around and the talent to make it function, then you were OK. You just downloaded HTTPd from NCSA, configured all the little text files, and edited the HTML. Most of the servers currently operating on the Net were set up in just this fashion.

But if you did not have a UNIX box, you had to get one. This proves difficult if you do not have any UNIX expertise because UNIX expertise is not cheap. Beyond just the expertise involved, the equipment needed is also not cheap. You can pay between $7,000 and $25,000 for a UNIX box and then more for software licences.

However, the most exciting news in the information age is that, if you use a Mac, you can pocket $90,000 while putting up a functionally identical server.

You can use WebSTAR on a Apple Work Group Server for under $3,000. Then you just have to run an installer and double-click on the WebSTAR application to be a site on the Net. Even at $120 an hour, the 15 minutes it takes to set up the server should not break you. Then it's just a matter of editing the HTML files and adding graphics. That's it.

Now you have a server that can be accessed by users from all over the world over 90,000 times a day without breaking a sweat.

Wow—90,000 hits. That's a bunch. Well, that's what happened to the *St. Petersburg Times* newspaper home page on a day in March when they were named the site of the day by Netscape.

What made this level of performance possible was Chuck Shotton's integration of the Thread Manager into the MacHTTP application.

I feel like a ginsu knife salesman saying this, but... But wait! There's more! What about interfacing to your favorite applications such as databases, spreadsheets, graphics, and multimedia applications? Well, thanks to AppleScript and Apple Events, making your Mac-based server interface with all manner of applications is a hell of a lot easier than trying to do the same thing on a UNIX box.

As a Mac user you already know that you have the very best library of software, bar none, of any computing platform out there. Well, with WebSTAR on a Mac, you can access any program that supports AppleScript. In fact lots of new stuff is coming out every day... new browsers, new CGI tools, new servers. The number of new tools developed in just the last six months is breathtaking. And we expect to see even more in the next six months.

I am going to go out on a limb and make a prediction. That is always a dangerous thing to do in print. But here goes. Today there are only about 30,000 Web servers on the Internet. If current trends continue, this number should top 300,000. So here is my prediction: By the summer of 1996, over half of the servers on the Net will be Macintoshes.

So now is your chance to catch this wave before it crests. This book is your best guide to taking advantage of this unique moment on the planet. The Mac makes Web servers affordable. MacHTTP makes web servers accessible. This book makes Web servers understandable.

While this book does not give you many suggestions about what to do with the $90,000 dollars you're going to save by not going with a UNIX web server, I want to give you a few:

1) Up your own salary.
2) Upgrade the Mac on your desk.
3) Get a better, faster pipe to the Net than you planned.

4) Go on a really killer vacation.
5) Hire an assistant to do your real job, and play with the server full time.

Carl de Cordova

Introduction

In the Beginning

The Authors Speak...

By now you've probably read or seen dozens if not hundreds of books about the Internet—what it's all about, how to get started, and where to go once you get there. You've spent the past few months moving up the admittedly steep learning curve of the Net, getting your SLIP or PPP account configured just so, reading books and magazine articles, and firing up your cool tools like Eudora, TurboGopher, NewsWatcher, and Netscape. (Well, OK, maybe you prefer Mosaic and are still fiddling around with different signature lines in Eudora... Still, you know what we mean.)

WebMaster Macintosh is something completely different, something new, something exciting — a book about building your own WWW (World Wide Web) server without really trying. So welcome to our book. In just a few hundred easy-to-understand pages we promise to teach you everything you need to know (and we mean *everything*) to turn your Mac into a World Wide Web site on the Internet. Without really trying. Really.

And, we'll provide the URLs of dozens of interesting Web sites you can check out for inspiration. So don't touch that dial.

Acknowledgments

Before we go any further, there are several acknowledgments we have to make. You'll be sick of hearing it by the time this book is through, but we can't say enough nice things about Chuck Shotton, the developer of MacHTTP. In a word, MacHTTP is awesome. And so's Chuck for making it.

Without MacHTTP, WebMastery would be excruciating. Painful, too. Chuck has done the near impossible and written one of those absolutely essentially unbelievable Mac programs that only comes along once in a blue moon. Really. MacHTTP is one of those amazing pieces of enabling technology — like the spreadsheet or the word processor — that opens up entire new computing vistas. It lets you do things you never thought you'd do — namely, design and run a Web site with a minimum of fuss and bother, and NO UNIX WHATSOEVER!

In a nutshell, MacHTTP is what's going to let you get your Web site up and running without really trying. So, of course, we've included the latest version — at press time — of MacHTTP on the CD that accompanies this book. You can always pick up the latest version of this most essential tool from the WebMaster Macintosh WWW server at **http://webmastermac.com.**

While we're on the subject of "thank-yous," we'd be remiss if we didn't also thank Peter Lewis, the developer of FTPd for the Mac (which you'll also find on the CD).

Thanks to all the master-blaster WebMasters who helped make this book possible, in particular, Jon Weiderspan at the University of Washington, Stephen Collins at the University of Minnesota College of Education, and Carl de Cordova at the Direct Response Center of Apple Computer in Austin, Texas. These three were instrumental in helping us learn the stuff we're about to teach. Be sure to check out their WWW and FTP sites listed in our hot lists, which are also included on the CD that came with this book.

Thanks to Apple Computer, Inc., for providing the ultimate platform and operating system for operating your own Web site.

And last, but most definitely not least, thanks to Chuck Glaser and the entire gang at AP PROFESSIONAL for giving us this opportunity. You've been an absolute pleasure to work with.

WHO THIS BOOK IS FOR

WebMaster Macintosh is for anyone who is reasonably proficient with their Mac and wants their own Web page. You don't have to be a power user. (OK, we are. Well, Bob is for sure. But you don't have to be.) You don't have to be a programmer. We're definitely not. Jeff has never been able to figure out HyperCard, let alone AppleScript, and Bob isn't that

much better. You don't have to be a big-time network administrator or UNIX geek. We're not. All you need is a little imagination, an Internet account, and this book and CD.

If you know how to turn on your Mac and launch files, just kick back, follow along, and you'll be mastering your own Web site RealSoon-Now (as they say in the software business). Trust us — we're trained professionals.

How to Use This Book

If you already have a SLIP, PPP, or dedicated Internet account and are ready to jump right in, skip forward a few chapters and do just that. It wouldn't hurt you to at least skim the earlier chapters, though. There's a lot in them; you may not know it all. We wrote this book so you'd have a great tutorial and great reference book. (It's two, two, two books in one.) The tutorials are totally original, made up by the two of us, but a lot of the reference material can also be found on the Net — somewhere. If you know where to look. We've tried to recap the best of it here to save you the trouble.

But... (and isn't there always a "but"?)

But there are a few excellent sites MacWebMasters should visit regularly; we'll tell you about them, and why you should visit often, later in this tome. When *we* started to set up our Macs as Web servers, it was a pain in the you-know-what to surf all over the place on the Internet trying to find everything we needed and wanted to know. We found it easy to get lost or distracted, and, depending on your service plan and degree of your addiction, it can get expensive surfing the ether at 14.4 baud. So, throughout this book, we'll try our best to save you the trouble and only steer you toward the best, brightest, and most worthwhile sites.

In Defense of Words on Paper...

It's embarrassing how many times we printed out documentation for one of these Internet tools, so we could read through it at our leisure, off-line and sans computer. Hypertext on screen is well and good, but you can lose your place easily when scrolling and clicking through screen after screen. In other words, there's something to be said for a good old-fashioned book like this one. (OK, old-fashioned books didn't come with CD-ROMs, but you know what we mean.)

A book is a wonderful thing. It feels good to hold. It doesn't require electricity or batteries. You can slap a Post-It™ note on a key page or passage. You can grab it off the shelf and refer to it easily, without waiting for all your Extensions and Control Panels to load. You can throw it at the dog. And it's easier to use in the bathroom than a PowerBook.

On the other hand, portions of this book can be found on the WebMaster Mac server (**http://webmaster-mac.com**, if you've forgotten) formatted as Web pages, complete with links. Let's see a paperback do that!

Bottom line: Hypertext is nifty, but books are cool, too. There's plenty of room for both in the world.

Overview (with a Point of View)

We'll start with a quick (we promise) overview of the Internet in general and the Web in particular. What makes our viewpoint unique, we think, is that it focuses on the origin and growth of the World Wide Web, rather than telling the hackneyed old "you've heard it a million times" tale of the history of the Internet (i.e., ARPA, Department of Defense, university researchers, huge growth rates, cover of *Newsweek*, blah, blah, blah). Try it — you might even enjoy it.

From the history and growth of the WWW we move on to more concrete issues such as server connectivity with your Internet provider and an overview of the Web client-server architecture. It's more nutsy-and-boltsy than the aforementioned history lesson, but it's not as boring as it sounds.

Then we'll start surfing in earnest, with a discussion about the setup and configuration of Web browsers — Netscape, MacWeb, and Mosaic to name a few — with an eye toward the differences between them that affect your site's design.

After that we delve deeper into Web page design. We've lined up some great contributions from experts on both aesthetic and information design, so even if you consider yourself artistically challenged, like Bob, you'll pick up valuable pointers here.

That's followed by an intense lesson in writing great HTML code with tips and hints from master WebMasters. We'll show you some great examples of how it's done, and some great examples of how *not* to do it.

Then there's the step-by-step walkthrough: how to set up your Web server using MacHTTP one step at a time. Don't worry; it's easier than you could possibly believe. If you can read it, you can do it. You'll be up and running in no time. In the closing stretch we'll do a quick bit of geeky stuff, like how to fine-tune MacHTTP with AppleScript, Butler SQL, and AppleSearch. These tools and apps let you add things like remote server administration, eMail, database access, and document search and retrieval functions to your Web server. Using AppleScript you can connect your Web server to any Mac application that supports AppleEvents.

Finally, we wrap it all up with a glimpse of the future, and what we think the Web might be like a few years from now.

Pay the Piper on That Shareware

Many of us have used shareware that we didn't pay for. We're all guilty of it at one time or another. Let's turn over a new leaf. In other words, we're asking you to please pay the developer if you regularly use any of the shareware that comes with this book.

We don't know of any developers making a living off shareware fees, so give 'em a break. They worked hard on that software. If you use the shareware, pay the registration fee. Please.

We know it's a hassle to write a check out and mail it. (Just think, soon you'll be able to do it all via the Web.) But for now, you'll have to do it the old-fashioned way — one envelope at a time. But please do at least consider doing it. It'll make you, as well as the developer, feel better.

OUR PROMISE

Here's our promise to you, faithful reader. This book is the easiest, most understandable, most logical, and most convenient way to get a Mac Web site up and running. We promise.

KEEPING IN TOUCH WITH US

You can stay current by accessing the WebMaster Macintosh WWW server at **http://webmastermac.com**. Relax, breathe — we'll be there with the latest, greatest versions of programs we mention in this book, all the relevant newsgroups, listserves, URLs, our monthly hot lists — you name it. If it has to do with running a better Web site, it will be there. All in one place! So stop by often.

So there you have it. Have fun, and don't forget to send us the address of your WWW site when it's up and running!

Last but not least, please let us know how you liked this book.

Peace.

Bob LeVitus
levitus@onr.com

and

Jeff Evans
jevans@outer.net

Chapter 1

THE HISTORY OF THE WORLD WIDE WEB (WWW)

What It Is, How Big It Is, and Why You Need to Know This Stuff

Though it's sometimes said that size doesn't matter, when it comes to the Internet and World Wide Web, there are some compelling reasons for knowing approximately how many denizens are out there. And although it's a moving target, it's important for you to have some sort of handle on just how big the Internet is and what part of the overall Internet traffic is Web traffic, to help you plan your own Web site. After all, just as you wouldn't want to throw a party and not know how many people might show up, you don't want to open shop on the Web without some idea of how much activity to expect.

So we'll tackle that issue in this chapter, and while we're at it, we'll take a look at the brief history of the Web as we know it.

But First, a Quick Brag about Our Home Town

We are fortunate to live in Austin, Texas. Besides being a fabulous place to live, Austin is a hotbed of Internet activity, Internet providers, and World Wide Web sites.

For what it's worth, Austin has the highest number of Macintoshes per capita in America and is the home of:

- Microelectronics and Computer Technology Corporation (MCC), where MacWeb was developed by John Hardin (and others) and home of EINet Galaxy Web Server.

- HAL Computer Services, where David Connolly is instrumental in putting together the specifications for future versions of HTML (and where you can test your own HTML; more on that later).
- Power Computing Corporation, the first licensee of the Mac OS, where Bob serves as "Director of Evangelism."
- A dozen or more Internet providers.
- A couple of Internet Special Interest groups including the CapMac Internet SIG at Apple, a rockin' good time one night a month, with Macs and high-speed Internet connections for our surfing pleasure.
- The University of Texas (Hook 'em Horns!), the single largest installation of Macintoshes in the USA and itself a hotbed of Internet activity in general and WWW in particular...
- The Iron Works, perhaps the finest BBQ restaurant on the planet.

THE COMPLETE (HA HA) HISTORY OF GROWTH

In researching the history of the Web and its growth rate in relation to the Internet itself, we turned to local experts John Quarterman and Smoot Carl-Mitchell of Texas Internet Consulting (TIC). John and Smoot may be Austinites, but they are recognized worldwide as authorities on quantifying the size and growth of the Internet.

Their work is prolific — each year they put together the TIC/MIDS Internet Demographic Study, the result of months and months of "pinging" sites and digging through Internet IP traffic reports. Their approach is both scientific and technical. Surf to: **http://www.tic.com.**

How Big Is It? (The Internet)

There has been more than a little discussion of just how big the Internet actually is. (For that matter, there's been more than a little discussion of *what* the Internet actually is, but that's a topic best tackled in someone else's book.) For now, here's one definition: The Users Services Working Group of the Internet Engineering Task Force (IETF) made one of the first "official" stabs at defining the Internet in one of its Requests for Comments (RFCs), way back in May 1993. Here's what they said:

> The Internet can be thought about in relation to its common protocols, as a physical collection of routers and circuits, as a set of shared resources, or even as an attitude about interconnecting and intercommunications. Some common definitions in the past include:
>
> - a network of networks based on the TCP/IP protocols
> - a community of people who use and develop those networks
> - a collection of resources that can be reached from those networks
>
> Today's Internet is a global resource connecting millions of users that began as an experiment over 20 years ago by the US Department of Defense. While the networks that make up the Internet are based on a standard set of protocols (a mutually agreed upon method of communication between parties), the Internet also has gateways to networks and services that are based on other protocols.
>
> — RFC 1462

By the way, if you want more than that by way of definition, we recommend Ed Krol's excellent book, *The Whole Internet User's Guide and Catalog* (O'Reilly & Associates; $29.95).

But What's It Done Lately?

What's been happening since RFC 1462 was written two years ago? Gateways to networks and services based on other protocols have been changing dramatically. Private, corporate, and commercial online networks are gaining more and better access to the full range of Internet protocols and services.

These days you have to include commercial services like CompuServe, America Online (AOL), Genie, and Delphi — and all their users — as part of the Internet. You also have to count the other networks, subnetworks, and BBSs included in John and Smoot's so-called "matrix" of networks — things like FidoNet and BitNet, and BBS systems like FirstClass and ResNova with Internet eMail, USENET, Telnet, FTP, and WWW server capability. Then you have to include the private enterprise networks of large corporations such as Exxon and Motorola that use the Internet Protocol (IP) for their internal network services and data communications. And finally, don't forget to count all the university and college students on campus-wide networks that are plugged into the Internet.

The Internet Index

The Internet Index is a lighthearted look at things Internet, inspired by "Harper's Index" and compiled by Win Treese (**treese@OpenMarket.com**). Here's Internet Index #5, reproduced by permission:

Growth of Gopher traffic in 1993: 1076%

Growth of Gopher traffic in 1994: 197%

Growth of WWW traffic in 1993: 443,931%

Growth of WWW traffic in 1994: 1713%

Growth in NSFnet traffic in 1994: 110%

Advertised network numbers in November 1993: 19,664

Advertised network numbers in November 1994: 42,883

Source: Merit Statistics

Number of Internet access providers with service in Quito, Ecuador: 1

Source: Network USA Internet Service Providers Catalog

Approximate number of countries reachable by electronic mail in 1993: 137

Approximate number of countries reachable by electronic mail in 1994: 159

Source: Larry Landweber's International Connectivity Table

Number of countries on the Internet in 1993: 60

Number of countries on the Internet in 1994: 81

Source: Larry Landweber's International Connectivity Table

Number of sites participating in "First Night in Cyberspace": 10

Source: Cyberfoo

Number of peer-reviewed scholarly journals published electronically on the, Internet: over 70

Source: Peter Fox and Jack Lancaster, "Neuroscience on the Net," Science, 11/11/94, pp 994-996

Number of attendees at Internet World, Jan. 1992: 272

Number of attendees at Internet World, Dec. 1994: over 10,000

Source: Alan Meckler

Number of "cybercafes" in the United Kingdom: 1

Source: Cyberia

Number of subscribers to the ClariNet electronic newspaper: 80,000

Source: ClariNet

Number of FBI cases officially publicized on the Internet: 1

Source: Federal Bureau of Investigation

Percentage increase in weight of the 2nd edition of the Canadian Internet Handbook: 280

Source: Jim Carroll

Number of meetings of the Internet Engineering Task Force, through 1994: 31

Source: IETF

Number of US Youth Soccer teams with web pages: 2

Source: US Soccer WWW Page (suggested by John Kinney)

Average number of days between issuing of new Internet RFCs in 1994: 2

Number of phone calls to InterNIC Registration Services Nov. 1994: 1,964

Source: November Internet Monthly Report

Number of Internet messages causing Microsoft to issue a press release denying them: 1

The Almost Last Word

And so, depending on how you define the Internet, it is generally agreed these days that the number of Internet users is somewhere between 20 and 30 million. (Though there are several advertising and PR firms that haven't figured it out yet, and are still quoting the front page *New York Times* article from the Fall of '94, which put the number of Internet users — erroneously — at about 2 million.)

No, *You* Do It!

If you don't believe us and the higher number we tout, check out D. C. Dennison and the staff of the Global Network Navigator's *GNN U-DO-It Internet Estimator.* You'll see that it agrees with us.

U-DO-It Internet Estimator

The *GNN U-Do-It Estimator* is a quick and dirty way to sketch out the size of the Internet. Follow GNN's advice and use a pencil. They take the total number of hosts on the Internet—about 3.8 million according to Mark Lottor's monthly Internet Domain Survey. Then they get conservative and subtract all the host computers behind corporate "firewall" security schemes— minus 2.5 million. Then they add back that 2.5 million since they agree with Quarterman that corporate users consume Internet bandwidth resources and can still do eMail, net news, and the like even though you and I can't get in. So we're still at 3.8 million, right? (What an exercise they make of all this...).

Next the *U-Do-It Estimator* takes Quarterman's estimate of the number of users per host machine on the Net such as friends, students, and dogs on the Net when the owner is not around of 3.8 times (the 3.8). Then you add 4–5 million or so for people who are on all the online and eMail services such as AOL, CompuServe, and MCIMail and 4–5 million for all the other types of network users that can and do use Internet resources (FidoNet, BITNET, UUCP, and the like). When your head is done spinning you end up with a number around 20–25 million. They should call it the *U-Go-Ahead-and-Do-It-For-Me Internet Estimator.*

Of course, if your computer had an early-model Pentium chip, you would probably get a different number entirely. Aren't you glad you bought a Mac?

Sorry. We couldn't resist.

Between the time we finish writing this book and it gets from the publisher to you, the definition of what it means to be a user on the Internet will become even more blurred and the number of users will be even higher. And just wait until the cable television companies figure it out and are approved to bring high-speed Internet access right to your TV.... You see what we mean. Suffice it to say that the Internet is huge and growing bigger by the minute.

HOW BIG IS IT? (THE WORLD WIDE WEB)

It's big. If the Internet is growing fast, the World Wide Web is growing as fast or faster. The generally accepted growth rate for the Internet is about 100% per year. At that rate, according to John Quarterman, everyone will be connected to the Internet by the year 2003.

> **Get Real, John...**
>
> OK, hang on. Let's get real. It's not going to continue at that rate and we won't all be connected in eight years. But, as we've already demonstrated, the Internet is big and growing fast. And the point we're about to make is that the World Wide Web is big and is growing as fast as or faster than the Internet as a whole.
>
> We'll get back to exactly how big after a brief discussion of what the Web actually is...

Brief Discourse: What Exactly Is This Web Thing, Anyway?

So what's the big deal about this here World Wide Web? What has made it so popular? We'd say it's that the Web provides a point and click, graphical interface that can take you from place to place in cyberspace. (Unless, of course, you're browsing the Web with Lynx, like an old-time UNIX geek).

The Web is easy and the Web is fun. It's also the most coherent and easily accessible way to get information on the Internet. And these days, just about everything on the Net can be viewed through the glasses of the Web. Modern Web browsers can do Gopher, WAIS, FTP, USENET, Tel-

net, electronic mail, and more. We've even seen a variation of Chat on the Web.

The Web was started by some scientists at CERN, who set out to build a big distributed hypermedia system and have succeeded in a manner beyond their wildest dreams.

CERN Who?

The more-or-less official birth of the Web was in 1989 at CERN, the European Center for Particle Physics, located near Geneva, Switzerland. From the get-go, the needs of particle physicists have made CERN a leader in networking technologies. All those physicists from around the world collaborating on complex physics and engineering and information- handling projects needed a way to share info efficiently. The Web was the result.

Most recently CERN and MIT have gotten together to help provide guidelines for the continued development of standards for the Web. Out of that has come the W3 Consortium. The W3C is an international industry-sponsored effort that has basically the same goals as CERN and MIT. Some of the early companies to join include AT&T, Digital Equipment Corporation, Enterprise Integration Technologies, FTP Software, Hummingbird Communication, IBM, IXI, MCI, Mosaic Communications, NCSA, Open Market, O'Reilly Associates, Spyglass, and Sun Microsystems.

You can find out more about the W3 Consortium at **http://www10.w3.org/hypertext/WWW/Consortium/Prospectus/.**

Hypertext Is the Key

What makes the Web so incredibly cool is hypertext and its stepsister, hypermedia (pictures, sound, movies, etc.). This hyper stuff is what makes the Web what it is.

When you're surfing the Web and you need more information on a subject you're reading about, you just click on a word or image that is a hyperlink and you're automatically transported to another page or site on the Internet that has additional information. It's kind of like footnotes with hard-wired cross-references to the actual referenced document — words, pictures, movies, or sounds. The really cool part is that the linked document can reside on the other side of the world on a different computer.

The "hyper-ness" of the Web makes the whole more than the sum of the parts. Imagine a James Michener novel on a Web site with hypertext links to all the historical references and places mentioned in each chapter. The possibilities are endless — especially as the nature of the hypermedia itself gets extended.

Hold that thought for later; we'll be talking a lot about designing good hypertext and hypermedia documents in just a few chapters. For now, the bottom line is that with the use of hypertext and hypermedia the Internet instantly became a really fun place to hang out.

With the advent of the Web, the Internet became more than eMail and file transfers and command-line interfaces. It became more than UNIX. Suddenly, using the Internet was like using a Mac. You just point and click and it works. Cool.

OK, So How Big *Is* the Web???

The reason we dwelt on the size of the Internet is to blow your mind about what's happening with the size and growth of the Web. In relation to the amount of Net activity comprised of eMail, FTP, USENET, and so on, the Web size and growth curve are awesome. In November 1994 alone, the number of WWW packets traveling over the Internet increased by 48%, or 10% per week! No other type of traffic — eMail, FTP, Gopher, WAIS, etc. — on the Internet came close to the traffic of the Web.

Growing Like a WWWeed

We asked Stephen Collins at the University of Minnesota to give us an idea of how much increase in Web traffic there was on the Net last year. Here's what he told us:

> The data I have charted is from samples taken from the NSF backbone. The NSFnet represents a portion of the main Internet backbone. Any calculation of "total" Internet traffic would be guestimates at best. However, the percentages of NSFnet traffic are probably a pretty good indication of the percentage on the Internet as a whole.
>
> You can measure traffic in terms of "bytes" or "packets" and get very different figures.
>
> For example, with the Telnet protocol, there are a large number of packets with only one character.
>
> With the FTP protocol, packets are almost always full (~1500 bytes).
>
> The dismantling of NSFnet began in the final months of 1994, so the NSF total statistics actually showed a decline. Most certainly, the numbers below do not reflect the Internet as a whole, which has seen nothing but sharp increases in traffic.
>
> All told, I think my figures and graphs from the NSF statistics for 1994 are a fairly accurate picture of the Internet.
>
> Here are some of the WWW totals from the NSF statistics:

	Bytes		Packets	
Month	**Total**	**%**	**Total**	**%**
1 Jan 93	511,450	0.00	122,440,450	0.00
1 Jan 94	822,317,950	1.49	269,129,084,100	2.61
1 Jan 95	11,518,306,800	13.12	3,382,697,720,400	17.69

I'd measure by bytes myself, which gives you an 880% increase in the percentage, and about a 1400% increase in total WWW traffic.

Stephen E. Collins
University of Minnesota
sec@boombox.micro.umn.edu

Looked at another way, at the end of 1994 there were more than 10,000 Web servers on the Internet according to Matthew Gray's World Wide Web Wanderer; two years before that there were probably fewer than 500. (There'll be at least 10,001 once yours is up and running.)

Since Web browsing software such as Mosaic, Netscape, and MacWeb were introduced and distributed freely, millions and millions of these and other popular "browsers" have been downloaded or distributed. That means millions and millions of users surfing the Web. Doug Colbeth, the president of the commercial software company Spyglass (the master licensee of Mosaic) said last year that they alone had licensed 10 million copies of Mosaic to more than 20 commercial concerns.

It's much easier to get a handle on the size of Web traffic on the Net in relation to the total amount of Net traffic and by means of a graph. Measuring Net traffic by protocol type — FTP (file transfers), SMTP (mail), NNTP (news) — is easy if you have the time. Just go to **gopher://nic.merit.edu** and get buried in the latest numbers and make some graphs like Stephen Collins did (Figures 1.1 to 1.5).

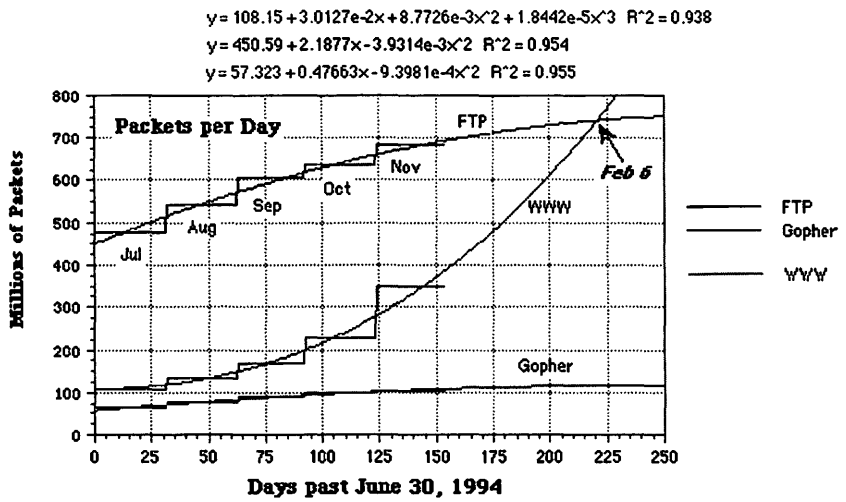


Figure 1.1: Average Daily IP Packet Count.
Source: Stephen Collins, University of Minnesota, December 1994.

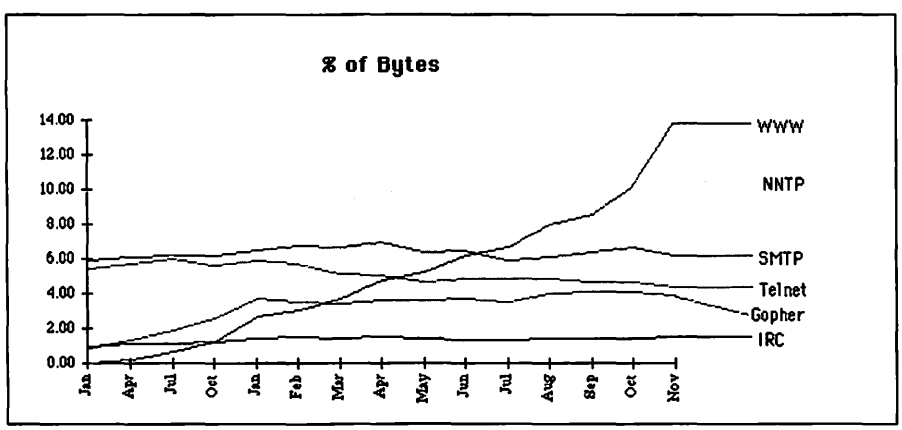


Figure 1.2: Total IP Traffic by Type in Bytes.
Source: Stephen Collins, University of Minnesota, December 1994.

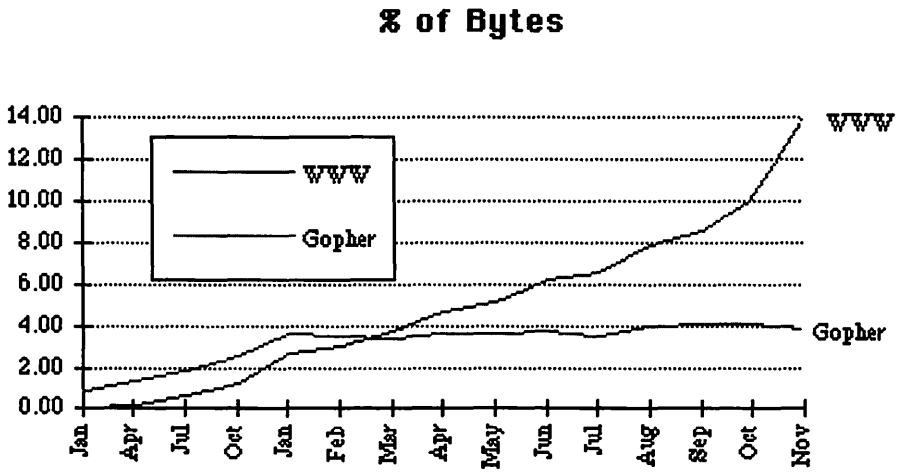


Figure 1.3: WWW vs. Gopher in Bytes.
Source: Stephen Collins, University of Minnesota, December 1994.

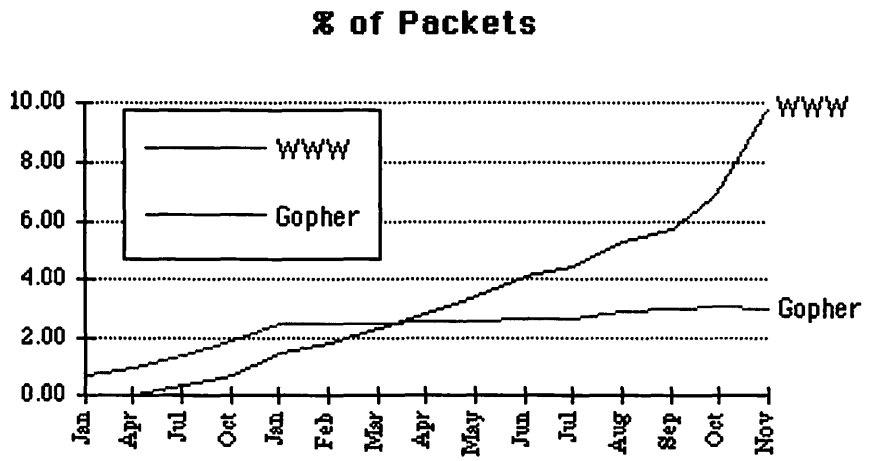


Figure 1.4: WWW vs. Gopher in Packets.
Source: Stephen Collins, University of Minnesota, December 1994.

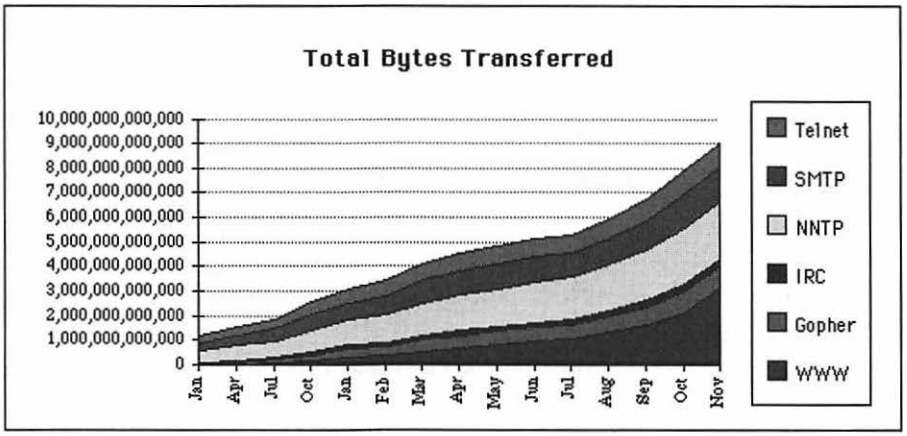


Figure 1.5: Total Bytes Transferred in Absolute Numbers.
Source: Stephen Collins, University of Minnesota, December 1994.

SUMMARY AND CONCLUSION

That wasn't so bad, was it? Well that's it. You're done with the history lesson. You got the idea, right? It's big. And a lot of people are surfing it.

If you need more on the size and growth of the Internet and WWW then take a cruise to the World Wide Web Wanderer Web site at **http://www.mit.edu:8001/people/mkgray/web-growth.html.** You can also check out the CyberWeb Virtual Library sponsored by the Maryland-based Internet provider, CharmNet. The URL is **http://www.charm.net/~web/Vlib/Misc/Statistics.html.** If you're really twisted and just can't live without the raw data you can find it at **gopher://nic.merit.edu:7043/1/nsfnet/statistics.**

The whole history of the Net and WWW is fascinating. People do nothing but study and collect data on it every day. Not us — and probably not you. It's there if you want it and you can use it to help justify or build a marketing story for what's the most exciting thing happening on the Net — setting up and running a WWW server.

In the next chapter you're going to get your feet wet. We're going to review the basic system requirements you need to have in place and take a look at the connectivity side of things. You'll learn what to ask for from your Internet provider and you'll pick up some tips on managing the relationship. You're also going to learn what the whole client-server story is all about.

(By the way, it's not the big mystery you thought it was. We'll take a break at the end of that chapter and take a look at the Web browsers and review some of the setups and configurations for Netscape, MacWeb, and the "Helper Apps"—Sparkle, Sound Machine, StuffIt Expander, JPEG Viewer, and NCSA Telnet.)

Sound like a plan? OK, then kick back, relax, and turn the page.

Chapter 2

GETTING YOUR FEET WET

Connections, Providers, Speed, TCP/IP, a (Brief) Client-Server Story, and a Hands-On Tutorial

This chapter is about Internet providers, setting up your connection with an Internet provider, and your speed options for connecting. Along the way we'll tell you a little about network protocols and the client-server story and a bit about the UNIX operating system and what makes UNIX lovers tick.

Then (and only then) we'll start our hands-on festivities with a delightful little ditty we call the MacHTTP Quick Start.

TO SURF AND PROVIDE: THE SERVICE PROVIDER STORY

If you're going to be a WebMaster, the most important person in your life is your Internet service provider. This section will tell you a little about the species, but in a nutshell, here's our advice: Cherish him and treat him right.

> **On Dealing with UNIX and UNIX Geeks**
>
> UNIX is the indecipherable multi-user, multi-tasking, multi-platform operating system that's more or less the standard for the Internet. Most service providers use high-powered computer workstations running the UNIX operating system.
>
> Traditionally, Macintosh users have sneered at UNIX users and vice versa. "UNIX is impossible for mere mortals to use" vs. "The Mac is just a toy." But given the fact

that many of the machines on the Net are running UNIX, it's obvious that we Mac enthusiasts must learn to get along with the UNIX community.

Just between us, the Mac interface is better than UNIX's cryptic commands. And Macs are a zillion times easier to use. The truth is, there's almost nothing you can do with a UNIX box that you can't do just as well or better on a Mac. But you'll never convince a dyed-in-the-wool UNIX geek of that. And chances are good that your service provider is dyed-in-the-wool.

So if your service provider is comfortable with his UNIX command line, don't try to convince him he should be running his site on a Mac. Trust us, you can't.

Follow this simple rule and you'll have a long, mutually beneficial relationship with your provider; disobey it and you'll have a screaming match which neither of you has the slightest possibility of winning.

(Note: Feel free to change "he" to "she.")

The Early Days When Life Was Simple (Not!)

Let's say it right up front: TCP/IP and UNIX are ugly. They're decidedly un-Maclike. In the early days, we mostly got our Internet accounts from UNIX geeks who didn't get it. They figured that us "little computer" users would know how to configure TCP/IP. Wrong! They thought we would be able to write our own modem scripts for our SLIP or PPP software. Wrong again. Many of them didn't even have a Mac in house! It wasn't long before they were swamped with support calls from Mac and PC users.

Things are better these days. Most Internet providers give you a nice, customized Mac installer disk with everything preconfigured for your SLIP or PPP dial-up connection. Just double-click and the installer sets all the IP addressing information in the MacTCP control panel auto-

matically; it prompts you to choose your modem brand and type from a pop-up menu and enter your user ID and password. Then, just click the Connect button and away you surf!

OK, not every Internet provider is that Mac savvy. But things today are much better. Our advice: If your provider can't at least preconfigure your MacTCP and SLIP or PPP software for you, think about finding a different provider.

Your Internet Provider Is Your Friend

Have you ever been over to your Internet provider's offices and seen their hardware? Have you ever met them in person? If not, you should. Have them show you their setup and explain which box does what. Get to know the people who work there. Take a UNIX geek to lunch. Trust us, it'll be worth it someday when you've got major problems on your end.

All kidding aside, we can't stress enough how important it is to be on good terms with your Internet provider or network administrator. There is much they can do to make setting up and running your Web site go even easier; you'll be depending on them for configuring MacTCP, getting a permanent address on the Internet, and maintaining the physical connection between your site and theirs. So be nice.

Believe us, as soon as your network or ISDN connection goes down, you're going to be calling the expert for help. The whole thing is so much easier if that expert is also your friend. We can't begin to emphasize this enough.

A Typical Internet Site

Here's the setup for one of the Internet providers we know in Austin. The machine names and IP numbers have been changed to protect the innocent.

It's not that complicated really. Just gaze at Figure 2.1 and in your mind, add about a ton of wires running all over the place and some guys who seem a little out-there and bug-eyed wild wearing jeans and T-shirts who haven't slept or combed their hair in days because they are setting up so many new accounts. Nice picture, eh?

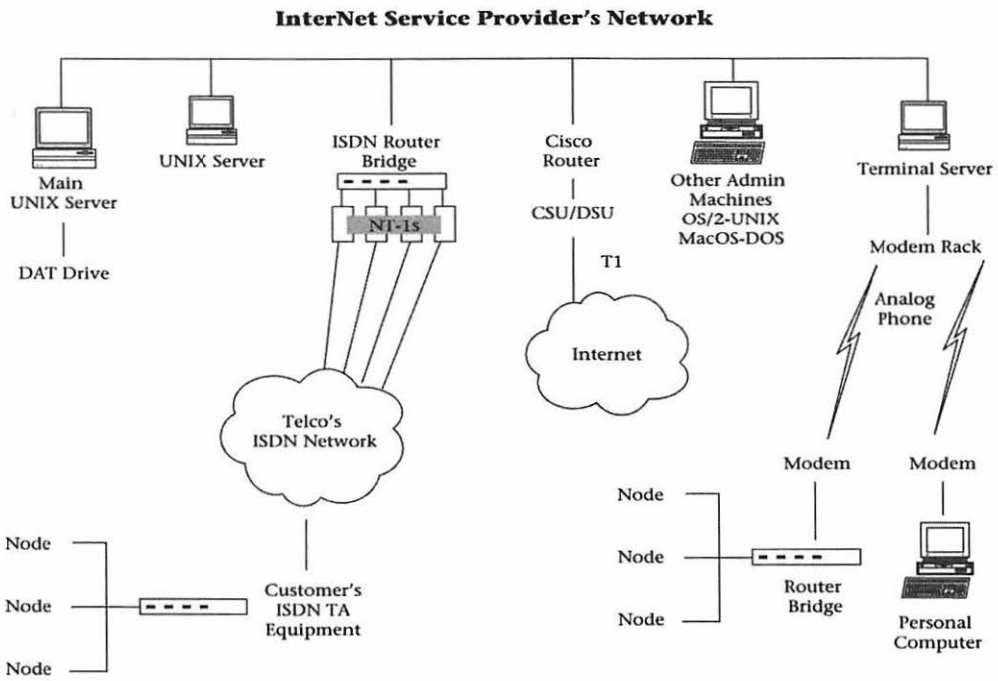


Figure 2.1: Internet Site Layout, Tomorrow's Technologies, Austin, TX, January 1995. © 1995 Tomorrow's Technologies, Inc.

Getting What You Need from Your Internet Provider to Get Started

There are two things you need to get from your Internet provider or network administrator to get your Web page up and running. The first is a dedicated, permanent connection to the Internet; the second is a permanent address on the Internet itself. If you haven't got both, these are two items you must have if you want to run a real full-time Web site. Realistically it takes two to three weeks to get this stuff in place, so read what follows, then contact your Internet provider and get started ASAP.

If you're already connected to the Net full time and have a permanent Internet address, skim the next couple of pages (but don't miss the jokes and secret passwords).

INEXPENSIVE CONNECTION CHOICES

Like we said, if you are serious about setting up a Web site you are going to need to get some sort of dedicated 24-hour connection with your Internet provider. Unless you're doing an in-house Web site for your company and only need the site to be available during specific hours, it's kind of bogus to set up a Web site, announce it to the world, and then not have it available 24 hours a day.

There are a couple of ways to go, and several things to keep in mind when you're deciding what type of connection to get. Basically your connection choices are as follows:

- An analog line using a 14.4 or 28.8 baud modem

  or

- A digital line such as a 56K dedicated line, Frame Relay, ISDN, or a T-1 (faster than that and you're really talking big money).

The cost for a dedicated connection will range from $75 a month to more than $5000 per month, depending on which type of line you choose. Prices keep dropping, so ask your Internet provider for the latest rates.

We've included a list of Internet providers by major metropolitan area as an appendix. Better still, you can usually get a current list off the Net itself by posting to a USENET newsgroup for the city you live in; in our case we'd ask in "Austin.general." Post a message something like this:

> I'm looking for a local Internet provider who offers ISDN [or whatever type of connection you want] at reasonable rates. I'd appreciate any recommendations. Thanks.

The type of connectivity you go with depends on your need for speed, your budget, and how many "hits" a day you anticipate your site is going to get. We'll talk about your options in depth in a moment.

Keep in mind that many people hitting your site will be using a 14.4 or 28.8 modem connection at their end. If there are a lot of people hitting your site simultaneously or if you are serving up large images, you're going to want a faster connection to handle the traffic and maintain performance. Choose a slow connection and people will come to your site, wait around too long for some text or image to load, say "later," and surf away.

It's easy just to click away from a slow site — easier than flipping channels, watching six shows at once, and driving your wife crazy. Tuck this little tidbit away in the back of your mind for now; we'll talk more about it later.

Anyway, the need for speed is something that's hard to predict. We recommend you start out modestly. You can always add bandwidth later. On the other hand, let's get real and stop beating around the bush: Get as much bandwidth as you can possibly afford. Like RAM and hard-disk space, more is always better.

Dedicated Modem

We recommend starting out with a dedicated 28.8 modem connection. Check with your Internet provider to see if they offer it; almost all do these days. You'll need a V.32bis modem, a phone line, and a dedicated SLIP or PPP account with your service provider.

This type of connection is typically the least expensive. Bearing in mind that prices are always changing (usually dropping), V.32bis modems are going for around $200 as we write this in early 1995. Most Internet providers are offering dedicated 28.8 connections for under $100 a month. By the way, 28.8 is probably as fast as it's going to get over analog phone lines. For higher performance you'll need to move to digital lines such as ISDN or T-1 (explained in the next section).

To sum up the costs: a V.32bis modem, a standard analog phone line that you don't use for anything else, plus setup and monthly fees you'll pay to your Internet provider. All in all, a dedicated 28.8 connection should cost you a couple of hundred dollars to get up and running, and a couple of hundred a month to maintain.

The Great Debate: SLIP vs. PPP

We recommend that if you are going the dedicated 28.8 modem route, that you stay away from SLIP (Serial Line Interface Protocol) and look for a provider who offers PPP (Point to Point Protocol). You don't need to know much about them but here's what you *do* need to know.

PPP allows multiple protocols concurrently over a dial-up link. You want that. We prefer PPP over SLIP because PPP gives you more flexibility, is more robust, and has more new development being done for it than SLIP. In fact, we'd venture to say, SLIP is becoming less prevalent and is usually more of a nuisance to configure.

So our advice if you don't already have a satisfactory connection is to find a provider who offers dedicated PPP connections for V.32bis modems.

The Importance of a Permanent Address

Macintosh SLIP and PPP implementations (MacSLIP, InterSLIP, and MacPPP) provide equivalent access to the Internet by allowing MacTCP to be used with a modem to turn your Mac into a machine that is "on the Net."

When you connect to go surfing the Net with a SLIP or PPP connection, you receive an "IP address" assigned by the UNIX box at your Internet provider. That address is in effect for the duration of your dial-up session. (Really. You can check it out any time by opening the MacTCP Control Panel during your session and seeing what IP address has been assigned.) Each time you dialed in you may have received a different IP address.

Since in the past you were *surfing* the Net and not serving a Web page on the Net, it didn't really matter whether or not you had a permanent IP address. But now that you are setting up a Web site, you want to have a permanent IP address so people can find you! Your service provider will assign you a permanent address. Just ask.

Two Reasons You Might Park Your Page on Someone Else's Server

Reason 1: If you're short of funds, you may be able to park your Web pages on somebody else's Web server — usually your Internet provider's UNIX box — for a lot less money than any of the solutions in this chapter.

But what fun is that? Part of the joy of WebMastery is to have it there on your desk, to fiddle with endlessly until you get it right. Still, this may be the most practical thing to do until you can afford a full-time connection of your own. Talk to your provider about your options.

Reason 2: At the other end of the spectrum is the problem of having so much traffic on your site that your Mac is overloaded or bandwidth becomes the bottleneck. We should all be so lucky. This means you've become successful! It also means you may have to move your Web pages on to a faster UNIX box and turn the WebMaster role over to a UNIX systems administrator to handle. (Or, of course, learn UNIX yourself...)

For instance, the Apple site at **http://www.apple.com** receives 300,000–400,000 hits a week and sits on a UNIX host computer with a dedicated T-1 connection. There are times when that machine receives 200 simultaneous hits per second! The fastest Power Macintosh can handle that load, but poor old MacTCP just can't keep up.

For what it's worth, the Penthouse WWW site gets 800,000 hits a day. The URL is **http://www.penthousemag.com**. Sex sells on the Net too.

(Of course, when Open Transport replaces MacTCP and the Mac OS supports true multi-tasking and several other UNIX-like features (code name: Copeland; available in 1996) you should be able to handle the big-time traffic.)

In the meantime, we hope this is a problem you encounter. It means you're a huge success and can pay a team of UNIX geeks to do it for you while you clip zero-coupon bonds.

OK. Now that that's out of the way, let's look at those aforementioned faster (and more expensive) connections.

ISDN AND OTHER DIGITAL CONNECTIONS

There are a couple of ways to go if you decide to go for a dedicated digital connection instead of the analog modem route. They all work pretty much the same way as far as you are concerned: You've got a Combinet box or router hanging off your network connected to a terminal box. The terminal box in turn is connected to the line from your local phone. Your Internet provider's side of things is set up the same way in reverse. Sometimes they have a lot of money and a setup that can handle multiple types of digital and analog connections from different types of customers.

If you ever want to see every type of router, terminal server, and type of network and digital connectivity on Earth, go to the networking event of the year — Networld+Interop. The big one is held in the Spring in Vegas and is a blast. You'll get an unbelievable dose of media, protocols, Internet, infrastructure, network software, wireless/mobile, multimedia, and carrier/telecommunications. The show is one of the first to have a virtual trade show on the Web at **http://www.interop.com**.

Whether you get a 56K line, ISDN, Frame Relay, fractional T-1, or T-1 (don't worry — all will be explained soon) is a matter of budget, your Internet provider's offerings, and your local phone company's technology. In some parts of the country the local phone companies are pushing Frame Relay, in others it's flat-rate ISDN; it just depends on where you live.

We like ISDN. OK, we're biased. That's what we have. Southwestern Bell has a great deal on ISDN in Austin — installation is cheap (under $100) and we pay a flat rate of $50 a month. Your mileage may vary.

Because of its speed, reliability, and cost-effectiveness, many Internet providers are strongly committed to ISDN as the preferred method of connection for customers. In addition, the global Internet community is moving toward a standardized entry-level topology, and ISDN is rapidly emerging as the most universally accepted candidate.

Almost all Internet providers offer ISDN. There's a reason. ISDN operates at 64Kbps or 128Kbps, two to four times faster than a 28.8 modem's highest theoretical connect speed. ISDN is digital, therefore its bandwidth is guaranteed and error-free. Finally, it's relatively easy to set up for both Internet providers and telephone companies.

If you decide on ISDN (or any other type of high-speed digital connection for that matter), get the advice of your Internet provider and telephone company before you do anything. Ask them what you need and how they recommend you go about getting it. You don't even need to know what ISDN stands for. ("Integrated Services Digital Network." There. Did that make you feel better? For what it's worth, many netizens say it stands for "I Still Don't Know.")

ISDN is nothing more than a digital telephone line. You get one by placing an order with your local phone company, assuming they offer ISDN service. Not all do. Yet. Be aware that there are different types of ISDN, ranging from ISDN on demand to bonded ISDN.

The one you care about is Basic Rate Interface ISDN or BRI. BRI is comprised of two "bearer" (B) 64K channels for user data plus one "data-link" (D) 16K channel for control and signaling information. Many Internet providers just set you up on one B channel, which gives you a 64Kbps connection to the Net — about two to four times faster than a 28.8 modem.

Here's where it gets interesting: You can "bond" the other 64K channel with the first to make a 128K connection. And you can just keep on bonding away with ISDN and end up with a PRI (Primary Rate Interface). PRI provides 23 data channels of 64Kbps each operating simultaneously and a 64Kbps signaling channel for communicating over an ISDN Primary Rate Service. By providing digital communication over twenty-three 64Kbps channels, many individual basic rate ISDN channels can be

serviced over a single telephone company connection line. In North America and Japan, PRI consists of 24 channels, usually divided into 23 B channels and one D channel, and runs over the same physical interface as T-1. Elsewhere it's 30 B channels and one D channel. Ask your Internet provider and your phone company; all you need is cash....

Monthly rates for a 64K dedicated ISDN connection vary widely. Just like the example of a dedicated modem connection, you're going to pay for both the phone line — in this case it's a digital one — and for the connection to your Internet provider. Prices on ISDN and other types of digital connections vary widely depending on your location. There are promotional rates, flat rates, measured rates, "on-demand" rates, rates by byte usage, rates, rates, and more rates. You have to be a bean counter to compare offerings and rate packages and figure it all out. Look for something that's simple to understand, and as close to a flat rate as possible — from both phone company and service provider. Figure it's going to run you somewhere between $250 to $500 a month for a 64K ISDN connection by the time you're done.

Ask your Internet provider how to get your ISDN line installed. They'll probably know a contact person at the local phone company with whom they have a good relationship or at least took to lunch once.

One last thing: When you're going the ISDN route, it may be better to determine a budget, then leave it up to your Internet provider to tell you what to buy equipment-wise, and what speed connection you can get with that budget.

Typical Monthly Dedicated Rates

The following pages detail the April 1995 rates for UUNET (see coupon at the back of this book). You may not get as good a deal in your neck of the woods; you may do better. As always, your mileage may vary.

UUNET PRICE LIST

AlterDial® Dial-up SLIP or PPP Internet Access via High-Speed Modems

Metered Client Cost (3-month commitment required)
$30/month basic service fee, including 15 hours of local usage, Internet mail, and USENET news for one user
$2.25/hour connection cost to local hub (after first 15 hours),
$6/hour surcharge to 800 number (at all times)
$10/month optional charge for each additional POP e-mail account
$25 one-time start-up fee

Metered LAN Cost
$49/month basic service fee
$3/hour connection cost to local hub, $9/hour to 800 number
$10/month optional charge each for Internet mail and USENET news via UUCP/TCP
$10/month optional charge for each POP account or $20 per concurrent NNTP session
$499 one-time start-up fee

ISDN Workgroup(SM) 64 Kbps or 128 Kbps Internet Service

For all standard features, including unlimited connect time:

	1 B Channel	2 B Channels
Startup charge	$395	$495
Standard monthly charge	$295	$495
Monthly charge with 12-month commitment	$280	$475
Monthly charge with 24-month commitment	$265	$450

For additional options:
POP accounts: $10 per month per mailbox
NNTP sessions: $10 per month per simultaneous session.

Equipment: Ascend Pipeline 50 ISDN Router: $1050 (when purchased with service)

56K Leased Line Internet Access

Cost: $795 one-time start-up fee
Monthly service fee: $695
Monthly service fee with 1-year term commitment:[1] $645
Monthly service fee with 2-year term commitment:[1] $595
$100/month optional charge for a USENET news feed and/or Internet mail via UUCP over TCP. There is no extra charge for Internet mail via SMTP.

[1]At the conclusion of the term, pricing will revert to the standard rates in effect. Any customer wishing to cancel service before the completion of the term will be required to pay 75% of the monthly charges for the months remaining on the contract.

56 Kbps Frame Relay Internet Access

$495 one-time start-up fee
Monthly service fee: $595
Monthly service fee with 1-year term commitment:[1] $545
Monthly service fee with 2-year term commitment:[1] $495
$100/month optional charge for a USENET news feed and/or Internet mail via UUCP over TCP. There is no extra charge for Internet mail via SMTP.

[1]At the conclusion of the term, pricing will revert to the standard rates in effect. Any customer wishing to cancel service before the completion of the term will be required to pay 75% of the monthly charges for the months remaining on the contract.

Fractional T-1 (128 Kbps) Frame Relay Internet Access*

$3,000 one-time start-up fee
Monthly service fee: $995
Monthly service fee with 1-year term commitment:[1] $895

[1]At the conclusion of the term, pricing will revert to the standard rates in effect. Any customer wishing to cancel service before the

completion of the term will be required to pay 75% of the monthly charges for the months remaining on the contract.
*Available in selected service areas — please call for details.

T-1 Plus(SM)

$5,000 one-time start-up fee

T-1 Usage Level	Burstable Service Monthly Rate	Fractional Service Monthly Rate
0 to 128 Kbps	$1,250	$1,100
128 Kbps to 256 Kbps	$1,750	$1,600
256 Kbps to 384 Kbps	$2,200	$2,050
384 Kbps to 512 Kbps	$2,500	$2,350
Over 512 Kbps	$3,000	N/A

T-1 Term Commitment: Discounted monthly rates with a 1-, 2-, or 3-year commitment. For a 1-year term, a 5% discount; for a 2-year term, a 10% discount; for a 3-year term, a 15% discount.

T-1 Price Protected: $5,000 startup; $2,200 monthly. This product "locks" the price of a full T-1 into a rate of $2,200 for one year. It is intended for thos organizations preferring nonvariable rates and is not based on bandwith utilization.

Optional 56K and T-1 equipment (available only with service):

Cisco 2501 router and 56K CSU/DSU: $1,895	
Kentrox T-1 CSU/DSU:	$1,325
Cisco 2501 router:	$1,595
ATL 56K CSU/DSU:	$365
Ascend 56K Pipeline LS56 with internal CSU/DSU:	$1,325
Imatek 56K router with internal CSU/DSU:	$995

10Plus®

$5,000 one-time start-up fee

Usage Level*	Monthly Usage Charge
0 to 128 Kbps	$1,500
128 Kbps to 256 Kbps	$2,000
256 Kbps to 512 Kbps	$2,500
Over 512 Kbps	$3,000

* Monthly charges based on typical usage level during the month. Please call for availability and detailed pricing.
Line costs are not included in any service costs; please contact AlterNet Sales for pricing.

UUNET's Web Server Hosting Services

Standard (server connected via T-1 link)
One-time start-up fee: $400; Monthly charge (1st month): $300

Monthly charge, after first month:

MB of data retrieved	Monthly cost	Approximate maximum (per day) number of inquiries*
0–250	$300	2,000
251–500	$400	4,000
501–1000	$500	8,000
1001–1500	$600	12,000
1501–2500	$750	20,000

Premium (server connected via 10 Mbps link)
One-time start-up fee: $850; Monthly charge (1st mo.): $900

Monthly charge, after first month:

MB of data retrieved	Monthly cost	Approximate maximum (per day) number of inquiries*
0–2,500	$900	20,000
2,501–5,000	$1,200	40,000
5,001–10,000	$1,800	80,000
10,001–15,000	$2,300	125,000
15,001–20,000	$2,800	150,000
20,001–30,000	$3,600	250,000
30,001–40,000	$4,400	300,000
40,001 and up	$5,400	500,000 and up

* Approximate maximum number of inquiries per day based on a typical mix of text and graphical information on the server. Pages with extensive graphic or multimedia content will consume more bandwidth per inquiry. Please call for details.

UUNET currently offers three security products. The first of these products is the TIS Gauntlet Internet Firewall, the premier price-performance Internet firewall available today. This device uses advanced application proxy technology—the best technology available to protect customer networks from other networks over which they have no control. The second is LanGuardian, an advanced selective IP encryption device. LanGuardian can encrypt data between a range of IP addresses, while allowing data to other IP addresses to pass unencrypted; thus, LanGuardian users can exchange encrypted traffic with other corporate sites (or strategic business partners) while retaining access where desired to other Internet information resources. LanGuardian supports both single-DES and triple-DES encryption for maximum data security. The third service is security consulting. In particular, it is critically important that users of security products have a written security policy to support their use of those products; UUNET can help your customers develop such policies or offer general security consulting and security site audits.

SELECTING YOUR CONNECTION TYPE

There are several different kinds of local access options for physically connecting from your location to AlterNet, and beyond to the Internet. The connection type you select will depend on the type of services you are seeking and on the size of the user base you are trying to serve.

Mail and News Access Only

For occasional Internet users, who only wish to send and receive electronic mail and participate in special-interest newsgroup discussions, a dial-up UUCP connection is all that is needed.

Indirect IP Access
A dial-up link to a provider's public-access machine, commonly dubbed a shell account. This account allows you to reach all Internet services that your provider makes available. However, some text and graphical applications, such as Mosaic, may not be used with a shell account. All files are stored on the provider's machine; you will need additional software to download them to your local computer.

Full Dial-up IP Access
Your computer becomes a official Internet host, with its own domain name, Internet address, and the ability to use the complete range of IP services with a standard modem. AlterNet's AlterDial(R) and ISDN Workgroup(SM) dial-up services are perfect for users and organizations who want full Internet access without the expense of a full-time dedicated link.

Full Dedicated IP Access
Full-time, full-service IP connectivity (mail, news, FTP, Telnet, Gopher, WAIS, WWW, Archie, Veronica, Mosaic, etc.) over a dedicated digital telephone line. Speeds range from 14.4 Kbps to T-1 (1.5 Mbps) and up. You choose the appropriate speed based on the number of users and the amount of traffic you expect.

ALTERNET'S FAMILY OF PRODUCTS

PHONE$HOME is a cost-effective way to stay in touch while traveling. All across the continental US, you can dial in to one of UUNET's local-access numbers and work on your Internet hosts quickly and inexpensively.

AlterNet's UUCP services offer you reliable Internet mail and USENET news across the US and around the world. In addition to UUNET's mail and news ser-

vices, UUCP offers proxy Internet file-transfer capabilities and access to our extensive on-line software and documentation archives.

AlterDial gives you direct TCP/IP access to the Internet and AlterNet's network. AlterDial provides unrestricted commercial access to Internet services such as mail, news, Telnet, FTP, Gopher, WAIS, Mosaic, the World Wide Web, and more. AlterDial customers can receive their own domain name and multiple POP mailboxes.

ISDN Workgroup service is designed for business LANs that need high-speed dialup access to the Internet. Connection speeds match those of digital leased lines (approximately four times faster than analog lines), at a cost which can be considerably lower.

56 Kbps service offers mid-sized businesses full access to the Internet, via either leased lines or Frame Relay technology. Frame Relay offers higher performance than "full-time" dialup access, at a lower cost than traditional leased-line solutions.

T-1 Plus is targeted toward subscribers with a variety of throughput requirements. You may choose from either the Burstable Service, which allows you to periodically "burst" up to the full 1.5 Mbps, or the Fractional Service, which permits speeds up to the upper limit of the tier you choose.

UUNET's 10Plus service offers extremely high throughput Internet access to customers with demanding bandwidth requirements. UUNET's 10 Mbps access offers the simplicity of a physical Ethernet interface, with prices usually associated with traditional T-1 access. You pay only for the bandwidth that you use on a sustained basis 95% of the month, as determined by traffic samples taken over UUNET's network every five minutes throughout the day.

UUNET'S WEB SERVER SERVICES

UUNET's Web server hosting services provide you with an easy and affordable way to establish and maintain your company's presence on the World Wide Web. You may choose from the Standard service, which places your information on a server connected to our 45 Mbps ATM backbone at T-1 (1.544 Mbps) speed; or the Premium service, in which your information is stored on a server connected to our backbone via a 10 Mbps link. Both options offer you 24-hours-a-day, 7-days-a-week monitoring, and reliable, fast access to your corporate information by anyone on the Internet. UUNET can also develop your Web server content, providing a fully integrated service. Please call for details.

UUNET. The Internet Business Solution.
For more information, please contact:

UUNET Technologies, Inc.
3060 Williams Drive
Fairfax, Virginia 22031-4648 USA
+1 800 488 6383 (voice)
+1 703 206 5600 (voice)
+1 703 206 5601 (fax)
info@uu.net
http://www.uu.net

For a Single Mac (Not on a Network)

If your site is just a single Mac that's not on a network, we like the Planet ISDN card. It's pricey, at $950, but from what we've looked at and heard from our service provider friends, the Planet ISDN seems to be the one that is most compatible with the different flavors of ISDN-capable boxes Internet providers have at their end. But don't just run out and order one; as always, ask the advice of your service provider before you buy anything. Because sooner or later (we hope sooner), some company is going to come out with an ISDN board priced around $400 and sew up the whole ISDN market.

At the current prices for ISDN cards like the Planet ISDN, you may be better off with something like a Combinet box or a router from Network Express, which is what you use to set up a local-area network (LAN) on the Internet. Remember, an ISDN card will only allow one machine to be connected at a time. That leaves no room for growth if you decide later you'd like to add multiple Web servers or an FTP server, or if you want to use another Mac to surf around the Net over that nice high-speed connection you're paying so dearly for.

For a LAN (Local-Area Network)

If you are on a LAN, your Internet provider will probably connect your network to theirs using a router box that works with ISDN. Go with their equipment recommendation. The cost at your end will run somewhere between $750 and $1500 for all the hardware, and another $50 to $150 a month for the ISDN line.

Your network administrator (that may be you, so pay attention) and Internet provider need to put their heads together and address the security issues that a LAN connection to the Net has the potential to create. There are issues you definitely need to consider. The main one, of course, is that you don't want outside traffic having access to your whole network. Your Internet provider can best recommend what "firewalls" can be put in place to restrict inbound access to your network to just the machine that is the Web server. They can also restrict outbound traffic for you so that other users on the LAN aren't just goofing off surfing the Net all day. If you are going to connect a LAN to the Internet you need

to spend time looking at security issues that are beyond the scope of this book. You can learn more about the "firewall" world by reading *Firewalls and Internet Security* by William R. Cheswick and Steven M. Bellovin. It's published by Addison Wesley and goes for about $25. Firewalls are an arcane study in and of themselves.

What You Need to Know If You ARE Firewalled

Here's the deal. If the LAN you are on is "firewalled" from the Internet, get your network administrator and Internet provider to set your Web server up outside the firewall. Just remember, that makes it a sacrificial lamb. So don't keep company secrets on your Web server. And always, always have at least one backup of your entire server, so that you can restore it after a catastrophe.

There is a very big advantage to having your LAN connected to the Internet. Besides letting multiple users surf the Net over the same connection to the Internet, you can run multiple Web servers, FTP servers, and so on, all on different machines on your local network, and all connected to the Internet with one device.

If your LAN is already on the Net, just ask your network administrator to assign you a permanent IP address (and while you're at it, a host machine name for the Mac you'll be using as a Web server) and you're in business!

Look Ma! No SLIP or PPP From Home Either!

Here's a cool trick we recently came across — a way of giving remote users access to the Internet via your network. All the remote user needs on his or her machine is MacTCP, the AppleTalk Remote Access (ARA) Client, and any Internet apps they want to run.

On your LAN you need TCP/IP, some spare IP addresses, a dedicated connection to the Internet, the ARA Personal Server, a modem and phone line to answer the incoming ARA Client, and the Apple IP Gateway software.

You dial in from home and connect to the network at your office using the ARA Client. (Remember, ARA is AppleTalk.) With the ARA connection established, you fire up one of your Net apps such as Netscape, TurboGopher, or Fetch. (Remember, Net apps like these use TCP/IP.)

The first thing that happens is that the Apple IP Gateway software recognizes that you are sending IP packets "encapsulated" inside the AppleTalk connection you have established with ARA. It then automatically assigns an IP address to your Mac at home — just like when you dial into your Internet provider with SLIP or PPP and establish an IP session on the Net for the duration of the call.

Once the IP address is assigned, the Net app's IP pack shoots over the LAN at the office and out onto the Internet through the LAN's dedicated connection to the Net.

Cool, huh?

No SLIP or PPP at all. Later in the book we'll show you something even cooler you can do along these lines using a box from Tribe Computer. But we couldn't stop ourselves from interrupting and telling you about this nifty trick right now.

Unless you are a serious telecommunications enthusiast or are the type who needs to know everything about everything before moving forward, leave the configuration of any digital connection to the Internet, such as ISDN, to your Internet provider and your local phone company. Let them have the joy of getting it all set up and running. It's their responsibility anyway. You can always come back when the dust settles and learn more.

By the way, an excellent resource on the Net for info on ISDN is at Dan Kegel's ISDN page at **http://alumni.caltech.edu:80/~dank/isdn/**. Everything under the sun about ISDN is there. Knock yourself out.

IP EVERYWHERE: UNDERSTANDING NETWORK PROTOCOLS

Besides the physical network or modem connection with your Internet provider, you need a permanent address on the Internet. (Remember a few pages ago we told you to get started on it ASAP?) If you don't already have one, it's time to get a domain name and IP address.

Getting Your Domain Name and Permanent IP Address

To be a Web site — or any other type of server on the Internet —you have to have a permanent IP address. You should also have a domain name, though strictly speaking, it's not mandatory. An IP address is a number — 204.96.111.158. A domain name is a name, like "outer.net" or "webmastermac.com."

Because most humans don't deal well with numbers, the Internet uses names as "aliases" for the numbers. That way you can type the name of a site instead of trying to remember the number. You can, of course, use the IP number instead, but let's face it, **http://www.info.apple.com** is easier to remember than **http://204.96.4.2** for most people.

URLs

You know what URL stands for, right? If you've ever surfed, you've seen URLs. But sometimes you see something over and over but still aren't sure what it means.

URL stands for Universal Resource Locator. It's pronounced You-Are-Ell, not EARL. OK?

Web browsers navigate using URLs. You can always tell from the URL what type of site it is. For example, if the URL starts with "**telnet://**", as in **telnet://internic.net**, you can tell it's a Telnet site. If the URL starts with "**ftp://**", as in **ftp://apple.com**, it is an FTP site. If it starts with "**http://**", it's a Web server.

A Detour into DNS

The Internet keeps track of all the domain names and associated IP numbers with DNS, the Domain Name System. DNS was developed by Sun Microsystems in the early 1980s and is a vital part of the Internet. The World Wide Web is heavily dependent on a fully functioning Domain Name System. In a nutshell, DNS is a hierarchical distributed database system that lets Internet providers (also known as "hosts") keep track of IP addresses and host names in "Host Tables." The Host Table "resolves" a domain name into an IP number to be transmitted over the Internet.

Because there are so many IP addresses and domain names on the Internet, it's impractical for a provider's Host Table to keep a list of all of them. DNS deals with this by distributing the names up and over the Net chain from host site to host site. This, coupled with a system of "lookups" when a host site or provider is unable to resolve a domain name into an IP number, makes the whole addressing scheme over the Internet function smoothly.

There's obviously more to how DNS actually works, but that should hold you. If you're really interested, you can talk about it over that lunch you are going to have with your service provider.

Before You Request...

One thing you should do before requesting a domain name and IP number from your Internet provider is Telnet to the InterNIC and search the "Whois" database to see if the domain name you want is available. If you've never done this, here's how:

After making a connection to the Internet if necessary, Telnet to **internic.net**. When you get there you'll see a menu that looks like Figure 2.3.

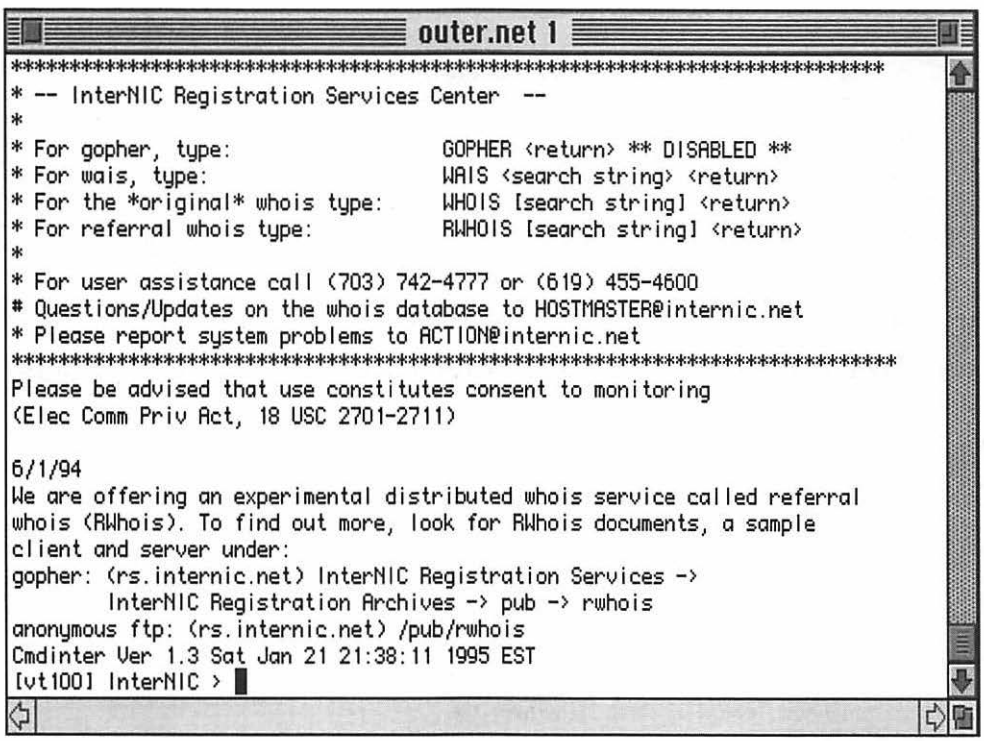


Figure 2.2: The InterNIC "Whois" Menu, January 1995.

You don't have to select anything from the choices presented. The prompt has you at the right place to begin with. Just type the domain name you are interested in having to see if it is available, as shown in Figure 2.3.

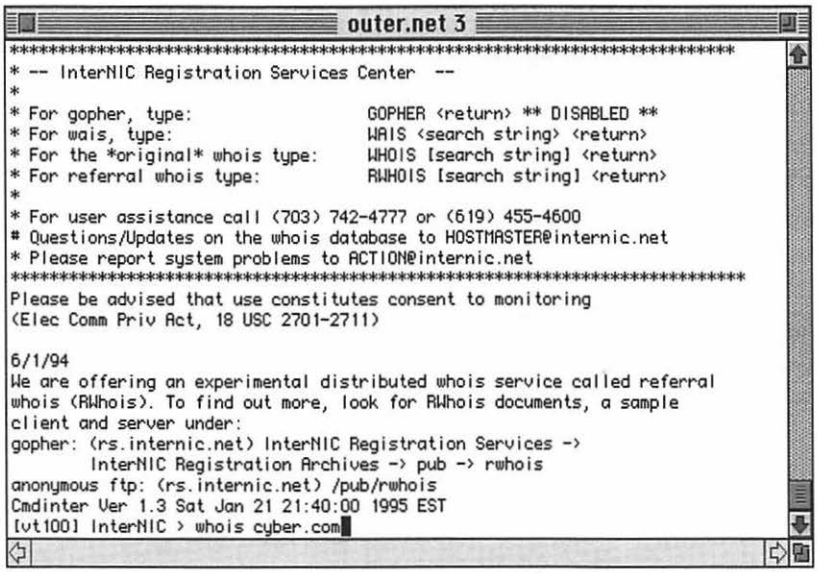


Figure 2.3: The InterNIC "Whois" Search, January 1995.

If you typed **cyber.com**, you got back the information shown in Figure 2.4.

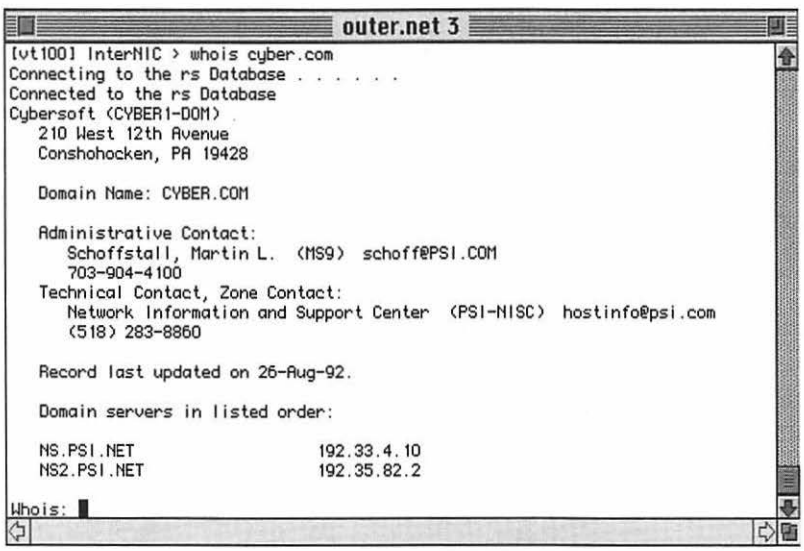


Figure 2.4: The InterNIC "Whois" Results, January 1995.

Bummer. Try again — you need to be just a little more original. Remember that you're probably going to be a .com (pronounced "dot com"), which means you're a commercial organization. Here are some other common "dot" codes:

- .edu = educational
- .org = nonprofit organization
- .net = Internet providers
- .gov = government
- .mil = millionaires (and military)

What's the InterNIC? Here's What "They" Say

The InterNIC is a collaborative project of three organizations that work together to offer the Internet community a full scope of network information services. These services include providing information about accessing and using the Internet, assistance in locating resources on the network, and registering network components for Internet connectivity. The overall goal of the InterNIC is to make networking and networked information more easily accessible to researchers, educators, and the general public.

The InterNIC name signifies the cooperation between Network Information Centers, or NICs. The entire Internet community, in fact, is brimming with cooperative ventures, resource sharing, and collaborations. It is fitting, then, that the InterNIC is managed by three organizations operating under one umbrella.

General Atomics in San Diego, California, manages Information Services, providing the Scout Report, net-happenings, NSF Network News, InterNIC, InfoGuide, InterNIC Briefcase, Reference Desk, seminars and other services that respond to the needs of the scientific and education community.

Phone: +1-619-455-4600
Fax: +1-619-455-4640
eMail: **refdesk@is.internic.net**

AT&T in South Plainfield, New Jersey, manages Directory and Database Services, providing the Directory of Directories, Directory Services, Database Services, and a help desk to assist both people making use of their services and additional support organizations who want to know more about offering these services to their own audiences.

Phone: +1-908-668-6587
Fax: +1-908-668-3763
eMail: **admin@ds.internic.net**

Network Solutions, Inc., in Herndon, Virginia, manages Registration Services, providing Internet Protocol (IP) address allocation, domain registration, Autonomous System Number (ASN) assignment, and inverse addresses.

Phone: +1-703-742-4777
Fax: +1-703-742-4811
eMail: **hostmaster@rs.internic.net**

For general information about the InterNIC, please send mail to info@internic.net.

Source: InterNIC Home Page
http://www.internic.net/

Once you've found a domain name you like that isn't already taken, call or eMail your Internet provider and get it registered. They should only charge you about $25 to $50 for this, if anything. Keep in mind that just because your search at the InterNIC didn't show that your domain name wasn't already taken, it doesn't guarantee that it's available. You're probably OK, but don't run off and spend money or time getting new business cards printed with that URL until you get a confirmation that the name is actually still available.

There is usually a big queue at the InterNIC. The name you like may be tucked in the queue somewhere and someone may get it before you. The last time we checked, the backlog at the InterNIC was over 3000 names, and the wait to get registered was over two weeks. (You can probably get it done sooner if your Internet provider is on a first-name basis with someone at the InterNIC.)

You can download the application from the InterNIC and apply yourself by going to the InterNIC registration services off the InterNIC Home Page at **http://www.internic.net**. But why bother? Your Internet provider has the same form and theirs is probably even easier to fill out. Unless they are charging extra to fill it out, have them do it; you've got enough work ahead of you. Do check out the InterNIC Home Page though. There's a ton of interesting Net info there.

How Not to Make Big Bucks...

Do everyone a favor: Don't try to register domain names for big companies like Coke, McDonalds, or Circuit City. It won't fly, and they won't pay you big bucks for it someday.

The InterNIC is wise to this and doesn't dig it. They can spot those bogus applications a mile away. Besides, don't you have any better ideas for your creativity? Knuckleheads who do those kinds of things on the Net are wasting bandwidth and making it take longer for anyone else to get a domain name. Besides, that kind of stuff will almost certainly get you mailbombed by the UNIX cybergods if they find out.

Here's a nifty tip. You don't have to wait for your domain name to come through to turn on your Web site. Your Internet provider or network provider can give you a permanent IP number from their set, and then you can turn your Web site on for all the world to see (assuming you have your connection in place and something to show).

When the domain name finally comes through from the InterNIC, your provider will associate it with the IP number they gave you earlier, put it in their Host Table, and it will automatically get distributed through the entire DNS system of the Internet in the course of a few days. (See! Now you know why we included that seemingly incongruous section on DNS a few pages back.)

If you are part of an organization that already has a domain name set up, you can contact your network administrator or Internet provider

and obtain a host name (as well as an IP number) for the Mac you will be using as a Web server. That would be everything you need — you lucky dog. If you're in this boat, you can make your request for a permanent IP number and host name for your machine in a single call.

Now What?

Assuming you're connected, took our tip, went ahead and got a permanent IP number and have put in your application for a domain name ASAP as we suggested, what do you do next? Learn to set up MacHTTP, of course!

MACHTTP: A FIRST LOOK

Now we're going to show you how to get MacHTTP up and running and be a network of one — and you don't even need a connection to the Internet. If you already have a dial-up SLIP or PPP account, we'll even show you what you can do with the IP number off the MacTCP Control Panel, the one we told you about before that is assigned automatically by your Internet provider during a connection. With that number, you can be a Web site on the Net, at least for as long as your current connection remains open. Of course, no one will know unless you call them up, tell them the IP number, and tell them to point their browser at that IP address. This Web site won't have a domain name yet, and if you end this connection, when you reconnect you may not have the same IP number. But you can definitely call your friends and have them check out your site with their Web browser.

But that's just an added thrill. This section is designed to put you ahead of the game, so that when you've got your dedicated connection and permanent IP and domain name, you'll have some idea of what it takes to get a Web site up and running.

MacHTTP: Your First Time

It's time to jump in and fire up MacHTTP. Make sure you are not connected to the Internet for this, at least for now. The approach we're going to take is just to build a Web page quickly. We'll go back and tweak

it and add the bells and whistles and show you how to "turn it on" for the general public in upcoming chapters.

At the end of this tutorial, you'll have set up your Mac as a WWW server. OK, here's what you need.

System Requirements

To set your Mac up as a WWW site you'll need the following:

- at least 8 megs of RAM
- System 7.x
- a 68030-based Mac or better
- MacTCP
- AppleScript

MacHTTP runs fine with all versions of System 7, including 7.5. The version of MacHTTP included with this book is fat binary and runs on any 680x0 Mac and is also accelerated for the Power Macintosh.

You'll also need MacTCP installed on your machine. MacTCP 2.04 comes with System 7.5; we've included an updater that converts that version into version 2.06 on the CD that came with this book (be sure to read its "readme" file so you understand why some people never read "readme" files.)

If you are running an earlier version than System 7.5, it's time to upgrade. While you're at it, buy Bob's other book, *Macintosh System 7.5 for Dummies*. Yes, this is a shameless plug but that book is at least a zillion times better than the Apple documentation. System 7.5 only costs $99 on the street, and it's even cheaper if you belong to any Mac user group. *Macintosh System 7.5 for Dummies* is published by IDG and is $20.

You can get by with System 7.1 and find some ancient FTP site that still has version 1.something-old and download MacTCP for free. We're not even going to tell you where to begin looking, though, because we don't want to help you do the wrong thing. But if you insist on running anything earlier than System 7.5, you're also going to want to dig up a copy of the new Thread Manager so your Web server can run faster and handle multiple requests from Web browsers more effectively.

Lastly, you'll need AppleScript installed on your Mac if you want to do some of the advanced stuff like forms and clickable GIFs that we'll talk about in upcoming chapters and appendices. Chuck Shotton got

Apple to let him bundle AppleScript with MacHTTP 2.0. It also comes with System 7.5.

MacTCP, Thread Manager, and AppleScript all come with System 7.5; if you're going to be a WebMaster, that's reason enough to upgrade. So get with the program. Join a Mac user group if you need the bargain price (not a bad idea anyway).

MacHTTP Quick Start: The Movie

If you haven't already done so, insert the CD that came with this book and copy the file **MacHTTP.sit** to your hard drive. You'll need about 4MB of free disk space. Double-click **MacHTTP.sit** on your hard drive to expand it; a folder will be created on your hard disk called "MacHTTP 2.0". Everything you need is in this folder. Open up the folder "MacHTTP 2.0". It will look like Figure 2.5.

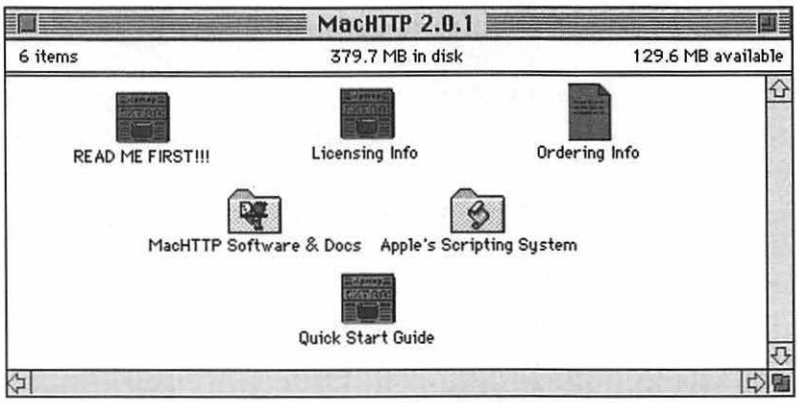


Figure 2.5: MacHTTP 2.0 Folder, January 1995.

You can read the MacHTTP Quick Start Guide if you want, but hey, that's why you bought this book, right?

Do read the "READ ME FIRST!!!" file for the license info on MacHTTP. The deal in a nutshell is that if you're not affiliated with any commercial, government, or educational organization and are running your WWW site for "free," then there's nothing you have to do except read the license info before you fire up MacHTTP. If you're subject to the

license, do the right thing and pay for the license by the end of the 30-day evaluation period. Chuck Shotton has done an awesome job with MacHTTP. Pay him if you're supposed to. It's peanuts for what you get.

Next, move the entire folder called "MacHTTP Software & Docs" to the root level of your hard drive. Forget about the "Apple's Scripting System" folder for now. We'll come back to it later.

Setting the IP Address

You configure your IP network connection to the Internet using the MacTCP Control Panel. Knowing some things about TCP/IP and the MacTCP Control Panel may earn you some points with your Internet provider, though it probably won't get you a break on your monthly fees. (Service provision isn't the high-margin business it was in the early days.)

Open the MacTCP Control Panel. It will look like Figure 2.6 if you don't already have a permanent address.

We need to do a little configuring of MacTCP. Before you do anything — jot down everything you have already configured with MacTCP so that you can always get back to it if you need to.

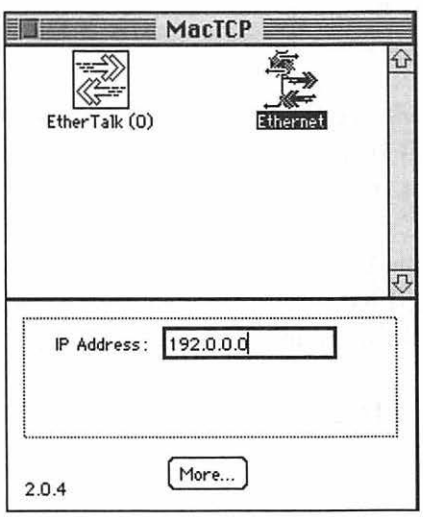


Figure 2.6: MacTCP Control Panel—IP Address Without Net Connection.

Click on "More...". You should see the dialog box shown in Figure 2.7.

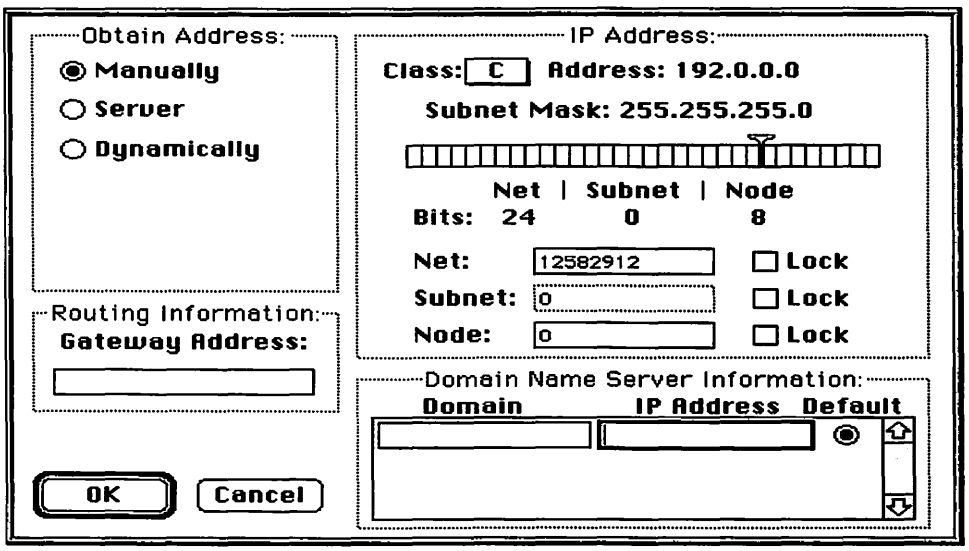


Figure 2.7: MacTCP Control Panel Configured for Manual Addressing without Net Connection.

Set all the settings in your own MacTCP panel just as you see them above. You're going to modify and fill in many of these later when you use a permanent IP address — not now, though. Just set "Obtain Address" to "Server" and "Class" to "C."

Close the window and you're back at Figure 2.6. Type in the IP number: 192.1.1.1 in place of 192.0.0.0 (or whatever else may be there).

Select either LocalTalk or EtherTalk depending on your network, close MacTCP, and restart your Mac.

Now go back and open the MacHTTP 2.0 folder. It will look like Figure 2.8.

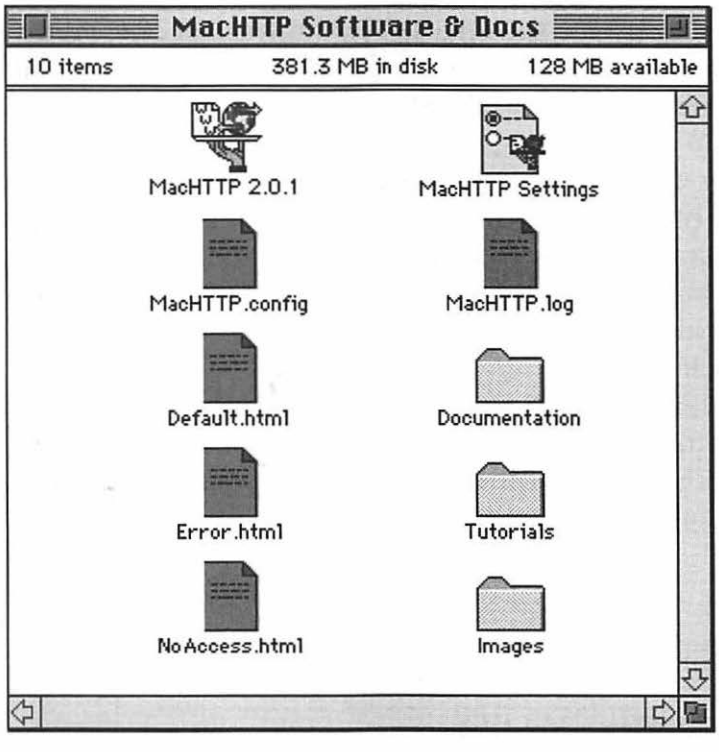


Figure 2.8: MacHTTP 2.0 Folder Contents.

Double-click the MacHTTP icon to launch your private Web server. You'll see the window shown in Figure 2.9.

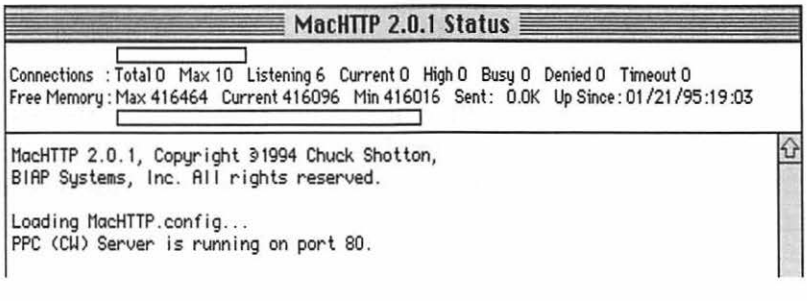


Figure 2.9: MacHTTP Status Window.

Congratulations!

You are now your own Web site on a TCP/IP network of one. Pretty easy, huh? Feel like you missed something? You didn't. It's just that simple. Try that on a PC or a UNIX server! Ha!

But don't run off shooting eMail to everyone you know and posting the location of your WWW site all over the Net. At least not yet. To start with, you're not connected to the Net. Further, you don't have a permanent IP address that works outside your little world. In other words, you still have a few housekeeping chores to do. But it was time to get your Web site up and running, and with those last few steps, you've done just that. Give yourself a hand.

Let's see what your Web site looks like and then we'll start some lightweight fine-tuning. Launch your favorite WWW browser, and go to the URL address of your brand-new WWW site. If you're using Netscape, that would be Command-L, then type **http://192.1.1.1** and press Return (Figure 2.10).

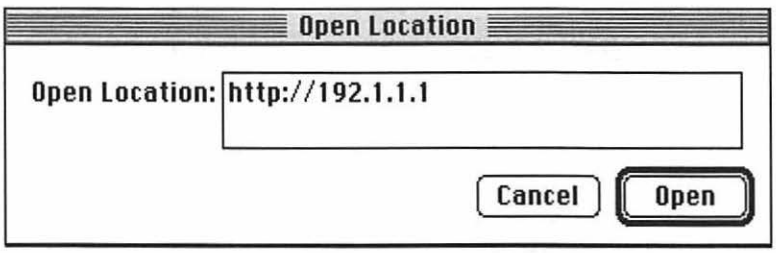


Figure 2.10: Netscape Go To URL Window.

Cool, huh? The MacHTTP default Home Page that Chuck Shotton put together loads right up. Which means that you've built a home page, more or less! Take a bow, then take a look at the MacHTTP window and see the fruit of your labors (Figure 2.11).

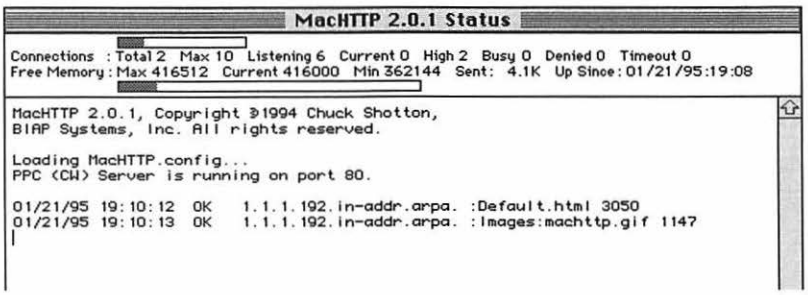


Figure 2.11: MacHTTP Window with Client.

More About Configuring MacHTTP

Now let's delve a little deeper and configure MacHTTP a little bit. Quit MacHTTP for the moment and open the file **MacHTTP.config**, which you'll find in the "MacHTTP Software" folder. This file is a big long text file (you can open it with SimpleText) and it starts off something like:

```
#MacHTTP Configuration file, v. 2.0.1
#
#The format of this file is free form, with a few exceptions. Lines not
#starting with a recognized keyword are ignored.
#Note, all entries are converted to upper case by MacHTTP, so the config file
#isn't case sensitive, with the exception of Mac file types and creators
#below.
#
#Note that any text styles like bold or underline in this file are purely
#cosmetic and are only intended to make the file easier to read. The styles
#are not required.
#Legal config file keywords will appear as BOLD text if this file is viewed
#with SimpleText or TeachText.
#
# The version number below must match MacHTTP's version number
VERSION 2.0.1
```

```
############################################################
# "Special" Files
#
#The following line defines the default file type if a suffix match isn't
#found. The syntax is: DEFAULT <default transfer type> <default MIME type>
#DEFAULT TEXT text/html
#
#The following lines specify where to find HTML files for error messages, the
#default home (or index) page, the name of the log file, and the message
#returned for security violations. Any of these three file directives point
#to a HTML document, script, or CGI application.
#
#NOTE!!! INDEX must be a simple file name, not a path like the other files.
INDEX  Default.html
ERROR  :Error.html
NOACCESS :NoAccess.html
```

There's a lot more to the file itself and the best thing to do at this stage is take a look at it or even print it so that it's in front of you for editing as we go.

If you've never seen anything like this in your life, don't be intimidated. It looks more complicated than it is. Most of it is very straightforward. Basically, the **MacHTTP.config** file tells MacHTTP how to behave, what to listen for, and how to respond to requests from WWW browsers such as Netscape, MacWeb, Mosaic, and others. In client-server speak, the browsers are the clients and MacHTTP is the server.

Most of the lines in the config file are comments. Those are the ones that begin with a # symbol. As you can see, the config file is chock-full of comments. The comment lines describe what the configuration options are and what will happen if you edit them. The whole thing is way easier than it could be and is nothing like the relative nightmare you get if you had to do this on a PC or UNIX box. Believe us, we've done it. We're going to walk you through the whole thing in a couple of chapters. (Chuck may even crush the last vestige of text configuration and turn all of what appears in the config file into pop-down choices from the menu bar. Chuck lives to crush vestiges.)

At this stage, our advice is just to use the config file as it came. We'll wait until later chapters to edit **MacHTTP.config**. There are many other fine-tuning adjustments to the **MacHTTP.config** file that can be made to optimize performance and implement security features. We'll take those up in a few chapters too.

If You Already Have an IP Address....

If you already have an IP number from your Internet provider or network administrator and your machine is already on the Net, all you have to do is enter the right settings in the MacTCP Control Panel and then fire up MacHTTP as described above. You'll need to plug in the following info into the MacTCP Control Panel:

- Obtain address (select manual)
- Gateway address (goes in Routing Information box)
- IP address (set to Class C — leave the rest as is)
- Domain Name Server Information (ask your Internet provider or network administrator)

When you're done it should look like Figure 2.12.

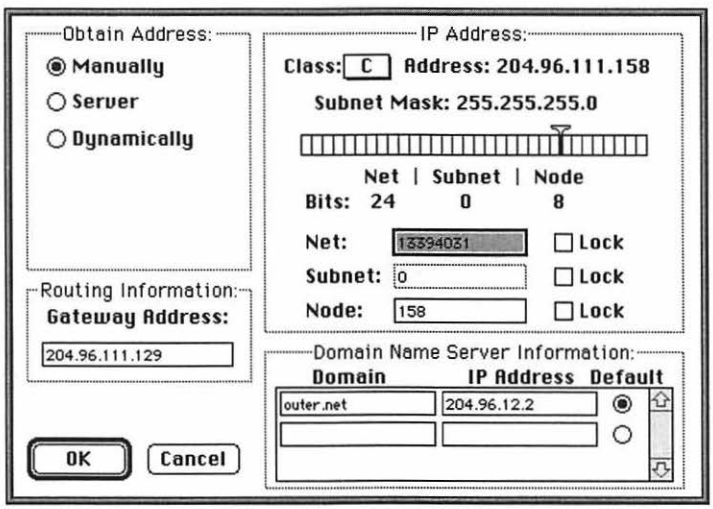


Figure 2.12: MacTCP Control Panel Configuration with Live Net Connection.

The last step you need to do to configure MacTCP is to close the window you have just set and fill in the rest of MacTCP on the first and smaller MacTCP panel — the one you see when you first open MacTCP. Select your network connection and enter the IP address of your machine. When that's all done it will look like Figure 2.13.

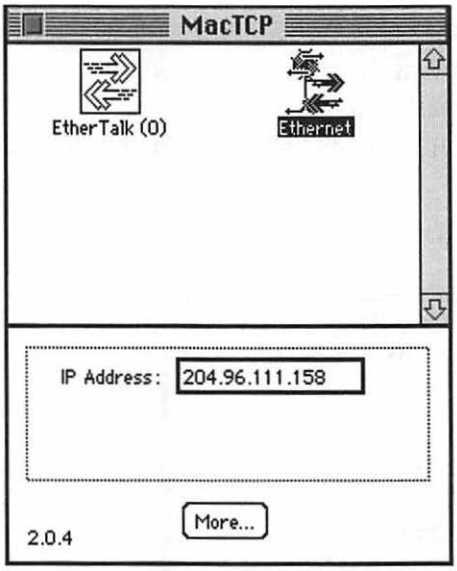


Figure 2.13: MacTCP Control Panel IP address and Network Type with Live Network Connection.

That's it. Launch MacHTTP. Fire up Netscape or Mosaic and load the URL of your Web server just like in the previous example.

If you have a SLIP or PPP dial-up connection and are still waiting to get your IP address, you too can tell your friends to come take a look — at least you can as long as you maintain that connection. Just open up MacTCP when you're dialed in with SLIP or PPP and jot down the IP number. It's the one that appears in the first of the two MacTCP Control Panel slots (as in Figure 2.12). Launch MacHTTP. Fire up Netscape or Mosaic and load the URL of your temporary Web server. Then call everyone you know who surfs the Net and tell them to come visit quick.

A Final Note about MacTCP

MacTCP works. It may, however, be the worst example ever of an interface from Apple.

It is unclear to the uninitiated how to configure MacTCP. We've never seen any published documentation. The "Obtain Address" portion of the Control Panel — with its Manual, Server, and Dynamically radio buttons — is completely nuts, not to mention counter-intuitive. The reality is that you either obtain an address by manually putting it in (once you've been told what it is) or it is assigned dynamically (and automatically) by the Internet provider's server or by an in-house network server running something like the Apple IP Gateway software. There is never a time when you select the "Dynamically" radio button.

And why two panels to begin with? And what about that bar that moves back and forth in the IP Address portion? How are you supposed to figure that one out on your own? And have you looked at the numbers in the three boxes labeled net, subnet, and node? How are you supposed to know what the heck these do?! Actually, Carl told us that they are decimal representations of the binary subnetting. There, now you know.

We hate messing around with MacTCP. Eudora has a clever error message that says something like, "That pesky MacTCP is acting up again." We know what it means!

EVERYTHING YOU NEED TO KNOW ABOUT CLIENT-SERVER ARCHITECTURE

We almost forgot. We promised an overview of the client-server thing. Let's recap: You have your Web server up, even if you're just a "network of one" for now. That was no big deal, right? In fact it was easy. So now we'll step back and take a bird's-eye look at what we've done so far.

You've got MacTCP, MacHTTP, and your Web browser of choice, all humming along perfectly. Let's finish with a bang, with a diagram that explains these elements and brings the whole client-server story into focus (Figure 2.14). Why are we doing this? Because they say a picture is worth a thousand words and neither of us feels like writing another thousand words about something as boring as client-server architecture.

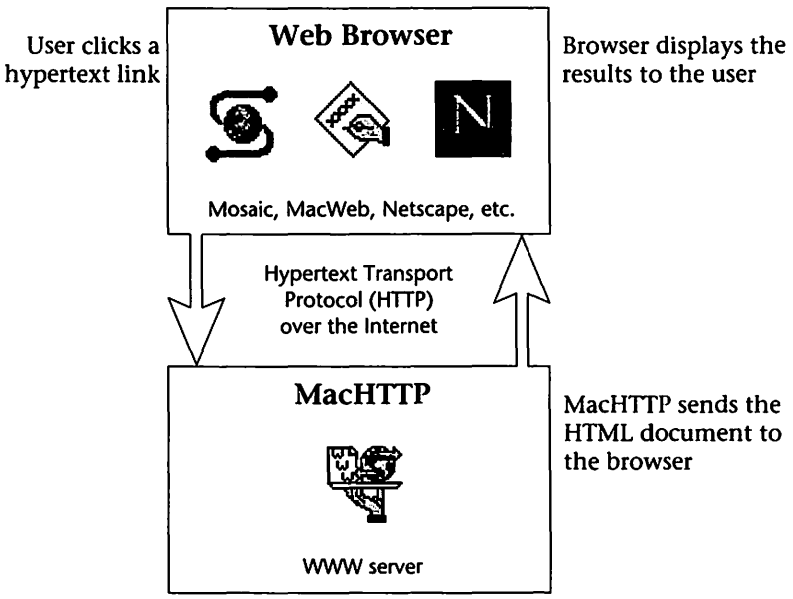


Figure 2.14: The Client-Server Story.

WRAP IT UP

It's a wrap. You've now learned all you need to know about getting connected. Don't forget our most important advice: Go with what your Internet provider recommends given your budget.

You now know how to configure MacTCP (let's hope you never have to look at it again) and bring a Web page up using MacHTTP. That's enough left-hemisphere behavior for one sitting.

If you want to take a break, you're excused. Go surf the Net for a while. Better yet, go outside and take a walk. In the next chapter you're going to learn everything you need to know about HyperText Markup Language, or HTML. So we suggest a long walk and something creative. This next section is a little bit technical. On the other hand, you have us, your hapless authors, who can make even HTML exciting!

So take that break, then turn the page.

Chapter 3

Introduction to HTML

HTML: Not Just an Acronym for "How to Make Levitusburgers" Anymore

OK — let's face the music. In order to serve up pages, text, graphics, and everything else on your Web server, you're going to have to learn some HTML (HyperText Markup Language). It's not rocket science; in fact, it's not bad at all as languages go — ever program in PostScript? That means it's not going to take hundreds of pages to explain. All we ask from you is that you learn a few rules. Soon, you'll be an expert just like us. <grin>

Introducing HTML...You're Gonna Love It

Learning HTML is going to be a little like going back to the early days of word processing when you had to type "bold" before and after the word you wanted to be bold.

Lovely, eh?

Until someone comes out with a great HTML editor that does it all and treats all your pages as one big integrated document — more along the lines of PageMaker or FrameMaker — you're just going to have to labor with typing out HTML codes. As we are writing this, the best and easiest way to create HTML code is to use the HTML "plug-ins" with BBEdit (more on this later). FrameMaker and Word have HTML filters coming out soon that look promising but were unavailable for testing. Hopefully by the time you read this you'll be able to take advantage of *the* killer HTML editor. Until then, you're going to have to tough it out, learn the markup elements of HTML, and obey the HTML cybergods.

What Makes HTML Worth the Hassle?

HTML is a subset of SGML (Standard Generalized Markup Language), which is a subset of ancient Greek. (We're just kidding but mark our words, all these acronyms are eventually going to ruin some crossword puzzler's Sunday morning.) For all its kludgy-ness and programmer's code-like look, HTML is actually a pretty nifty authoring language for WWW. In a nutshell, it's what allows the same Web pages to be viewed by a Mac, a PC, an Amiga, or even a UNIX box. That's very cool.

An HTML document consists of text, formatting, defined hypertext links, and anchors, and adheres to a strict set of document formatting rules. Stick with us and you'll learn them all.

When you create a page with HTML, you don't specify whether the title of a document should be displayed in a particular font such as Times Roman with a point size of 14. Instead you use HTML code to instruct the browser how to display your page's parts — its title, text, images, and so on. It's left entirely to the user's browser — Netscape, Mosaic, or whatever — to recognize HTML tags and attributes and display the pages.

The defaults and/or preferences set in the client's browser determine how things are going to look. So relax, don't worry, and try to do your HTML right. Everything else is up to the browser.

(Most people don't bother or even know that they can change the style preferences in their Web browser and do all sorts of wild-looking things with your boring old HTML. For that matter, most users never change the default URL to something other than the NCSA Home Page or Netscape Communications.)

Rest assured. If you use proper HTML markup codes, your pages will look great in any browser on any platform.

HTML: The Good, the Bad, and the Ugly

HTML makes life simple. Imagine what it would be like if you had to do a separate Web page for every type and version of Web browser out there — each one with its own defaults and preferences for displaying HTML. You'd be up all night for a month of Sundays.

On the other hand, HTML limits what you can do creatively with formatting, leaving your page wide open for users to wreak havoc on by specifying weird fonts and things in their Web browser's preferences. (Though, as we said before, most don't.)

Like everything in life, there is Good, Bad, and Ugly when you write HTML code. What's Good is that HTML makes it possible for Mac, PC, and UNIX users to all see each other's stuff with a nice graphical, hypertext interface. It's the Esperanto that never was — or something like that.

The Bad is that it's still kind of geeky and requires a certain amount of concentration to get right.

The Ugly? Surf the Web for five minutes and we promise you'll come across examples of Ugly.

Steal This HTML

One of the best ways to learn HTML is to take a look at other people's HTML by choosing to view the source with your Web browser. It's OK to beg, borrow, and steal, but don't forget to thank the WebMasters whose source HTML you use. You should know that there is a slight pitfall to nabbing HTML and modifying it to suit your purposes: Not every WebMaster adheres to the accepted HTML standards. (We're sticklers for accurate HTML formatting, so stick with us and we'll teach you how to do it right.)

When you see HTML that looks clean, it's obvious that the WebMaster cared about the work he or she did. We like seeing HTML where all the markup code is in uppercase (or all in lowercase — just as long as it's consistent). We like seeing HTML where the layout is easy to follow and care was taken with line wrapping.

Unfortunately, a lot of the HTML we've seen is sloppy. When you are looking at other people's source HTML, take a look at large commercial sites, where the HTML code is more likely to adhere to the guidelines. Some of the best work out there is the stuff done by the HTML troops of the O'Reilly Global Network Navigator site. We recommend you at least take a peek at their HTML source at **http://www.gnn.com**. That's what good HTML should look like.

Since Web browsers are becoming increasing particular about what they recognize as correct HTML, it is becoming more and more important to adhere to standard HTML. That's what we'll show you in this

chapter. (And, of course, you can always get the *latest* version of HTML standards from the WebMaster Mac server at **http://webmastermac.com.**)

HTML QUICK START

All you need to get started creating HTML is any text editor. SimpleText will do, as will Microsoft Word (Bob says: Word 6.0 is slow and slothlike; earlier versions are faster and better). We both like BBEdit Lite 3.0. It runs about $100 (there's a link to a demo version of it as well as its HTML plug-ins on the WebMaster Mac WWW server). Try it out. If you have a Power Macintosh, spend another 20 bucks and get the full-blown commercial version of BBEdit, BBEdit 3.1. Unlike 3.0, 3.1 runs native on the Power Macintosh and supports Macintosh drag and drop.

BBEdit has a great set of "plug-in" extensions that make creating HTML a blast — at least compared to what it used to be when your only option was to type everything out. It forces you to stay within accepted HTML conventions too. You can just type out some plain text, highlight it, select "HTML Format" from the menu bar and blammo, instant HTML. You can grab BBEdit at:

ftp://webmastermac.com/devtools/bbedit_lite3.0.hqx

Do that now, launch it, and we'll take a look at what HTML looks like (Figure 3.1).

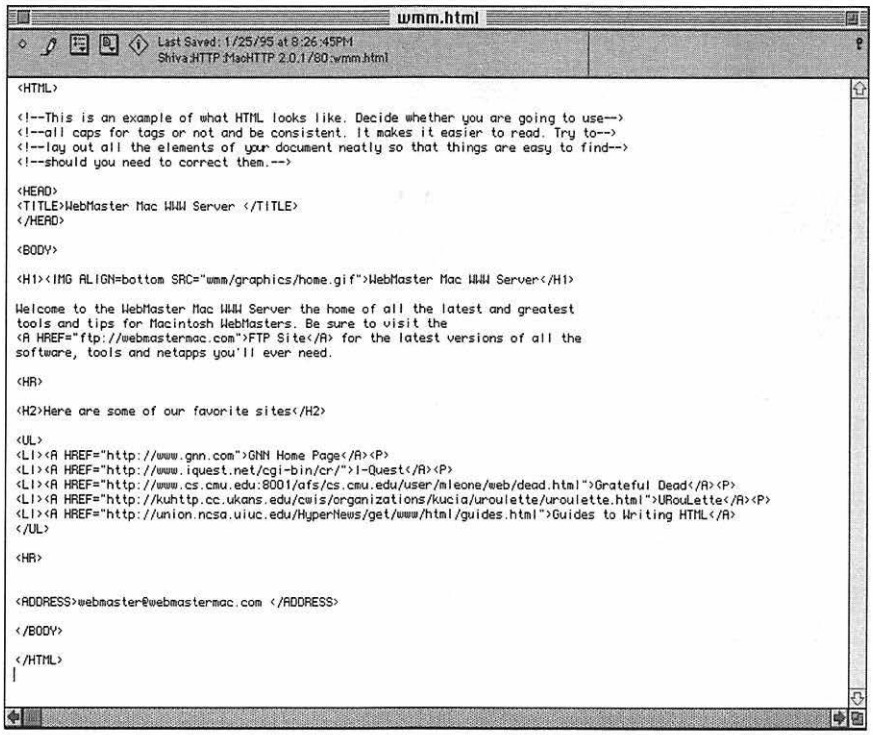


Figure 3.1: Sample HTML Home Page (Text).

If this is the first time you've looked closely at HTML, don't flip out. You'll get over it. We did. The best thing you can do to learn HTML is, as the Nike commercials say, "Just Do It." We make life easier by including many of the HTML examples in the book on the CD.

Go ahead and type everything you see in Figure 3.1. You can leave out the lines that begin with <! if you're a lousy typist. Those are called *comment lines* and won't appear to a Web browser unless a user views your source HTML. They're used to make notes to yourself in your HTML document. Start each line of comment with <! and end it with >. (You just learned how to code a comment line in HTML. That wasn't so bad, was it?)

After you've typed out or copied over all that HTML, save the file as **wmm.html**. Now fire up your Web browser (you don't have to be on the Net to do this) and open the file named **wmm.html**. It should look something like Figure 3.2 (our browser-of-choice is Netscape, by the way).

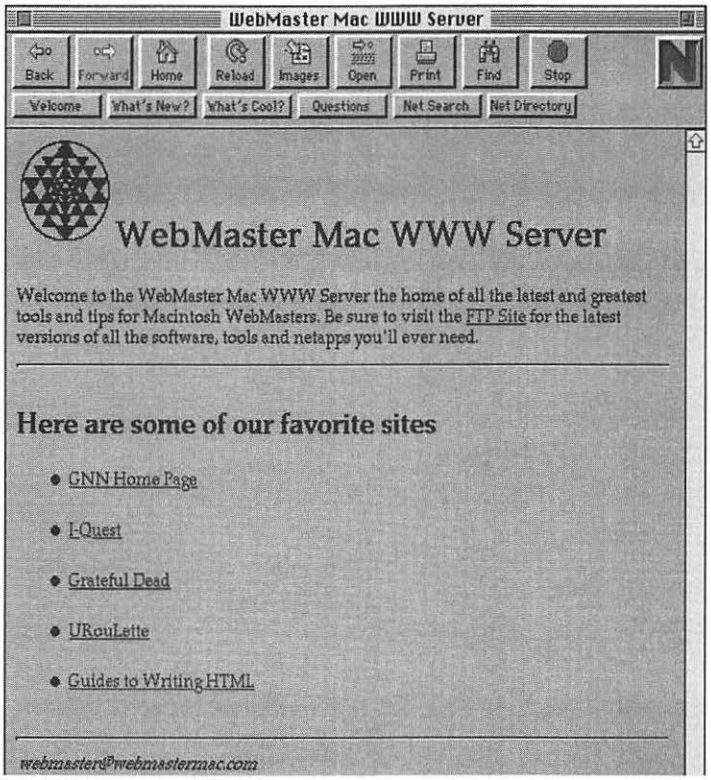


Figure 3.2: HTML Code in Figure 3.1 Viewed with a Browser (Netscape).

Whenever you are coding HTML, it's a good idea to have your Web browser open to check your work. In fact, it's a good idea to take a look at your HTML with *as many popular browsers as you can*. Take a look at it on a Mac, a PC, and on a UNIX machine if possible.

OK. That's a quick look at creating HTML code and then viewing it. Now let's get serious...

TAGS (YOU'RE IT)

There are three main components to HTML: tags, attributes, and URLs. By way of general definition:

- Tags tell the Web browser how to display text.
- Attributes tell the tag what action to take and are different depending on the Tag with which they are associated.
- URLs are used to specify the location of linked files and indicate the type and address of resource being accessed (e.g., WWW, Gopher, WAIS, FTP, etc.).

In this section we'll talk about tags — technical tags, formatting and styling tags, and even logical tags. Mostly, tags tell the browser how to display your words (and pictures, but we'll do pictures a little later).

There are all sorts of tags. Tags usually come in pairs and are bracketed by the < (less than) and > (greater than) signs. Of the pair, one tag starts the action and the other ends it. For instance, take the line:

<TITLE>WebMaster Mac WWW Server </TITLE>

You spotted the set of tags, right? <TITLE> and </TITLE>.

> **Just in Case: Upper, Lower, or Mixed Case?**
>
> Tags aren't case sensitive. You could just as easily use **<title>** and **</title>**. We recommend you use upper case however; we think it makes tags easier to identify and is the accepted HTML standard. Most plug-ins and HTML editors we've seen are also doing it that way.

Those Wild, Wacky Technical Tags

There are a couple of basic markup tags that every HTML document has that have nothing to do with how the content of the Web page is displayed. These tags are ones that give important information to the browser.

The HTML Tag

The first such tag is the document identifier tag, **<HTML>** and **</HTML>**. This tag tells the Web browser that the document is an HTML document. Everything else that goes to make up a Web page is nested inside this tag. Some browsers don't require that you have the **<HTML> </HTML>** tag in

your Web page. Pretend they do. It will probably be important in the future. Quite a bit of HTML markup is like this: It doesn't matter now but it may in the future. There are also a bunch of ways you can use different tags to make the browser display your HTML the same way. We're only going to show you the way that works and is the right way, not the way that just works.

The HEAD and TITLE Tags

Every HTML document should also have a Title tag, **<HEAD>** and **</HEAD>**. You can put comments in the header if you want using the **<!** and **>** tag. Other than that, the only thing that usually goes between **<HEAD> </HEAD>** is the Title tag, **<TITLE>** and **</TITLE>**. The title (and head) is used for document identification. It is what appears in a browser's title, its hot list, and when View History is selected. There should only be one Title tag per document. Choose a descriptive title — one that is short and to the point.

Right:

```
<TITLE>WebMaster Mac WWW Server</TITLE>
```

Wrong:

```
<TITLE>My Home Page</TITLE>
```

And keep it as short as you can: Depending on the browser, a long title may get truncated when displayed in hot lists and menu items like View History.

The BODY Tag

The fourth type of technical tag that gives information to the browser and doesn't affect the content of your document is the **<BODY> </BODY>** tag. It's another tag that isn't required by most browsers to do their thing... yet. This tag indicates the beginning and end of the document contents you are going to be serving up.

Technical Tag Recap

Here are the pieces of an HTML document that we have learned so far; you should consider all except the comment tag — <! > — to be required.

```
<HTML>
<! >
<HEAD>
<TITLE> </TITLE>
</HEAD>
    <BODY>
    </BODY>
</HTML>
```

Formatting and Style Tags

Are you beginning to get a feel for how formal HTML is? You better have interesting content, because it ain't gonna be the HTML that keeps people coming back for more. There are a few things you can do with HTML to dress up your pages and play with the visuals, but not many.

Heading Tags

Heading tags are written with the format **<H1>** and **</H1>** and are of types H1 to H6 with descending font size and emphasis for each type. It is completely up to the browser's configuration as to how an H1 header, an H2 header, and so on, look. Generally H1 will be seen as the largest, boldest text, H2 somewhat smaller, and so on. Try to use heading tags, at least two or three levels deep, to organize your documents.

Let's take a look at Figures 3.1 and 3.2 again. Do you see the HTML line that begins with **<H1>** and ends with **</H1>**? (Ignore the IMG ALIGN stuff for a sec — that's an attribute; we'll cover them in full in a few pages.) Find the line in Figure 3.1 that begins with **<H2>**. Now take a look at Figure 3.2 and see how that line appears in our browser.

By the way, it's not uncommon for the text that appears between the **<H1>** and **</H1>** the first time to be the same as that which appears between **<TITLE>** and **</TITLE>**. It may even be a good idea to do it this way. You decide.

OK, so that's the story on the header markup stuff. It's kind of like an outline. Play around with it a bit if you like, creating some text with each of the tags, **<H1>** and **</H1>**, **<H2>** and **</H2>**, and so on. You'll see that it's pretty straightforward.

The Paragraph, Line Break, and Rule Tags

Two things browsers don't care about when they see an HTML document are white space and carriage returns. If you want to separate paragraphs of text, use the Paragraph tag, **<P>**. Put it at the beginning or end of the text you would like to separate as a distinct paragraph.

If what you want is a line break, not a new paragraph, use the Line Break tag, **
**.

One more quickie and then we'll show you another example that includes most of what you know so far.

If you want to have a line under some text or use a line to separate text you use the Rules tag, **<HR>**. The **<HR>** tag also causes a paragraph break. Keep that in mind when you are using it.

The **<P>**, **
, and **<HR> tags don't require the corresponding ending tags of **</P>**, **</BR>**, and **</HR>**.

We're starting to see **</P>** being used at the beginning and at the end of a paragraph; this usage forms a "container" of text in HTML+, the successor to HTML. This is yet another one of those tagging conventions that doesn't matter now but probably will in the future when the next iteration of HTML is out and about.

Recap II

So here's where we're at. You've got the following tags down:

<HTML> </HTML>
<HEAD> </HEAD>
<! >
<TITLE> </TITLE>
<BODY> </BODY>
<H1> </H1>
<H2> </H2>
<P> </P>
**
**
<HR>

Figure 3.3 is an example of what it all looks like so far.

```
<HTML>

<HEAD>

<!--This is an example of the use of the Header, Paragraph,-->
<!--Break and Rules tags-->

<TITLE>American Diabetes Association</TITLE>
</HEAD>

<BODY>

<H1>Diabetes Texas WWW Server</H1>
<H2>American Diabetes Association<BR>
9430 Research Blvd. Austin, Texas</H2>

<HR>

The American Diabetes Association, Texas Affiliate is the leading diabetes-related
volunteer organization in Texas as well as the authoritative source of information
for the Texas diabetes commmunity - for people with diabetes, their families and
their health care providers.<P>

<H2>Current Research</H2>


<UL>
<LI><A href="ada/adaresearch/gennid.html">The GENNID Study</A>
<LI><A href="ada/adaresearch/dcct.html">DCCT</A>
<LI><A href="ada/adaresearch/type1trial.html">Type I Prevention Trial</A>
<LI><A href="ada/adaresearch/adafunded.html">ADA Funded Research</A>
<LI><A href="ada/adaresearch/volopps.html">Volunteer Opportunities</A>
<LI><A href="ada/adaresearch/news.html">Addtional Research News</A>
</UL>


<HR>
<ADDRESS>webmaster@dtx.org
</ADDRESS>

</BODY>

</HTML>
```

Figure 3.3: Using the Header <H1>, Paragraph <P>, Line Break <BR> and Rules <HR> Tags (Text).

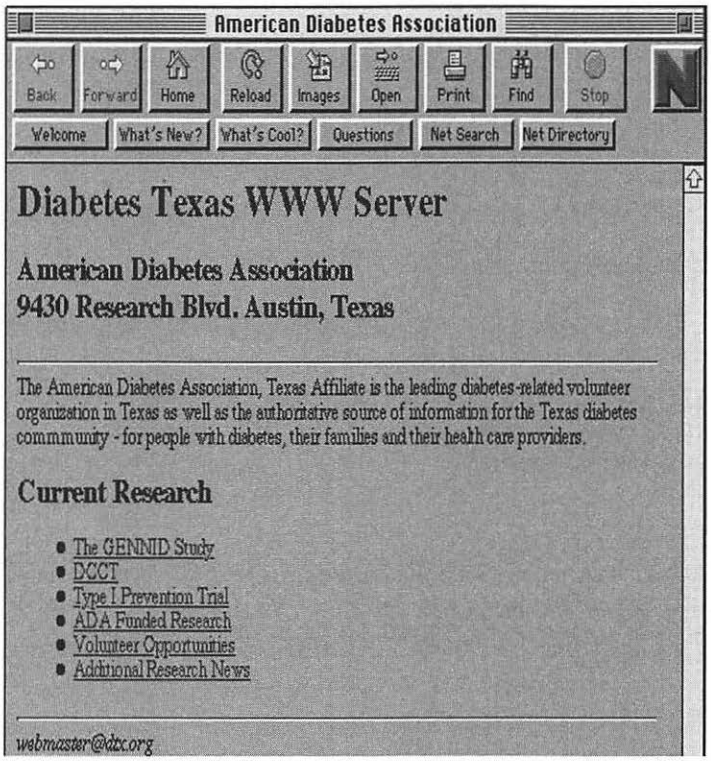


Figure 3.4: HTML code in Figure 3.3 Viewed with a Browser (Netscape).

Logical Tags

If you want to play around with text attributes in a block of text and make it bold or italic there are a couple of tags to use. Many WebMasters have been using the Bold <B> </B> and Italic <I> </I> tags as well as the Underline <U> </U> tag to accomplish this. Don't. If you have, consider changing your evil ways. This is one of those cases where we're going to teach you the right way to do it.

Italic, Bold, and Underline tags are referred to in the HTML world as "physical" styles. Physical styles are bad unless you have a good reason for using them. Instead, use what are referred to as "logical" styles and use the Strong <STRONG> </STRONG> tag for bold and the Emphasis <EM> </EM> tag for italics, instead. Logical tags help enforce consistency

in your documents and offer the browser the flexibility to display such tags with the user's preference.

In our humble opinion, you should never use the Underline <U> </U> tag. Just Don't Do It. Why? 'Cause the Underline command creates massive confusion, since most browsers display hypertext links with an underline.

Logical Markup Codes

Here's the complete list of logical markup types, their names as well as what they are used for, and how they typically appear:

- **<CITE> </CITE>** = Citation; use for titles of books, films, etc. (Usually appears as italics when viewed in a browser.)
- **<CODE> </CODE>** = Code; use to show an example of computer code. (Usually appears as a fixed-width font when viewed in a browser.)
- **<DFN> </DFN>** = Defining instance; use for word being defined. (Usually appears as italics when viewed in a browser.)
- **<EM> </EM>** = Emphasis; use for emphasis. (Usually appears as italics when viewed in a browser.)
- **<KBD> </KBD>** = Keyboard input; use for user keyboard entry. (Usually appears as bold or plain fixed-width font when viewed in a browser.)
- **<SAMP> </SAMP>** = Literal characters; use for computer status messages. (Usually displayed as fixed-width font when viewed in a browser.)
- **<STRIKE> </STRIKE>** = Strike-out; use to show strike-out text. (Usually displayed as, you guessed it, text with a strike line through it when viewed in a browser. This tag is often used for legal documents.)
- **<STRONG> </STRONG>** = Strong emphasis; use for strong emphasis. (Usually appears as bold when viewed in a browser.)

Try these out on your own. Try them all. For the most part, you can just take the same text and use the different tags.

There's another physical style in addition to the Bold, Italic, and Underline we mentioned earlier. It's the infrequently seen Fixed Width <TT> </TT> tag. This tag gives a fixed-width typewriter text font. Like the other physical styles, we don't recommend it. Instead, use the logical tags **<CODE>** or **<SAMP>**.

The PRE, BLOCKQUOTE, and ADDRESS Tags

Sometimes it is necessary to display text just as you want it to look, in a preformatted manner. You can do this with the Preformatted tags **<PRE> </PRE>**. Other tags, attributes, and links can be used within the **<PRE>** tag. Figure 3.5 and 3.6 show an example.

```
<PRE>
                                      12 weeks ended

                                     July 3        July 4
                                      1994          1993


Sales                                $97,362       $79,518

Cost of good sold and
 occupancy costs                      65,871        53,773
Direct expenses                       23,716        19,075
Pre-opening costs                        384           416
Amortization expense                     261           194
General and administrative
 expenses                              3,464         3,084
Non-recurring expenses
 related to earthquake                    --            --

    Income from operations             3,666         2,976

</PRE>
```

Figure 3.5: Using the Preformatted Tag **<PRE> </PRE>** (Text).

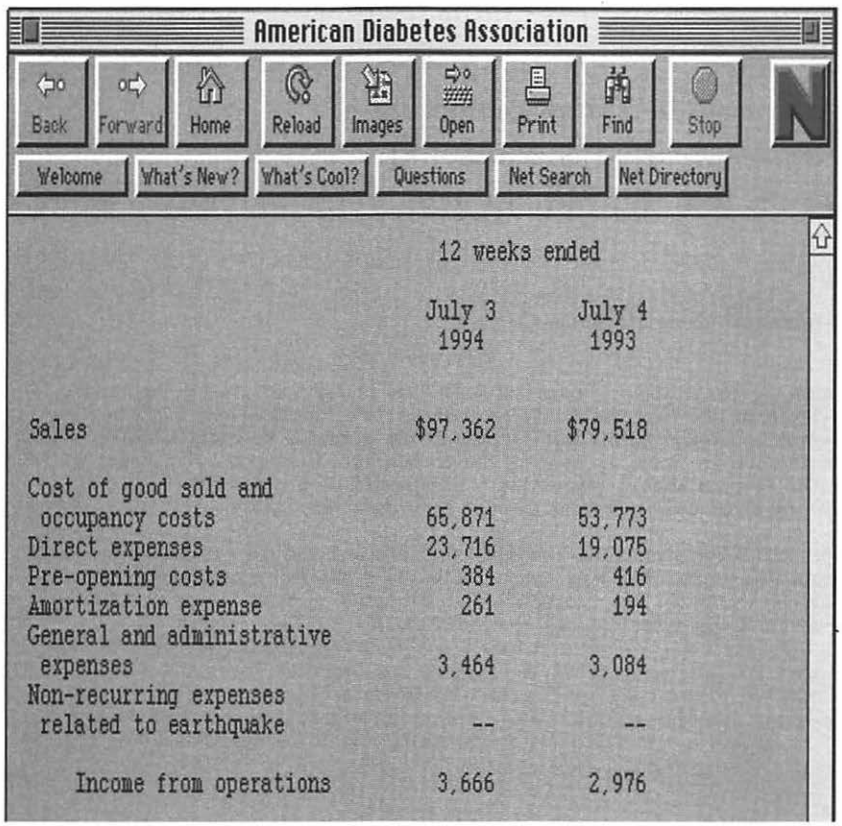


Figure 3.6: HTML Code in Figure 3.5 Viewed with a Browser (Netscape).

You can also use the Block Quote command **<BLOCKQUOTE> </BLOCKQUOTE>** to display quoted text. Figures 3.7 and 3.8 give an example of that.

```
<HTML>

<HEAD>

<!--This is an example of the BLOCKQUOTE Tag-->

<TITLE>American Diabetes Association</TITLE>

</HEAD>


<BODY>

<H1>Diabetes Texas WWW Server</H1>


Welcome to the Diabetes Texas WWW Home Page brought to you by the Texas
Affiliate of the Amercian Diabetes Association. The American Diabetes
Association, Texas Affiliate is the leading diabetes-related volunteer
organization in Texas as well as the authoritative source of information
for the Texas diabetes commmunity - for people with diabetes,
their families and their health care providers.<P>

<BLOCKQUOTE>The mission of the ADA is to prevent and cure diabetes and to
improve the lives of all people affected by diabetes.  Each year tens of
thousands of Texans are diagnosed with diabetes or find themselves facing
one of its many complications.<P></BLOCKQUOTE>

Our most innovative programs to date are the Diabetes Texas WWW Home Page
and our FirstClass based online service Diabetes Texas Online. Both are
timely and powerful ways to obtain reliable information and referrals as
well as access our extensivedatabases using the Texas Resource Directory.<P>


<HR>


<ADDRESS>webmaster@dtx.org</ADDRESS>

</BODY>

</HTML>
```

Figure 3.7: Using the <BLOCKQUOTE> </BLOCKQUOTE> Tag (Text).

The ADDRESS Tag

There's one last tag that doesn't fall into any family — the Address tag <ADDRESS> </ADDRESS>. Normally this is found at the end of page and holds contact information such as **webmaster@webmastermac.com** (see Figures 3.7 and 3.8).

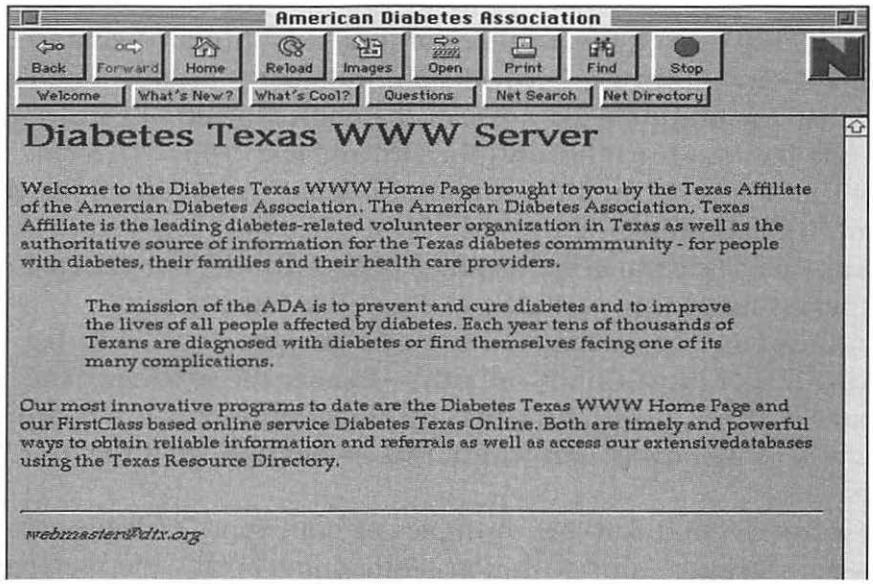


Figure 3.8: HTML Code in Figure 3.7 Viewed with a Browser (Netscape).

Entities and ISO Latin-1 Characters

There is a small set of character combinations referred to in HTML as entities. Entities are used to represent characters that have special meanings in HTML such as <, >, &, and ". There is also a large set of character combinations called the ISO Latin-1 characters that is used to represent things like the ñ, ø, and so on. The list of both — entities and Latin-1 characters — can be found in the appendices at the end of the book.

By the way: Unlike the rest of HTML, entities and ISO Latin-1 characters *are* case sensitive.

Lists, Lists, and Lists — How Many Ways Can You Make a List Already?

OK, we're getting close to the end of the tag stuff. Lists are next — a tag element you can get creative with. Kind of.

HTML supports three flavors of lists: unnumbered, numbered, and definition. You can also do lists within lists, which can be a very good thing.

Let's create two different types of lists, the unnumbered and numbered. The only difference between an unnumbered list and a numbered list is that the unnumbered list uses the Unnumbered List <UL> </UL> tag while the numbered list uses the Ordered List <OL> </OL> tag. Items in the numbered list are then numbered; items in the unnumbered list are not. It's that simple.

The items listed in both instances begin with the List <LI> tag. Note that there is no need to include a closing </LI> tag.

You can use the Paragraph <P> tag or the Line Break <BR> tag — by inserting one or more, either, or both — to alter the spacing of the items in your list. You should know that by all accounts this is a completely bogus way of fooling around with HTML — but hey, what the heck, give it a try.

Figures 3.9 and 3.10 are examples of both types of legitimate lists and some playing around with nonstandard line breaks and spacing.

```
<UL>
<LI>capers
<LI>olives
<LI>olive oil
<LI>red wine vinegar
<LI>pepper and salt
<LI>cayenne
<LI>italian seasoning
<LI>garlic and red onion
</UL>


<OL>
<LI>capers
<LI>olives
<LI>olive oil
<LI>red wine vinegar
<LI>pepper and salt
<LI>cayenne
<LI>italian seasoning
<LI>garlic and red onion
</OL>


<A HREF="http://www.gnn.com">GNN Home Page</A><BR>
<A HREF="http://www.iquest.net/cgi-bin/cr/">I-Quest</A><BR>
<A HREF="http://www.txinfinet.com">MahaInfinet WWW Server</A><BR><P>

<OL>
<A HREF="http://www.gnn.com">GNN Home Page</A><BR>
<A HREF="http://www.iquest.net/cgi-bin/cr/">I-Quest</A><BR>
<A HREF="http://www.txinfinet.com">MahaInfinet WWW Server</A><BR>
</OL>

<UL>
<A HREF="http://www.gnn.com">GNN Home Page</A><BR>
<A HREF="http://www.iquest.net/cgi-bin/cr/">I-Quest</A><BR>
<A HREF="http://www.txinfinet.com">MahaInfinet WWW Server</A><BR>
</UL>
```

Figure 3.9: Using the Unnumbered <UL> and Numbered List <OL> Tags (Text).

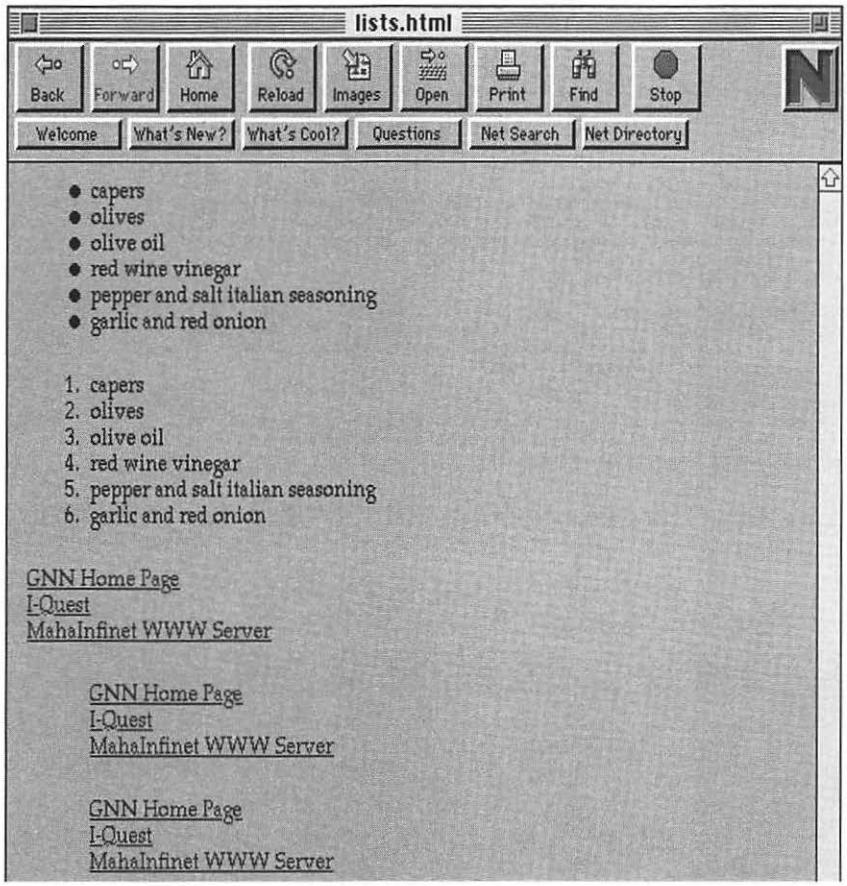


Figure 3.10: HTML Code in Figure 3.9 Viewed with a Browser (Netscape).

Another type of list is the definition list, which is sometimes called a "glossary." Definition lists consist of alternating a term and a definition. Within a definition list you can use all the formatting tags except the heading tag and you can nest unnumbered and numbered lists within definition lists.

You use the <DL> </DL> tag to start and end a glossary, the <DT> tag for the main entry or subject line, and the <DD> tag for the descriptive text itself. The <DT> and <DD> tags have no end tags.

You got all that? Figures 3.11 and 3.12 show a great example.

```
<H2>Diabetes Texas WWW Services</H2>

<DL>
<DT>Diabetes Information and Action Line (D.I.A.L.) Manual
<DD>The D.I.A.L. Manual contains answers to your most frequently
asked questions about diabetes, as well as referrals to relevant
publications.<P>

<DT>Texas Resource Directory
<DD>The Texas Resource Directory is an easily searchable database
with over 6,000 sources of information and assistance for Texans
affected by diabetes.  These resources can be found using dozens of
categories organized into major topic groups as well as by city,
county and ADA Chapter.<P>

<DT>Diabetes Texas Online
<DD>Diabetes Texas Online is a FirstClass online service that offers a
graphical user interface for both Mac and Windows-based PC users over
dial up modem lines throughout Texas.<P>

<DT>Publications Center
<DD>A comprehensive listing of the publications available from the
American Diabetes Association, Texas Affiliate and others including a brief
summary of each together with ordering information.<P>
```

Figure 3.11: The Definition List <DL>, </DL>, <DT>, and <DD> Tags (Text).

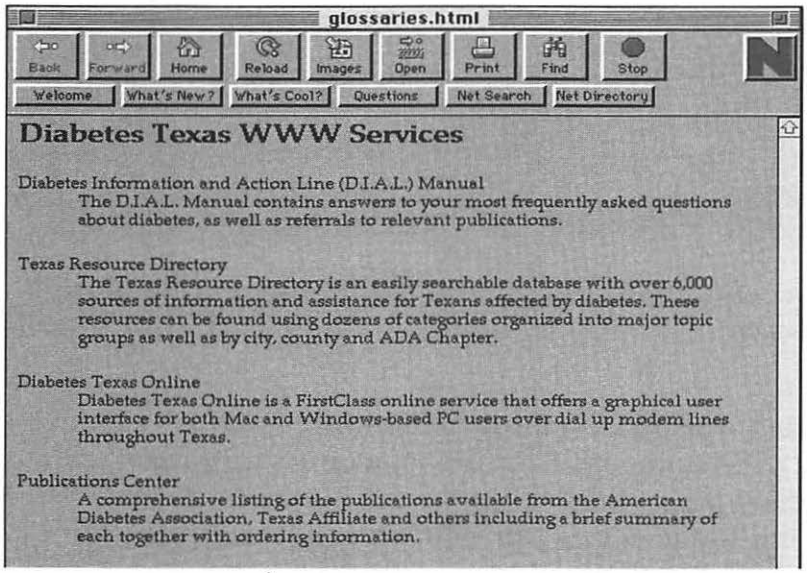


Figure 3.12: HTML Code in Figure 3.11 Viewed with a Browser (Netscape).

The last two tags you need to know are the Anchor <A> </A> tag and the Image <IMG> tag. Both of these tags require that you define what you want them to do. That's what attributes are all about and we'll talk about them right after we look at...

YOU ARE WHAT YOU L: THE LOW-DOWN ON URLS

We said that there were three parts to HTML: tags, attributes and URLs. We're done with tags. Next we'll knock off URLs. We'll cover the slightly more complicated subject of attributes in a moment.

You know by now that URL stands for Uniform Resource Locator (It's not Uniform Resource Location or Uniform Record Locator. It's Uniform Resource Locator. We've heard and seen all these and worse.)

The Web uses URLs to find files on servers. If you want to place links to other sites or files on the Net in your Web pages, you need to know how to use URLs.

A URL "address" has the following defined format:

scheme://host.domain [:port]/path/filename

Yum. It's that UNIX stuff, back to haunt us. You need to understand it, though, if you're going to do links properly. And since links are what it's all about, take a deep breath, and dive in.

Here's what that gobbledygook means. The scheme can be one of the following:

- **file://**
  a file on your local system or on an anonymous FTP server
- **http://**
  a file on a WWW server
- **ftp://**
  a file on an anonymous FTP server
- **gopher://**
  a file on a Gopher server
- **WAIS://**
  a file on a WAIS server

- **news://**
  USENET newsgroup
- **telnet://**
  a connection to a Telnet-based service

The host.domain part refers to the machine name (if any) and domain name of the server you want to link to.

The [:port] part of the URL syntax is usually not included when you create a link. (The default port for WWW servers is 80, by the way.) But it's part of the official spec, so we had to tell you about it. Unless you see it indicated in the URL of a place you want to link to, leave it out.

The path/filename is everything after the domain name (and port, if any). It shows what directories or folders you have to nest your way down through to get to the file you want to link to. Sometimes all you'll want to link to is the "Home Page" of a WWW server, or the top-level directory of an FTP site, for instance. In that case you don't have to concern yourself with the path and filename.

The Absolute Theory of HTML Relativity

There are two types of links that are often confusing to beginning HTMLers — the relative and the absolute links.

A relative URL is a URL that doesn't contain all the parts of an absolute URL — such as the type of scheme (**http://**, **ftp://**, **telnet://**, etc.) and host name and full path to the document. You can safely use relative links to documents when the linked document is in the same directory or when the document is on the same level as the server itself. An example of using a relative URL is:

```
<A HREF = "bestsites.html">My Hostlist</A>
```

An absolute link to that page would be:

```
<A HREF = "http://webmastermac.com/
bestsites.html>My Hotlist</A>
```

ATTRIBUTES

To create a link you need to use the Anchor <A> </A> tag. Anchor tags are used when you want to link to another location or when you want to use hypertext links to navigate through large documents. The Anchor tag requires an attribute to tell it exactly what to do. There are two attributes for the Anchor tag: the HREF attribute and the NAME attribute. Let's take HREF first.

HREFs

The following format is used for an anchor tag with a HREF attribute (the spaces before and after the "equals" signs aren't necessary):

```
<A HREF = "filename">HYPERTEXT</A>
```

For example:

```
<A HREF = "http://www.wholefoods.com/wf.html">Whole
Foods</A>
```

Notice how the beginning of the Anchor tag isn't just <A>. The whole thing up to the "w" in "wholefoods" is the first part of the Anchor tag. Then comes the word or words that the user will see as a hypertext link with their browser (e.g., **Whole Foods**), followed by the rest of the Anchor tag, </A>. When the user clicks on the words **Whole Foods** on the screen, the browser will surf to **http://www.wholefoods.com/wf.html**. Neat, huh?

Just stare at it a couple of times if you didn't get that the first time through.

NAMEs

The other type of attribute to use with an Anchor tag is the NAME attribute. You use a NAME attribute and an HREF attribute together to make it easier and more interesting and fun to navigate through large text documents. The format for an anchor with NAME attribute is as follows:

```
<A NAME = "name">TEXT</A>
```

For example, say you have a long text file named "WF.food.html" (Document A) with a section we want to link to from another document (Document B). You use the NAME attribute as follows:

```
<A NAME = "cafedeli">
The Whole Foods Cafe and Deli satisfies the concerns of even the
pickiest eaters at a glance. Easy-to-read nutrition cards itemize
calories, fat, protein, carbohydrates, sodium, and cholesterol, and
indicate food exchange equivalents. You get fresh food fast at the
Whole Foods Deli. Whether you eat it on the run or in our Cafe,
you'll enjoy fantastic flavors and creative combinations of natural
foods.</A>
```

In this example, you define the section of Document A, beginning with the words "The Whole Foods Cafe" as the place that will be linked to from another document.

On another document about Whole Foods (or any document for that matter) we create a link to the "NAMEd" part of Document A by using an anchor with the HREF attribute. Wherever we want a link to the section that has been "NAMEd", all we have to do is follow the format:

```
<A HREF = "linked file" NAME = "#name">Hypertext Anchor</A>
```

For example:

```
<A HREF = "WF.food.html"#cafedeli>Cafe and Deli</A>
```

If the link to a "NAMEd" section is within the same document (e.g. from one place in Document A to another place in Document A) it is not necessary to include the "linked file" part of the HREF attribute. In that case, do this:

```
<A HREF = #cafedeli>Cafe and Deli</A>
```

Don't forget to include the # mark before the text used to define the NAME! Oh yeah, we almost forgot. The text you use to define your NAME has to be a one-word name.

Whew!

If you got that first time around either your mind is a sponge or we are beyond awesome... and less than humble. It's a bit tricky. Anchors with the NAME attribute are probably the hardest part of HTML to grok at first. Whenever we use them we always go back and read up on them quietly. There's no point memorizing this stuff when you have a book to help you, right?

Ultimately, practice makes perfect, so play around with anchors and attributes a bit before we continue with images.

IMAGES <IMGS>

Graphics are a cool addition to your Web pages. We'll show you the things you can do with HTML to place your graphics on your page and even show you how to use a picture as a link.

Just for laughs, you should know that if your graphics stink we won't be able to help you. Neither one of us can draw to save our lives. Hopefully you, or someone you know, can create decent graphics.

It's not just a matter of taste by the way. Keep in mind that not everyone has an ISDN line to their house or office. Most people seeing your Web pages will be doing it via 14.4 or 28.8 modems. Put another way: If you have graphics that are huge it's OK if they are ugly, because no one is going to wait for them to load anyway.

Inline vs. External (Linked) Images

You can present graphics on your page as either inline or linked. Inline images are images that appear within the page, automatically. Linked images are images that stand alone and are linked to and reside elsewhere — either on the same server or elsewhere on the Internet. You have to click to see a linked image; an inline image appears automatically when a browser comes across it (assuming that preference is turned on in the user's browser). Inline images must be either in GIF or XBM format since that's all the image types that HTML currently supports.

In general you want to use external images when the images are too large to be inline. We think that anything over 100K is way too big to be

an inline image — unless your site is a big-time image site and you warn people in advance. If it's such a great image and you really need it in order to get your message across, make it an external image so that it doesn't slow down the loading of your text.

There are really only a couple of other reasons to link to an external image. An image may need to be linked as an external image if it resides on another server for reasons such as Web server performance optimization or storage location, or because the image is not a GIF or XBM image. Remember that when you link to other types of graphic images such as JPEG, you are dependent on the user having the helper app that will launch to show it. It's tough nugies for the user who doesn't have the helper apps and pity on those who use a browser — like Lynx — that doesn't display images at all.

Inserting Inline Images

You can insert an inline image by using the Image **<IMG>** tag. There's no end tag involved. It's just **<IMG>**. The **<IMG>** tag is another one of those tags that requires an attribute. In this case it's the **SRC** (source) attribute, which is used to define the filename of... the source! The **SRC** attribute can be any URL. The format for the **<IMG>** tag is:

```
<IMG SRC = image_URL>
```

For example:

```
<IMG SRC = sriyantra.gif>
```

Remember, you have to use the **.gif** or **.xbm** if the file is a GIF or XBM file and you want it to load as an inline image.

Inserting Linked Images

If you wanted to create a linked image, linking to a graphic stored on another server, it would look something like this:

```
<A HREF="http://www.txinfinet.com/gifs/sriyantra.gif">Wealth
and Prosperity</A>
```

There are a couple of other attributes you can add to the <IMG> tag that define how text is displayed after an image. As HTML formally stands, you can use the ALIGN attribute and position text adjacent to an image using values of TOP, MIDDLE, or BOTTOM. The formats for doing this are as follows:

<IMG ALIGN = TOP SRC = bob.gif>My name is Bob Levitus.
<IMG ALIGN = MIDDLE SRC = bob.gif>My name is Bob Levitus.
<IMG ALIGN = BOTTOM SRC = bob.gif> My name is Bob Levitus.

Figure 3.13 shows the HTML and Figure 3.14 shows how those three examples turn out on the screen.

```
<H2>Will the Real Bob Levitus Please Stand Up</H2>


<A HREF="bobsbio.html"><IMG SRC="bob.gif" ALIGN=TOP></A>My name is Bob Levitus<P>
<A HREF="bobsbio.html"><IMG SRC="bob.gif" ALIGN=MIDDLE></A>My name is Bob Levitus<P>
<A HREF="bobsbio.html"><IMG SRC="bob.gif" ALIGN=BOTTOM></A>My name is Bob Levitus<P>
```

Figure 3.13: Will the Real Bob Levitus Please Stand Up (Text).

If for some sick reason you feel you need to do something for the Lynx users out there (Lynx is a semi-ancient text-only browser rarely used these days), you can use the ALT attribute to show some text as an alternative to the image. You can let them see the words "meow" instead of a picture of Socks, the White House cat and most frequented part of the WWW server at the White House (**http://www.whitehouse.gov**)

The format for the ALT attribute that goes with the IMG tag is:

<IMG SRC = "cat.gif" ALT = "meow">

Finally, there are plenty of tricks in the following sidebar, which was written for us by one of our favorite WebMistresses, Julie Gomoll.

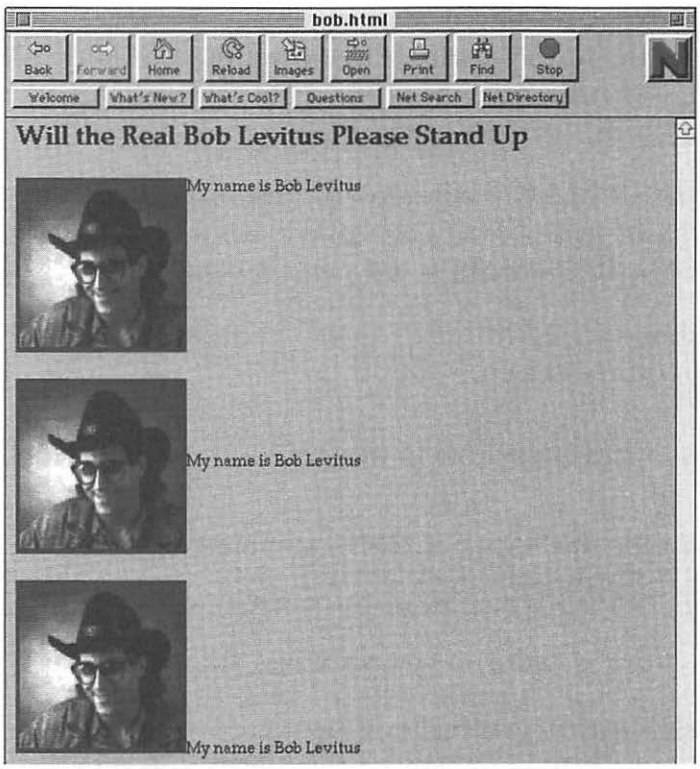


Figure 3.14: HTML code in Figure 3.13 Viewed with a Browser (Netscape).

Designing for the Web
by Julie Gomoll, President of Go Media

Your computer screen is not a piece of paper. Most printed material is presented in a fashion that reflects decades of graphic design knowledge. We know what point size and leading make for the most readable novel, and we know that the brain assimilates information differently when it encounters sans serif type on glossy white paper than it does bright red type on neon

yellow paper. Photographers know that part of the impact of a picture assumes the viewer will be seeing it at a certain size, from a specific distance, under controlled lighting. Information in a brochure is designed with the confidence that the reader will approach the material in a linear fashion, starting here and ending there.

Needless to say, the Web is another story. People are designing Web pages as though the screen is just another print medium. Online magazines are laid out like news-rack publications. Electronic catalogs appear to be direct adaptations of their printed counterparts.

Think about the information you are designing. First of all, the amount of information you can present in a square inch of screen real estate is practically nothing compared to the amount you can put in the same amount of paper space. You're limited in the fonts you can use, and you can't assume everyone will be looking at your site using the same quality monitor you've been using. Your audience may not even be bothering to load your graphics. You can count on the fact that if you're using more than a couple of hypertext links, users at the other end will be meandering through your masterpiece in a manner that makes sense to them. So—a few tips:

- Keep your home page simple. Give the viewer a chance to choose the area at your site that is most interesting. Don't assume they'll be willing to scroll through several feet of screen in hopes of finding something interesting.
- If you use a clickable map, offer a text version of the same links for the poor souls trying to surf with their trusty old 9600 baud modems.
- Put the size of the graphic in parentheses near the icon, so users know what they're in for if they decide to look at it.

- Offer a return route. Dead-end pages can be a pain, and some folks bounce around in such a way that the "back" button in their browser isn't a very efficient option.
- Keep your graphics small. (More on that in a moment...)

A Couple of More Fun Things Before We Leave

There are a couple of fun things you can do with anchors, graphics and external links that we'd like to touch on before turning you loose to try some of this out on your own.

You can make a graphic an anchor for a hypertext link. You already know all the pieces of HTML markup that are used to do this. But if you're like us, it's easier if we just show it to you. Here it is:

<A HREF = "CyberSurfing.html"><IMG SRC="ganesh.gif"</A>

If I click on the image of Lord Ganesh, it takes me to the document called CyberSurfing.html. You can see an example of this function on the examples page of the WebMaster Mac WWW server.

The other cool thing you can do now that you know the syntax for external images is to create links on your documents to other types of externals. The only difference is the file type of the linked file. Here's an example:

<A HREF = "MyFavoriteSound.au>Listen to this!</A>

(As our friend Carl pointed out, this is also an example of bad form. It's almost as bad as a button that says "Click here!" Say what the sound is. At least be a little more creative than "Listen to this!")

You can link to all sorts of things — audio, QuickTime movies, graphics, or sounds (as shown above). You can have loads of fun with linked files. Try some of these:

Linked File Types and Their Extensions

- Plain text = **.txt**
- HTML document = **.html**
- GIF image = **.gif**
- TIFF image = **.tiff**
- XBM bitmap image = **.xbm**
- JPEG image = **.jpeg**
- PostScript file = **.ps**
- AIFF sound = **.aiff**
- AU sound = **.au**
- •QuickTime movie = **.mov**
- MPEG movie = **.mpeg** -or- **.mpg**

Just remember: As we mentioned earlier, not everyone will have the proper external apps — such as a JPEG viewer or sound player — to enjoy these multimedia links. But hey, we say push the envelope and leave it up to them to get the helper apps they need. (If you want to be a really cool WebMaster, give them a link to an FTP site that has your favorites.)

HTML: TOP 10 TIPS AND AVOIDING COMMON ERRORS

There are all sorts of pitfalls to avoid when you're doing HTML. Some of them are technical, others are aesthetic. We're going to zero in on the technical ones — the ones where you screw up and write some HTML that either doesn't come out right when you look at it with your browser, or is impossible to decipher when you go back to look at months from now.

It's kind of a rule with HTML. Everyone gets nailed eventually. You will too. But rest assured, the more you play with HTML the less frequently it'll happen to you.

In a blatant attempt to save you at least a bit of time and trouble, here's our Top 10 list of tips and things to avoid:

10. Avoid improper use of the <P> tag to force a line break and carriage return.

Don't use the <P> tag before or after tags that carry their own line break such as <HR>, <PRE>, <ADDRESS>, <BLOCKQUOTE>, <UL>, <OL> and the Header tag <H1....H6>. You'll end up with extra white space on your page.

9. Be consistent with your styles.

Use uppercase in your HTML tags. Even though you still can get away with mixing cases, use uppercase. It may even be important in the future. Be consistent. Add some space in your HTML code when you use a <P> or a <HR>. (Browsers ignore white space!) And try to keep list items on single lines in your HTML code. Habits like these make it easier to read your HTML and you'll be less likely to forget the ending tags if your code is easy to read.

8. Don't forget the trailing semicolon after entities and ISO Latin-1 characters.

Remember — if you need to use the & entity to display the & (ampersand) to a browser, don't forget the semicolon that goes after the letters amp.

It's

```
&amp;amp;
```

not

```
&amp;amp
```

This rule also applies to ASCII or other character combinations used to represent the Latin-1 characters.

7. **Use fully qualified domain names in URLs.**

   When you use a URL in a link be sure to use the entire fully qualified domain name (or IP number if that is all there is). Not everyone in the world is on the local network your Web server is on. You can't just give the link to the machine followed by the path to the document. For instance, if you were to use the link:

   <A HREF="http://www/~sonja/graphics">Sonjya's Graphics</A>

   only the in-house troops on your local-area network will be able to see Sonjya's awesome design work. You need to use the whole domain name to let the world in:

   <A HREF="http://www.gomedia.com/~sonjya/graphics">Sonja's Graphics</A>

6. **Avoid improper use of relative and absolute links.**

   In general, it is a good idea to use relative links — if you know what you are doing. They save typing because they're shorter. They also make it easier to move groups of HTML documents to another server since the path names to the documents are... relative, so you don't have to type in a new domain name for every linked document you are serving up.

   The rule in a nutshell is to use absolute links when you are linking to other sites or sets of documents that are not grouped logically together.

5. **Avoid improper use of the trailing / slash on URLs**

   Here's one that no one seems to know about outside the orbit of the HTML and HTTP gods. Sometimes you see a trailing slash (/) at the end of a URL. Contrary to what we heard at first, this has nothing to do whether or not the WWW server is a UNIX box or that a trailing slash forces the server to return a file named index.html. The fact that some servers generate an index of a directory is a result of NCSA's concession to gopher administrators. It's not an accepted URL standard. It's completely up to the server how to interpret URLs ending in slash.

If you really do want the index.html or if you want the server and directory you link to to return the contents of the directory on the fly, use the trailing slash at the end of the URL. Otherwise don't.

The URL:

http://www.gomedia.com/graphics

refers to a file named "graphics." The URL:

http://www.gomedia.com/graphics/

refers to either the default.html file or a directory index, depending on how the server interprets it.

Though most WWW servers will redirect you to the proper URL, not all browsers support such transparent redirection. Once again, we suggest you do it right in the first place.

4. **Use headings properly.**

Be sure to use only heading levels that are one level below the level above. In other words, go from **<H1>** to **<H2>**. Don't follow an H1 heading with an H3 unless that's part of your design. In general, don't skip a level if you are using multiple heading elements. Besides being bad style, it is strictly verboten by Herr Shultz and HTML specification.

3. **Avoid missing quotes in URL links.**

   Here's Jeff's biggie: Don't forget both the start and end quotes in your URL links — before and after! It's:

   <A HREF="http://www.gomedia.com">

   not

   <A HREF="http://www.gomedia.com>

   or

   <A HREF=http://www.gomedia.com">

   Look carefully until you see the difference. Only the first example will work. Try it for yourself and see what happens.

2. **Don't forget end tags.**

   Here's Jeff's other biggie: Don't forget the end tags. Actually, Jeff doesn't usually forget the end tag. He forgets the / in the end tag.

   Right:

   <H1>My Favorite Web Sites</H1>

   Wrong:

   <H1>My Favorite Web Sites<H1>

   You usually won't goof on one like this one. It's the ones where you have a long list of URL links that catch you. That's usually where you'll forget to use a / to close the tag. Those are also the links in which you forget the quotes in the URL link address — especially when you have a big old list of 'em.

   We guess the moral of the story is to test your links thoroughly before unleashing your pages on the World (Wide Web).

   And our number one tip?

1. **Buy BBEdit and use the HTML extensions so you don't have to type so much HTML! Or at least check out the demo version.**

   'Nuff said. It's wonderful.

One More Thing (Fixing Anchors)...

We didn't want to upset the symmetry of our Top 10 list but we do have one last piece of advice for you. Whenever you are doing an anchor, be sure you open and close the quotes of the HREF attribute, be sure you have all the less than (<) and greater than (>) symbols for the Anchor tags, and be sure you have the forward slash (/) before the closing Anchor tag. Nine times out of ten, one of those points will be the reason your HTML doesn't look right when viewed with your browser.

PARTING SHOTS

You can go off on all sorts of esoteric tangents talking about HTML. If you really want to tap into the HTML scene go visit the HAL Computer Systems HTML validation server at **http://www.hal.com/users/connolly/html-test/service/about.html**. You can submit your HTML to their server and automatically get back a response telling you whether or not you've done it right. If you pass you are an absolute ace, an HTML guru.

The other thing you can do if you want to go deep in to the transcendent with HTML is read an HTML book such as *HTML Programming For Dummies* by Ed Tittel (IDG Books, 1995) or *Publish it on the Web* by Bryan Pfaffenberger (AP PROFESSIONAL, 1995).

You don't have to memorize everything in this chapter but you have to know how to do it to be a Master. You know where to turn if you forget. Feel free to take a highlighter to this section of the book if it helps (hey, you paid for it already). It reminds us of the old joke:

Q. How do you get to Carnegie Hall?
A. Practice, practice, practice.

Today, it's more like:

Q. How do you become a Master-Blaster-Ultra-WebMaster?
A. Practice, practice, practice.

The point we're making is that it's a lot easier to do than it is to read about. For best results, if you haven't already, try each of the examples in this chapter. We deliberately kept the examples short to encourage you to experiment with your own stuff, not just type in our boring example text.

Just take a byte at a time (groan) and learn to enjoy it. But learn to enjoy it soon, because we'll be delving much deeper into HTML when we tackle forms and .cgi. So before you move on, be sure and crank out some HTML.

At the very least, we'd like to see you get a simple home page together so you can use it in the next chapter. It will be much more fun (and far more productive) if you're working on your own stuff when we show you how to configure and optimize a Web server.

Chapter 4

MAXIMIZING YOUR WEB SERVER

Bells, Whistles, and Screaming Daemons

It's showtime! Time to pull it all together and bring your server online. In the last chapters we brought you along and showed you how to get your server connected to the Net, how to fire it up, and how to bang out a bit of HTML. In this chapter we're going to tweak the configuration of your Web server and drop in all that HTML that you've slaved over late at night. We're also going to show you how to set up and use WebStat to monitor how much and what type of traffic your Web server is getting. And since you've been so patient, we'll even show you how to create transparent GIFs, interlaced GIFs, and clickable maps.

THE MACHTTP.CONFIG FILE REVISITED...

The first thing we need to do is go back and take a look at that **MacHTTP.config** file. Go grab it and open it up with your favorite text editor. It's in the MacHTTP Software & Docs folder along with MacHTTP. We'll hold...

OK, you back? It's time to read the fine print. You need to do this now. Sorry. Unlike a lot of things on the Mac, this task absolutely requires you to read the documentation. Go slowly. Read it twice, even.

Don't forget, everything after a # symbol is a comment and everything without a # is part of the actual configuration code for your server. Chuck has done a good job of describing everything clearly, thank heavens. Here's how it read at press time. (If you have a later or different version than the one included with this book, you may see something slightly different.):

```
#MacHTTP Configuration file, v. 2.0.2tm
#
#The format of this file is free form, with a few exceptions. Lines not
#starting with a recognized keyword are ignored. Note, all entries are
#converted to upper case by MacHTTP, so the config file isn't case sensitive,
#with the exception of Mac file types and creators below. Note that any text
#styles like bold or underline in this file are purely cosmetic and are only
#intended to make the file easier to read. The styles are not required. Legal
#config file keywords will appear as BOLD text if this file is viewed with
#SimpleText or TeachText.

# The version number below must match MacHTTP's version number
VERSION 2.0.2tm

##########################################################
# "Special" Files
#
#The following line defines the default file type if a suffix match isn't
#found. The syntax is: DEFAULT <default transfer type> <default MIME type>
DEFAULT TEXT text/html

#The following lines specify where to find HTML files for error messages, the
#default home (or index) page, the name of the log file, and the message
#returned for security violations. Any of these three file directives point
#to a HTML document, script, or CGI application.
#
#NOTE!!! INDEX must be a simple file name, not a path like the other files.
INDEX    Index.html
ERROR    :Error.html
NOACCESS :NoAccess.html

#If the LOG file directive is missing or commented out, no logging will occur.
LOG      :MacHTTP.log

########################################################
# The following commands can be used to adjust MacHTTP's behavior and
# performance. Most of them can be adjusted via AppleScript and AppleEvents
# as well.

#Sets the timeout for inactive connections to 60 seconds
TIMEOUT 60

#Sets the max number of simultaneous users to 10.
#The minimum value is 3, the maximum is 48
#For larger values, you should monitor memory usage and increase
#MacHTTP's memory allocation in the Finder accordingly.
MAXUSERS 32

#Sets the number of "listens" MacHTTP performs simultaneously. For busy
#servers with clients that report "Unable to connect" errors, this number
#should be increased. If the "Listening" statistic in the status window ever
#drops to 1, some clients may miss connecting.
```

```
# Default is 5, minimum is 3, maximum is 48. Never set the number of listens
#to be more than the MAXUSERS!
MAXLISTENS 32

# A single copy of MacHTTP only listens on a single port for multiple
# connections. The HTTP standard port is 80. Users may define any port
# they'd like to listen on, but internet standards say that ports
# numbered 1024 and below are reserved for "Well known services" that
# are pre-defined. That means if you change MacHTTP's port from 80,
# you should pick a number greater than 1024 to avoid conflicting with
# things like telnet, gopher, ftp, nfs, pop, etc. that all have ports
# assigned below 1024.
PORT 80

# This is the number of ticks that MacHTTP will "steal" from other processes
# while sending data to clients. This equates directly to how much time
# MacHTTP will spend processing connections. Your Mac will effectively be
# dedicated to MacHTTP for this period of time. The argument is in "ticks",
# which are 60ths of a second. The default is .5 seconds. (30 ticks) Values
# can range between 0 and 120.
PIG_DELAY 30

# This is the chunk size that MacHTTP will divide file transfers into. The
# larger the chunk, the longer it will take to transmit over slow connections.
# The smaller it is, the more times MacHTTP will be able to swap between
# servicing multiple connections and freeing the Mac to work on other
# processes. The argument represents the max number of bytes to be sent in a
# single MacTCP write to the client. The min is 256, the max is 10240.
DUMP_BUF_SIZE 4096

# MacHTTP can be configured to eliminate DNS accesses. For best performance,
# uncomment the following line. MacHTTP will log IP addresses rather than
# host names, but the software will perform much faster when used with
# slow domain name servers.
NO_DNS

########################################################
#These lines define the suffix and file type mappings for MIME types.
#The syntax is <type> <suffix> <mac file type> <mac creator> <mime type>
#
#Unspecified parameters should be replaced with "*".  MacHTTP tries to match
#a file suffix first. Failing that, it tries to match Mac file type info,
#and if it can, Mac creator info as well. Matching either suffix or
#type/creator determines the transfer type and MIME type. If the client
#supports HTTP/1.0, the appropriate MIME header will be constructed and
#returned, based on the info below.
#Scripts are responsible for generating their own HTTP/1.0 headers!!!

TEXT    .HTML TEXT * text/html
BINARY  .GIF  GIFf * image/gif
CGI .CGI APPL * text/html
ACGI .ACGI APPL * text/html
```

```
SCRIPT  .SCRIPT TEXT * text/html
SCRIPT  * TEXT ToyS text/html
APPL    .EXE APPL * text/html
TEXT    .TEXT TEXT * text/plain
TEXT    .TXT TEXT * text/plain
TEXT   .HQX TEXT * application/mac-binhex40
BINARY  .JPG JPEG * image/jpeg
BINARY  .JPEG JPEG * image/jpeg
BINARY  .PICT PICT * image/pict
BINARY .AU * * audio/basic
BINARY .AIFF * * audio/x-aiff
BINARY .XBM * * image/x-xbm
BINARY  .MOV MOOV * video/quicktime
BINARY .MPEG MPEG * video/mpeg
BINARY .WORD WDBN MSWD application/msword
BINARY .XL XLS3 * application/excel
BINARY .SIT SITD * application/x-stuffit
BINARY .PDF PDF%20 * application/pdf

########################################################
# Security configuration
#
# Security realms - see the Security tutorial for more details
#REALM workers Co-Workers
#REALM cust Customers

#IP and domain name security. There is an implied "DENY *" that is evaluated
#prior to any address security specifications if they are present. Otherwise,
#the default is an implied "ALLOW *". End complete host IP addresses with
#a "." for an exact match. Otherwise a statement like "ALLOW 129.106.3"
#would match hosts 129.106.30.*, 129.106.31.*, 129.106.32.*, etc.
#
#You may also specify domain names for ALLOW and DENY statements. The domain
#names are matched from right to left, as opposed to the left to right matches
#done for IP address ALLOW and DENY statements. Also, the domain names you
#specify are case-sensitive and MUST end with a period (.).
#For example:
#  ALLOW abc.edu.
#  DENY mac22.abc.edu.
#would deny all hosts (implicit DENY *), allow any abc.edu node, and deny the
#specific host, mac22.abc.edu.

#NOTE!!! "ALLOW *" and "DENY *" are not valid syntax!

#ALLOW 123.45.6.
#DENY 123.45.6.7.
```

Let's go in and mess with this a little. There really isn't that much you need to change; in fact, you can almost use it just the way it is.

Go ahead and mark up this book. Don't worry, you won't hurt our feelings. Highlight all the config lines above — the ones in bold — so that they jump off the page and you can spot 'em easily.

Minor Detour

If you haven't already done so, now would be a good time to edit the **Error.html** and **NoAccess.html** files in the MacHTTP Software folder to make them more relevant to *your* site. Also, you should probably substitute *your* home page for the **Default.html** file.

Changing the name of the file named **Default.html** to something more meaningful means making your first change to the MacHTTP configuration file. In your text editor, find the following lines:

```
INDEX    Index.html
ERROR    :Error.html
NOACCESS :NoAccess.html
```

Change "Index.html" to the filename of your home page — for example "Loka.html".

```
INDEX    Loka.html
ERROR    :Error.html
NOACCESS :NoAccess.html
```

WARNING, WARNING

Pay close attention to the fact that there is no colon preceding the filename, unlike the ERROR and NOACCESS lines. You'll feel like a knuckle head if you make the same mistake we once did.

Other Tweaks to Consider (Subtitle: TIMEOUT and MAXUSERS and MAXLISTENS, Oh My!)

The only other changes you may want to consider at this time are bumping up the TIMEOUT and MAXUSERS and MAXLISTENS entries. Where you end up is a matter of trial and error and paying attention to the information that appears on the MacHTTP Status window.

If you are running System 7.5 on a Power Macintosh and are using a version of MacHTTP that takes advantage of the Thread Manager (version 2.0.2 or higher), then you can go ahead and really jack up the MAX-

USERS and MAXLISTENS. The maximum number of simultaneous hits that the Mac OS allows with MacTCP is 48. That will all change with OpenTransport of course; check out the sidebar for more details.

If you are noticing a lot of timeouts, then bump up the TIMEOUT number. Hard to be more straightforward than that. You can fool around with PIG_DELAY and DUMP_BUFF_SIZE if you want to do a little more fine-tuning. We never mess with 'em. From time to time there'll be a discussion thread on the MacHTTP Listserve describing different ways to tweak these puppies. In fact, now is a good time for you to subscribe to the MacHTTP mailing list. Allow us to digress...

Subscribing to the MacHTTP Listserve

It's crucial that you subscribe to the MacHTTP listserve. Here's how.

You can subscribe by sending mail to **majordomo@academ.com** or **machttp-talk-request@academ.com** with the single line:

subscribe machttp-talk

in the body of the message. IMPORTANT! Never send subscription requests or other list commands to the machttp-talk address. If you have a question about how the mailing list itself works or who is subscribed, send an eMail to **majordomo@academ.com** with the list command (i.e., "help") in the body of the message.

To unsubscribe, send

unsubscribe machttp-talk

to **majordomo@academ.com.**

To receive only a digest of the messages, subscribe to the machttp-talk-digest list instead, by typing

subscribe machttp-talk digest

as the entire body of the message.

Several other options are available from the listserve. Send the message:

help

to **majordomo@academ.com** for more info.

The big tip of the day here is to subscribe to a digested version of the machttp-talk list. (The digest is a sort of "greatest hits" version of the machttp-talk newsgroup.) Machttp-talk has tons of messages each day, so even if you paid for the commercial version of Eudora (it's worth the $60, by the way) and set up all kinds of filters, the digested version of the mailing list is a lot easier to deal with.

Back to the Show — ALLOW & DENY

OK, we're back. There are two more parts of the config file to take a look at. They have to do with security. You can set MacHTTP so that it only allows certain domain addresses to access your server. You can also set your server up so that a page or group of pages requires a password to be accessed.

The ALLOW and DENY lines of the config file can be uncommmented and modified to — you guessed it — allow and deny access. They do this based on IP and domain name. This is a useful feature if you want to set up your Web server for internal use only.

You can also create an area or "realm" on your server that requires a name and a password to access. Take a look at the following example where the REALM lines of the config file have been uncommented:

```
# Security configuration
#

# Security realms - see the Security tutorial for more details

REALM ada ADA
REALM test Test
```

What we've done in this example is to set up two areas of our Web server that are private. Setting up a secure realm on your server involves three things actually — modifying the config file, editing the password preferences from the menu bar of MacHTTP, and making sure that all of the files that you want to have included in the realm have the realm name somewhere in their path/filename. Let's try it.

Realms

Suppose you want to create a private area on your server for testing purposes (Jeff has something like this set up on one of his servers for testing all the new stuff that comes out for MacHTTP).

The first thing to do is to create a realm in the config file. This is easy. There are two steps. Step one, uncomment one or more **REALM** lines of the config file by deleting the "#" symbol. Second, take a look at the URL path name(s) for the document(s) you want to make private. Hopefully you've done an orderly job organizing your HTML files and all the related documents have a word in common in their path name. In the line below, the word "test" is in the path name of every HTML document that has to do with beta testing pages on one of Jeff's WWW servers:

REALM test Test

You have to follow the convention of "test Test" when you do this. One little word. One big word. Another one might read:

REALM geek Geek

You can have as many "Realms" as you want. If you need more than the two lines worth that you uncomment and modify in the config file, add 'em.

Be sure to quit and restart MacHTTP after you do these first steps or none of this will work!

The next step is to choose **Passwords** from the MacHTTP **Edit** menu. Your server needs to be up to do this. A dialog box like the one shown in Figure 4.1 will appear.

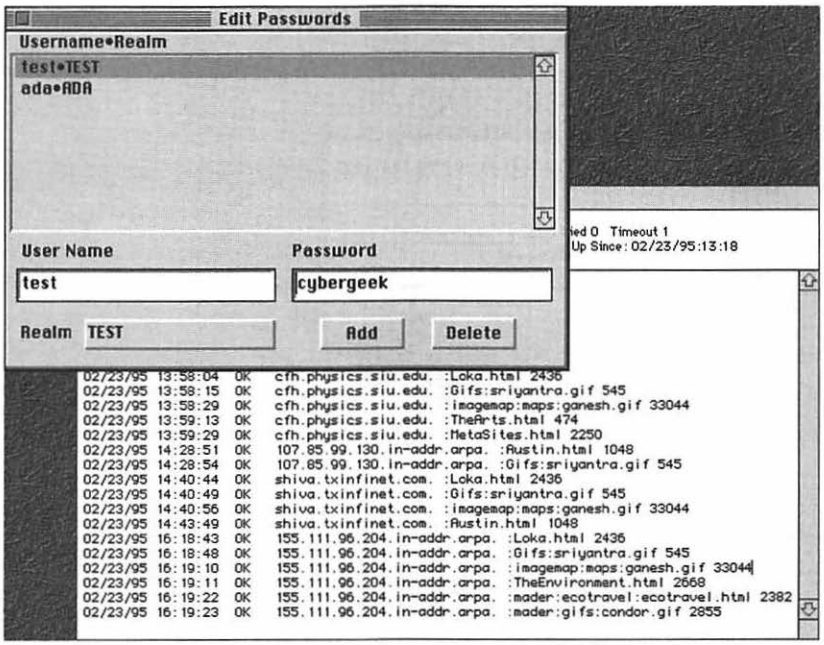


Figure 4.1: Creating Realms with MacHTTP.

Press (like a click, only longer) on the **Realm** button and the realms that you defined in the MacHTTP config file will be displayed as a pop-up menu. Choose the one you have created in the config file. Type in a name and password and select the **Add** button. Do the same thing for each realm you created. You can set up as many names with passwords as you want for each realm you created in the config file. Keep it simple, though, or you'll have a maze of realms, names, and passwords to manage. For the sake of the example we created a name "test" and password of "cybergeek" for the test realm.

In order for all this to work, the document(s) you want to restrict access to must have the name of the realm in their location — either in the path to or in the name of the file itself. In this case the realm "test" is in the HTML name of the file we want to restrict access to, **test.html**.

When a user selects a link on a Web page that has restricted access, they'll see the screen shown in Figure 4.2.

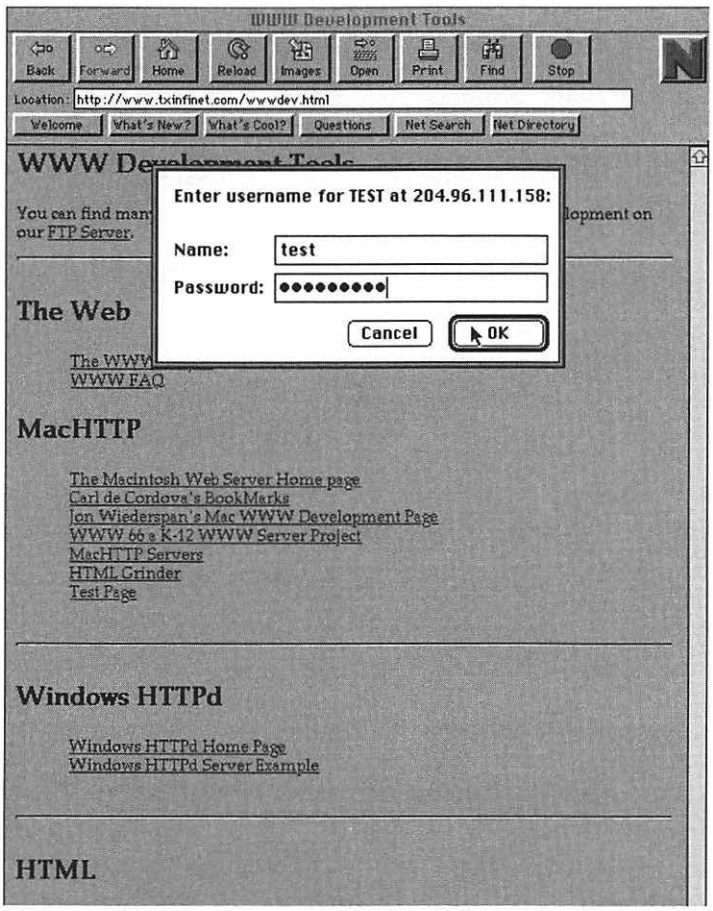


Figure 4.2: MacHTTP Login Window.

Once they have entered a name and password (somewhere in the space-time continuum you have to furnish them with it, of course), they can log in to the secure area and get something as brilliant as the text shown in Figure 4.3.

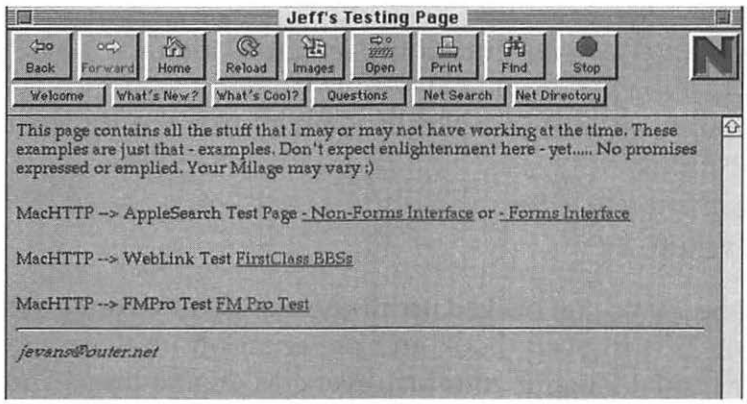


Figure 4.3: Jeff's MacHTTP Test Page.

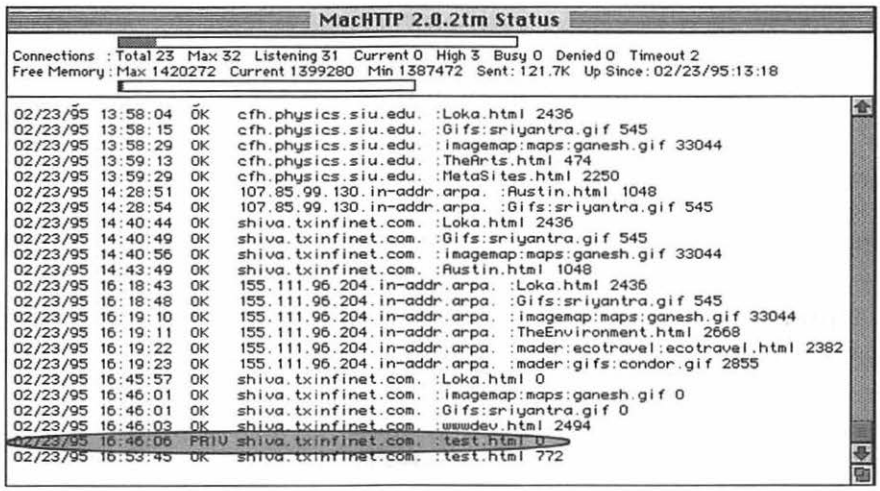


Figure 4.4: MacHTTP Status Window Showing PRIV Access (highlighted).

Figure 4.4 shows what's going on behind the scenes with MacHTTP as this happens.

You got all that so far? Good. It's a good idea to have every document that you want included in a "realm" have the name of the realm somewhere in its path or actual filename. Don't think that just because a

user can't see a document that they won't stumble across it accidentally — or deliberately.

Try going to a WWW site and adding **test.html** after the home page URL and see what you get. If you have stuff you don't want people nosing around through, set it up in a "Realm" and let the security feature of MacHTTP do its thing.

Back to the Backup...

It's pop quiz time. Have you backed up all the work you've done yet? Do it. Bob harps on this in every book and presentation he does. And for good reason. It's bad enough if your Mac ever crashes and burns and you lose everything; it would be even more of a drag to have to redo all that HTML and reconfigure your server now that you are this far along.

So if you haven't done so yet, now's the time to back it up! And continue to do so regularly — especially after any major changes to your site. You've worked too hard to lose everything; learn to back up regularly. While both Mac and MacHTTP are pretty reliable, sooner or later something bad will happen and you'll thank us for reminding you.

Getting Your Files Organized

While we're on the topic of WebMaster behavior, we'd like to make another suggestion. It's one that will save you a ton of work down the road. Think carefully about how you organize your HTML files, images, utilities and other tools, scripts, and CGIs (we'll get to these, we promise) within your MacHTTP folder.

Everything — except for AppleScript and its scripting additions — has to reside somewhere in the MacHTTP folder. Over time that can become an enormous number of files!! Jeff's first Web site — his personal one — is such a mish-mash of HTML files and GIFs and old versions of things that it's almost too much work to go back and clean it all up.

So start out on the right foot. Think about organizing files into folders. Put all your GIFs and other images into one folder — or into a folder within a folder of related HTML files. If you have completely different types of content on your server — say personal stuff and Web pages for several clients — group those things within their own folders.

Using folder names and nested folders and files in URL paths is no big deal. It doesn't noticeably slow down server performance. BELIEVE US! If you ever want to move a set of files to another server it will be far easier to do so if all the related files are in a single folder or nested folders.

Let's take a look through a set of exhibits and follow the organizational flow of the folder named "mader".

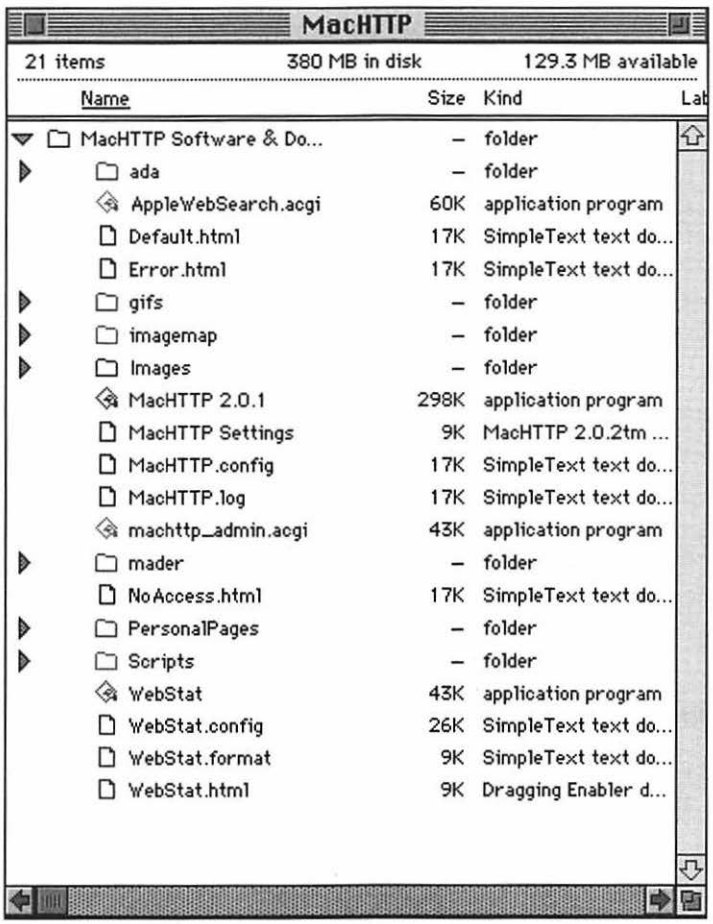


Figure 4.5: Example of Orderly MacHTTP Software Folder.

Tucked away on one of Jeff's WWW servers at **http://www.txinfinet.com/mader/ecotravel/ecotravel.html** is a collection of pages and

graphics that make up the **Eco-Travels in Latin America WWW Site**, by Ron Mader. You would never know by looking at the MacHTTP server folder in Figure 4.5 that there are a ton of documents inside the "mader" folder.

Ron's Eco-Travels is organized in such a way that the whole folder and everything in it could easily be moved to another site. All he would have to do is move the "mader" folder and create a new link to his "home page." It would take a total of 2 to 3 minutes, tops.

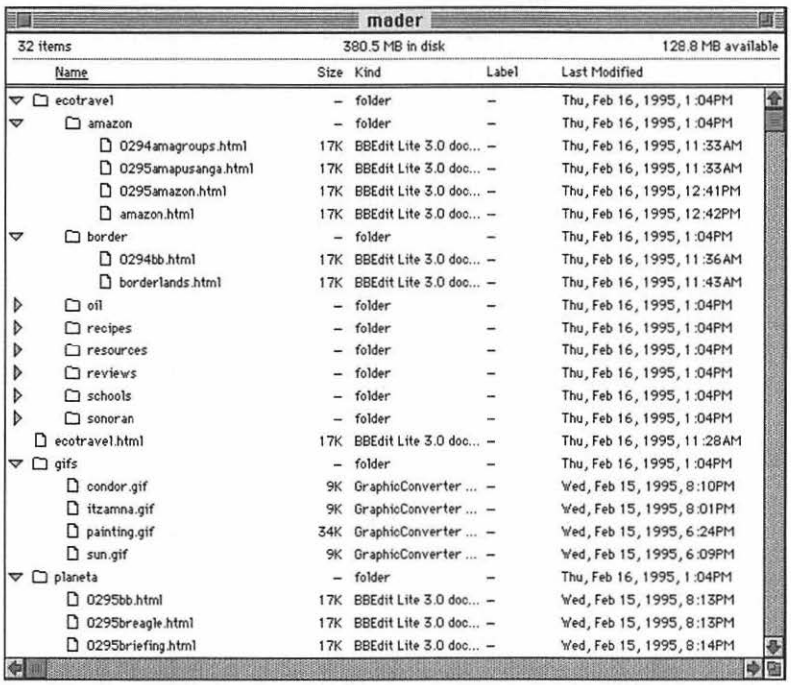


Figure 4.6: Eco-Travels in Latin America Folder Hierarchy.

Ron did a great job organizing his site ahead of time (Figure 4.6). In fact, he probably spent as much time thinking that through as he did converting the text files he had written for magazines and periodicals to HTML. The reason Ron's documents are so easily transportable is that he used relative URL links everywhere (Figure 4.7).

```
<HTML>

<HEAD> <TITLE>Eco Travel in Latin America</TITLE> </HEAD>

<BODY>

<img src="/eader/gifs/condor.gif"><BR>

<H1>Eco Travel in Latin America</H1>
This site is dedicated to informing Internet travelers of environmental issues in Latin America.
You don't necessarily have to be an ecotraveler to be interested in all you'll find here.

<H2>El Planeta Platica</H2>
In the spring of 1994 I created an environmental newsletter to address the needs
of conscientious travelers who want to know more about Latin America, thus the Spanish
title, which translates as "The Earth Speaks."  This is a collection of materials - news,
resources, poems, recipes and it allows me to stay in touch with good friends across the
hemisphere who work on environmentally sustainable projects.<P>

<UL>
<LI><A HREF="/eader/planeta/planeta_current.html">El Planeta Platica - Current Issue</A><BR>
<LI><A HREF="gopher://csf.Colorado.EDU:70/11/environment/orgs/El_Planeta_Platica/Archive%20-%20Text%20Files">
</UL>

<H2>Destinations</H2><P>

<UL>
<LI><A HREF="/eader/ecotravel/amazon/amazon.html">The Amazon</A><BR>
<LI><A HREF="/eader/ecotravel/sonoran/sonoran.html">Sonoran Desert</A><BR>
<LI><A HREF="/eader/ecotravel/border/borderlands.html">US-Mexico Borderlands</A><BR>
<LI><A HREF="/eader/ecotravel/oil/oil.html">Oil & the Environment</A><BR>
<LI><A HREF="/eader/ecotravel/schools/schools.html">Spanish Language Schools</A><BR>
<LI><A HREF="/eader/ecotravel/resources/resources.html">Eco Travel Resources</A><BR>
<LI><A HREF="/eader/ecotravel/reviews/reviews.html">Book Reviews</A><BR>
<LI><A HREF="/eader/ecotravel/recipes/recipes.html">Recipes</A><BR>
</UL>

<H2>Related Sites</H2>
<UL>
<LI><A HREF="gopher://csf.Colorado.EDU:70/11/environment/Conservation-Biology/docs/countries/americas">
<LI><A HREF="http://csf.colorado.edu/elan/index.html">Environment & Latin America</A><BR>
<LI><A HREF="/TheEnvironment.html">MahaInfinet Environment Links</A><BR>
<LI><A HREF="http://lanic.utexas.edu/">Latin America Background - UT LANIC</A><BR>
</UL>

<HR>
I welcome your <A HREF="mailto:ron@txinfinet.com">suggestions</A>.


<HR>
<ADDRESS>webmaster@txinfinet.com </ADDRESS>
```

Figure 4.7: Eco-Travels in Latin America — Home Page Source HTML Showing Extensive Use of Relative Links.

Rather than throw all his HTML files in one folder, Ron grouped related documents together (Figures 4.8 and 4.9). That way, whenever he needs to update anything or make a change, it's just a matter of minutes and he's done.

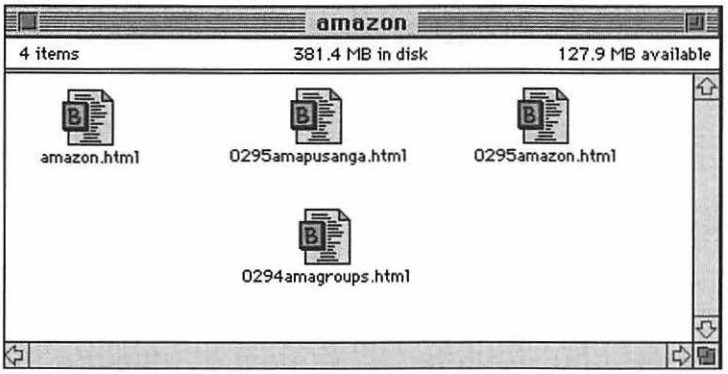


Figure 4.8: Eco-Travels in Latin America — Keeping Related Documents Together.

All of Ron's HTML pages are simple and uncluttered. His work is a great example of how to keep everything efficient and well managed.

```
<HTML>

<HEAD>
<TITLE>Amazon
</TITLE>
</HEAD>

<BODY>
<img src="/mader/gifs/sun.gif"><H1>The Amazon</H1>

<UL>
<LI><A HREF="/mader/ecotravel/amazon/0294amagroups.html">Amazonian Organizations</A><BR>
<LI><A HREF="/mader/ecotravel/amazon/0295amazon.html">Keeping the Amazon Green</A><BR>
<LI><A HREF="/mader/ecotravel/amazon/0295amapusanga.html">Pusanga: Making the Forest a Home</A><BR>
</UL>

<HR>

Return to <a href="/mader/ecotravel.html">Eco Travel in Latin American</a><BR>
Return to <a href="/mader/planeta/planeta_current.html">El Planeta Platica</a><BR>

<HR>
<ADDRESS>webmaster@txinfinet.com
</ADDRESS>

</BODY>

</HTML>
```

Figure 4.9: Eco-Travels in Latin America — Links to the Amazon Directory.

The Dark Side

The last exhibit in the story, Figure 4.10, is a glimpse into the dark side of WebMastery. It's like that drawer you have in the kitchen or hallway where you just throw everything. It's the first server Jeff ever did and he was acting like Eddie Haskell from "Leave it to Beaver" — orderly and polite on the surface, wild and unruly to the core.

Don't let your server start to get this disorganized. You're asking for trouble if you ever have to move it to another machine or reorganize it to make room for more archives. Even though it may not look it at first, this one is too late to correct without a major reorganization.

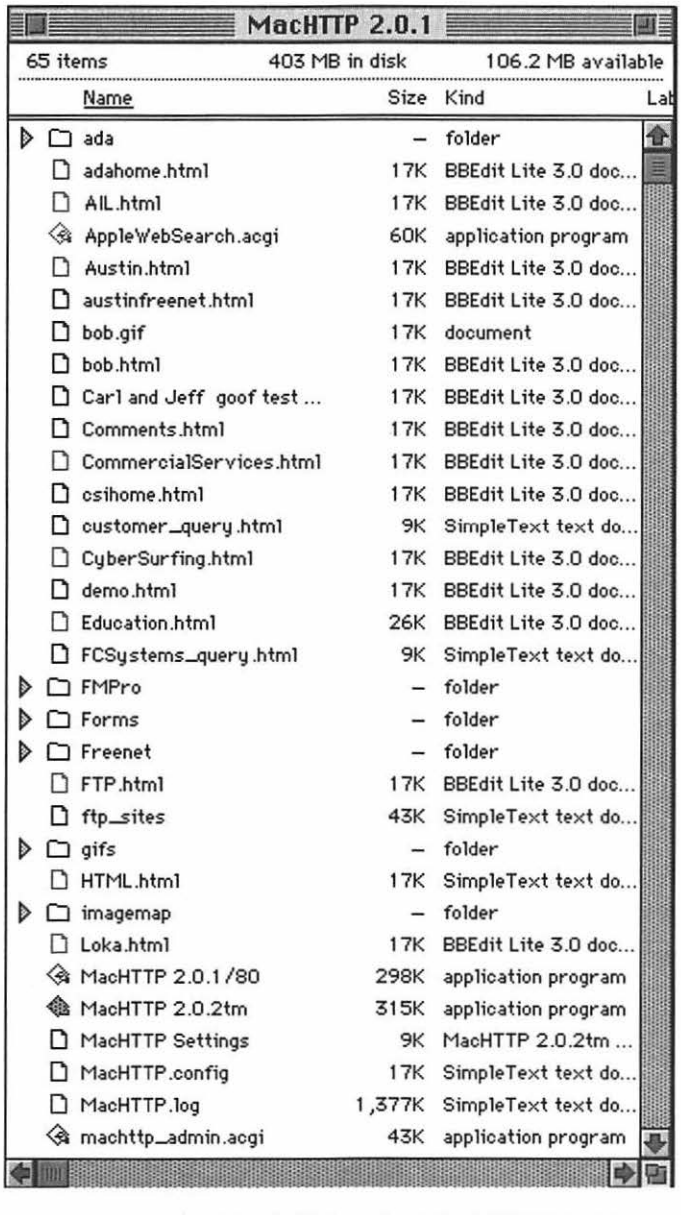


Figure 4.10: The Dark Side of a MacHTTP Folder Structure (a.k.a. Jeff's First Mess).

HITS ON A CLICKABLE MAP — USING WEBSTAT AND IMAGEMAP FOR JOB SECURITY

When you took a look at the nice tidy MacHTTP folder in Figure 4.5 you may have noticed a couple of items you're not familiar with. One of them, **AppleWebSearch.acgi**, we'll talk about in the next chapter. For now, let's take a look at two of the first utilities written specifically for MacHTTP — WebStat and ImageMap. We'll start with WebStat.

WebStat

WebStat is a nifty little item developed by Phil Harvey from Queen's University in Ontario, Canada. It keeps track of how often your Web server is accessed. It also lets you know what domain types (com, edu, org, gov, or mil) hit it and when, as well as which files were accessed most frequently. It's a great way to justify all the hours you put into setting up your Web server and can serve as the basis for reports on server activity and marketing presentations to existing and potential clients. It may also help you keep your job doing cool WebMasterey stuff.

(It will definitely let you know, in no uncertain terms, that no one is interested in that great page you put up on the relationship between the 7-layer OSI model, MacTCP, Open Transport, and the Grand Unified Field Theory of quantum physics.)

Here's how the WebStat program works in a nutshell (Figure 4.11). When you fire it up it looks for the **MACHTTP.LOG** file that sits in your MacHTTP folder. WebStat reads that file and automatically creates a file in your MacHTTP folder called **WebStat.html**. **WebStat.html** is an HTML file, right? Right. That means you can look at it with your favorite browser. So can others if you let them.

The WebStat program lets you modify the look of **WebStat.html** if you want to. You do this by opening up the text file called **WebStat** and going at it. Our recommendation is to leave everything else as is because it does a fine job as is.

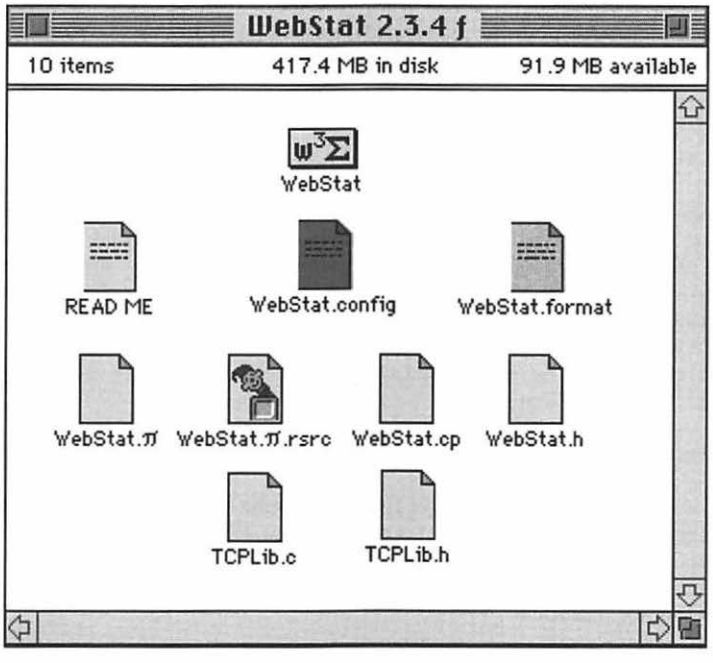


Figure 4.11: WebStat Folder.

Warning Will Robinson

To set up WebStat properly, you do need to go muck around in the **WebStat.config** file. It's another beautiful text file with comment lines and configuration lines just like the **MacHTTP.config** file. Bummer. Here's a real-life one in all its beauty — the actual configuration lines are way at the bottom before the list of countries starts:

```
############################################################
#    This is the WebStat Configuration File (WebStat.config)
#
#    This file must exist in the same directory as the WebStat program.
#
# Commands:
#
#    LOG <filename>
#
#    Specify MacHTTP log file name for WebStat input (paths allowed). The
#    file name must be inside quotation marks if it contains spaces.
#    Default is "MACHTTP.LOG".
```

```
#
# OUTPUT <filename>
#
#    Specify output filename for WebStat summary (html format).  The file
#    name must be inside quotation marks if it contains spaces.
#    Default is "WebStat.html".
#
# FORMAT <filename>
#
#    Specify format of WebStat output.  Lines in this file are copied
#    directly to the output file.  Special commands are embedded in this file
#    which tell WebStat where to place the output statistics.  These commands
#    must be in square brackets at the start of a new line.  Only a single
     command (and no other text) is allowed on this line.  The commands are:
#    Summary, Day, Hour, Weekday, Domain, Subdomain, and Archive. See the
#    WebStat.format file for an example of how to set up a format file.
#    Default is "WebStat.format".
#
# EXCLUDE <address>
#
#    Specify a computer/network to exclude from the summary.  The format of
#    the name should be reversed as in the Client Reversed Subdomain table
#    in the output file. A '*' is allowed at the end of the name to match
#    any subsequent characters.
#    Default is no exclusions.
#
# MESSAGES <On | Off>
#
#    Specify whether or not WebStat sends routine messages to a console
#    window while it is running.  Error messages will always be output.  If
#    you have problems or questions about what WebStat is doing, try turning
#    MESSAGES On to take a look at messages produced.
#    Default is "Off".
#
# ADDRESSES <Long | Short>
#
#    Specify length of subdomain addresses.  If "Short", the last field of
#    the Reversed Client Subdomain address is not shown.
#    Default is "Long".
#
# DOMAIN <code> <name>
#
#    Specify domain names for the Client Domain list.  Domain codes are
#    matched like EXCLUDE addresses, assuming a '*' after the domain code.
#    Codes are searched in the order of entry. The name must be inside
#    quotation marks if it contains spaces.
#    Default is no domain names.
```

```
# DNSLOOKUP <On | Off>
#
#    If this option is "On", WebStat will translate numerical IP addresses
#    found in the log file using your domain name server.  To use this option,
#    you must be connected to a network with a domain name server, and you
#    must be running MacTCPand have it configured properly for DNS lookups.
#    Default is "Off"
#
# # <comment>
#
#    Comment lines start with a '#'.
#
# Examples:
#
#    LOG  MyLogFile.Log
#    LOG  "My Hard Disk:My Folder:My Log File"
#
#    OUTPUT  WebStat.html
#    OUTPUT ":My Folder:WebStat output file"
#
#     FORMAT WebStat.format
#    FORMAT "My Format File"
#
#    EXCLUDE ca.queensu.phy.snodaq    - exclude Phil Harvey's computer
#    EXCLUDE ca.queensu.*               - exclude all computers at Queen's
#    EXCLUDE mil.*                      - exclude all US military computers
#
#    MESSAGES On
#
#    ADDRESSES Short
#
#    DOMAIN  ca.queensu.  "Queen's University"
#     DOMAIN  ca.  Canada
#    DOMAIN  128.  "A Domain Name"
#
#     DNSLOOKUP On
#
###################### End Comments ##########################

LOG          MACHTTP.LOG
OUTPUT    WebStat.html
FORMAT    WebStat.format
MESSAGES Off
ADDRESSES Long
DNSLOOKUP Off

DOMAIN ad. "Andorra"
DOMAIN ae. "United Arab Emirates"
DOMAIN af. "Afghanistan"
DOMAIN ag. "Antigua and Barbuda"
DOMAIN ai. "Anguilla"
DOMAIN al. "Albania"
DOMAIN am. "Armenia"
```

We didn't list all the countries that are part of the configuration for WebStat — the list would be too long. Suffice it to say that some poor soul typed them all, as you'll soon see for yourself when you install it.

Put the **WebStat** program (the one with the cool icon), **WebStat.config**, and **WebStat.format** files in the MacHTTP folder, at the same folder level as the MacHTTP application. The rest of the stuff in the WebStat folder can go anywhere on your hard drive.

You can make your server stats visible to the world by just making a link to the output file **WebStat.html**. Alternatively, you can just keep it tucked away somewhere in a realm that is set up with security or with a path name that only you know. **WebStat.html** is an HTML file, so you can view your server statistics with any browser.

We wanted to show you an example of one we have in real life but the output file is huge and it would have taken too many screen shots and pages to do. So, because we knew how much it would mean to you, we set up a quick little server one afternoon and asked all our friends to hit it over a weekend and run through all the documents. That's how we got the dummy **WebStat.html** file shown in Figure 4.12.

When you open up the **WebStat.html** file with your browser (we used Netscape) the first thing you see is summary information. There is also a list of links to other parts of the **WebStat.html** file so you can jump quickly to different parts instead of scrolling through the whole thing.

If you start scrolling down through the **WebStat.html** page you start to see a breakout of your server's statistics. First you see it on a daily basis, as in Figure 4.12. Then you get the hourly traffic report on your server activity measured in the total number of files and bytes transferred, as shown in Figure 4.13.

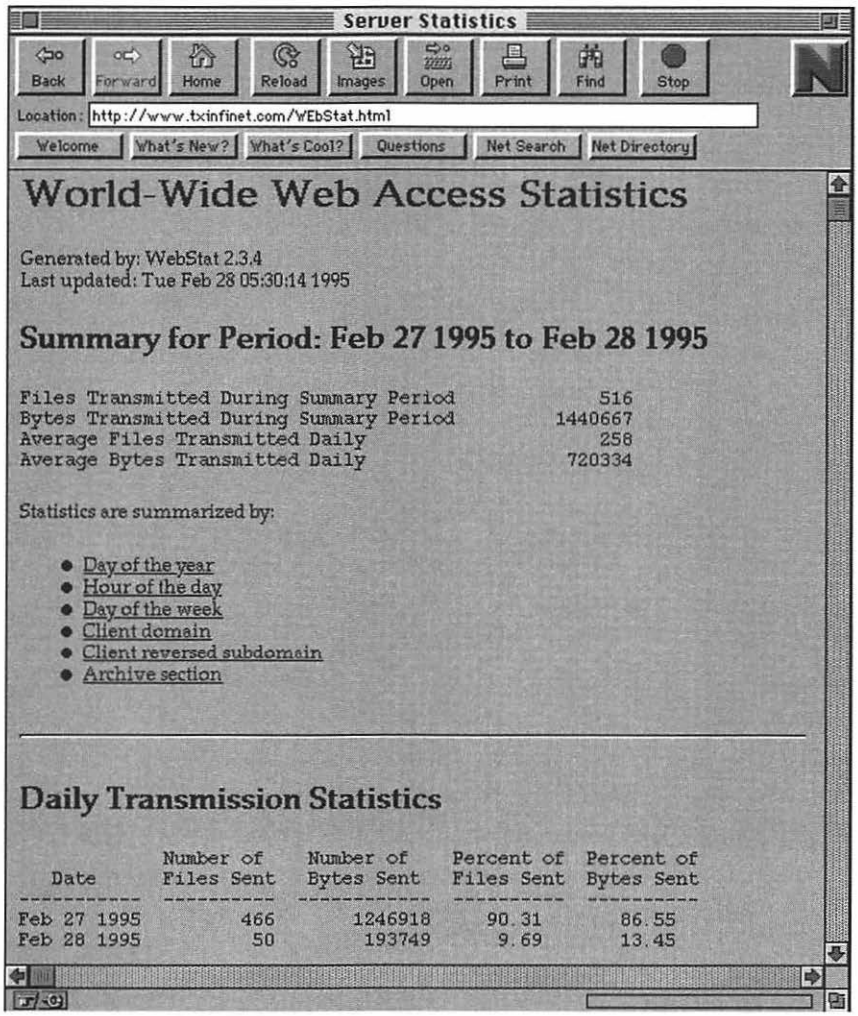
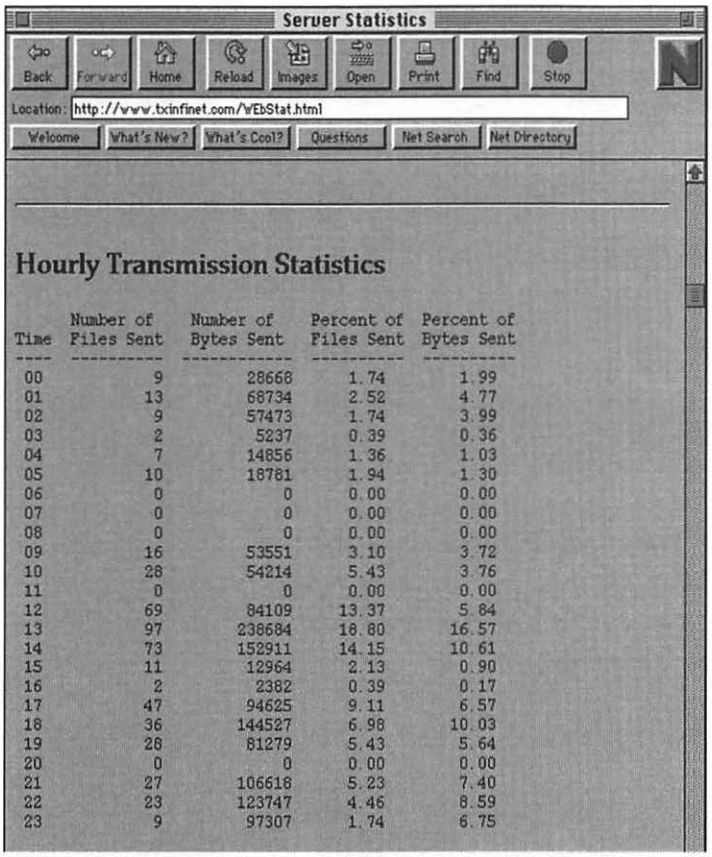


Figure 4.12: WebStat.html Page Showing Summary Statistics, Links, and Daily Transmission Statistics.

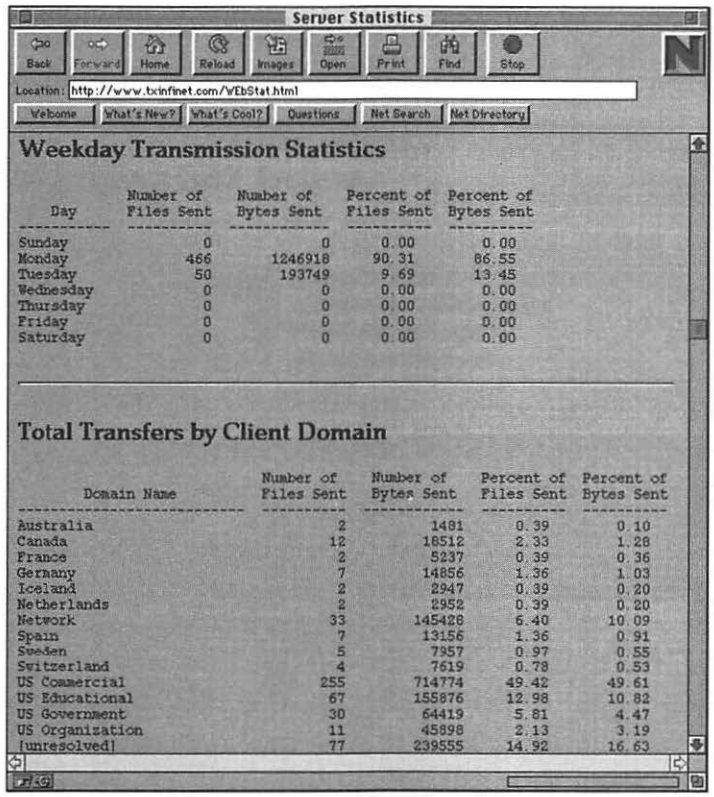


Hourly Transmission Statistics

Time	Number of Files Sent	Number of Bytes Sent	Percent of Files Sent	Percent of Bytes Sent
00	9	28668	1.74	1.99
01	13	68734	2.52	4.77
02	9	57473	1.74	3.99
03	2	5237	0.39	0.36
04	7	14856	1.36	1.03
05	10	18781	1.94	1.30
06	0	0	0.00	0.00
07	0	0	0.00	0.00
08	0	0	0.00	0.00
09	16	53551	3.10	3.72
10	28	54214	5.43	3.76
11	0	0	0.00	0.00
12	69	84109	13.37	5.84
13	97	238684	18.80	16.57
14	73	152911	14.15	10.61
15	11	12964	2.13	0.90
16	2	2382	0.39	0.17
17	47	94625	9.11	6.57
18	36	144527	6.98	10.03
19	28	81279	5.43	5.64
20	0	0	0.00	0.00
21	27	106618	5.23	7.40
22	23	123747	4.46	8.59
23	9	97307	1.74	6.75

Figure 4.13: WebStat.html Page Showing Hourly Statistics and Number of Files and Bytes Transferred.

Next comes the Weekly Transmission Statistics part of the **WebStat.html** file. It also shows traffic on your server measured in the number of files and bytes transferred. It's followed by the total transfers by Client Domain. Besides showing you the number of the different types of commercial, educational, and other U.S. types of users that have hit your server, this is where you look to see if anyone from Outer Mongolia is surfing the Net.



Weekday Transmission Statistics

Day	Number of Files Sent	Number of Bytes Sent	Percent of Files Sent	Percent of Bytes Sent
Sunday	0	0	0.00	0.00
Monday	466	1246918	90.31	86.55
Tuesday	50	193749	9.69	13.45
Wednesday	0	0	0.00	0.00
Thursday	0	0	0.00	0.00
Friday	0	0	0.00	0.00
Saturday	0	0	0.00	0.00

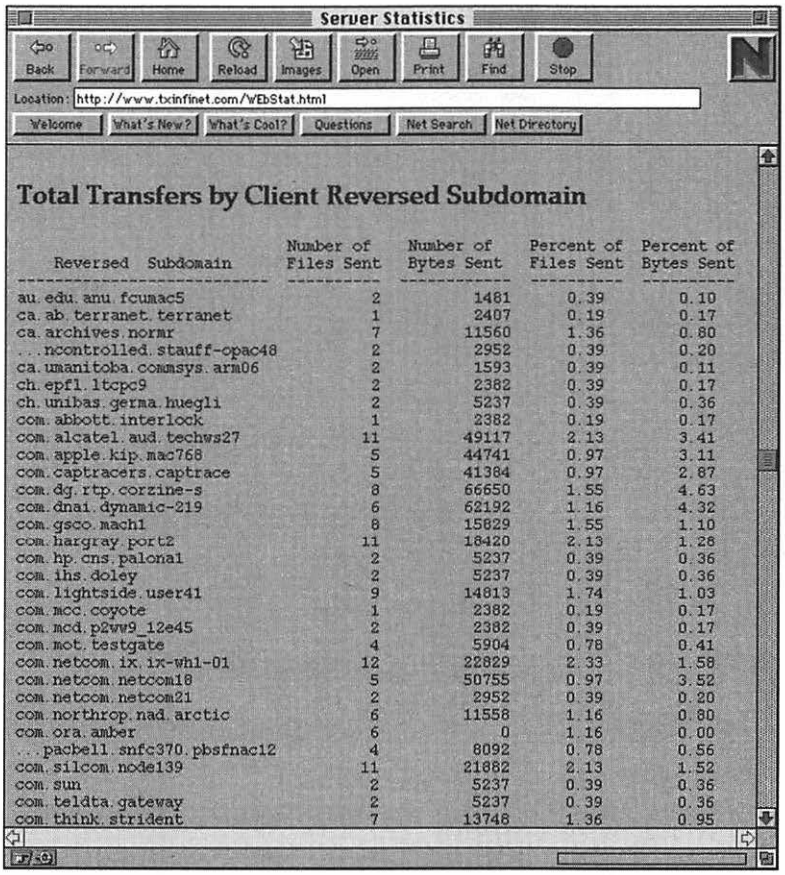
Total Transfers by Client Domain

Domain Name	Number of Files Sent	Number of Bytes Sent	Percent of Files Sent	Percent of Bytes Sent
Australia	2	1481	0.39	0.10
Canada	12	18512	2.33	1.28
France	2	5237	0.39	0.36
Germany	7	14856	1.36	1.03
Iceland	2	2947	0.39	0.20
Netherlands	2	2952	0.39	0.20
Network	33	145428	6.40	10.09
Spain	7	13156	1.36	0.91
Sweden	5	7957	0.97	0.55
Switzerland	4	7619	0.78	0.53
US Commercial	255	714774	49.42	49.61
US Educational	67	155876	12.98	10.82
US Government	30	64419	5.81	4.47
US Organization	11	45898	2.13	3.19
[unresolved]	77	239555	14.92	16.63

Figure 4.14: WebStat.hmtl Page — Weekly Statistics and Transfers by Domain Type.

You hope that WebStat would take you all the way and show you the name of the person who came to your site. Unfortunately it's just not possible. To do that you need to set up a "Form" page on your WWW server that lets a user enter that information themselves. The closest you are going to get to seeing who is hitting your site is to take a look at that part of the **WebStat.html** page that shows the total number of files and bytes sent by client reversed subdomain (Figure 4.15). That's gobbledygook for the Internet name of the machine that visited your site. For example, one of Jeff's in-house Mac networks uses the domain name txinfinet.com. One of his Macs has the name "shiva." If Jeff surfs over to your WWW site on that particular Mac you'll see shiva.txinfinet.com show up in your **WebStat.html** file.

There's no way you can tell how many other people use Jeff's Mac when he's not looking, so you'll never know how many individual users actually come to your site.

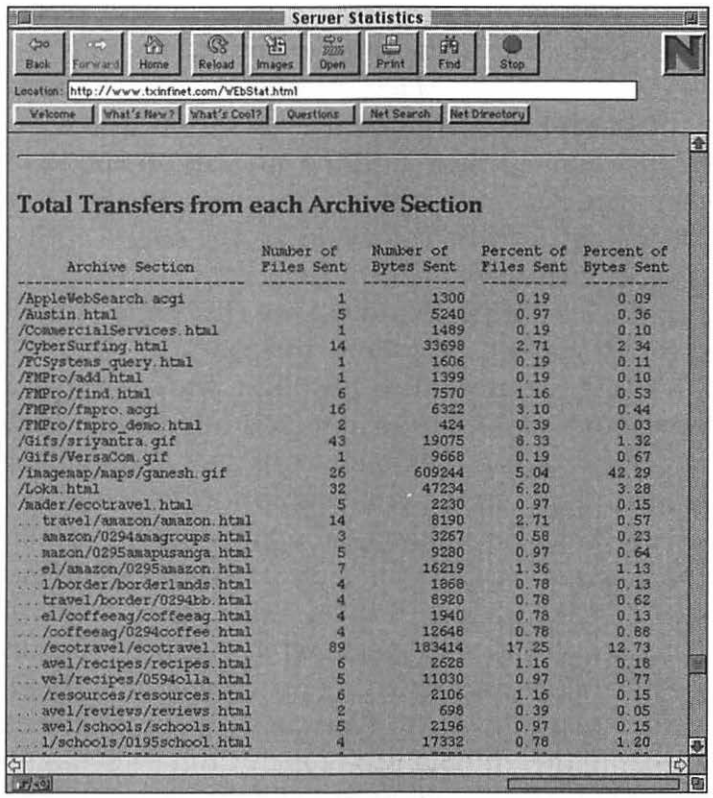


Total Transfers by Client Reversed Subdomain

Reversed Subdomain	Number of Files Sent	Number of Bytes Sent	Percent of Files Sent	Percent of Bytes Sent
au.edu.anu.fcumac5	2	1481	0.39	0.10
ca.ab.terranet.terranet	1	2407	0.19	0.17
ca.archives.normr	7	11560	1.36	0.80
...ncontrolled.stauff-opac48	2	2952	0.39	0.20
ca.umanitoba.commsys.arm06	2	1593	0.39	0.11
ch.epfl.ltcpc9	2	2382	0.39	0.17
ch.unibas.germa.huegli	2	5237	0.39	0.36
com.abbott.interlock	1	2382	0.19	0.17
com.alcatel.aud.techws27	11	49117	2.13	3.41
com.apple.kip.mac768	5	44741	0.97	3.11
com.captracers.captrace	5	41384	0.97	2.87
com.dg.rtp.corzine-s	8	66650	1.55	4.63
com.dnai.dynamic-219	6	62192	1.16	4.32
com.gsco.mach1	8	15829	1.55	1.10
com.hargray.port2	11	18420	2.13	1.28
com.hp.cns.palona1	2	5237	0.39	0.36
com.ihs.doley	2	5237	0.39	0.36
com.lightside.user41	9	14813	1.74	1.03
com.mcc.coyote	1	2382	0.19	0.17
com.mcd.p2ww9_12e45	2	2382	0.39	0.17
com.mot.testgate	4	5904	0.78	0.41
com.netcom.ix.ix-wh1-01	12	22829	2.33	1.58
com.netcom.netcom18	5	50755	0.97	3.52
com.netcom.netcom21	2	2952	0.39	0.20
com.northrop.nad.arctic	6	11558	1.16	0.80
com.ora.amber	6	0	1.16	0.00
...pacbell.snfc370.pbsfnac12	4	8092	0.78	0.56
com.silcom.node139	11	21882	2.13	1.52
com.sun	2	5237	0.39	0.36
com.teldta.gateway	2	5237	0.39	0.36
com.think.strident	7	13748	1.36	0.95

Figure 4.15: WebStat.html Page — Files and Bytes Transferred by Client Reversed Subdomain.

The last part of the **WebStat.html** page is probably the most useful. It shows which of your pages is getting the most hits. This is where you find out if anyone is interested in invertebrate cave-dwelling octapods (Figure 4.16).



Total Transfers from each Archive Section

Archive Section	Number of Files Sent	Number of Bytes Sent	Percent of Files Sent	Percent of Bytes Sent
/AppleWebSearch.acgi	1	1300	0.19	0.09
/Austin.html	5	5240	0.97	0.36
/CommercialServices.html	1	1489	0.19	0.10
/CyberSurfing.html	14	33698	2.71	2.34
/FCSystems_query.html	1	1606	0.19	0.11
/FMPro/add.html	1	1399	0.19	0.10
/FMPro/find.html	6	7570	1.16	0.53
/FMPro/fmpro.acgi	16	6322	3.10	0.44
/FMPro/fmpro_demo.html	2	424	0.39	0.03
/Gifs/sriyantra.gif	43	19075	8.33	1.32
/Gifs/VersaCom.gif	1	9668	0.19	0.67
/imagemap/maps/ganesh.gif	26	609244	5.04	42.29
/Loka.html	32	47234	6.20	3.28
/mader/ecotravel.html	5	2230	0.97	0.15
...travel/amazon/amazon.html	14	8190	2.71	0.57
...amazon/0294amagroups.html	3	3267	0.58	0.23
...mazon/0295amapusanga.html	5	9280	0.97	0.64
...el/amazon/0295amazon.html	7	16219	1.36	1.13
...1/border/borderlands.html	4	1868	0.78	0.13
...travel/border/0294bb.html	4	8920	0.78	0.62
...el/coffeeag/coffeeag.html	4	1940	0.78	0.13
.../coffeeag/0294coffee.html	4	12648	0.78	0.88
.../ecotravel/ecotravel.html	89	183414	17.25	12.73
...avel/recipes/recipes.html	6	2628	1.16	0.18
...vel/recipes/0594olla.html	5	11030	0.97	0.77
.../resources/resources.html	6	2106	1.16	0.15
...avel/reviews/reviews.html	2	698	0.39	0.05
...avel/schools/schools.html	5	2196	0.97	0.15
...1/schools/0195school.html	4	17332	0.78	1.20

Figure 4.16: WebStat.html Page — Total File and Byte Transfers by Section and Filename.

A Few More Tips

There are always all sorts of **ReadMe** and other sometimes-useless files lying around in folders here and there once you get MacHTPP (or any server) set up. Your server's hard drive can start to look like a rummage sale gone bad...

For simplicity's sake, we keep all the folders that hold "leftover" files and previous versions in a single folder, which brings us to another great tip: Don't delete an old file until you're absolutely certain its replacement works properly! Stick it in the "leftover" folder until you're sure.

The other thing we do to keep things orderly is to color-label everything that is crucial. It makes it really easy to drop in new versions of anything if you can quickly locate the three or four related files that are to be updated. We do this for both MacHTTP and WebStat, for instance. In the early days of MacHTTP it seemed like there was a new version coming out every week. Being able to find the application and its associated config files, then quickly and easily replace them, made our lives a whole lot easier. Pick your favorite color and do the color-coded thing using the Finder's Label menu.

For what it's worth, we both prefer to use the "by Name" view in the Finder; there are just too many files to view "by Icon."

By the way, we just got around to putting a little number called Cron by Chris Johnson on our servers to automatically launch and close WebStat everyday. After you read the quick set of docs that are included with Cron and get it all set up, make sure you've put an alias of Webstat in the Cron directory. Both WebStat and Cron are included on the CD-ROM that comes with the book. Give 'em a try. They are easy to install.

When you set up Cron you'll wish there were a specific example in the docs for WebStat. Leave it to your great Webmaster-blaster gurus Bob and Jeff to bail you out! When you're setting up Cron and you get to the part of the instructions that asks you to type out a line in the file named **crontab**, the line that follows in bold will launch Cron everyday at 5:30 AM. It won't mean much until you go to install it, so highlight what follows now so you can find it easily when the time comes. Here's our **crontab** file. The bold line is the actual configuration line. The rest are comments.

```
///  QuickRef: The fields of a crontab entry specify the following:
//
//   Field        Meaning
//   ----------------------------------------------------------
//     1          Minute (0-59)
//     2          Hour (0-23)
//     3          Day of the month (1-31)
//     4          Month of the year (1-12)
//     5          Day of the week (1-7, with 1=Monday)
//     6          User name (unused, so we always set it to "nobody")
//     7+         Command name and arguments (if command is preceded by
//                "-b" cron executes it in the background).

30 5 * * * nobody -l webstat
```

Cron is great. You can automate all sorts of routine things on your Mac with it. WebStat, on the other hand, has a bit of a drawback — you can't keep statistics on your server on different sets of documents independently. In English that means that if you have two clients parked on your Web server, you won't be able to use the same WebStat app to automatically generate two separate reports on the number of hits different clients' respective documents received. You'll have to set up two versions of MacHTTP and WebStat on the same machine (and give the second MacHTTP a different port number in the MacHTTP config file). No big deal. After all, you should almost be an expert now anyway. What's a few Web servers on the same machine to a master-blaster like you. Nada. You can have as many as you like, as long as you remember to pay Chuck for each license!

Don't get us wrong. We really like WebStat. It's trés cool and it does a nice job. But if you need an industrial-strength report generator for server statistics, your best bet is probably to hire a programmer to write you some code. (Do us a favor and put a copy up on the Net for everyone to enjoy if you do!)

While we're on the topic of server stats, let's make sure we're all clear about what is meant by the number of "hits" your server gets each day. We hear things like "Oh, we get 6000 hits a day" or "We're getting 100s of hits a minute." It's time to be a little more honest about what this means, folks. The way it comes across to someone who doesn't know doodley-squat about it is that 6000 people visited your site. Technically this may be true since MacHTTP is a "stateless" application. Stateless means that as soon as MacHTTP serves up the document that a client such as Netscape requests, the connection between the client and server is dropped. If the same client selects another document on your server a few seconds later, it counts as another hit.

The fact that MacHTTP is stateless makes the whole operation between client and server very efficient. Someday we'd love to read a more technical book about the inner workings of MacHTTP. Maybe Chuck Shotton will write one. Here's the point though: If you are consciously trying to pull the wool over someone's eyes and paint a picture of enormous server activity, or if you can't back up your words with an explanation of what you mean by "hits" on your server, sooner or later the rest of the world is going to wise up and you'll get busted. So be straight about it and don't let someone go off with the wrong impression.

THE GRAPHIC ZONE

Now it's time to dress up your Web site by introducing some graphics. We'll start with the easy stuff and show you how to make transparent and interlaced GIFs. From there we'll dive into the advanced stuff like clickable maps.

GIFs: Transparent and Interlaced

Let's start with how to make a simple GIF blend in with the default screen color of the Web browsers that are out there. GIF files are the way to go, by the way, because most Web browsers don't support other file formats without requiring a helper app. Netscape will read JPEGs, but as of this writing it's the only browser that will. There's a big advantage to JPEGs: the compression ratio for most images is much higher. You give up a little in quality with JPEG's "lossy" compression method, but your image will be a whole lot smaller. Most browsers will probably have built-in support for JPEG in the near future. Until then, use GIFs because, as good as Netscape is, not everyone uses it.

Try a Little Transparency

Many graphics on Web pages blend right into gray backgrounds unless you, the browser user, messed around with your browser's preferences. You've seen that, right? And once in a while you see a Web page where there is white space behind the image... But for the most part, WebMasters have figured out how to make their GIFs transparent. Here's how you do it.

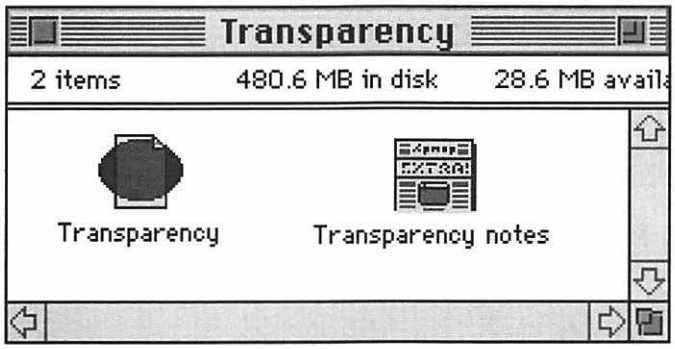


Figure 4.17: Transparency Folder.

You need a little utility called Transparency by Aaron Giles. It's free and is included on the CD-ROM that came with this book. Transparency is a "drag & drop" application; just drag a GIF file onto the Transparency icon (Figure 4.17). Or, you can launch the program and use the Open command to open your GIF files. Once your GIF file is open, click anywhere inside the image window and a palette of colors will appear. Select the background color you want. (*Hint:* The upper left corner of the palette will get you the gray you're probably looking for.) When you've done that, choose **Save as GIF89...** from the **File** menu, save the resulting file, and type in a filename.

That's it. Figures 4.18 through 4.21 show some before and after examples.

Figure 4.18: A Nontransparent GIF.

Figure 4.19: A Transparent GIF.

Figure 4.20: A Nontransparent GIF Being Browsed with Netscape.

Figure 4.21: A Transparent GIF Being Browsed with Netscape.

GraphicConverter: Interlaced GIFs and Much, Much More!

Another great utility to have in your WebMaster bag of tricks is GraphicConverter. This is a great little program that converts graphics to and from almost every type of format you can imagine (Figure 4.22). We include GraphicConverter with the book too. It's shareware and costs $35; it's well worth the money.

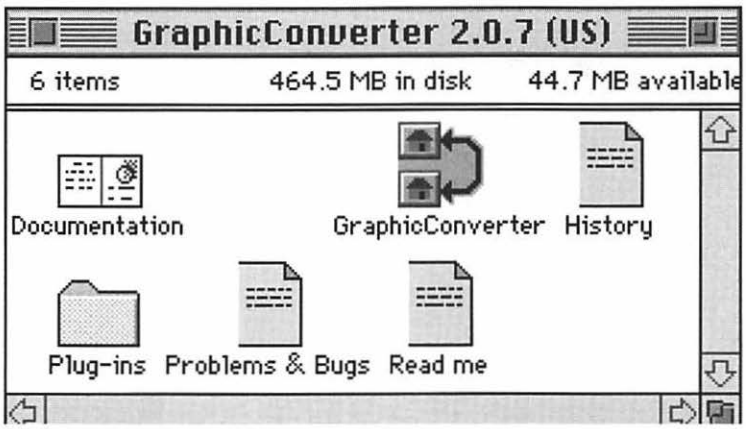


Figure 4.22: GraphicConverter Folder.

The documentation that comes with GraphicConverter is very thorough. In other words, you can use GraphicConverter to design graphics, even if you're not a Photoshop whiz.

We like GraphicConverter because we haven't found anything so far that it can't translate. Jeff got hooked on it in the early WebMaster days when it was the first and easiest way to create clickable images that redirected you to other Web pages.

The main thing we use GraphicConverter for these days is to create those nifty interlaced GIFs that Netscape "fades" in when it comes across them. You've seen 'em a million times — that kind of venetian blind effect as the picture appears. They're pretty, at least the first few times...

On the other hand, interlaced GIFs may be getting a little overdone (probably, for the most part, by people like us who can't do much else original in the way of graphics). Our advice: Use them sparingly.

Sigh. Of course, you want to create one. Who are we to deny you? Here's how: Choose **Save as...** from the **File** menu. Choose the GIF file format from the pop-up menu, then click the **Options...** button. You'll see a window that looks like the one in Figure 4.23.

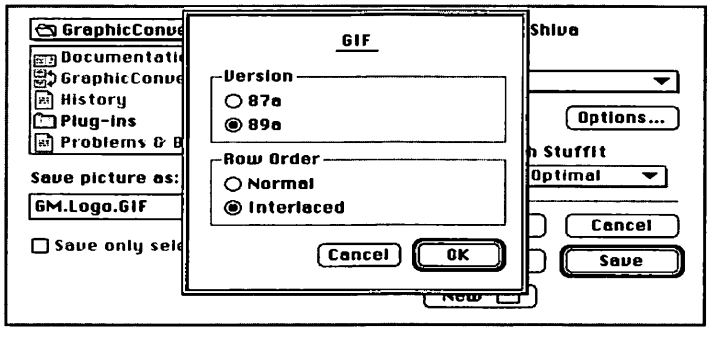


Figure 4.23: Saving a GIF as Interlaced.

Click the **Interlaced** radio button to make it interlaced. (Also click the **89a** button if it's not already clicked. Don't ask why, it's not important. Just do it.) How's that for easy? Keep in mind that once you select the option to save your GIF as interlaced, GraphicConverter will stay in the interlaced mode, and will save each succeeding GIF as interlaced until you change it using the **Options...** button again.

If you want to make a GIF transparent *and* interlaced you have to save it as an interlaced GIF first, and then make it transparent.

It seems like there is a new version of GraphicConverter coming out every week or so and the procedure has changed at least once that we are aware of. You may have to hunt for the option a little if you have a different version. (*Hint:* If you can't find it quickly you can always post a question to the MacHTTP Listserve like we did.)

Another thing that GraphicConverter is handy for is making thumbnail pictures on your pages. Thumbnails are reduced-size pictures that are linked to their larger original. They provide a quick preview. If a user is interested in seeing more they can just click on the thumbnail and load the bigger version. Thumbnails are great on a site that is heavy on graphics. We've seen them used effectively on servers showing off a museum or collection of art, for instance. Check out these URLs for examples. They're from the FineArt Forum WWW Resource Directory:

http://www.msstate.edu/Fineart_Online/art-resources-museums.html

http://www.msstate.edu/Fineart_Online/art-resources-commercial.html

Creating Thumbnails with GraphicConverter

To create a thumbnail, first drag your graphic onto the GraphicConverter icon. Next, press **Size** from the **Picture** menu and select **Scale** from the pop-up submenu in the same fell swoop. Click the **Size** radio button and type in the new picture width and height. The rule of thumb...nails (sorry, we couldn't resist) is to make them 75 pixels high by 75 pixels wide.

Once you've done that, GraphicConverter will show you the new image, in its new size. Save it with a new name and use that filename in combo with the filename of the larger original file in your HTML like this:

```
<A HREF="GM1.GIF"><img src="GM2.GIF"></A>
```

GM1 is the big daddy (Figure 4.24) and **GM2** is the little bambino (Figure 4.25).

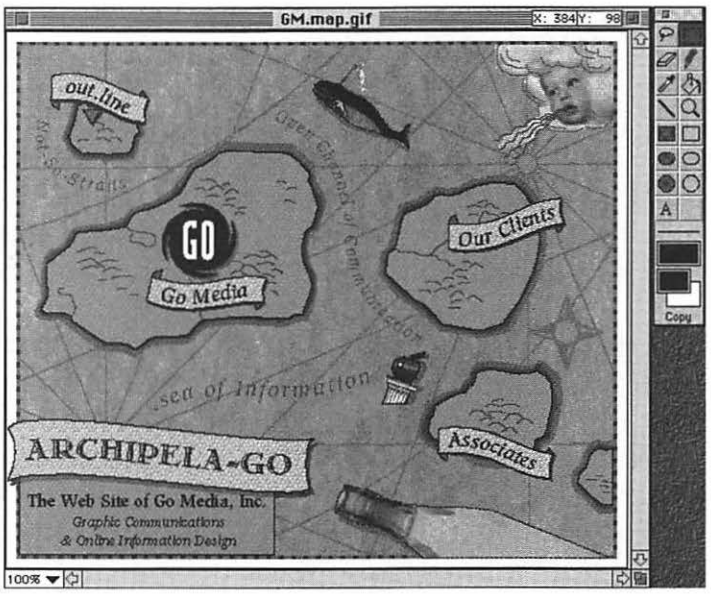


Figure 4.24: Thumbnail GIFs — Papasito.

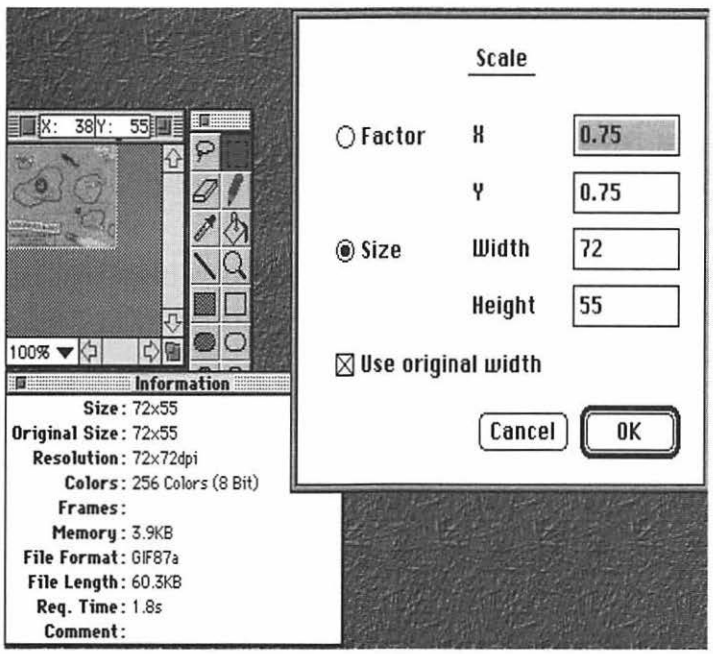


Figure 4.25: Thumbnail GIFs — Bambino.

Our thanks to Stephen Collins at the University of Minnesota for showing us and so many others how to work magic with GraphicConverter.

Netscape is the browser that currently pushes the envelope with tricks you can do when it comes to displaying graphics. There are all sorts of other things you can do to create high-impact documents if Netscape is the browser you're creating for.

(To find out more about some of the other bells and whistles that Netscape supports when you're messing around with graphics, go to the Netscape Communications Server at **http://home.mcom.com/** and wander around. Remember, however, that not every browser out there will jump up and down with delight when it comes across HTML written specifically for Netscape.)

Tricks with Photoshop — Julie Gomoll

Here are a few tips to make your color Photoshop images small while ensuring that they continue looking great.

- If you're scanning a photo, scan it at a relatively high resolution—at least 300 dpi. Make it look the way you want at this higher resolution.
- Do your image editing in RGB mode. When your image looks the way you want it to, size it down to the dimension you want for your Web page.
- Crop the image as closely as you can.
- Go to "image size" and set the resolution to 72 dpi. This is the resolution of most monitors.
- Go to "mode" and select "indexed color." You have a resolution choice of 3, 4, 5, 6, 7, or 8 bits/pixel. 8 bits per pixel is the default, as is adaptive palette and diffusion dither. Your file size will be the same no matter which resolution you choose. Eight bits will give you the fullest color.
- Now save it as a GIF. The file size should remain the same.

An example: I opened a 300 dpi scan that showed up on my desktop as a 1,684K Photoshop file. It opened as a 1.19Mb RGB file. When I changed the image resolution, the file size dropped to 71K. Changing the mode to indexed color reduced it to 24K.

Who Let the Cat In? Or, Putting Sounds on Your Pages

One other thing... Wanna know how to put sound on your Web page? You know, like that cat at the White House Web page at **http://www.whitehouse.gov/White_House/Family/other/socks.au.**

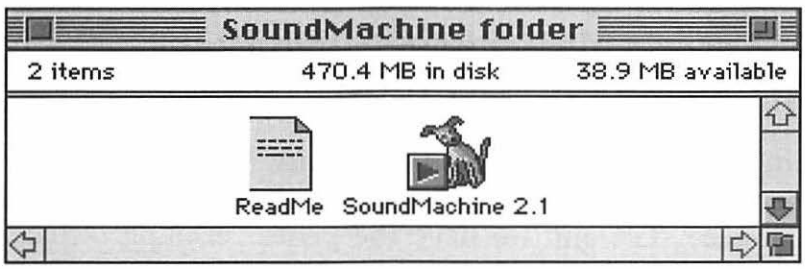


Figure 4.26: SoundMachine Folder

We've thoughtfully included SoundMachine on the CD-ROM you got with the book, so get ready to record (Figure 4.26). It's freeware. (We like how when you launch it you get the "Welcome to SoundMachine" voice.)

Ready? Let's go. Choose **AU Record** from the **Recording menu** and end your filename with the suffix **.au**. To record your sound, click on the **Record** button. When you're done, click **Stop**. Is English a great language, or what? Save your sound (don't forget the **.au** suffix) and you're done.

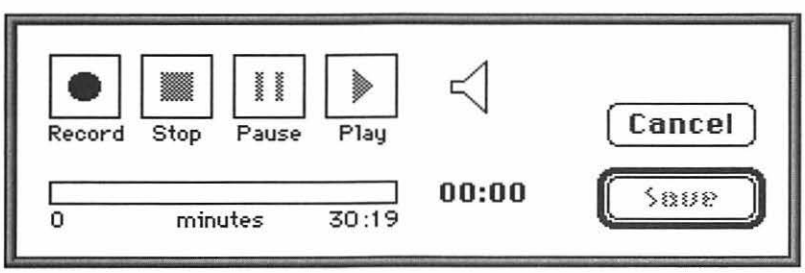


Figure 4.27: SoundMachine Record Panel.

The deal with sounds and the Web is that sounds are big-time memory and bandwidth hogs. Keep that in mind. There are plenty of sounds on BBSs and on the online services too. Pay attention to the copyright laws and have fun.

Rather than make you look back to the HTML chapter, here's the HTML tag for adding sound:

```
<A HREF="SomeEnchantedEvening.au">You may meet a
stranger</A>
```

MAKING CLICKABLE MAPS: INTRODUCTION TO THE COMMON GATEWAY INTERFACE (CGI)

We've mentioned CGI plenty. More than plenty. You've been kind to let us keep mentioning it and forever promising you we would explain it. Well, it's time to start and we have the perfect example — clickable graphics. The first CGI you will probably use will be a clickable graphic — also known as a clickable map.

CGI stands for Common Gateway Interface. Big deal. What does it mean and what can it do for me, you ask? If you really want to know, read on. A word of caution, though. You may really like this stuff and turn into a programmer. This is where Jeff started getting sucked into the abyss and almost stopped having a life.

But we digress....

The Common Gateway Interface provides a standard for how external programs interface with information servers such as MacHTTP. There are different CGI definitions for different platforms — Mac, UNIX, etc. There's a big detailed explanation of CGI that NCSA has put together at **http://hoohoo.ncsa.uiuc.edu/cgi/overview.html.**

To understand the role of CGI with MacHTTP, you need to step back and get an overview of the whole client-server story — briefly. In a nutshell, CGIs can really make a Web site powerful. Figure 4.28 helps illustrate CGIs as we present the pieces in a coherent fashion.

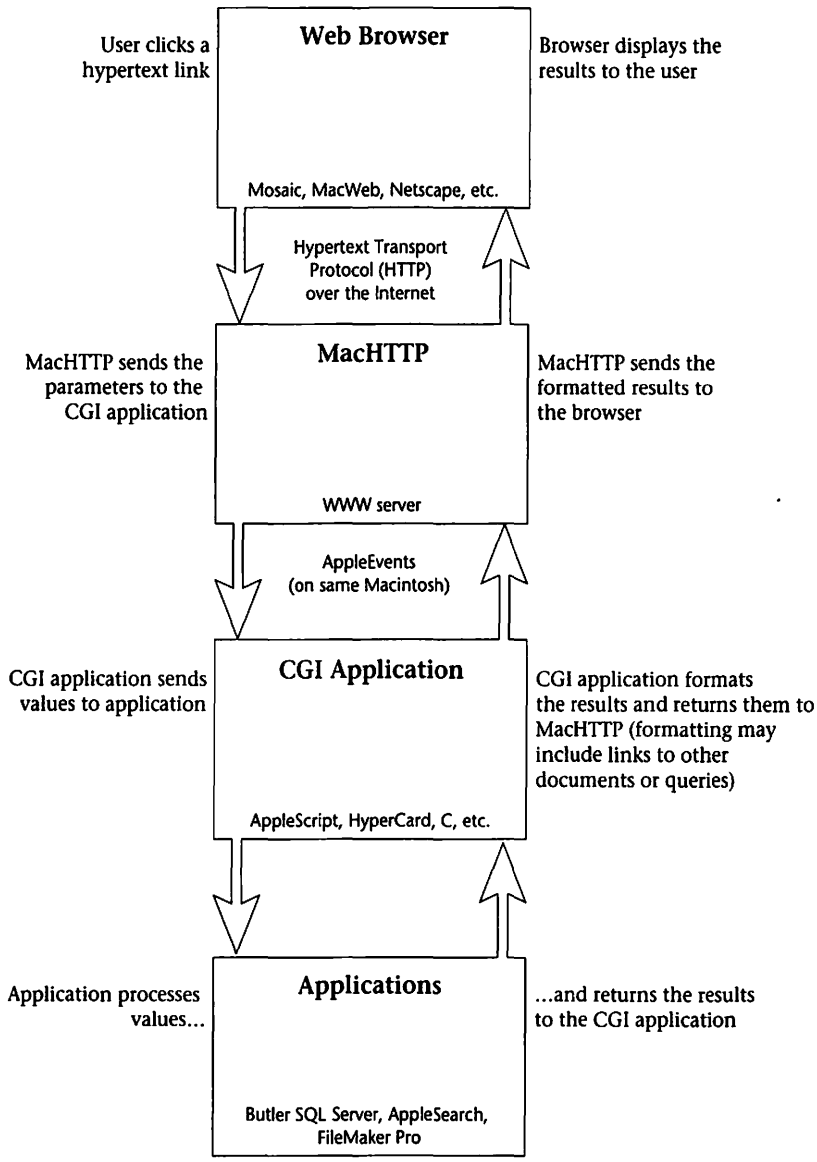


Figure 4.28: The CGI Story — Client/Server and Common Gateway Interface.

MacHTTP is the server, right? Netscape, Mosaic, MacWeb, etc., are the clients. MacHTTP "listens" for clients. When one comes along and requests a URL, MacHTTP goes and finds the document and passes it back to the client.

The client can also request a URL that fires off an application that does something with the info that the client has sent (such as clicking on a specific range of coordinates of an image or sending responses to a survey form). That application is called a CGI. It can be an AppleScript, a C program, or a HyperCard routine, for example. For what it's worth, on a UNIX box CGIs are often written in PERL. (Perls before swine???)

Example Time: You Too Can Do CGI...

Let's get right to an example before this gets too complicated. There'll be plenty of time to wrap your head around the CGI thing in the next chapter, which is almost entirely about CGIs. We could have used the example of a map of the United States or Europe or Asia — but since this book is going around the world we didn't want to leave anyone out... So we're going to use a simple graphic and the Go Media home page as our examples. (You can check them out in real life at **http://www.gomedia.com.**)

To create a clickable map, you need to use a CGI that allows MacHTTP to serve clickable graphics with "hotspots" that can call any other document that has a URL — either on your server or out there on the big wide web of the Net. To make this all work you need to do two things: First you have to define what parts of your graphic are going to be clickable and where those clicks are going to take you. Second, you need to install a CGI in your MacHTTP folder to make it happen.

Step 1: WebMap

To define the "hot" regions of your graphic the hard way, you figure out all of the x,y coordinates using Photoshop or GraphicConverter and crank out the HTML in the standard HTML map file format. But why bother? Take the easy way out and use WebMap. WebMap is nectar.

WebMap reads GIF and PICT files and allows you to draw various shapes such as rectangles, circles, and polygons that represent the hot areas on an HTML graphic. Cool, huh?

Wait until you see this baby work for yourself. Is it on the CD-ROM? You bet, thanks to Rowland Smith, the developer. It's another freebie. (Many people have created cool add-ons for MacHTTP and released them as freeware as a way of giving something back to the Net community. We love it, don't you?)

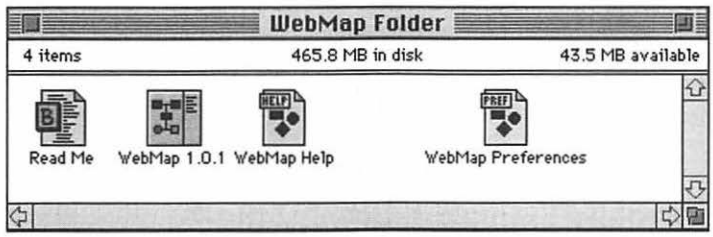


Figure 4.29: WebMap Folder.

Launch WebMap (Figure 4.29), then open a graphic file. Use the tool palette to pick a shape to draw around the part or parts of the graphic you want to be clickable for a Web browser. (*Hint:* You may want to set the preferences of WebMap to draw circles from the center — we found that easier.)

Once you've drawn the shape where you want it to be, double-click the word "undefined" at the right and enter the URL you want the user to be taken to when he/she selects the defined region (Figure 4.30). You also need to set the default URL in case the user clicks outside your defined region (Figure 4.31).

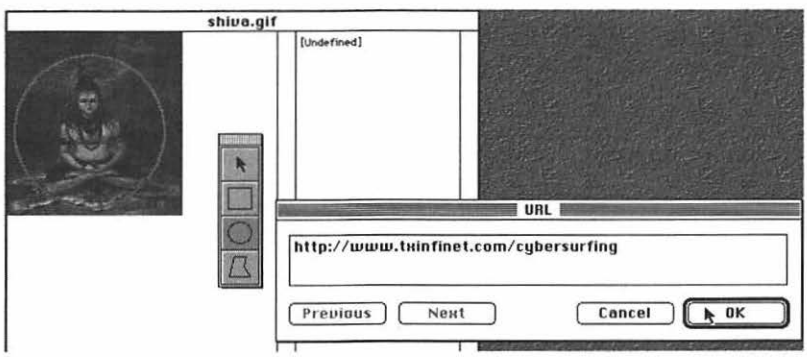


Figure 4.30: WebMap Example — Defining the Destination URL.

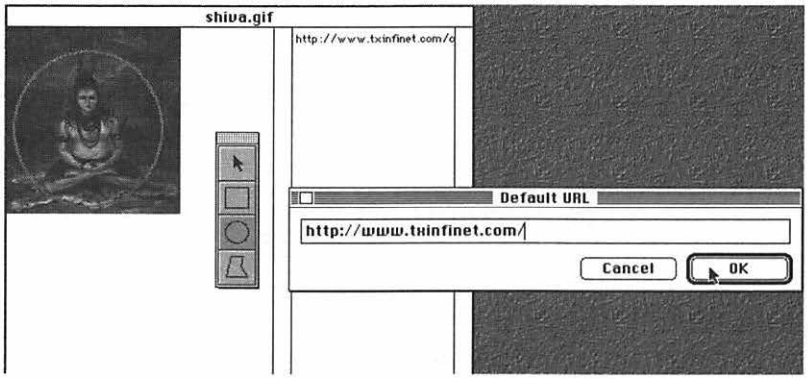


Figure 4.31: WebMap Example — Defining the Default URL.

Once you've gone through and defined all the regions you want to be clickable and have assigned the destination URLS, choose **Export as Text...** from the **File** menu and save the **.map** file in a handy place (Figure 4.32). (This filename must end with the suffix **.map**.)

For what it's worth, you can choose CERN or NCSA format when you save; there's supposed to be a difference but we haven't noticed any. CERN and NCSA are duking it out on a bunch of things. This is one of them. It will all get resolved someday. From what we can tell, the difference as far as the imagemap format goes still won't matter. (We choose NCSA because CERN is on the other side of the Atlantic and we like to buy American.)

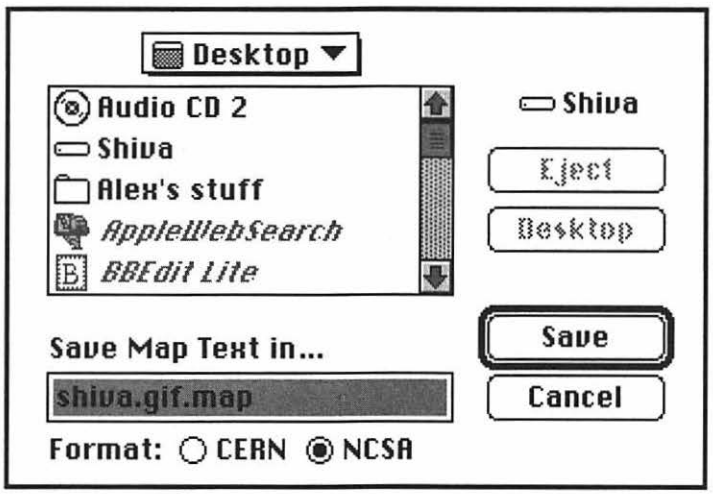


Figure 4.32: WebMap Example — Exporting as Text.

Once you've exported, you end up with a text file that looks like the one shown in Figure 4.33.

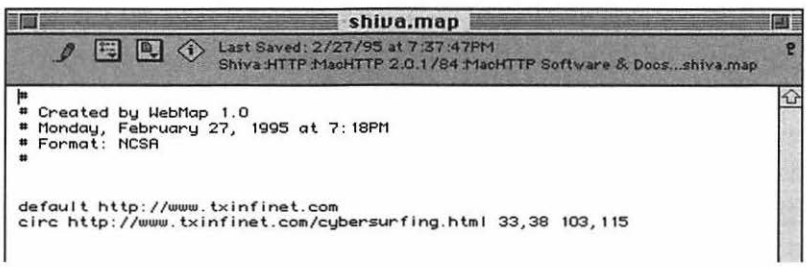


Figure 4.33: WebMap Example — The **.map** File.

Figure 4.34 shows the **.map** file for the Go Media home page. To see this home page in action, and check out the clickable graphics on it, point your browser at **http://www.gomedia.com**.

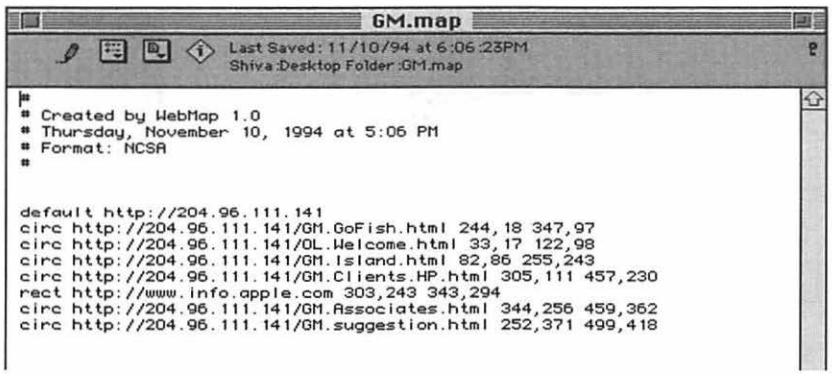


Figure 4.34: WebMap Example — Go Media Home Page **.map** File.

OK. We're done with the first part of making this puppy. Now it's time to set the CGI up and write the HTML that will start the whole thing happening.

Step 2: MapServe

There are a few canned CGIs around for doing the imagemap thing. For us, it's a toss-up between two.

(Don't worry about trying to understand how you make a CGI yet. We're going to use one that was written by somebody else first; in the next chapter you'll learn about writing your own.)

The very first CGI we had experience with is called Mac-ImageMap, written by Lutz Weimann from Germany. It's freeware and does the job of clickable graphic magic nicely. You can get the latest version of Mac-ImageMap at

http://wezl.zib-berlin.de/imagemap/MacImageMap.html

(There's also a pointer to it on the WebMaster Mac WWW server.) The other CGI out there that does the job is called MapServe. The home page for MapServe is: **http://www.spub.ksu.edu/other/machttp tools/mapserve/mapserve.html.**

We included MapServe (Figure 4.35) with the book only because it was quicker to download when we went to nab the latest versions of each. We admit that that's not much of a reason — but it's the only one

we could think of. Either MapServe or Mac-ImageMap will serve your needs nicely (pun intended). Kelly Campbell is the developer of MapServe and the documentation she did for MapServe is a little easier to follow than the documentation for ImageMap. Kelly asks for a small contribution of $20. Go ahead and send the money. It will help her through college at KSU.

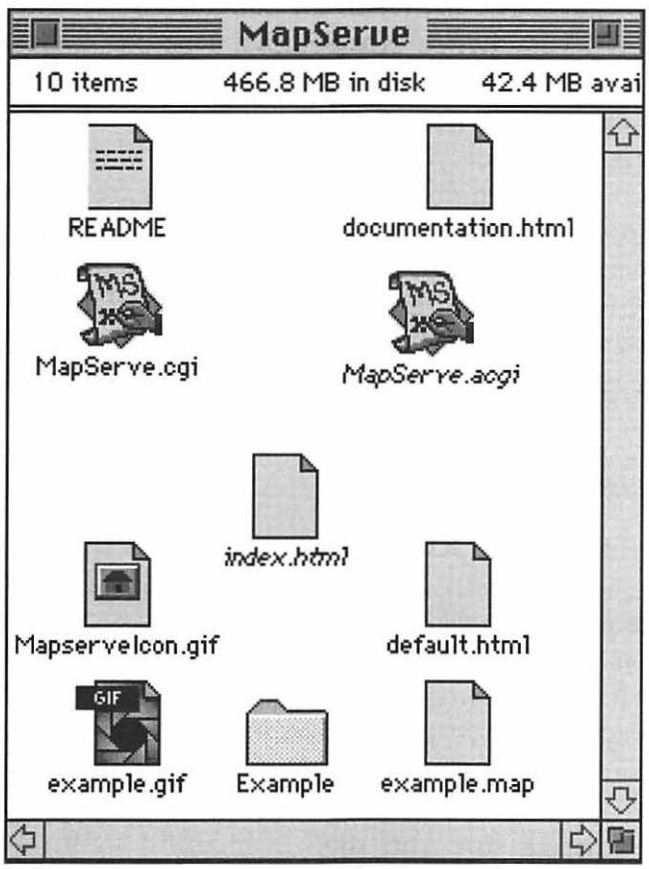


Figure 4.35: MapServe Folder.

Drop the whole MapServe folder in your MacHTTP folder. Now go grab the **.map** file you created in Step 1 of this deal and drop it in the MapServe folder (it needs to be in the same folder as **MapServe.acgi**.)

The Next Step (Step 3)

Write the HTML that is going to cause the whole deal to fire up and happen. It should be in the following format:

```
<A HREF="/mapserve/mapserve.acgi$shiva.map"><IMG SRC="/
mapserve/shiva.gif" ISMAP></A>
```

Here's what's going on in this example. You've got an inline graphic called **shiva.gif**. You've also got an HTML tag called **ISMAP** whose sole purpose in life is to tell the client's browser that it needs to send the coordinates the user clicked on to the program called **mapserve.acgi**.

You see that **shiva.map** after the **$** sign: **$shiva.map**? That part of the HTML refers to the **.map** file you created with WebMap. The **.map** file holds all the coordinates that you defined as clickable, remember?

Well, when **mapserve.acgi** is launched, it looks for that **.map** file and, depending on what coordinates were "passed" to the **.acgi**, the client (the user) will get redirected to the URL that is associated with those coordinates as defined in the **.map** file.

You got all that? It may be a little confusing at first. It was to us first time we tried it.

Now Here's the Really Important Part...

You have to be careful that the actual location of the **.acgi**, **.map** file, and **.gif** file are where you specify them to be in HTML. Remember, it doesn't matter where in the MacHTTP folder they reside — they can be nested deep within if you like as long as the subfolder is somewhere inside the MacHTTP folder. We suggest you keep everything together in the same folder, though. It's up to you to organize your folders and files in a way that makes sense to you, but remember to do the paths properly in the HTML.

One other thing: Make sure you spell everything right. We had everything in the MapServe folder and had the MapServe folder inside the MacHTTP folder. But when we did the HTML, we ("we" meaning Jeff) typed the path out as "/mapserver/mapserve.acgi". We kept getting the dreaded "Error: File Not Found" message when we clicked on the image with Netscape. The correct path for the HTML should have been "/mapserve/mapserve.acgi" — the extra "r" killed us.

A Great Example of Doing It Right

The Go Media WWW server at **http://www.gomedia.com** is one of the cooler sites on the Net, the fact that Jeff works on projects with Go Media notwithstanding. It does a really nice job with a clickable map, if we do say so ourselves.

When you land at the Go Media WWW Server you're greeted by a big clickable map with all sorts of underlying URL redirections waiting to take you on an interactive adventure (Figure 4.36).

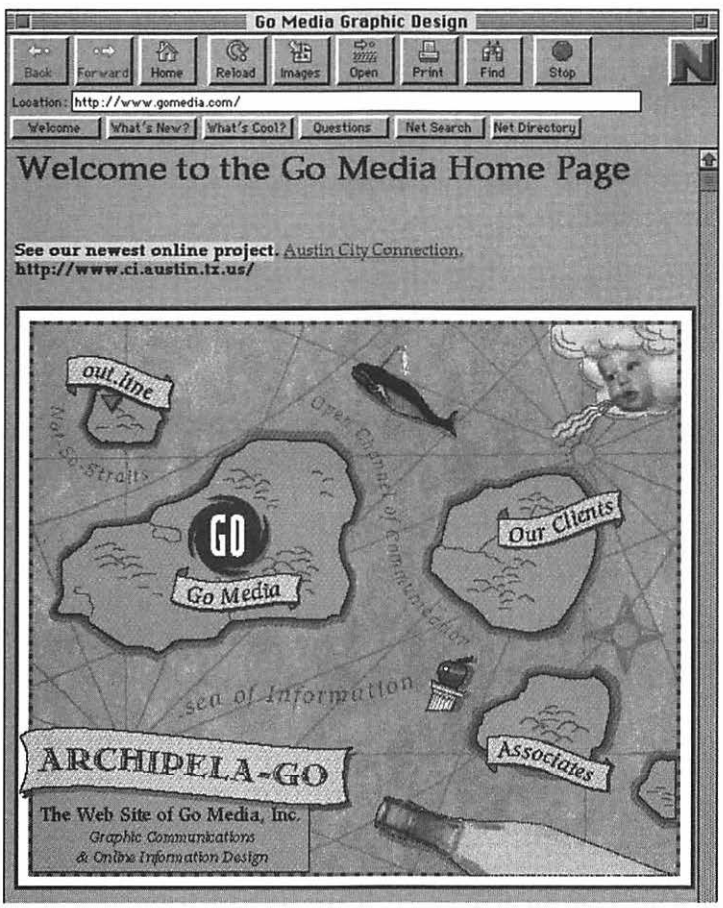


Figure 4.36: Go Media Home Page.

If you took a look at the **GM.map** file that got created by WebMap when they built their home page, you would see the file in Figure 4.37 that stores the clickable coordinates and the URLs you get redirected to.

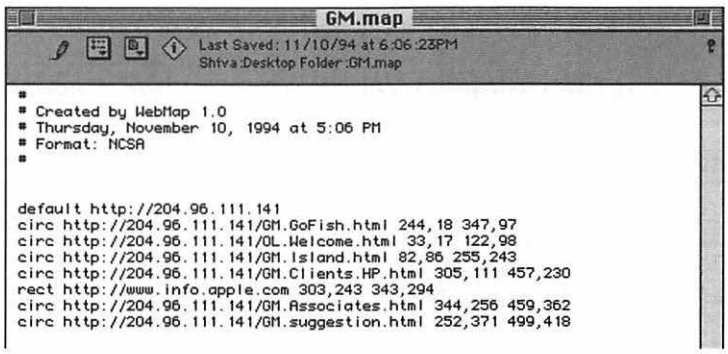


Figure 4.37: Go Media **.map** File Showing URL Redirects and Clickable Coordinates.

The screen shot in Figure 4.38 shows the source HTML for the Go Media home page with the HTML tag that makes the entire graphic a clickable map.

Figure 4.38: Go Media Home Page — Source HTML.

When you click on that portion of the home page that's an island with the "Go" logo on it, MapServe does its thing (Figure 4.39). Your browser passes the coordinates you clicked on to the **mapserve.acgi** and you get redirected to the URL that is associated with the coordinates of the region you clicked on.

It's a really fun way to make your WWW site interesting. Remember, though, don't let those graphics get too large!

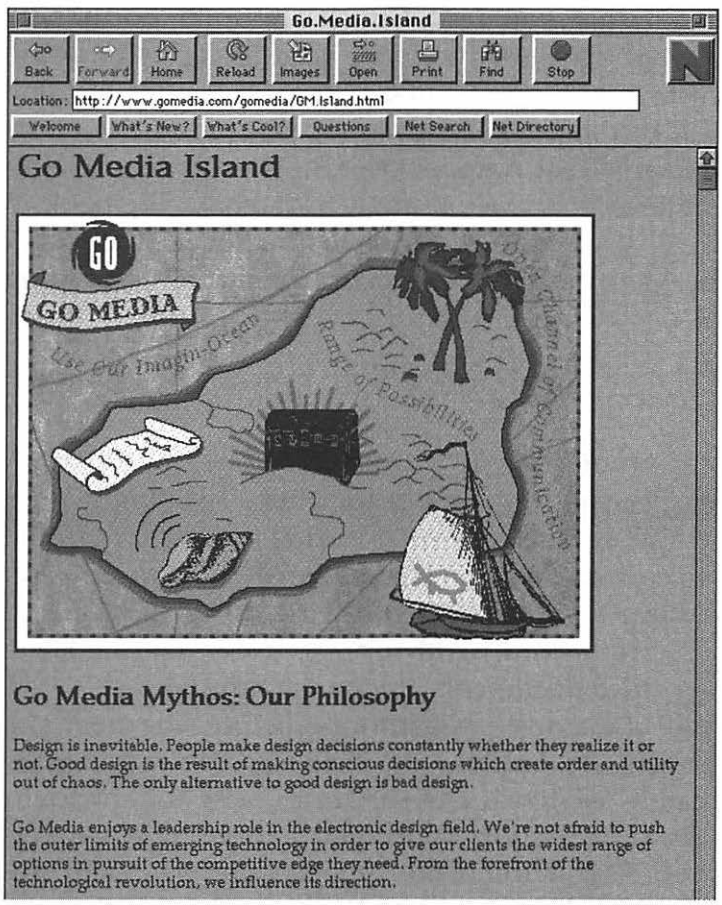


Figure 4.39: Clickable Map Results — The Go Media Island.

A FINAL WORD

We've covered a lot of ground in this chapter, from tweaking your MacHTTP server to introducing you to CGI programming. For many of you it will be a lot to digest. Now would be a good time to take a break and really get comfortable with everything you've learned so far. Practice makes perfect.

There's a lot of detail we deliberately didn't cover. Once again, we strongly recommend that you read Chuck Shotton's documentation carefully. We've given you everything you need to get up and running and well on your way to this point, but Chuck is the Master. Just do it. In fact, go do it now if you haven't already.

We also recommend that you look over the developers' own instructions for the WebMaster tools we mention throughout the book. We're bound to miss things you might be interested in. Believe it or not, even though this *is* the Macintosh you can learn something by reading the docs.

What's Next?

In the next chapter we are going to go way off into gearhead land. With the help of Jon Weiderspan we have put together a spellbinding tutorial on CGI. We have also assembled several working examples of different ways to use CGI to connect MacHTTP and your Web browser to applications such as FileMaker Pro, Butler SQL database, and AppleSearch. Plus, you'll learn how to create forms and surveys, search and update databases, retrieve keyworded documents, and much, much more.

If you've never done any programming before, the next chapter may be a little challenging. But take the challenge. We're going to. The two of us have barely scratched the surface of how to craft our own CGIs. But never fear: We'll show you enough to get you started, and make it work for you. As a matter of fact, you may never actually need to learn how to do anything but copy from the examples you'll work with in the next chapter!

The great thing about MacHTTP and the path of the WebMaster is that you don't have to reinvent the wheel to get to cyber-enlightenment.

Onward!

Chapter 5

CGIs, Forms, Bells, and Whistles

Everything You Need to Know to Extend MacHTTP

Everything you've learned and put to practice with MacHTTP was lightweight compared to what's coming up next. It's time to learn how to extend MacHTTP and connect users browsing your WWW site using CGIs and back-end applications such as FileMaker Pro, Butler SQL database, and AppleSearch. We told you that this chapter was going to get a little more complicated, right? Hang in there with us on this stuff. It's what will separate you from the zillion other Web sites that are popping up these days. We'll try to break things up with screen shots and a sidebar here and there for you, but for the most part you're just going to have to go off someplace quiet and delve deep into the mysteries of CGI. But learning this stuff will definitely be worth it in terms of cool stuff you can do on *your* Web site.

An effective Web site often requires a way for users to interact with the information that you are serving up. It's not always enough to be able to present interesting hyperlinked text files, graphics, and clickable maps. We've all seen examples of great interactive sites — whether it's searching a database of vegetarian recipes, filling out a survey online, pulling up the weather for your city, or online ordering — that demonstrate the myriad of ways you can make your site sparkle and shine. What all these sites have in common is a method of communication between the client's browser, the Web server, and some application that shoots back or accepts information based on the client's request.

To interact with a surfer out there, you need to create applications that will work with MacHTTP to extend its capabilities. These applications handle things like maps (as you saw in the last chapter), forms,

automatic eMail replies, document and database searching and input, plus a whole array of wild and innovative ideas that WebMasters are just starting to implement such as interactive video and real-time chat.

WELCOME TO CGI LAND!

What you need to get going with CGIs is an understanding of the pieces that go together to make the whole interactive transaction happen. To that end you need to understand a few terms and you need some basic tutorials on creating a CGI. Most important, you need some real-life examples that you can customize and make your own. Well guess what... You're going to get it all.

We defined what a CGI is in the last chapter and showed you a nice diagram of how the client, the server, and the CGI all fit together and talk to each other. You may want to go back and take a look at it again. It's on page 143. It made it a lot easier for us to go into the details of CGIs when we had that visual in mind.

A great tutorial will help enormously. Rather than try to create a great tutorial ourselves, we've received permission to include Jon Wiederspan's truly great tutorials on the CD-ROM that came with this book. After you cruise through this chapter and get a better handle on how CGIs are used, we strongly recommend you go through Jon's tutorials.

A Word about Jon Wiederspan's CGI Tutorials

First, Jon welcomes eMail and questions about the tutorials. We think that's really cool. Next to Chuck Shotton, Jon is probably the most generous person on earth with his time and help. You can reach him at **jonwd@tjp.washington.edu**.

He did ask us to make two requests, though. First, please read the tutorials carefully before firing off questions to him. Second, do the tutorials in order. They build on each other. Jon says more than 95% of the questions he gets are addressed fully in the tutorials. So read them carefully and do the examples before firing off eMail to Jon.

You can also post questions to the MacHTTP mailing list. There are a lot of experts there to help, too.

From time to time Jon updates his tutorials. You can find the latest version from the pointer we have to them on the Webmaster Mac WWW server. You can also get there directly by going to **http://www.uwtc.washington.edu/Computing/WWW/Macintosh.html**.

Our gratitude goes to Jon for the work he did putting together the CGI tutorials for WebMaster Mac and for his contributions to this chapter. His tutorials have really helped make MacHTTP the fabulous Web server that it is. What can we say? Thank goodness we didn't have to figure this all out on our own!

Two Methods and Why CGI Is Better

Chuck Shotton has built two methods into MacHTTP for communicating with external applications. The first method uses the "search" interface, the second uses CGI, the Common Gateway Interface. Both of these are common to a wide base of servers and are very nearly a standard, although the method of communicating with them varies from server to server. In MacHTTP's case, AppleEvents are used to communicate. AppleEvents are a method available in the Mac OS for applications to pass information back and forth, including instructions for another application, like "please open up and print this document."

Both of these methods work in a similar manner. First, the information that needs to be sent to the application is appended to the URL.

With the "search" method, a dollar sign ($) is used to indicate the start of the search method data and the end of the filename: **http://www.uwtc.washington.edu/Computing/searchme$text_to_find.**

With the CGI method, a question mark (?) is used instead: **http://www.uwtc.washington.edu/Computing/searchme?text_to_find.**

The second step is to create an AppleEvent message containing the URL and the data.

The third step is to send this AppleEvent to the correct application. The URL is used to determine what the correct application is. Instead of

specifying a document in the URL, you specify the CGI: **/Computing/WWW/TestApp.cgi**.

The final step, for MacHTTP, is to wait for the application to send an AppleEvent back. That instruction will give MacHTTP directions on what to do next. The AppleEvent that is returned can contain either HTML text (which is passed right back to the user) or the URL to a document to be opened (this is called "URL redirection"), or one of several other codes to indicate an error of some kind.

The two methods seem very similar, but the CGI method is far superior. It can handle quite a bit of data, whereas the "search" method is limited to only very short strings. The CGI method can also be run asynchronously, so that the server isn't tied up handling just one request at a time. The CGI application gets much more information from MacHTTP, like a username and machine address, a password (if needed), the call pathway, and even the name of the client software being used. For these reasons, the search interface is no longer useful and may even disappear entirely. In the following pages we will focus entirely on CGIs.

AppleScript and You

The examples in this chapter will show you some ways to use CGI applications to extend the capabilities of your MacHTTP server. The primary requirements for getting a handle on CGIs are a willingness to learn, patience, and a MacHTTP server to practice on. But it's also a good idea to have some familiarity with AppleScript, though it's not absolutely necessary.

For experienced AppleScripters, the information in Jon's tutorials should be sufficient to get going. For everyone else we strongly recommend the 2nd edition of *Danny Goodman's AppleScript Handbook*, published by Random House. It sells for $39 in the U.S.

If you are not experienced in scripting, try the examples in this chapter and Jon's tutorials anyway. The more examples you work through, the better.

We recommend that you also subscribe to the MacScript mailing list. To subscribe, send mail to: **LISTSERV@Dartmouth.edu**. Don't bother

putting anything in the subject line of your message. Instead, put the following in the body: **SUB MACSCRPT your name.**

For example, Jeff would subscribe by typing **SUB MACSCRPT Jeff Evans** in the body of the message. If you have any further questions, you can send mail to **MacScripting-Request@dartmouth.edu.**

If you are just starting out with AppleScript, find someone on the Net off the MacHTTP mailing list or, better yet, someone in real life who will help you. We've been lucky to have friends in Austin like Alex Sirota, who knows AppleScript inside and out. He is also a database über-guru, so we really lucked out. For the most part our experience has been that the key to being successful in the early days of your life with CGIs lies in modifying existing ones rather than reinventing the wheel. There are a lot of CGIs for MacHTTP out there and more are coming every week. We've included a bunch of them on the CD-ROM that comes with this book. It's a good bet that one of them will do the job for you.

Also, be sure to check out the WebEdge WWW server at **http://www.webedge.com**. It's the best source for working examples of CGIs for MacHTTP.

What You Need to Get Started...

Before you run off and try to make a CGI application, you need to get some items set up on your server. The following is a list of what is required to complete all the tutorials. These items will also be useful for the examples we give later in the chapter, as well as in any future scripts you write.

WebSTAR™ PS (formerly known as MacHTTP)
AppleScript 1.1 or later
ScriptTools 1.3.1 OSAX
Tokenize OSAX
Decode URL OSAX
DePlus OSAX

You might as well get everything you need to do CGIs properly installed on your server's hard drive before you begin.

AppleScript comes with System 7.5. It's also included with the aforementioned Goodman book. Everything else on the list is included on the *WebMaster Mac* CD-ROM.

What's an OSAX?

What's an OSAX you ask? Good question. OSAXen are AppleScript Language Extensions. An OSAX is to AppleScript what an XCMD is to HyperCard, basically.

The **Script Tools OSAXen** package is an extremely useful addition for any work with AppleScript. The **Tokenize OSAX** separates text blocks into lists based on the specified list of delimiters. The **Decode URL OSAX** extension translates the encoding used by HTTP clients to represent special characters (written by Chuck Shotton, who has also made the source code public domain). The **DePlus OSAX** converts all +'s that get passed by HTTP clients to spaces in a string (written by Jon Wiederspan, based on Chuck Shotton's code example).

If none of that makes any sense to you, don't worry. It will make sense when you work through the tutorials. So copy it all to your hard disk before you continue.

If you need to install AppleScript, follow the instructions in whatever package you bought to get AppleScript. If you're using System 7.5 or later, use the Installer that came with your System software CD-ROM or floppy disks to install it if you haven't done that already.

After a successful installation, you will have a new folder in your Extensions folder (which is in your System folder) called Scripting Additions. It contains extensions to your AppleScript system including libraries and OSAXs. This is a very important folder and over time you're going to end up with a ton of stuff in it.

To install an OSAX, you drag it to the folder named Scripting Additions in the Extensions folder of your System folder. There is no need to restart the computer — AppleScript additions are loaded dynamically.

THE BASICS: HOW CGIS WORK

When trying to understand how CGI applications work, the first thing to keep in mind is the difference between the server and the client. In case you love being bored to tears or have been hit on the head since starting this book and now have amnesia, MacHTTP is the server.

Mosaic, MacWeb, and Netscape are clients. When a user clicks on a link, the client sends a request to the server to see the file that the link points to. Here's an example of some typical client-server interaction with an HTML page:

> **Netscape:** I want to look at the page with the URL **http://www.uwtc.washington.edu/UWHome.html**.
>
> **MacHTTP:** Ah! That's one of my pages. Here is all the data from the URL you sent. I'm sending it as text.
>
> **Netscape:** Ah! I see the text begins with an HTML tag that I recognize. I'll interpret this as HTML then and display it correctly for my user.
>
> **MacHTTP:** I couldn't care less how you display it. I'm done dumping the data so good-bye.

Both the server and client have their respective responsibilities in the interaction. The server decides what the client is requesting and how to feed it back to the client. The client decides what to do with what is returned to display it for the user.

When the user requests a URL that involves a CGI application — like clicking on a map or submitting a form — the interaction is a little different. In this case there is also an interaction between the server and the CGI. Let's take a look now at the conversation that might occur between a server and client when handling a map click (assuming they speak English, of course).

> **User:** Hmmmm. A map of Washington state. There's a star in the upper right-hand corner. I wonder what that's for. Let's click on it.
>
> **Netscape:** Let's see, that click was at 287,48. I'll add that to the URL that was given with the map and send it back to the server.
>
> **MacHTTP:** Hey, someone sent me a URL with some extra data. That URL is for a CGI application on this machine. Well, I'll just send an AppleEvent to that

CGI application with the data enclosed in the **post_args** argument. I'm glad I don't have to do anything to the data myself.

CGI app: Finally, an AppleEvent! Let's see, first I decode the information in the **post_args** argument. Now I can use these map click coordinates to figure out what page to return to the client. Here it is — I'll send the server an AppleEvent reply containing the URL for the new page and a code to redirect the client to that page.

MacHTTP: Finally. I've been waiting for this AppleEvent reply. This code says I should redirect the client to this other page instead. I'll send back the new page then.

Netscape: Here comes another HTML page. Better display it nicely for my user.

User: That star represented Republic, Washington. I've never even heard of that place before!

That was a bit long-winded, but it gives you an idea of the complex interactions that go on when you're using a CGI application.

There are a couple of important points to remember from these dialogues:

1. The client is responsible for packaging data to be passed to the CGI application. With maps, this is easy; but when you begin trying to do forms, there will be significant differences in how clients package the information from a form.

2. The server generally does not do any processing of the actual data going to the CGI application or of pages being returned to the client. The server's only job is to handle connections, use the proper method to return pages and files, and handle communications between the client and CGI applications. The CGI application is responsible for either returning a *complete* page, including MacHTTP header, or for returning a code to redirect to a new URL and the new URL. The server has no idea what is going on

and will generate an error if the CGI fails to provide the needed information.

3. An AppleEvent is used for communications between MacHTTP and outside applications. There is a special AppleEvent used for communicating with CGI applications. This type of AppleEvent contains quite a bit of information in addition to information passed from the user. Any application that can accept such an AppleEvent and reply to it can be a CGI application.

Synchronicity

Not so long ago, MacHTTP was only able to handle CGI applications *synchronously*. That meant that every time a CGI application was called, MacHTTP waited for a response. Therefore, no one else could get any information until that CGI application finished and returned some information. A very slow connection, sending lots of data to a slow application, could tie up your server for several minutes or cause an error when MacHTTP got tired of waiting.

Chuck has since added the ability to run the CGI applications *asynchronously*. This means that MacHTTP doesn't have to wait uselessly while the CGI application runs. Instead, it passes an AppleEvent to the application and then goes on processing other things until a reply is received. To distinguish between the two, MacHTTP recognizes two different file extensions: ".cgi" means the file should be handled synchronously and ".acgi" means to handle it asynchronously. The latter is almost always preferable and is what you'll find out there for the most part, but a well-written CGI application can run either way. The key is to remember that AppleEvents might be queued up in your CGI application while it is processing the current event. Therefore, you want to design the application so it doesn't quit immediately after processing an event or it might miss the next one. Instead, give it time to test to see if there is another event queued up.

You can write a CGI application in any language you want as long as it accepts and returns the proper AppleEvent information. There are tons of examples out there and shells available in C, MacPerl, and Prograph. You could even use an application like 4D, HyperCard, SuperCard, or HyperApp. For most of us, the easiest way to write a CGI application is by using AppleScript.

Tutorial Detour — Show Me Something Already

This is a good place to detour off to Jon Wiederspan's CGI tutorials if you're hankering to get started. When you need a break from them you can come back here for some easy reading. If you're feeling like you've already had enough, hang out. There are examples and screen shots galore coming right up.

The rest of the chapter has examples of CGIs in practice. There are some really cool things being done with CGIs as we're writing. In fact, some really innovative ones came out of WebEdge, the first Mac WWW Developer's Conference.

More than 100 of some the world's best Mac WebMasters got together in Austin, Texas, for three days for presentations, CGI "hack sessions," and fun at Apple Computer. It was a frenzy of brain power and the beginning of what promises to be a popular event. Check out the WebEdge server at **http://www.webedge.com** for current information about WebEdge conferences and the latest in CGI land.

THE MAXUM MAN

So far, we've told you a lot about CGI applications, what they are, and how they work. If you took the time out to go through Jon Wiederspan's tutorials you may have even set up some of your own already. The CGI interface is also being used by third parties to create MacHTTP "plug-ins" that add unique new features to your Web server without the need for programming on your part. The best of these plug-in applications are as stable, reliable, and configurable as MacHTTP itself, and provide a variety of extremely useful functions.

The first such company to come out with a commercial CGI is Maxum Development. John O'Fallon is the founder and lead programmer at Maxum. Their "professional tools for creative World Wide Web applications" are easy enough for beginning WebMasters to set up and

start using right away, and also save the advanced WebMaster the effort of writing a new CGI for every new function of their server. They've got two great products for starters — NetCloak and NetForms. A bunch more are on the way. You can find NetCloak, NetForms, and a boatload of commercial, shareware, and free CGIs on the WebEdge WWW server at **http://www.webedge.com.** NetCloak and NetForms are included on a trial basis on the CD-ROM that came with this book.

NetCloak

NetCloak is a "server preprocessor" that extends the HTML language with a wide variety of new functionality. The limitation of standard HTML is that MacHTTP (and all HTTP servers, for that matter) is a data pump. The browser requests a document, and MacHTTP pumps it out, without looking at it at all. It's up to the browser to figure out what to do with the data. In the case of an HTML document, the browser displays the text in a clean, formatted manner.

NetCloak allows the server to look at the document while it is being sent out, and process certain HTML extension commands on the server before they are even sent to the browser. This is what is meant by the term "server preprocessor".

NetCloak uses the non-HTML standard **<SHOW>** and **<HIDE>** commands to turn on (**<SHOW>**) and turn off (**<HIDE>**) the MacHTTP "pump," effectively allowing you to tailor exactly which portions of the HTML document the browser will display. For example, if your organization was on the TCP/IP domain "192.1.2.", you might have the HTML code shown in Figure 5.1 on your home page. When someone from your organization loads this page, they would see the screen shown in Figure 5.2.

```
<HTML><TITLE>Welcome!</TITLE>
<H2>Welcome to our Web Site!</H2>
<HIDE><SHOW 192.1.2.>
<HR>
<B>Notice To Employees...</B>
This is a private message, which should be read only by our employees.
<SHOW>
<HR>
Here is a list of link to services on this site...
<UL>
<LI><A HREF="SubPage1.html">Public Discussion Forum</A>
<HIDE><SHOW 192.1.2.>
<LI><A HREF="SalesInfo.html">Q3 Sales Information (Employees Only)</A>
<SHOW>
<LI><A HREF="SubPage2.html">Product Literature</A>
</UL>
</HTML>
```

Figure 5.1: NetCloak Home Page Example in HTML.

Figure 5.2: NetCloaked Home Page Example Seen Internally.

But from anywhere outside your domain, someone who accessed the page would see the Figure 5.3 screen.

Figure 5.3: NetCloaked Home Page Example Seen Externally.

Your server is more secure because not only is sensitive information password protected, but unauthorized users don't even know that the sensitive information exists. Your server is also easier for the users coming in, because they see only the information and links that are applicable to them.

Domain-based security is just one of a dozen ways to tailor your documents. NetCloak also lets you create custom sections of your documents based on the browser (so you can use the latest features of new browsers without forsaking the older ones), day of week, time, username, password, etc. You can even have portions of your documents displayed randomly.

NetForms

The other early entrant into the commercial CGI market is John O'Fallon's NetForms. We wish it had been out sooner — it would have saved figuring out what CGI to use to handle information on a feedback page!

In a nutshell, NetForms lets users add pages to your server themselves. This is extremely powerful and useful. With NetForms, you can take the information entered by your users and turn it into formatted, clean HTML files automatically and immediately. With NetForms you can also create navigational hypertext links so that other users can easily find the comments and/or pages that have been entered by other users.

For example, let's say you were creating a pizza guide. You might want to do reviews of pizza restaurants, but you'd also like users to be able to add their own comments. Your home page might look like the one shown in Figure 5.4.

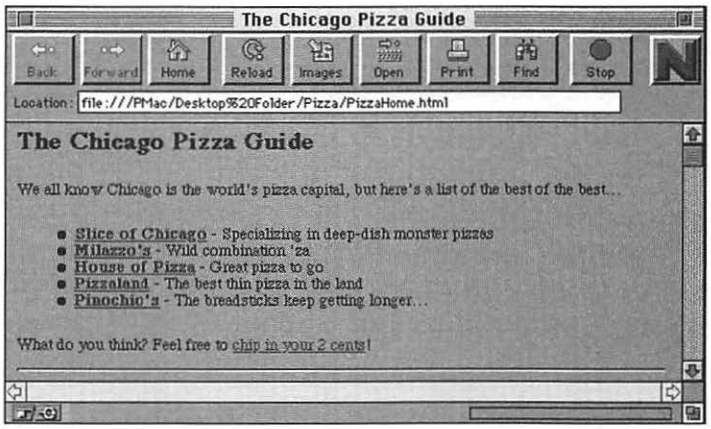


Figure 5.4: NetForms Example — Pizza Reviews Home Page.

When someone clicks on the "chip in your 2 cents" link, they would get a form that looks something like Figure 5.5.

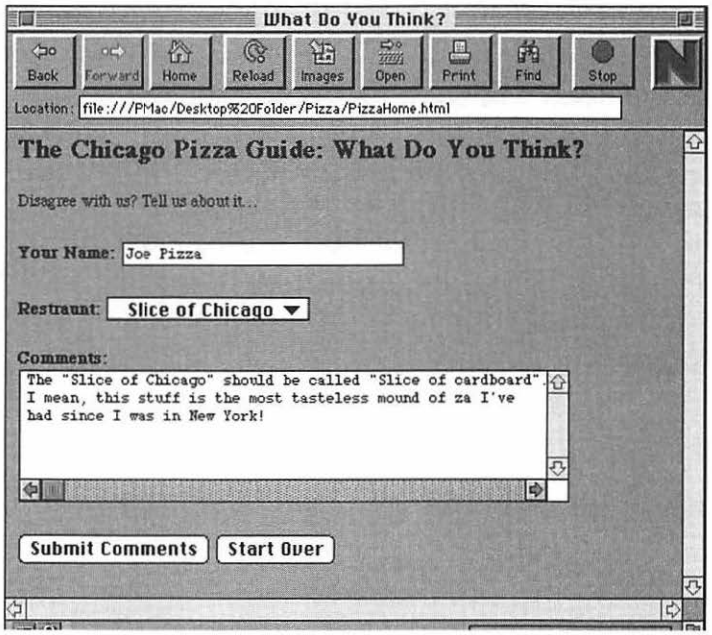


Figure 5.5: NetForms Page — Joe Reviews Slice of Chicago Pizza.

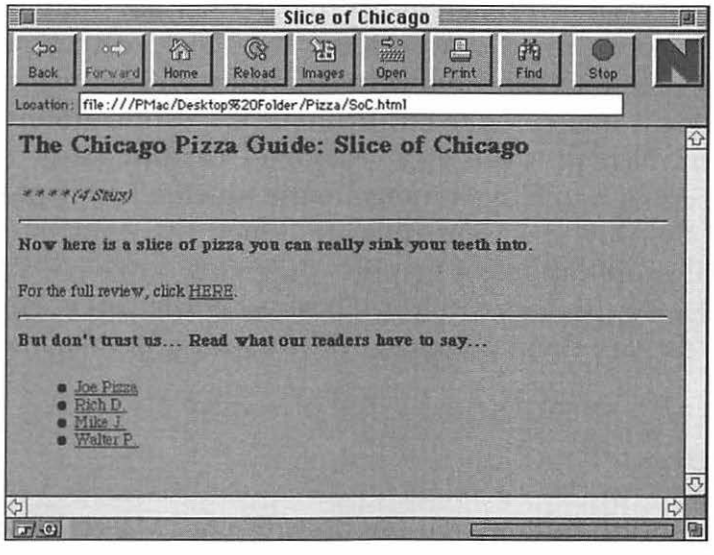


Figure 5.6: NetForms Example — Slice of Chicago Page.

The "Slice of Chicago" restaurant page would then show the Figure 5.6 screen, so that when someone clicks on the "Joe Pizza" link, they see the article Joe entered (Figure 5.7).

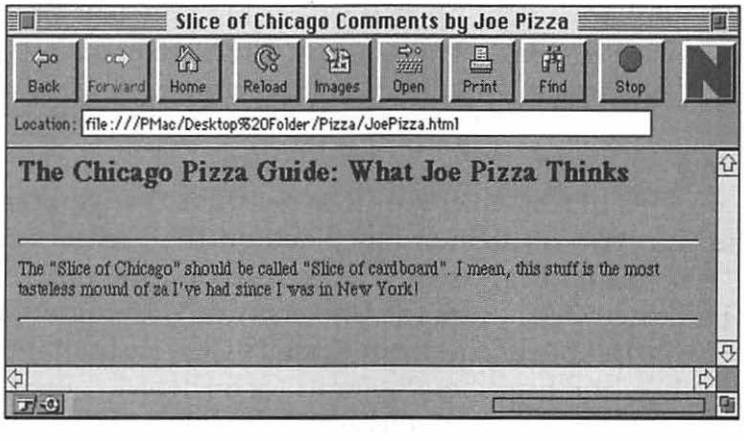


Figure 5.7: NetForms Example — Joe's Review.

Of course, this same server could have a "Pizza recipe forum" or a "The best pizza I ever had..." area. You, as the server administrator, set up the way the articles will look and feel, but it's the users who create the actual content.

NetForms can be used to allow people to enter just about any kind of article, from their own personal resumes to news articles, from recipes to book reviews. The messages can even be "threaded" so that people can post responses or additional comments to other, previously entered articles. This feature is often used to create online discussion groups.

NetCloak and NetForms are great examples of third-party tools. They are flexible enough to let you integrate the functions they provide into your server the way you want them to. Much like "plug-ins" for desktop publishing applications or graphics programs, a MacHTTP add-on such as NetCloak or NetForms gives your server entirely new capabilities... What you do with them is limited only by your own imagination.

FILEMAKER PRO

One of the most powerful things you can do with your MacHTTP server is hook into a database such as FileMaker Pro or Butler SQL from Everyware Development Corp. By the time you read this there will probably be CGIs written to hook MacHTTP into other databases such as 4th Dimension, Fox Pro, Sybase, and Oracle as well.

We're going to show you CGIs that work with both Butler and FileMaker Pro. We've included these CGIs on the CD-ROM that came with the book. You should know right off that we are big-time fans of Butler SQL, but let's start with FileMaker Pro. Many of you already have a copy, and it's relatively easy to get something up and running in just a few minutes that will integrate a FileMaker Pro database with MacHTTP and clients on the Web.

There are a couple of FileMaker Pro CGIs out there that you can use right out of the box. Two of them are included on the *WebMaster Mac* CD-ROM. The first, **fmpro.acgi**, is by Chuck Shotton. The other, **FileMaker.acgi**, is by Russell Owen from the University of Washington. Both are AppleScript CGIs. Definitely check out Russell's CGI. He built it based on Chuck's and others and it pushes the limits for a FileMaker Pro CGI.

The thing we like most about Russell's **FileMaker.acgi** is that it doesn't require editing of the CGI itself to get it to work with a specific database. Instead, all the database-specific information such as field names is done on the HTML forms that come with his CGI package. You can also use **FileMaker.acgi** to serve up multiple databases. Keep in

mind that no AppleScript CGI will truly give high performance with FileMaker Pro. If you have a large database you'll run into problems with speed. You can probably bump performance up if you use a CGI written in C or Perl but we don't know of any to pass along to you.

Russell's instructions that come with **FileMaker.acgi** are excellent — we had it up and running in less than 5 minutes!

When a user comes to insert a record to your FileMaker Pro database, they'll see a page like that shown in Figure 5.8.

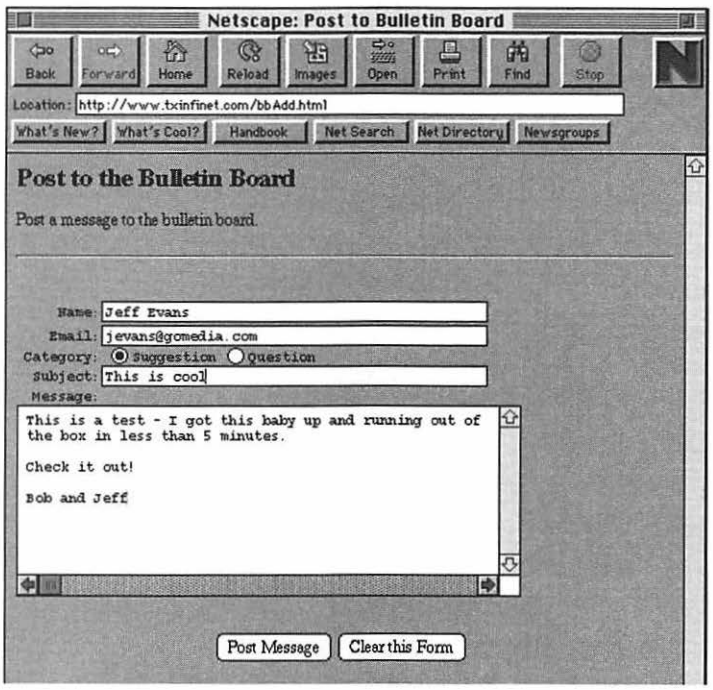


Figure 5.8: FileMaker.acgi Example — Adding a New Record to a FileMaker Pro Database from a Web Page.

In this example, when the user clicks on "Post Message," the **FileMaker.acgi** gets called, the CGI takes its cue and opens the FileMaker Pro database and a new record gets added.

We had FileMaker Pro running on the same machine as our WWW server. Figure 5.9 shows what it looks like when the example FileMaker Pro database that comes with Russell's CGI is launched and added as well.

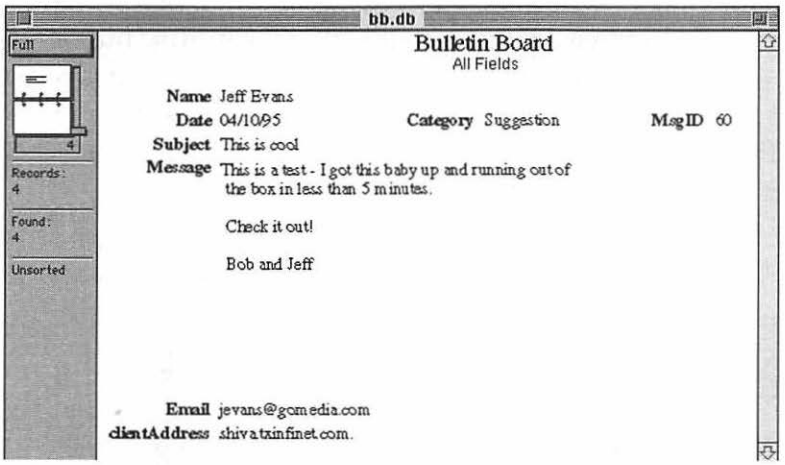


Figure 5.9: FileMaker.acgi Example — View of FileMaker Database Showing New Record Addition.

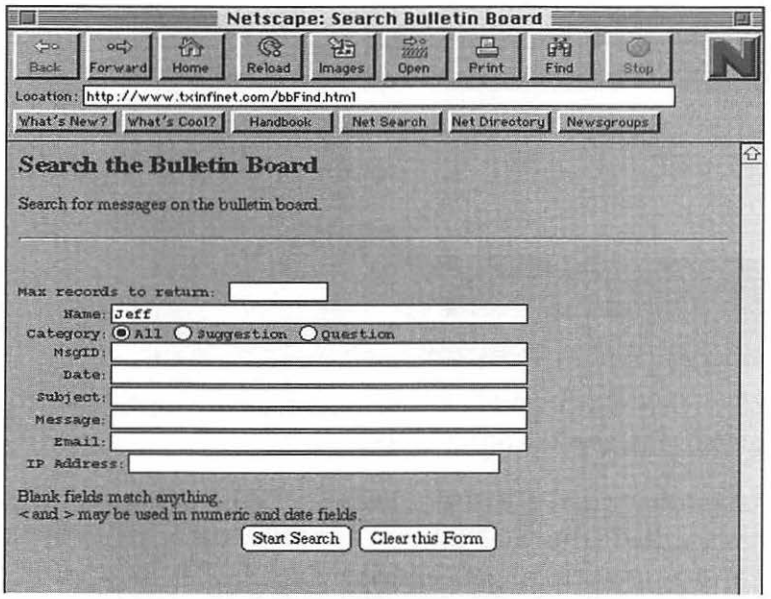


Figure 5.10: FileMaker.acgi Example — Searching a FileMaker Pro Database from a Web Page.

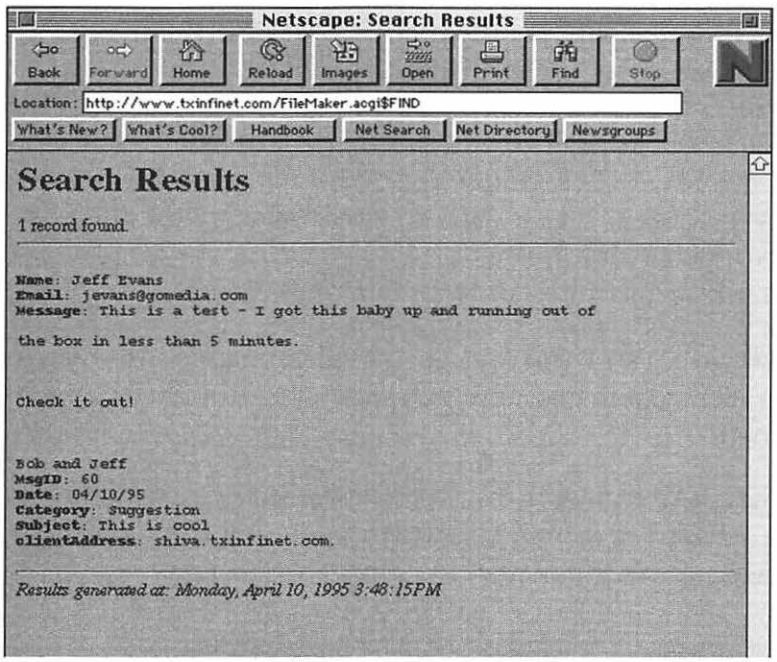


Figure 5.11: FileMaker.acgi Example — Searching Results from FileMaker Pro Database Displayed on a Web Page.

Using FileMaker Pro as a back-end database to your WWW site is a great way to get started using CGIs effectively. It's quick to set up, it's reliable, and it's relatively inexpensive. More than one person we talked to said that there are some limitations to the FileMaker Pro AppleEvent suite that make coding your own CGI challenging. We don't have first-hand experience to confirm that, but we trust the tip.

If you have a large database to work with and speed and performance become an issue, you probably need to consider using something other than FileMaker Pro. Which brings us to Butler SQL.

BUTLER SQL

Butler SQL is more expensive than FileMaker Pro. It is also faster and more powerful. The big surprise to us was that, even though it's a full-featured, full-blown database environment, it is remarkably easy to set up as a back-end to MacHTTP.

Everyware Development has put together a fantastic package called **ButlerLink/Web** that, in conjunction with MacHTTP, lets you easily create a World Wide Web interface to Butler SQL databases (Figure 5.12). It provides a full range of database functionality. Depending on how you wish to configure access, you can allow users to query, update, insert, and delete records. No programming, scripting, or HTML knowledge is required. Everything is set up with an easy-to-use, point and click interface. That worked great for us!

ButlerLink/Web consists of two main applications:

1. **ButlerLink/Web Form Maker** — used to create forms for accessing Butler SQL databases and to specify the format of the records returned.

2. **ButlerLinkWeb.acgi** — the application that translates queries sent by the Web client via MacHTTP, queries the Butler SQL server, retrieves results, and passes them back to the client via MacHTTP.

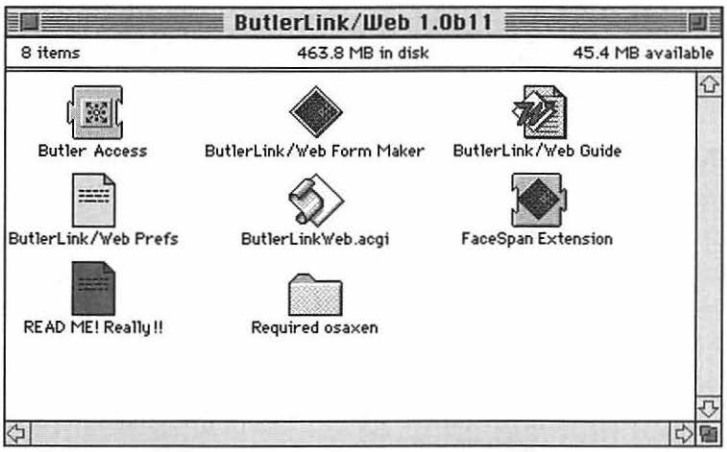


Figure 5.12: Contents of ButlerLink/Web Package.

To query a database, you just specify search criteria via a form created with the Form Maker (Figure 5.13). When the user submits the search, ButlerLink/Web queries Butler SQL and returns a list of matches (the "hit list") (Figure 5.14). Each record in the list is linked (if desired) to the full detail for the record. To see the detail, the user clicks on an item in the hit list (Figure 5.15). The detail form allows viewing only, updating of all or some fields, or the ability to delete the record.

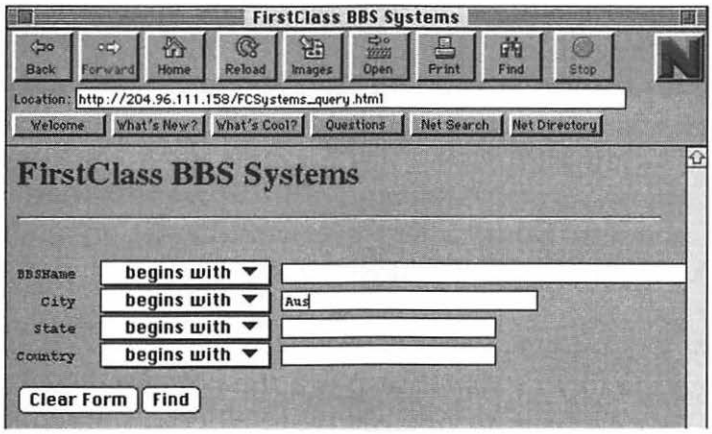


Figure 5.13: ButlerLink/Web Example — Searching a Butler SQL from a Web Page.

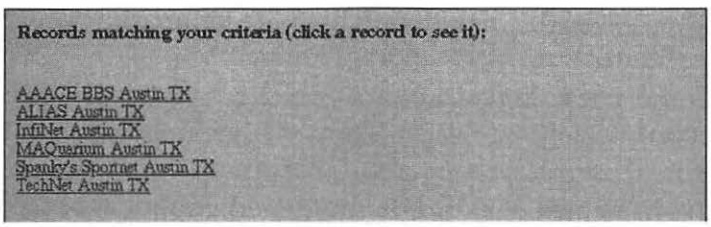


Figure 5.14: ButlerLink/Web Example — The Butler SQL Hit List on a Web Page.

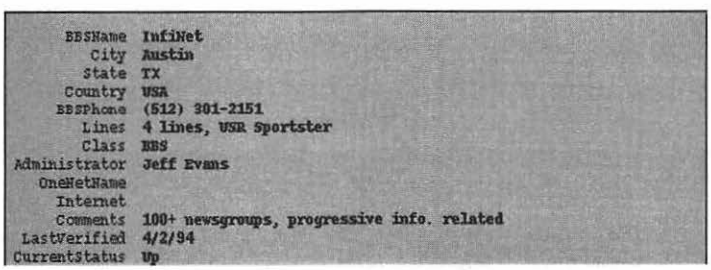


Figure 5.15: ButlerLink/Web Example — Record Retrieved from Butler SQL Displayed on a Web Page.

All of the ButlerLink/Web forms used to build your HTML pages (the query form, hit list, and detail) let you specify your own headers and footers for adding titles, graphics, links to other pages on your server, etc. The hit list form lets you put any text or HTML code before and after each record and between each field in each record.

You do have to jump through a few hoops to get your Butler SQL server set up properly, but it's easy enough if you read the Butler SQL manual. Ideally, you want your Butler server on Ethernet on a different Mac from your MacHTTP server. The docs for setting up ButlerLink/Web are a breeze.

Once you've set up the Butler SQL server the whole process of connecting your Web site to your database(s) is just a point and click affair. You don't even have to get into AppleScript at all. You simply use the Butler Form Maker application to create a query or insert a form that accesses a particular Butler SQL database table. The form gets saved as a text file and is then placed in your MacHTTP server's hierarchy so clients can access it. Only the initial query or insert form exists as a file on your server; all the other pages (hit list, detail, etc.) are generated "on the fly" and returned by the **ButlerLink/Web.acgi** application.

You launch the Form Maker application on your MacHTTP server and connect to your Butler SQL server. Then you create a form and specify parameters for a user search of a database table. You choose **New Query Form** from the **Form Maker File** menu and you're presented with the database(s) you have set up on your Butler server (Figure 5.16).

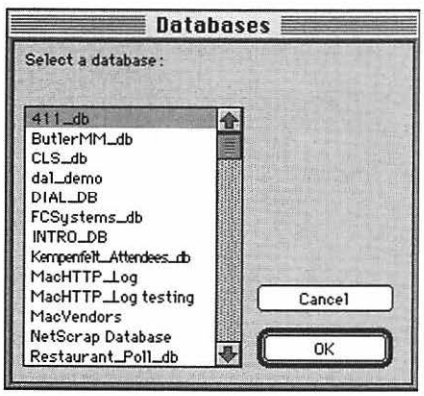


Figure 5.16: ButlerLink/Web Package — Connecting MacHTTP to Your SQL Database.

You simply work your way through the prompts that ButlerLink/Web gives you to create an HTML Query Form. Figure 5.17 shows its interface for designing the form; it connects all the pieces and spews out your HTML automatically.

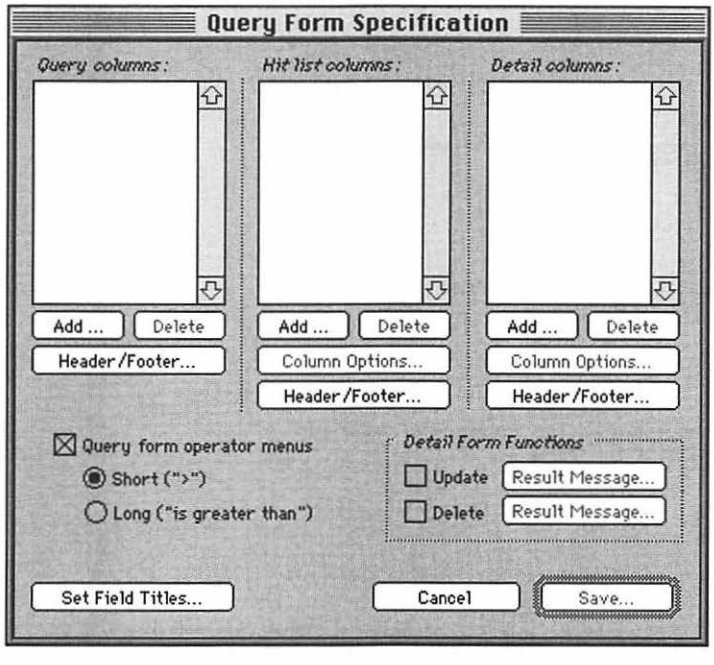


Figure 5.17: ButlerLink/Web Package — Designing the Query Form.

It's awesome. Aren't you glad ButlerLink/Web does it all for you so you don't have to type all the HTML? Figure 5.18 shows you just what you're missing!

FCSystems_query.html

Last Saved: 3/28/95 at 6:31:11PM
Shiva:Desktop Folder:WIP:FCSystems_query.html

```
<!-- This HTML document was generated by ButlerLink/Web Form Maker v1.0b3 -->
<!-- ©1995 EveryWare Development Corp. -->
<!-- e-mail: info@everyware.com    phone: 905-819-1173 -->
<TITLE>FirstClass BBS Systems</TITLE>
<H1>FirstClass BBS Systems</H1>
<FORM METHOD="POST" ACTION="/Scripts/ButlerLinkWeb.acgi$query"><HR>
<PRE>
BBSName   <SELECT NAME="@op|1" SIZE=1>
<OPTION>begins with
<OPTION>=
<OPTION>!=
<OPTION>&gt;
<OPTION>&lt;
<OPTION>&gt;=
<OPTION>&lt;=
<OPTION>contains
</SELECT> <INPUT NAME="@fd|1|BBSName" TYPE=TEXT MAXLENGTH = 50 SIZE = 50>
   City   <SELECT NAME="@op|2" SIZE=1>
<OPTION>begins with
<OPTION>=
<OPTION>!=
<OPTION>&gt;
<OPTION>&lt;
<OPTION>&gt;=
<OPTION>&lt;=
<OPTION>contains
</SELECT> <INPUT NAME="@fd|2|City" TYPE=TEXT MAXLENGTH = 30 SIZE = 30>
  State   <SELECT NAME="@op|3" SIZE=1>
<OPTION>begins with
<OPTION>=
<OPTION>!=
<OPTION>&gt;
<OPTION>&lt;
<OPTION>&gt;=
<OPTION>&lt;=
<OPTION>contains
</SELECT> <INPUT NAME="@fd|3|State" TYPE=TEXT MAXLENGTH = 25 SIZE = 25>
Country   <SELECT NAME="@op|4" SIZE=1>
<OPTION>begins with
<OPTION>=
<OPTION>!=
<OPTION>&gt;
<OPTION>&lt;
<OPTION>&gt;=
<OPTION>&lt;=
<OPTION>contains
</SELECT> <INPUT NAME="@fd|4|Country" TYPE=TEXT MAXLENGTH = 25 SIZE = 25>
</PRE>
<INPUT NAME="@sel" TYPE=hidden VALUE = "BBSName,City,State">
<INPUT NAME="@ord" TYPE=hidden VALUE = "BBSName">
<INPUT NAME="@det" TYPE=hidden VALUE = "BBSName,City,State,Country,BBSPhone,Lines,Class,Adminis
```

Figure 5.18: ButlerLink/Web Example — HTML Automatically Created with ButlerLink/Web Form Maker.

We're not going to steal Everyware's thunder and walk you through the entire setup. ButlerLink/Web is included on the CD-ROM that came with this book. So is a trial copy of Butler SQL with an example database. Take a look and give it a try for yourself. The documentation they put together for it is outstanding.

Everyware has made a big commitment to supporting MacHTTP with back-end solutions. In fact, they were probably the most exciting vendor at the first WebEdge conference. With a copy of the Butler SQL and the **ButlerLink/Web.acgi** package, it's easy to build the HTML pages that hook right into a Butler SQL database. Best of all, you don't have to write a single line of HTML to set it all up.

One of the most exciting things to happen at the first WebEdge conference was when Everyware's Mark Wickens and Henry Lach got together with Chuck Shotton. Together they built a CGI that traps the log info coming off of MacHTTP as an AppleEvent and throws it right into a Butler SQL database. The work they did — especially Mark's — gives you unbelievable access to data. With it you can graphically present server statistics or export the data into an accounting package for billing purposes. Think about it.

Keep an eye on the stuff that Everyware is up to with MacHTTP. We are confident that there will be more developments to come.

APPLESEARCH

There's one more CGI that you should know about and that's **AppleWebSearch.acgi**. As of this writing the version number is 1.5.

For those of you who don't know about AppleSearch yet, you need to. AppleSearch is a full-text and retrieval software that allows you to find information with keyword and boolean searches. **AppleWebSearch.acgi** lets you do this via the Web.

Before you go out and buy AppleSearch you need to determine that you have documents you want users to be able to search and retrieve from. The entire MacHTTP mailing list is up on AppleSearch at the WebEdge WWW server at **http://www.webedge.com**. Check it out.

At about $1500 — depending on where you get it — AppleSearch is up there in price. Make sure you need it or that you're rich and like to tinker. Keep your eyes open for WWW sites that use it to get some ideas on how it might be useful for you. We'll try to point you to ones we find from the Webmaster Mac WWW server. AppleSearch is well worth the price if it's what you're looking for. As of this writing **AppleWebSearch.acgi** is free (such a deal). It's also included with the CD-ROM that came with this book.

AppleWebSearch.acgi is another CGI that works under the covers and requires no programming or scripting skill unless you want to use ResEdit (and consider that programming) to modify the interface of the Web pages that get created by the CGI. But you don't ever need to look at the inside of the CGI itself to set it up and get it working. You do need

to read the AppleSearch manual and learn how to set an AppleSearch server up, however. Apple did a good job with the documentation on this, so it's no big deal when the time comes.

Figures 5.19, 5.20, and 5.21 show examples of what the Search, Hit List, and Document pages look like on the Web when you use **AppleWebSearch.acgi**.

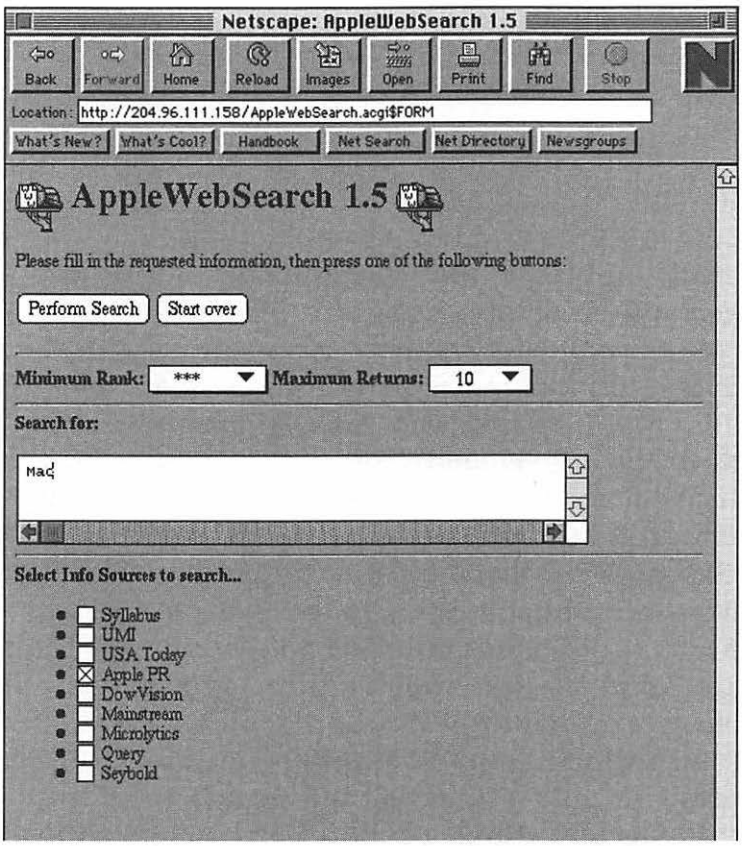


Figure 5.19: AppleWebSearch Example — The Search Page.

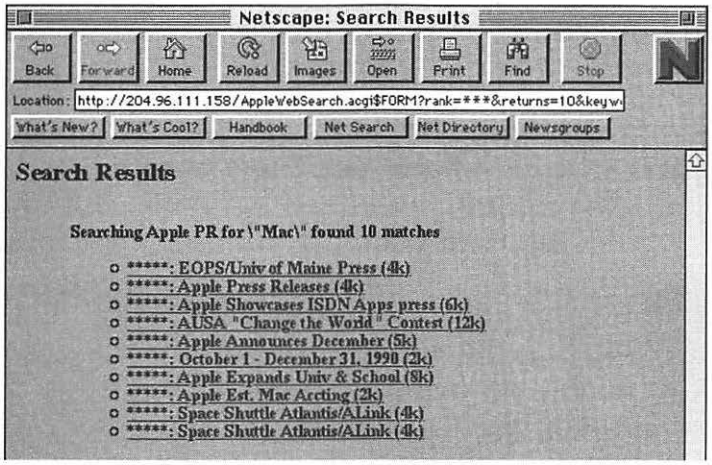


Figure 5.20: AppleWebSearch Example — The Hit List Page.

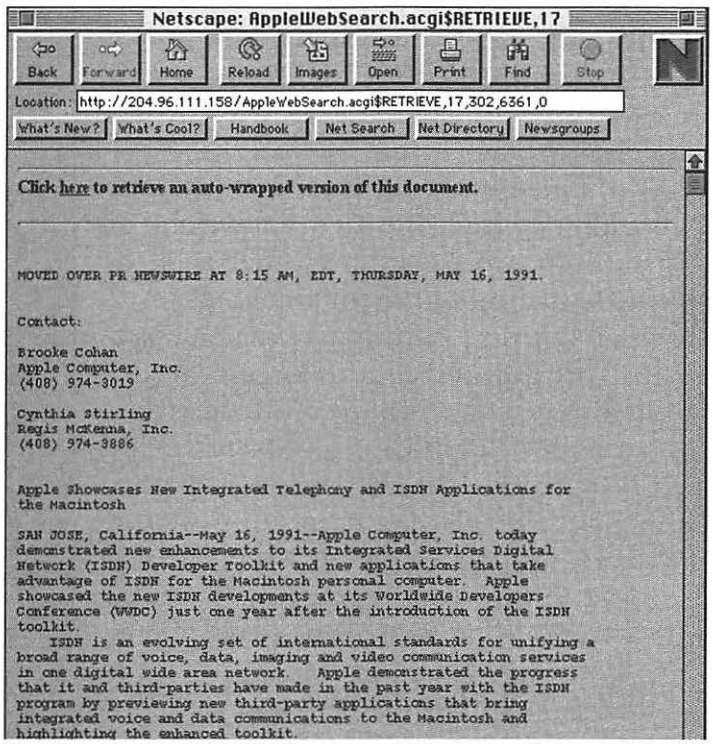


Figure 5.21: AppleWebSearch Example — The Results Page.

Before we let you move on, we'd like to pause for a moment of reflection from our friend Rob Hafernik, on the occasion of his first CGI...

My First Mac CGI, or, A Fortunate Experience
by
Rob Hafernik

Soon after being exposed to the World Wide Web, many people develop a fever to put up their own pages. They learn HTML, get access to a Web server, form a link collection, and start taking digital pictures of their pets. For many, this is as far as the fever goes.

For some, however, this fever can spread into a new set of symptoms. These people are not satisfied with merely publishing some pages. They want to set up their own server, publish the pages of others and, in severe cases, they want to write their own CGIs and be free of the tyranny of static content. For better or worse, I suffer from this affliction.

For my first CGI, I wanted to do something simple, but useful. It couldn't just be the CGI equivalent of "Hello World," it had to *do* something. After a bit of thought, I came up with a new form for an old idea.

In the UNIX world there is a traditional program, usually called "cookie" or "fortune" that users can execute to print a random fortune cookie saying on their screen. While working around UNIX boxes, I collected a file with thousands of pithy sayings, UNIX jokes, quotes, and various wordplay. This file had been lying around on my Mac for years, waiting for me to "Get a Round Tuit" and write a Mac fortune cookie program. Instead, I decided to write a World Wide Web fortune cookie program.

The task sounded simple: Write a CGI that picked a random saying from the cookie file and formatted it into a page of HTML. Lucky for me, it actually turned out to be just about as easy as it sounded.

WebSTAR™ (a.k.a. MacHTTP) uses AppleEvents to send information to CGIs. The structure of these events and the information they contain is documented in the server manual. The CGIs in turn reply with an AppleEvent that has a page of HTML text in the event's direct parameter (AppleEvents can get pretty complicated — refer to the AppleEvent chapter of *Inside Macintosh* if this isn't making sense). The Fortune CGI would be simpler than most, because it would not have to parse any incoming information from the server, only return a randomly generated page.

I already had a shell of a Mac application written that did nothing but put up a menu bar, respond to mouse clicks, display an About Box, and take the required AppleEvents (every programmer should keep one of these around). Starting from this shell I added a new function, **HandleWebStarEvent**, to handle the special class of events generated by WebSTAR and put in the code to register it with the Mac OS.

The **HandleWebStarEvent** routine just calls a **BuildReply** function to generate a handle to the HTML reply, then stuffs that information into the direct parameter of the reply AppleEvent and returns it.

The **BuildReply** function is a little more interesting. Since I wanted this CGI to work for anyone that wanted to use it, I had to provide a mechanism for third parties to add their own information before and after the fortune cookie text. This way, a third party could put a fortune on a page that had links to other pages and so on. The **BuildReply** function does this by concatenating three chunks of HTML together. From the resource fork of the Fortune CGI, it starts with a text resource (number 1000) that contains the top part of the page. To this it adds the randomly chosen fortune. Last, it adds a footer, taken from text resource 2000. Anyone using Fortune can use ResEdit to make the header and footer

text into anything they want (adding a link back to their home page, for example). Pulling the text from resources is a little faster than opening a file and reading it and it makes for less clutter in the folder. (The fact that it's also easier to program this way never entered consideration, I swear.)

The **BuildReply** function in turn calls **GetFortune** for a handle to a random cookie file entry. **GetFortune** is simple and uninteresting, just a hack to pick a random line from a file. (You could substitute your own file, containing any lines of text you like, if you're careful.)

There were a couple of additional details. Even though the Fortune application is small and fast, I wanted it to quit if it wasn't being used. I added a simple timer that keeps track of the ticks since the last request was handled and quits the app if no requests come in for three minutes. I also wanted a counter that would keep track of the number of fortunes the Fortune CGI had distributed, so I added a counter (also kept in the resource file) and a little code to display the counter when drawing the About Box.

That's it! Weighing in at a mere 350 lines, the Fortune program makes a small executable file (28K, with 22K of that coming from the About Box pict!) and runs in 64K of RAM (but give it 100K to be safe). It runs quickly and can format its result any way you like.

Of course, it won't end here. I've got the fever and it's not something you recover from easily. Now I have to write more CGIs that do more interesting things. Maybe something with a database...

Do you suppose there's a support group for this sort of thing?

YOU'RE DONE!

Congratulations! You're done with the CGI chapter! That wasn't so painful, was it?

We recommend you play around with CGIs that work right out of the box before attempting to write your own. Besides, someone else may have already done it and you can just work off of theirs. When you're ready or need to write your own CGIs, go bury your head in Jon Wiederspan's CGI tutorials and go for it.

By the way, we've included a couple of fun CGIs for you to play with. They're on the CD-ROM too. We'll keep posting them on the WebMaster Mac WWW server as we find 'em — even the dumb ones that have absolutely no purpose or redeeming value whatsoever. You know, the fun ones.

What's left? How about a great series of Q & A with some WebMaster luminaries? Turn the page, you're in for a real treat.

Chapter 6

Interviews with the Gods

The Best and Brightest WebMasters Speak Their Minds

Not so long ago Bob wrote a book entitled *Dr. Macintosh, Second Edition: How to Become a Macintosh Power User.* The final chapter was entitled *What Other Power Users Think You Should Know...* and it consisted of tips and techniques contributed by power users all over the globe. That chapter proved to be among the most popular in the book so we've ripped off the idea for *WebMaster Mac.*

On the following pages you'll hear from the best of the best: MacHTTP creator Chuck Shotton; Web66 cookbook creator and MacHTTP guru extraordinaire Stephen Collins; the king of CGI tutorials, Jon Wiederspan; and Apple's WebMaster supreme, Carl de Cordova.

News Flash: Just as we were going to press, Star Nine Technologies announced that they would be publishing a commercial version of MacHTTP, called WebSTAR, which eagle-eyed readers will remember is what Rob Hafernik called it a few pages ago. So we began our discussion with Chuck by asking about this new development.

Chuck Shotton (BIAP Systems)

Jeff & Bob: MacHTTP is now called WebSTAR. Tell us a little about your plans for WebSTAR with Star Nine Technologies.

Chuck Shotton: Global domination, a WebSTAR in every pot! er, server! Actually, our intent is to make

Web publishing available to the masses, not just the UNIX jockeys and MIS propeller-heads of the world. This means teachers, bankers, lawyers, retailers, hobbyists, secretaries, and anyone else that knows what a mouse is for.

Jeff & Bob: How many downloads of MacHTTP have there been to date? Any idea on the number of Mac WWW servers out there? How does that compare to other types of CPUs serving up Web?

Chuck Shotton: I've got no way to guess how many downloads there have been worldwide, because it is on every Info-Mac and UMich FTP mirror, plus a bunch of online services, CD-ROMs, and bulletin boards. If I had to guess, my guess would be that the software has been downloaded way over 100,000 times.

Some slightly stale stats showed that at the beginning of 1995, MacHTTP was the second most used Web server software, immediately behind NCSA and CERN httpd software. At this point, I'd guess that there are between 5000 and 8000 permanent MacHTTP sites on the Internet or on internal LANs today with about double that number in transient sites.

Jeff & Bob: What's on your wish list for future versions and plans for MacHTTP? Will you continue to offer a shareware version of MacHTTP as well as commercial versions?

Chuck Shotton: The big plans are support for Open Transport, secure transactions, an OpenDoc version, and then some special-purpose servers that work with

intelligent agents. All indications are that WebSTAR will pick up where MacHTTP left off, with a network-distributed version as well as the usual shrink-wrapped stuff. That means that people who want to get their hands on a basic Web server will still be able to.

Jeff & Bob: Tell us a little about First Virtual. What's it all about? How does WebSTAR fit in?

Chuck Shotton: First Virtual is electronic commerce for the rest of us. It doesn't need encryption. It doesn't even need the Web. If you can send eMail, you can buy stuff with First Virtual, have it charged to your credit card, and be completely protected against fraud. The neat thing about First Virtual is that it allows anyone with a checking account to open up shop on the Internet. WebSTAR has tightly coupled support for electronic commerce with First Virtual, meaning that all a seller has to do is write a few simple HTML pages to advertise and sell stuff on the Net.

Jeff & Bob: Do you ever find time to just surf around on the Net or are you always busy working on WebSTAR and responding to the MacHTTP mailing list?

Chuck Shotton: I don't get nearly as much time to surf as I'd like. This is going to become more necessary as the next generation of Web tools emerges with intelligent agents and smart servers. I expect I'll have a little more time to do this once business on the Web settles down a bit.

Jeff & Bob: Speaking of which — you must be really pleased with the MacHTTP mailing list. I remember in the early days you were the only one answering all

the questions. How many subscribers to the list are there now?

Chuck Shotton: There are close to 1000 subscribers on the list now. The list has had some growing pains, sort of like a reptile shedding its skin every so often. It's about time to find a bigger rock to scratch up against, too. The list has outgrown its current host again. This is such a phenomenal resource, it's hard to keep up with it. I think parts of it will spin off into an interactive Web page sometime soon.

Jeff & Bob: What are some of the innovative things you've seen people doing with MacHTTP and CGIs?

Chuck Shotton: I think the *real* potential of the CGI interface built into MacHTTP is just starting to be discovered. It is so much easier to hook a Mac server to a third-party application than on, say, a UNIX host. People have been a little bit intimidated by the learning curve, but we're starting to see a *lot* of development of serious CGI applications like easy-to-use database interfaces, online conferencing, real-time video capture, interactive entertainment, etc. The Mac is definitely the platform for this stuff to come out on first.

Jeff & Bob: What are your thoughts about Netscape? How do you feel about the extensions to HTML that Netscape Communications is pushing the envelope with?

Chuck Shotton: I think Netscape is driven a bit too much by their marketing department and management and less by good technical judgment some-

times. They've had some neat ideas, and they've had some really scary ideas. They are relying on the market to drive their direction and I think there are times when the market doesn't know what's good for it. On the whole, I think what Netscape represents is the future of how development will be done for the Internet. The days of university-based developers leading the way are over.

Jeff & Bob: How about the whole client-server HTTP architecture and HTML itself? What would you like to see unfold as the way data is handled?

Chuck Shotton: I'd like to see less dependence on UNIX in the standards. Too many compromises have been made to support the inadequacies of that OS and it is holding back development. Life is no longer a text stream. When people finally realize that, I think we'll see some cool developments. Things like active data, intelligent agents, cooperating servers, etc., will start to pop up when people realize that the Web should be a peer-to-peer architecture instead of a client-server architecture.

Chuck Shotton
cshotton@biap.com
http://www.biap.com/ cshotton@oac.hsc.uth.tmc.edu
"I am NOT here."

STEPHEN COLLINS (UNIVERSITY OF MINNESOTA)

Jeff & Bob: Your Web66 cookbook was how many of us got started with MacHTTP. What kind of traffic has your site been getting?

Stephen Collins: Web66 is getting about 7000 connections per day now, but it is increasing about 10% per month. Web66 is actually an extension of our Hillside Elementary project, in which we had sixth-grade students set up their own WWW server (MacHTTP is so easy sixth-graders can do it!). So many people asked how we did it that I decided to create Web66 to explain all the details, and also to link all those K–12 WWW sites to each other and to other resources on the net.

In raw numbers, between the two servers, we're now getting about 10,000 to 12,000 connections per day. But we're also getting visitors and publications almost every day now to our project.

Jeff & Bob: Tell us a little about what you're up to these days. Are you still pulling together statistics and doing those great charts on Internet traffic?

Stephen Collins: For some time, I charted the statistics on Internet traffic that were reported by the NSF on their backbone. These statistics provided concrete information about the tremendous increase in Internet use, and the skyrocketing use of the World Wide Web. Unfortunately, they've started dismantling the NSF backbone, and there is no longer any source of hard data on Internet use. While it's clear that Inter-

Net and Web use continues to grow at phenomenal rates, we can't prove it anymore!

Jeff & Bob: How have schools been using MacHTTP? What are some examples we should be sure to check out?

Stephen Collins: When we started the Hillside server in early 1994, there were only three other K–12 schools with a presence on the Web. When we started the Hillside server, we started a list of the schools with Web sites. In little more than a year, that list has grown to somewhat over 500 schools. About 30% of those schools are using MacHTTP.

Of course, I must recommend our own server at Hillside. There are so many good sites today, that I hesitate to recommend any — but don't miss Buckman school in Oregon and Patch American High School in Germany.

Jeff & Bob: What are some of your favorite sites on the Net?

Stephen Collins: I've had great fun with the Lion King movie clips from the Walt Disney home page. NASA is full of interesting sites, and they always have something new going up. It seems there is a great new site going up almost every day, and I keep a list of my daily favorites on the Web66 What's New page.

Jeff & Bob: So where are we headed? You're the visionary type.

Stephen Collins: In the not very distant future, I predict we will no longer have WWW servers or browsers.

The basic file system on every computer will be navigated with URL-type file specifications. Every program that reads information from files (word processors, drawing programs, spreadsheets, database, whatever!) will specify a URL to get at every single file.

We will no longer need separate browser programs, because every program will be able to obtain information from the Web with a URL.

In fact, programs will no longer be aware of where the data actually resides. It could be on your own hard disk or on a server on the other side of the world. All the program will know is the URL necessary to get the data.

This means there will no longer be a distinction between your computer and the Net. That physical and logical barrier will be erased, and all our computers will become a huge integrated distributed network, where information can and will be readily shared.

This change is really nothing more than a simple change to the operating system file structure. At the same time, programs will be changed to handle data as it arrives (much like UNIX pipes). It will no longer be necessary to store large files on your computer before you can use them, since you can just process the data as it arrives.

What next? Real-time information handling! For example, current network speeds are almost fast enough

for real-time video. If it takes two hours to download a movie, that's all the faster we need it if we can watch the movie as it downloads.

This change is going to be easier and come faster than anyone might predict.

Stephen E. Collins
sec@boombox.micro.umn.edu
http://www.micro.umn.edu/staff/sec

JON WIEDERSPAN (UNIVERSITY OF WASHINGTON)

Jeff & Bob: Jon, your tutorials on CGI were a tremendous contribution to the success of MacHTTP. What are you working on these days?

Jon Wiederspan: The tutorials need constant revision to keep up with changes in both WWW clients and servers. I am also working on several new examples to apply the lessons learned. I hope to have some CGIs released after MacHTTP 3.0 comes out that will do things people did not expect to be able to do.

Jeff & Bob: You're familiar with both — how does running a Web server on the Mac compare to UNIX? Besides it being easier, tell us a little about some of the performance issues that come up. The word is that you've been playing around with using an array of MacHTTP servers. Tell us more.

Jon Wiederspan: I am part of a sizable and growing pool of network managers who are only moderately familiar with UNIX, forced into it in order to provide network services like FTP, NNTP, DNS, and HTTP.

There is no comparison between the hours I spent with no real success in setting up httpd on our UNIX workstation and the minutes it took to get MacHTTP running perfectly. Performance was a big issue when I started (MacHTTP 1.3), but it is not really an issue now. Using the latest version (2.02 or later), a Macintosh computer can outperform any equivalently priced UNIX machine. That is ignoring the higher administration costs for a UNIX machine. There is also the security issue; a MacHTTP server is extremely secure, whereas UNIX servers take a lot of knowledge and effort to be very secure.

I have theorized that high performance could be gained by combining multiple Macintosh computers together in a virtual server, but I have yet to receive any gift in the mail of five or more Macs to try this out on.

Jeff & Bob: How about you? Are you doing any surfing just for fun? What are some of your favorite sites?

Jon Wiederspan: I don't get to do much surfing for fun, mostly for research. When I'm out for fun, I'm usually looking at movie sites like the Cardiff Movie database or new release movie sites (e.g., MCA/Paramount).

Jeff & Bob: What excites you most about the things you're seeing people do with MacHTTP?

Jon Wiederspan: Actually, the most exciting thing I'm seeing is with the new CGI solutions coming out. More and more people are creating generic solutions

(like MapServe and FileMaker.cgi) that can be used by nonprogrammers to get speedy responses out of their servers. I anticipate much more of this as MacHTTP 3.0 extends the CGI interface even further.

Jon Wiederspan
jonwd@tjp.washington.edu
http://www.uwtc.washington.edu/Computing/WWW/Macintosh.html

CARL DE CORDOVA (APPLE COMPUTER)

Jeff & Bob: So where are we headed? You're the visionary type. What are you going to have us banging around on next?

Carl de Cordova: There seem to be four major movements right now in the Macintosh Web community.

1. The Macintosh is becoming the platform of choice for small- to medium-size Web sites. It's inexpensive, easy to use, and integrates with popular programs. More people will be able to have their own personal servers and be in complete control of the content.
2. Unique tools are beginning to emerge. The Macintosh is the best multimedia and publishing platform on the planet. Tools such as QuickTime VR will be used in unique ways to create sites that are just not possible on any other platform.
3. Commerce tools are coming online.
4. Macintosh developers are starting to get the picture, particularly the database vendors.

Most sites on the net are getting less than 10,000 hits per day and they almost never get more than 1000 hits per hour. The Mac is the perfect platform for this kind of medium-volume site.

Right now we are seeing a lot of consolidation going on. The tools to run a medium-size Web server on the Mac are just not that old. The low cost and ease of use offered by the Macintosh platform will enable thousands of people to set up sites on the Net. The percentage of sites on the Net that are Macintoshes should increase rapidly over the next year or so.

In the short term we are going to be working on exploiting the commercial potential of the Net. While I think it will be some time before selling big ticket items on the net is a common event, 1996 may be remembered as the year when the micro transaction became a major source of income for thousands of WebMasters.

Jeff & Bob: What's still needed for us WebMasters? What type of tools and apps should developers be working on?

Carl de Cordova: Developers need to be concentrating on tools that take advantage of the Macintosh platform ease of use.

Someone is going to make a fortune marketing the first "Web page layout" program that integrates the functions of today's multitude of small tools into one monolithic program. This "PageMaker" for the Web will keep all of your pages together in one document and respond directly to requests from MacHTTP for individual Web pages. It will also contain tools for mak-

ing forms, tables, clickable maps, simple database queries and so forth. It will give you the ability to quickly make changes across multiple pages and to create common items for each page.

The big news is that all forms of interaction between MacHTTP and third-party programs should get faster and easier to use. The major developers are waking up to the fact that the WebMasters are a really important segment of the market. Almost all of the major vendors have some kind of tools in the works for Web development.

In the near future, many programs will have a menu item that says "save as HTML" that will take care of translating from their own file formats into marked-up text ready to add right into your server. Word processors, desktop publishing, forms, graphing, database, and spreadsheet programs will all have this feature to some extent. The first generation of these tools is just starting to appear as add-on tools and small conversion programs.

By this time next year, most programs will have the ability to pass data directly to and from MacHTTP. Many programs will be "MacHTTP aware" and will respond directly to requests from Web servers. WebMasters who want to take advantage of existing data should have a host of tools at their disposal for querying existing databases and finding chunks of information, without having to write custom AppleScripts by hand.

Of course there will also be a number of tools that will be developed specifically for use on the Web.

Jeff & Bob: What do you think the future for commercial transactions on the Net looks like?

Carl de Cordova: The cost of making a simple transaction has been coming down for a long time. It used to be that you could not afford to sell anything for less than $10 or $20 because that is how much it would cost to process the transaction and cover any bad debt or collection problems.

In just the last few months we have seen a number of new banking initiatives that promise to lower the cost of doing business on the Net. First Virtual's payment system costs about $0.31 per transaction. The digicash, ecash, and Visa/Microsoft systems should lower this cost even more.

A whole new type of product becomes viable when the cost of a transaction falls close to $0. Right now I will pay 25¢ for a newspaper to get information about the movies playing near me, find out how my sports team did last night, or chuckle at Dilbert.

I think it will become routine for your computer to regularly ask you if you are willing to pay a nickel or a dime to access a well-organized server that regularly has the information you find useful. Once the transaction systems are in place, providing obscure information for a very small fee may become a viable business.

A lot of the frantic activity we are seeing today is just a number of large companies staking out their place in cyberspace. But I think the most exciting opportunities are for the small, independent WebMaster who

has a highly specialized and well-maintained database of information about a specific subject.

For example, I have a friend who is really into tropical fish. He is one of the world's experts on cyclids. He maintains a USENET news group and an FAQ as well as a couple of HyperCard tools that present information that is essential to people who raise tropical fish. Using the Web and micro transaction software he can charge just a few cents to each fish enthusiast that finds his information useful. While this may not let him quit his day job, it may just pay for a significant portion of his fishy hobby.

It will be fun to watch the competition between various sites as they try to outdo each other to provide the best information to their customers.

Jeff & Bob: So what's the limit on a Mac? How much traffic can it really take without choking? How will Open Transport change things?

Carl de Cordova: Right now the biggest limitation to the performance of the Mac as a Web server is MacTCP. MacTCP was designed a long time ago as a developer's tool and was never really intended as a platform for high-performance Power Mac servers running on fast network connections.

Currently MacTCP only allows 64 "listeners" at the same time. This means that in the best case your Mac can only handle 64 simultaneous connections. Because there are other tasks that MacTCP must perform, only about 35 to 40 of those connections can be used

by MacHTTP. Any additional connections get a busy message. This number goes down even further if you want to run other TCP/IP programs such as FTPd, Cron, Timbuktu, or Netscape. When the Mac is being used primarily as a client, that was really not a problem, but it is a serious limitation for a busy server.

We hope to fix all these problems with the Open Transport version of MacTCP. Open Transport is PowerPC native, using a more modern streams-based protocol. The number of listeners is entirely RAM based so the number of simultaneous connections should be an order of magnitude higher.

Open Transport will also allow us to do some things that just are not possible with the current MacTCP. The most exciting of these for the WebMaster will be multi-homing, or the ability to use two EtherNet cards for a single connection.

Open Transport will also be easier to configure, more robust, and provide better support for new technologies. The interface has been completely redesigned and will be much more straightforward. You will not have to reboot after making changes.

Jeff & Bob: Apple Business Systems has a new Net server configuration coming out on their workservers. What's a good reason to go that route instead of getting a Power Mac 6100? We know that MacHTTP runs native on a Power Mac and uses the Thread Manager, so we understand why the Power Mac as opposed to a Mac II or something. But why a workserver?

Carl de Cordova: The idea behind the Internet bundles is to provide a single source that provides all the tools for building a Web server in one place.

The workgroup servers make an ideal platform for Web server development because they bring together all the pieces needed to create a compete server.

Jeff & Bob: What else needs to be done to round out the Mac as a full-fledged Internet server? What's in the works that you can tell us about that developers outside Apple are working on besides DNS? And what is DNS and what's the big deal about it?

Carl de Cordova: We are still missing a few tools, such as a software TCP/IP router; MacDNS should be out soon...

Jeff & Bob: We know from your home page that you stay right on top of new and interesting sites on the net. What are some of your favorites?

Carl de Cordova: I also have a weakness for cute hacks.... Simple, cool ideas that are well implemented. I think the most exciting thing about MacHTTP is that it lowers the cost and complexity of setting up a Web server so that people can spend their time coming up with cool Web tricks. As more and more people find out about using their Macs as Web servers we will see some really wonderful and creative pages come on line.

A number of the pages I have really liked are no longer with us... They were short-term projects that ran their course or served their purpose.

Although it is not around anymore, I liked the Internet candy dish that Notre Dame did for Valentines where the server randomly chose five GIFs of little candied hearts every time you accessed it.

I really enjoyed watching the Texas A & M bonfire on line. Every year, when Texas plays Texas A & M, the Aggies build a big bonfire. They have been doing this for about 70 years. This year they set up a camera that took snapshots of the bonfire every 15 minutes as they were building it. When they finally lit it, they took a picture every minute.

We just happened to be having a meeting of the Austin CapMac User Group Internet SIG the night of the bonfire. I have about 40 people that meet at Apple's Assistance Center every month to surf and talk about the Internet. We accessed the Bonfire page every five minutes or so and watched the event live across the Net.

One of my current favorite hacks is the electronic postcard rack at **http://postcards.www.media.mit.edu/Postcards/**. This server allows you to send beautiful artwork to your friends along with a personal message.

I also like the sites that help provide tools for finding information on the Net. The Net is such a crazy quilt of disconnected and uncoordinated sites that the various indexes are absolutely essential.

Yahoo at **http://akebono.stanford.edu:80/yahoo/** is by far and away the most useful search engine on the net. I always try there first and I usually do not have to try anywhere else.

I also like to play around on the sites that have robots out combing the Net and building indexes. You can check some of these out at: **http://www.biotech.washington.edu/WebCrawler/WebCrawler.html** and **ftp://netcom5.netcom.com/pub/ksedgwic/hotlist/hotlist.html.**

Carl de Cordova
carldec@apple.com
http://www.info.apple.com/carldec.html
"Any sufficiently advanced technology is indistinguishable from a well rigged demo."

Appendix A

GLOSSARY

Anarchie
A Mac shareware application by Peter Lewis that encompasses FTP/Archie clients. Available from **http://wwwhost.ots.utexas.edu/mac/internet-ftp.html.**

Anchor(s)
The place(s) in an HTML document where a hypertext link(s) occurs.

AppleTalk
A series of related communications protocols introduced and maintained by Apple Computer that allows computers and peripherals to communicate on a network.

Application
Software that performs a task or tasks. Usually a launchable program. Examples: a Web browser or word processor.

Application layer
The OSI Reference Model layer that provides protocols for various network applications, such as eMail or file transfer.

Archie
A search protocol by which many of the FTP sites on the Internet can be examined simultaneously for files whose names match arbitrary patterns: you can search for files whether you know all or just a part of the filename. Essential.

ARPAnet
A 1970s experimental network that fostered the early development of Internet software.

Asynchronous communication
Data transmission in which characters are sent one at a time, encapsulated in control bits. Also refers to commands that may be sent before a response to a previously sent command has been received.

Attributes
SGML (and HTML) tags can accept attributes that further define their usage (much as parameters are used with command-line options, for those of you lucky enough to be familiar with the good old command line). A tag is often followed by an attribute, which in turn is assigned a particular value.

AU sounds
A type of audio format. See "SoundMachine."

Backbone network
A network acting as a primary conduit for traffic that is often both sourced from, and destined for, other networks.

Bandwidth
A measure of the information-carrying capacity of a network, measured in bits per second. The greater the bandwidth, the greater the data transmission capacity of the network.

Baud
The number of times per second that a modem changes the signal it sends during data transmission. See also "Bits per second."

BinHex
A file conversion procedure used to convert binary files to ASCII text files and back again. Files are binhexed before being transmitted via FTP or as attachments to eMail messages.

Bits per second (bps)
The rate of bit transmission over a communications link.

BRI
Basic Rate Interface is an ISDN service that has two full-duplex, 64Kbps B channels for data and voice transmissions and one full-duplex, 16Kbps D channel for control and monitoring functions. BRI is often referred to symbolically as "2B+D" or as "basic."

Browser
See "Web browser."

BTW
A common eMail abbreviation for "by the way."

Cache
The process of storing data in select locations within memory for convenient retrieval.

CERN
The European Particle Physics Laboratory in Geneva, Switzerland. Their research on hypertext technology formed the basis for the World Wide Web.

CGI (Common Gateway Interface)
CGI is the standard for external gateway programs to interface with information servers such as HTTP servers.

CIX (Commercial Internet Exchange)
An organization founded to create an understanding between network providers regarding commercial traffic accounting methods.

Client
A computer that links to a server, utilizing the services it provides through a communication link.

Client-server architecture
The client-server relationship operates by having a client computer connect to a host (server) computer via its server program. The client connects to the server and transmits a request for information. The server then disconnects from the client, processes the information request, and reconnects to the client to deliver the result.

Cyberspace
A term commonly used to describe the Internet, probably coined by science fiction writer William Gibson.

Dedicated line
A private telephone line reserved for a communication link between locations. For example, the connection between a LAN and an Internet provider will typically run on a dedicated line.

Dial-up connection
The most common type of connection to the Net for home computer users, utilizing telephone lines to connect the host to an Internet-connected computer.

Direct connection
A permanent connection between a single computer (or LAN) and the Internet, sometimes referred to as a "leased line connection."

Document
In World Wide Web parlance, a document refers to any text, media, or hyperlink file that can be transmitted to a client from an HTTP server.

Document window
Commonly refers to the scrollable window in a client program in which HTML documents can be viewed.

DNS
The Domain Name System; a database tool that allows one to translate alphabetic computer names into Internet numeric addresses, freeing the Internet user from having to remember long lists of numbers.

Driver
Software that allows a peripheral or internal device to be used by a computer, such as an Ethernet card driver.

Electronic mail (eMail, EMail, e-mail, etc.)
The transmission of information, usually text messages, between users over various types of networks, allowing for personalized addressing and other options, such as binary file (i.e. programs, documents, etc.) transfer with the message. See "Eudora."

Enterprise computing
A large computer network typically linking a variety of locations operating under various protocols within a large corporation.

Ethernet
A network protocol, faster than AppleTalk, frequently used to connect computers using TCP/IP to the Internet.

EtherTalk
AppleTalk protocols running over Ethernet.

Eudora
An eMail program that operates over a MacTCP connection, originally developed by Steve Dorner at the University of Illinois. Freeware version is available from **http://www.qualcomm.com/quest/QuestMain.html.**

External viewer
A helper application — a program that allows graphics, audio, or movies to be displayed in conjunction with or via a Web browser.

E-zines
Typically small, do-it-yourself publications that are distributed electronically over computer networks. (For more info go to **http://www.acns.nwu.edu/ezines/.**)

FAQs/FAQ list (Frequently Asked Questions)
Internet text files that address common questions about a specific subject area; a handy source of Net knowledge typically maintained by USENET newsgroups.

Fetch
An Internet FTP file transfer tool for the Macintosh, utilizing a user-friendly graphic interface. Unlike Archie protocol, you must know the exact server address you want to access. Essential. Available from **http://wwwhost.ots.utexas.edu/mac/internet-ftp.html.**

File server
A computer on a network used specifically to store files that may be accessed and used by other computers on the network. A File server will often be "dedicated," meaning that it exists only for this purpose and is not used for any other tasks.

Firewall
See "Security firewall."

Flame
A response to a USENET post, usually in the form of a personal attack against the author or subject.

Frame
A group of information "bytes" transmitted over a data link, similar to a "packet."

Freenet
A provider of free Internet access to the public.

FTP (File Transfer Protocol)
A standard protocol used to transfer files from one computer to another, or the act of transferring files (using FTP).

FYI
An information sheet about the Internet, or a common eMail abbreviation meaning "for your information."

Gateway
A computer system that allows for the transfer of data between incompatible applications by reformatting the data before sending it to its destination.

GIF (Graphic Interchange Format)
A file compression format allowing transfer of graphics files through online services.

Gopher
A menu-driven tool (program) used to locate and link to online sources of information. Essential. See "TurboGopher."

Gopherserver
A server configured to offer Gopher information.

Gopherspace
A term used to describe the entire Internet Gopher network.

Groupware
Applications designed to address tasks that involve two or more members of a group, such as meeting scheduling.

Helper applications
See "External viewer."

History
A function of some Web browsers that tracks all the documents viewed during the current session and allows you to call them up again.

Hits
Web slang for the number of visits tracked by a program like WebStat. More is usually better.

Home page
The initial display on a Web site, through which one can access the other documents within the site.

Homer
IRC (Internet Relay Chat) client software for the Macintosh. Available from **http://wwwhost.ots.utexas.edu/mac/internet-misc.html.**

Host
Typically, a computer that provides resources to other computers (clients) that reach it through a communications link, such as an Internet host computer.

Hotlist
A compilation, commonly personalized, of frequently visited Web sites and URLs.

HoTMetaL
A SoftQuad program that uses HTML codes to format documents for use on a Web site.

HTML (Hypertext Markup Language)
A formatting language that instructs a Web browser on how to display a document's various components.

HTML document
A document labeled in the HTML format, which may also be referred to as a Web document if it is accessible to the World Wide Web.

HTTP server (Hypertext Transport Protocol)
A server computer that utilizes the communication protocol for Web document transfer.

Hyperlink
See "Links."

Hypermedia
Graphics, movies, and audio features linked to a document, using the same principles as hypertext links.

Hypertext
Refers to a method of linking documents within a Web site that allows the browser to jump back and forth between files by clicking on hypertext "links."

IETF
See "Internet Engineering Task Force."

IMHO
A common eMail or "chat" abbreviation for "in my humble opinion."

Inline images
Graphics placed in a Web document that can either be loaded automatically, or by clicking on an icon.

Internet Architecture Board (IAB)
The group that makes decisions about standards and other important issues.

Internet Engineering Task Force (IETF)
A panel of volunteers that investigates and solves technical problems and reports to the Internet Architecture Board.

Internet resources
Information available to the public via the Internet.

Internet service provider (ISP)
An organization that provides Internet connections to its clients.

IP (Internet Protocol)
The most vital set of protocols that determine the way in which data travels across multiple networks via the Internet. IP is the network addressing portion of the TCP/IP protocol stack.

IP address
An IP address is a number assigned to any Internet-connected computer.

IPX (Internet Packet Exchange)
A Novell NetWare protocol for transmitting and routing packets.

IRC (Internet Relay Chat)
Basically a huge multi-user live chat on the Internet. Chats can be public or private. There are a number of major IRC servers around the world that are linked to each other. See "Homer."

ISDN (Integrated Services Digital Network)
Communication protocols for the transfer of voice, data, and other media over telephone networks. Much, much faster than your v.34 modem.

ISP
See "Internet service provider."

JPEG (Joint Photographic Experts Group)
An image compression protocol used to assist in the transfer of color images over computer networks.

JPEGView
A program that allows the browser to view graphics embedded in a Web document via JPEG or GIF formatting. Available from **http://wwwhost.ots.utexas.edu/mac/pub-mac-graphics.html.**

LAN
See "Local-area network."

Layer
One set of network protocols that is part of a complete set of protocols.

Leased line
See "Dedicated line."

Linked image
See "Inline images."

Links
Encoded text that allows you to jump from document to document in the Web. Also called hyperlinks, hotlinks, or anchors.

Local-Area Network (LAN)
A group of computers, usually within a fairly limited space, that are physically connected to one another.

LocalTalk
Apple Computer's system for connecting computers and peripherals via shielded twisted-pair cables and inexpensive connection boxes that is built into all Macintosh computers.

Lynx
A character-based Web browsing tool developed at the University of Kansas.

MacTCP
The program that allows a Macintosh to speak the Internet communications protocol TCP/IP (Transmission Control Protocol/Internet Protocol).

MacWeb
Web browser (freeware) developed at Microelectronics and Computer Technology Corporation (MCC). Available from **http://galaxy.einet.net/EINet/MacWeb/MacWebHome.html** or **http://wwwhost.ots.utexas.edu/mac/internet-www.html.**

Mail reflector
An eMail address that forwards mail to many other locations, such as participants in a particular discussion group.

Mail server
A computer on a network that acts as a storage place for eMail messages. Similar to a file server, a mail server stores messages that have been sent by one user until the intended recipient retrieves them.

MIME (Multi-purpose Internet Mail Extensions)
A format that allows non-text media files to be attached to eMail messages, such as audio or graphic features.

Modem
A device that connects a computer to a communication link for data transmission.

Mosaic
A graphical user interface (browser) for the World Wide Web, developed at the University of Illinois, available free to the Internet community. Available from **http://wwwhost.ots.utexas.edu/mac/internet-www.html.**

MPEG (Moving Pictures Expert Group)
An internationally recognized protocol for video compression. A viewing tool is needed to watch MPEG "desktop movies" on your computer. See "Sparkle."

MUD/MOO
MUD refers to "Multi-User Dungeon," a series of computer games based on the "Dungeons and Dragons" game. MUDs have been modified for use as educational and conferencing tools. MOO refers to Object-Oriented MUD.

Multimedia
Generally refers to different forms of data, such as text, audio or video, that may appear in a document.

NCSA
The National Center for Supercomputing Applications at the University of Illionois. Developers of Mosaic, NCSA Telnet, and a number of other freeware applications.

NCSA Telnet
An application that allows users to remotely log into hosts running the TELNET protocol. Available from **http://wwwhost.ots.utexas.edu/mac/internet-misc.html.**

Netscape
Client and server software available from **http://www.netscape.com/.**

Network
A collection of computers and peripherals that are able to communicate with each other via some set of network protocols.

NewsWatcher
Freeware used for reading and responding to newsgroups. Available from **http://charlotte.acns.nwu.edu/jln/progs.html** or **http://wwwhost.ots.utexas.edu/mac/internet-news.html.**

NFS (Network File System)
A set of protocols that allows you to read, write, or edit a file that sits on another computer in your network, using the same commands that you would use if the file existed locally, eliminating the need to "FTP" transfer the file to your machine to perform these tasks.

NIC (Network Information Center)
An organization that provides information about networks.

Node
A term used to describe a device that can access a network, such as a computer or printer.

NSFnet (National Science Foundation Network)
The National Science Foundation Network is one of the networks that comprise the Internet.

Packet
A bundle of data that travels across the Internet independently, usually in sizes of 1500 bytes or less.

Port
A port number delineates the particular Net application that is being used by a computer when it sends packets of information to another computer. A port is also a "jack" on the back of a computer that is used to connect a hardware accessory, such as a modem. The default port for a Web server is port 80.

Posting
A message sent to a USENET newsgroup, or the act of sending the message.

PRI (Primary Rate Interface)
The ISDN telecommunications standard capacity. It has two definitions depending on where you live. In the U.S., Canada, and Japan it is equivalent to 23 B channels plus one D channel (23B+D, 1.544Mbps capacity); in Europe it is equivalent to 31 B channels plus one D channel (31B+D, 2.048Mbps capacity).

Protocol
A set of rules that defines the way in which computers will communicate with each other by providing set patterns and formats for data transfer. Use of standard protocols allows different kinds of computers and software programs to exchange information.

Protocol layers
The different layers of a protocol refer to the different steps the networking software must take to accomplish the variety of tasks required for successful data transmission.

Protocol stack
Related layers of protocol software that function together to implement a particular communications process, such as AppleTalk.

Proxy server
An Internet server that allows one to access systems indirectly that have security "firewalls" prohibiting direct Internet access.

PPP (Point-to-Point Protocol)
A direct connection between a computer and the Internet (bypassing a typical host connection) using a modem and telephone lines, requiring PPP software, TCP, and an IP address. PPP connections are usually obtained from an Internet provider.

QuickTime
An Apple Computer video standard that runs as an extension file in the system folder. You need an additional viewing application to watch QuickTime "movies" on your screen. See "Sparkle."

RFC (Request for Comments)
Published documents describing the Internet's accepted and proposed standards and norms.

Router
Software or hardware that is used to join separate networks into larger Internetworks. It can also be used to transfer information (packets) between two networks operating under the same protocols, but which may be mechanically different.

RTFM
A colorful abbreviation commonly used in eMail meaning "read the f%#@ing manual."

Security firewall
A method of preventing direct Internet access to computers on a network by filtering or blocking certain network protocols or addresses. It is used to maintain security and privacy. See also "Proxy server."

Server
A computer that provides a variety of services, such as text libraries, file transfer, or eMail, to client computers.

Service provider
See "Internet service provider."

Session
A related group of communications transactions between different nodes on a network.

SGML (Standard Generalized Markup Language)
A document-formatting protocol utilizing codes to define the parts of a document.

Shell
On a UNIX system, software that accepts and processes command lines from your terminal. UNIX has multiple shells available, each with slightly different command formats and facilities.

Signature
A file that some people place at the bottom of an eMail message or USENET posting as a personal identification. They usually contain the sender's name and eMail address, and often include quotations, simple drawings, or other information.

SLIP (Serial Line Internet Protocol)
A direct connection between a computer and the Internet (as an alternative to a typical host connection) using a modem and telephone lines, requiring SLIP software, TCP, and an IP address. SLIP connections are usually obtained from an Internet provider.

SMTP (Simple Mail Transfer Protocol)
The eMail protocol designated for use in TCP/IP networks.

SNMP (Simple Network Management Protocol)
The TCP/IP network protocol for managing the various devices on a network.

Socket
A particular software process operating as a communications endpoint within a network device. Sockets are permanently assigned to the software process that they have been designated to service.

Socket number
The number that identifies which socket within a node is assigned to carry out a particular software task.

SoundMachine
Sound shareware. This application plays and records SND/AU sound files. Available from **http://wwwhost.ots.utexas.edu/mac/pub-mac-sound.html.**

Sparkle
A player for MPEG and QuickTime movies. Available from **http://wwwhost.ots.utexas.edu/mac/pub-mac-graphics.html.**

SQL (Structured Query Language)
A data manipulation protocol standardized by ANSI, commonly used in relational databases.

Switched access
A temporary network connection that is initiated when needed and subsequently discarded, such as a SLIP or PPP connection.

Synchronous communication
A process of data transmission in which both the sender and receiver synchronize their clocks each time a packet of data is exchanged.

Tags
The formatting codes in an HTML document that instruct the browsing software how to display the document's text, graphics, or sounds.

TCP/IP (Transmission Control Protocol/Internet Protocol)
Two basic and important Internet protocols. TCP allows for the reliable transmission of packets of data. IP allows data to travel across different systems.

Telnet
An application that allows a computer to log into another system via the Telnet protocol. See "NCSA Telnet."

Terminal emulation
A popular network application in which a computer runs software that makes it appear to be a terminal across the network, allowing it to "speak the language" of a mainframe that it might not otherwise be able to exchange data with.

TIFF (Tagged Image File Format)
A graphic file format developed by Aldus and Microsoft commonly used as an image transfer protocol on computer networks.

Time out
An interruption in the connection between two computers transversing a communications link.

TurboGopher
Gopher client freeware from the University of Minnesota. Available from **http://wwwhost.ots.utexas.edu/mac/internet-gopher.html.**

UNIX
An operating system intimate to functions that occur on the Internet.

URL (Uniform Resource Locator)
A system of address identification for all materials on the World Wide Web, including the method of access, and host computer to be contacted.

USENET
A loosely structured, informal group that exchanges messages or "news" about a particular subject, usually over the Internet. See "Newswatcher."

UUCP (UNIX-to-UNIX copy)
A protocol by which files are transported between UNIX systems.

Viewer
A program launched by a Web browser when it encounters a file that it can't translate or display.

W3O
An organization overseeing the future development of the World Wide Web, under the auspices of CERN and the Massachusetts Institute of Technology.

WAIS (Wide Area Information Service)
A service that helps one locate libraries and databases across the Internet.

WAIS gateway
Networks that are not able to speak directly to WAIS use a WAIS gateway to translate the data into a compatible format.

WAN (Wide-Area Network)
A system of networked computers that communicate over a large geographic area.

Web browser
Software that allows a user to access and view HTML documents, such as Mosaic, Lynx, MacWeb, or Netscape.

WebMaster
The administrator of a World Wide Web site.

Web node
Another term for a Web site or Web server.

Web page
An HTML document that one accesses via the Web.

Webspace
A term used to describe the "space" on the World Wide Web.

Web spider
A program that shows a map of the links that one has crossed over while browsing the Web.

WebStat
Freeware that "reads" MacHTTP log files and reports server usage statistics in HTML format. Available from **http://wwwhost.ots.utexas.edu/mac/internet-www.html** or **http://snodaq.phy.queensu.ca/Phil/phil.html.**

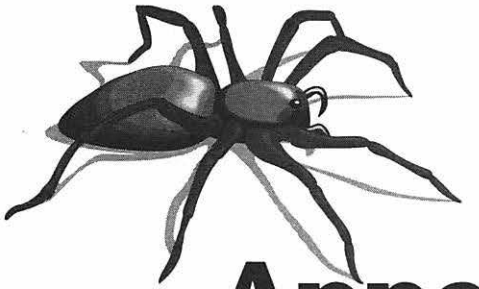
Web walking (Web surfing)
The act of traversing sites and documents on the World Wide Web via a Web browsing tool.

Workstation
A computer on a network.

World Wide Web
A method of locating and utilizing Internet resources via a graphical user interface (browser) and hyperlinks.

XBM
XBM files are X Window system bitmaps.

Zines
See "E-zines."



Appendix B

INTERNET PROVIDERS

We combed the Net and thumbed through back issues of Net magazines to come up with a comprehensive and up-to-date list of Internet providers for you. And basically, we give up. We've reproduced the best list we found, the one from Paul Celestin, on the next pages.

But better than any printed list are two excellent Web pages: The TAG Online Mall and Paul Celestin's comprehensive list of Internet providers (the Web page is guaranteed to be more up to date than the pages that follow). Check out:

http://www.teleport.com/~cci/directories/pocia/pocia.html

and

http://www.tagsys.com/

We promised our friends at TAG we'd put a plug in for them, so that's not the exact address of the page with the list. That URL will force you to start at the TAG home page, which will make you check out a nifty example of a Web mall on your way to the list, which is easy to find from the home page. By the way, this site has a great searchable database.

And now, here's Paul Celestin's list as of February 1995, reprinted here with his kind permission.

PROVIDERS OF COMMERCIAL INTERNET ACCESS (POCIA) DIRECTORY

Updated 7 February 1995. The information in this directory is provided as-is and without any expressed or implied warranties, including, without limitation, the implied warranties of merchantability and fitness for a particular purpose. You may use the information in this directory for non-commercial purposes only. Contact us if you wish to use the directory for a commercial purpose.

All of the information in this directory was supplied to Celestin Company directly by the service providers and is subject to change without notice.

This directory is brought to you as a public service. Celestin Company does not receive any compensation from the providers listed here. Since Internet service providers come and go, and frequently change their offers, we strongly urge you to contact them for additional information and/or restrictions.

The latest version of this document is available at the following location:

ftp://ftp.teleport.com/vendors/cci/pocia/pocia.txt

You may also retrieve the latest copy (as well as additional information on Celestin Company and its products) using eMail. For information on how this works, send a blank message to **cci@olympus.net**. If you have Web access, try **http://www.teleport.com/~cci/** for the hypertext version of this list, which includes addresses, telephone numbers, fax numbers, eMail addresses, and pricing.

DOMESTIC

A listing of Internet service providers in the U.S. and Canada, sorted by area code. Fields are area code, service provider name, voice phone number, and eMail address for more information.

Free Service Providers

Cyberspace (shell,slip,ppp)	modem -> 515 945 7000	info@cyberspace.com
Free.org (shell,slip,ppp)	modem -> 715 743 1600	info@free.org
Free.I.Net (via AT&T)	modem -> 801 471 2266	info@free.i.net
SLIPNET (shell,slip,ppp)	modem -> 217 792 2777	info@slip.net

Nationwide Service Providers

ANS	703 758 7700	info@ans.net
Global Connect, Inc.	804 229 4484	info@gc.net
Informed Access Technologies Holonet	510 704 0160	info@holonet.com
NETCOM On-Line Communications Services	408 554 8649	info@netcom.com
Network 99, Inc.	800 NET 99IP	net99@cluster.mcs.net
Performance Systems International800 827 7482	all-info@psi.com	

Toll-Free Service Providers

American Information Systems, Inc.	708 413 8400	info@ais.net
Association for Computing Machinery	817 776 6876	account-info@acm.org
CICNet, Inc.	313 998 6103	info@cic.net
Colorado SuperNet, Inc.	303 296 8202	info@csn.org
DataBank, Inc.	913 842 6699	info@databank.com
Global Connect, Inc.	804 229 4484	info@gc.net
Internet Express	719 592 1240	info@usa.net
Msen, Inc.	313 998 4562	info@msen.com
NeoSoft, Inc.	713 684 5969	info@neosoft.com
New Mexico Technet, Inc.	505 345 6555	granoff@technet.nm.org
Pacific Rim Network, Inc.	360 650 0442	info@pacificrim.net
Prometheus Information Network Group Inc.	404 399 1670	info@ping.com
Rocky Mountain Internet	800 900 7644	info@rmii.com

Regional Service Providers

201	Carroll-Net	201 488 1332	info@carroll.com
201	The Connection	201 435 4414	info@cnct.com
201	Digital Express Group	301 220 2020	info@digex.net
201	INTAC Access Corporation	800 504 6822	info@intac.com
201	InterCom Online	212 714 7183	info@intercom.com
201	Internet Online Services	201 928 1000 x226	help@ios.com
201	Mordor International BBS	201 433 4222	ritz@mordor.com
201	NETCOM On-Line Communications Services	408 554 8649	info@netcom.com
201	New York Net	718 776 6811	sales@new-york.net
201	NIC - Neighborhood Internet Connection	201 934 1445	info@nic.com
202	CAPCON Library Network	202 331 5771	info@capcon.net
202	Charm.Net	410 558 3900	info@charm.net
202	Digital Express Group	301 220 2020	info@digex.net
202	Genuine Computing Resources	703 878 4680	info@gcr.com
202	Quantum Networking Solutions, Inc.	805 538 2028	info@qnet.com
202	US Net, Incorporated	301 572 5926	info@us.net

203	Connix: Connecticut Internet Exchange	203 349 7059		info@connix.com
203	Futuris Networks, Inc.	203 359 8868		info@futuris.net
203	I-2000 Inc.	516 867 6379		info@i-2000.com
203	Paradigm Communications, Inc.	203 250 7397		info@pcnet.com
205	Community Internet Connect, Inc.	205 722 0199		info@cici.com
205	interQuest, Inc.	205 464 8280		info@iquest.com
206	Eskimo North	206 367 7457		nanook@eskimo.com
206	NETCOM On-Line Communications Services	408 554 8649		info@netcom.com
206	NorthWest CommLink	206 336 0103		info@nwcl.net
206	Northwest Nexus, Inc.	206 455 3505		info@nwnexus.wa.com
206	Pacifier Computers	206 254 3886		account@pacifier.com
206	Seanet Online Services	206 343 7828		info@seanet.com
206	SenseMedia	408 335 9400		sm@picosof.com
206	Skagit On-Line Services	206 755 0190		info@sos.net
206	Structured Network Systems, Inc.	503 656 3530		info@structured.net
206	Teleport, Inc.	503 223 4245		info@teleport.com
206	Transport Logic	503 243 1940		sales@transport.com
207	Agate Internet Services	207 947 8248		ais@agate.net
208	Minnesota Regional Network	612 342 2570		info@mr.net
208	NICOH Net	208 233 5802		info@nicoh.com
208	Primenet	602 870 1010		info@primenet.com
208	Transport Logic	503 243 1940		sales@transport.com
209	Cybergate Information Services	209 486 4283		cis@cybergate.com
209	West Coast Online	707 586 3060		info@calon.com
212	Alternet (UUNET Technologies, Inc.)	703 204 8000		info@alter.net
212	Blythe Systems	212 226 7171		infodesk@blythe.org
212	Creative Data Consultants (SILLY.COM)	718 229 0489		info@silly.com
212	Digital Express Group	301 220 2020		info@digex.net
212	Echo Communications Group	212 255 3839		info@echonyc.com
212	escape.com - Kazan Corp	212 888 8780		info@escape.com
212	Ingress Communications Inc.	212 679 2838		info@ingress.com
212	INTAC Access Corporation	800 504 6822		info@intac.com
212	InterCom Online	212 714 7183		info@intercom.com
212	Internet Online Services	201 928 1000	x226	help@ios.com
212	Interport Communications Corp.	212 989 1128		info@interport.net
212	Mordor International BBS	201 433 4222		ritz@mordor.com
212	NETCOM On-Line Communications Services	408 554 8649		info@netcom.com
212	New York Net	718 776 6811		sales@new-york.net
212	Phantom Access Technologies, Inc.	212 989 2418		bruce@phantom.com
212	The Pipeline Network	212 267 2626		info-info@pipeline.com
213	Abode Computer Service	818 287 5115		eric@abode.ttank.com
213	Delta Internet Services	714 778 0370		info@deltanet.com
213	DigiLink Network Services	310 542 7421		info@digilink.net
213	DirectNet	213 383 3144		info@directnet.com
213	EarthLink Network, Inc.	213 644 9500		info@earthlink.net
213	Electriciti	619 338 9000		info@powergrid.electriciti.com
213	KAIWAN Corporation	714 638 2139		info@kaiwan.com
213	Primenet	602 870 1010		info@primenet.com
213	ViaNet Communications	415 903 2242		info@via.net
214	Alternet (UUNET Technologies, Inc.)	703 204 8000		info@alter.net
214	DFW Internet Services, Inc.	817 332 5116		info@dfw.net
214	NETCOM On-Line Communications Services	408 554 8649		info@netcom.com
214	Texas Metronet, Inc.	214 705 2900		info@metronet.com
215	FishNet	610 337 9994		info@pond.com
215	Microserve Information Systems	717 779 4430		info@microserve.com
215	Oasis Telecommunications, Inc.	610 439 8560		staff@oasis.ot.com
215	YOU TOOLS Corporation	610 954 5910		info@youtools.com
216	APK Public Access UNI* Site	216 481 9436		info@wariat.org
216	Branch Information Services	313 741 4442		branch-info@branch.com
216	Exchange Network Services, Inc.	216 261 4593		info@en.com
216	OARnet (corporate clients only)	614 728 8100		info@oar.net
218	Red River Net	701 232 2227		info@rrnet.com
301	Charm.Net	410 558 3900		info@charm.net
301	Clark Internet Services, Inc. ClarkNet	410 995 0691		info@clark.net

301	Digital Express Group	301 220 2020	info@digex.net
301	Genuine Computing Resources	703 878 4680	info@gcr.com
301	Quantum Networking Solutions, Inc.	805 538 2028	info@qnet.com
301	SURAnet	301 982 4600	marketing@sura.net
301	US Net, Incorporated	301 572 5926	info@us.net
302	SSNet, Inc.	302 378 1386	info@ssnet.com
303	Colorado SuperNet, Inc.	303 296 8202	info@csn.org
303	Internet Express	719 592 1240	info@usa.net
303	NETCOM On-Line Communications Services	408 554 8649	info@netcom.com
303	New Mexico Technet, Inc.	505 345 6555	granoff@technet.nm.org
303	Rocky Mountain Internet	800 900 7644	info@rmii.com
305	Acquired Knowledge Systems Inc.	305 525 2574	info@aksi.net
305	CyberGate, Inc.	305 428 4283	sales@gate.net
305	InteleCom Data Systems, Inc.	401 885 6855	info@ids.net
305	SatelNET Communications	305 434 8738	admin@satelnet.org
310	Abode Computer Service	818 287 5115	eric@abode.ttank.com
310	Cloverleaf Communications	714 895 3075	sales@cloverleaf.com
310	Delta Internet Services	714 778 0370	info@deltanet.com
310	DigiLink Network Services	310 542 7421	info@digilink.net
310	EarthLink Network, Inc.	213 644 9500	info@earthlink.net
310	KAIWAN Corporation	714 638 2139	info@kaiwan.com
310	Lightside, Inc.	818 858 9261	info@lightside.com
310	NETCOM On-Line Communications Services	408 554 8649	info@netcom.com
310	ViaNet Communications	415 903 2242	info@via.net
312	American Information Systems, Inc.	708 413 8400	info@ais.net
312	CICNet, Inc.	313 998 6103	info@cic.net
312	InterAccess Co.	800 967 1580	info@interaccess.com
312	MCSNet	312 248 8649	info@mcs.net
312	NETCOM On-Line Communications Services	408 554 8649	info@netcom.com
312	Open Business Systems, Inc.	708 250 0260	info@obs.net
312	Ripco Communications, Inc.	312 477 6210	info@ripco.com
312	Tezcatlipoca, Inc.	312 850 0181	info@tezcat.com
312	WorldWide Access	708 367 1870	info@wwa.com
313	Branch Information Services	313 741 4442	branch-info@branch.com
313	CICNet, Inc.	313 998 6103	info@cic.net
313	ICNET / Innovative Concepts	313 998 0090	info@ic.net
313	Msen, Inc.	313 998 4562	info@msen.com
314	NeoSoft, Inc.	713 684 5969	info@neosoft.com
314	ThoughtPort, Inc.	314 474 6870	info@thoughtport.com
315	ServiceTech Inc. Cyber-Link	716 546 6908	dam@cyber1.servtech.com
316	SouthWind Internet Access, Inc.	316 263 7963	info@southwind.net
317	Branch Information Services	313 741 4442	branch-info@branch.com
317	IQuest Network Services	317 259 5050	info@iquest.net
360	Pacific Rim Network, Inc.	360 650 0442	info@pacificrim.net
360	Townsend Communications, Inc.	360 385 0464	info@olympus.net
360	Whidbey Connections, Inc.	360 678 1070	info@whidbey.net
401	InteleCom Data Systems, Inc.	401 885 6855	info@ids.net
402	Internet Nebraska	402 434 8680	info@inetnebr.com
403	Debug Computer Services	403 248 5798	root@debug.cuc.ab.ca
403	UUNET Canada, Inc.	416 368 6621	info@uunet.ca
404	Internet Atlanta	404 410 9000	info@atlanta.com
404	MindSpring	404 888 0725	info@mindspring.com
404	NETCOM On-Line Communications Services	408 554 8649	info@netcom.com
404	Prometheus Information Network Group Inc.	404 399 1670	info@ping.com
405	Internet Oklahoma	405 721 1580	info@ionet.net
405	Questar Network Services	405 848 3228	info@qns.net
406	Montana Online	406 721 4952	info@montana.com
407	CyberGate, Inc.	305 428 4283	sales@gate.net
407	Florida Online	407 635 8888	info@digital.net
407	InteleCom Data Systems, Inc.	401 885 6855	info@ids.net
407	InternetU	407 952 8487	info@iu.net
407	MagicNet, Inc.	407 657 2202	info@magicnet.net
408	Aimnet Information Services	408 257 0900	info@aimnet.com
408	Alternet (UUNET Technologies, Inc.)	703 204 8000	info@alter.net

408	BTR Communications Company	415 966 1429	support@btr.com
408	Direct Net Access Incorporated	510 649 6110	support@dnai.com
408	The Duck Pond Public Unix	modem -> 408 249 9630	postmaster@kfu.com
408	Electriciti	619 338 9000	info@powergrid.electriciti.com
408	Infoserv Connections	408 335 5600	root@infoserv.com
408	InterNex Information Services, Inc.	415 473 3060	info@internex.net
408	ISP Networks	408 653 0100	info@isp.net
408	NETCOM On-Line Communications Services	408 554 8649	info@netcom.com
408	Scruz-Net	408 457 5050	info@scruz.net
408	SenseMedia	408 335 9400	sm@picosof.com
408	South Valley Internet	408 683 4533	info@garlic.com
408	West Coast Online	707 586 3060	info@calon.com
408	zNET	619 755 7772	info@znet.com
408	Zocalo Engineering	510 540 8000	info@zocalo.net
409	Internet Connect Services, Inc.	512 572 9987	info@icsi.net
410	CAPCON Library Network	202 331 5771	info@capcon.net
410	Charm.Net	410 558 3900	info@charm.net
410	Clark Internet Services, Inc. ClarkNet	410 995 0691	info@clark.net
410	Digital Express Group	301 220 2020	info@digex.net
410	US Net, Incorporated	301 572 5926	info@us.net
412	Telerama Public Access Internet	412 481 3505	info@telerama.lm.com
413	Mallard Electronics, Inc.	413 732 0214	gheacock@map.com
413	ShaysNet.COM	413 772 3774	staff@shaysnet.com
414	Exec-PC, Inc.	414 789 4200	info@execpc.com
414	FullFeed Communications	608 246 4239	info@fullfeed.com
414	MIX Communications	414 351 1868	info@mixcom.com
415	Aimnet Information Services	408 257 0900	info@aimnet.com
415	Alternet (UUNET Technologies, Inc.)	703 204 8000	info@alter.net
415	BTR Communications Company	415 966 1429	support@btr.com
415	Community ConneXion - NEXUS-Berkeley	510 549 1383	info@c2.org
415	Direct Net Access Incorporated	510 649 6110	support@dnai.com
415	InterNex Information Services, Inc.	415 473 3060	info@internex.net
415	LineX Communications	415 455 1650	info@linex.com
415	NETCOM On-Line Communications Services	408 554 8649	info@netcom.com
415	QuakeNet	415 655 6607	info@quake.net
415	SLIPNET	415 281 3132	info@slip.net
415	ViaNet Communications	415 903 2242	info@via.net
415	The WELL	415 332 4335	info@well.com
415	West Coast Online	707 586 3060	info@calon.com
415	zNET	619 755 7772	info@znet.com
415	Zocalo Engineering	510 540 8000	info@zocalo.net
416	HookUp Communications	905 847 8000	info@hookup.net
416	InterLog Internet Services	416 975 2655	internet@interlog.com
416	UUNET Canada, Inc.	416 368 6621	info@uunet.ca
418	UUNET Canada, Inc.	416 368 6621	info@uunet.ca
419	Branch Information Services	313 741 4442	branch-info@branch.com
419	OARnet (corporate clients only)	614 728 8100	info@oar.net
501	Cloverleaf Technologies	903 832 1367	helpdesk@clover.cleaf.com
501	Sibylline, Inc.	501 521 4660	info@sibylline.com
502	IgLou Internet Services	800 436 4456	info@iglou.com
503	Alternet (UUNET Technologies, Inc.)	703 204 8000	info@alter.net
503	Data Research Group, Inc.	503 465 3282	info@ordata.com
503	Europa	503 222 9508	info@europa.com
503	Hevanet Communications	503 228 3520	info@hevanet.com
503	NETCOM On-Line Communications Services	408 554 8649	info@netcom.com
503	Open Door Networks, Inc.	503 488 4127	info@opendoor.com
503	RainDrop Laboraties	503 293 1772	info@agora.rdrop.com
503	Structured Network Systems, Inc.	503 656 3530	info@structured.net
503	Teleport, Inc.	503 223 4245	info@teleport.com
503	Transport Logic	503 243 1940	sales@transport.com
504	Communique Inc.	504 527 6200	info@communique.net
504	NeoSoft, Inc.	713 684 5969	info@neosoft.com
505	Computer Systems Consulting	505 984 0085	info@spy.org
505	Internet Express	719 592 1240	info@usa.net

505	New Mexico Technet, Inc.	505 345 6555	granoff@technet.nm.org
505	Southwest Cyberport	505 271 0009	info@swcp.com
505	ZyNet SouthWest	505 343 8846	zycor@zynet.com
506	Agate Internet Services	207 947 8248	ais@agate.net
507	Internet Connections, Inc.	507 625 7320	info@ic.mankato.mn.us
507	Millennium Communications, Inc.	612 338 5509	info@millcomm.com
507	Minnesota Regional Network	612 342 2570	info@mr.net
508	The Destek Group, Inc.	603 635 3857	inquire@destek.net
508	FOURnet Information Network	508 291 2900	info@four.net
508	The Internet Access Company (TIAC)	617 276 7200	info@tiac.net
508	intuitive information, inc.	508 341 1100	info@iii.net
508	North Shore Access	617 593 3110	info@shore.net
508	SCHUNIX	508 853 0258	schu@schunix.com
508	StarNet	508 922 8238	info@venus.star.net
508	UltraNet Communications, Inc.	508 229 8400	info@ultra.net.com
508	The World	617 739 0202	info@world.std.com
508	Wrentham Internet Services	508 384 1404	info@riva.com
508	Wilder Systems, Inc.	617 933 8810	info@id.wing.net
509	Internet On-Ramp	509 927 7267	info@on-ramp.ior.com
509	Transport Logic	503 243 1940	sales@transport.com
510	Alternet (UUNET Technologies, Inc.)	703 204 8000	info@alter.net
510	BTR Communications Company	415 966 1429	support@btr.com
510	Community ConneXion - NEXUS-Berkeley	510 549 1383	info@c2.org
510	Direct Net Access Incorporated	510 649 6110	support@dnai.com
510	InterNex Information Services, Inc.	415 473 3060	info@internex.net
510	LanMinds, Inc.	510 843 6389	info@lanminds.com
510	LineX Communications	415 455 1650	info@linex.com
510	NETCOM On-Line Communications Services	408 554 8649	info@netcom.com
510	SLIPNET	415 281 3132	info@slip.net
510	West Coast Online	707 586 3060	info@calon.com
510	Zocalo Engineering	510 540 8000	info@zocalo.net
512	@sig.net	512 306 0700	sales@aus.sig.net
512	Internet Connect Services, Inc.	512 572 9987	info@icsi.net
512	NETCOM On-Line Communications Services	408 554 8649	info@netcom.com
512	Real/Time Communications	512 451 0046	info@realtime.net
512	Zilker Internet Park, Inc.	512 206 3850	info@zilker.net
513	The Dayton Network Access Company	513 237 6868	info@dnaco.net
513	IgLou Internet Services	800 436 4456	info@iglou.com
513	Internet Access Cincinnati	513 887 8877	info@iac.net
513	Local Internet Gateway Co.	510 503 9227	sdw@lig.net
513	OARnet (corporate clients only)	614 728 8100	info@oar.net
514	Communication Accessibles Montreal	514 288 2581	info@cam.org
514	Communications Inter-Acces	514 367 0002	info@interax.net
514	UUNET Canada, Inc.	416 368 6621	info@uunet.ca
516	Creative Data Consultants (SILLY.COM)	718 229 0489	info@silly.com
516	Echo Communications Group	212 255 3839	info@echonyc.com
516	I-2000 Inc.	516 867 6379	info@i-2000.com
516	LI Net, Inc.	516 476 1168	info@li.net
516	Long Island Information, Inc.	516 294 0124	info@liii.com
516	Network Internet Services	516 543 0234	info@netusa.net
516	Phantom Access Technologies, Inc.	212 989 2418	bruce@phantom.com
516	The Pipeline Network	212 267 2626	info-info@pipeline.com
517	Msen, Inc.	313 998 4562	info@msen.com
517	Branch Information Services	313 741 4442	branch-info@branch.com
518	wizvax communications	518 271 0049	info@wizvax.com
519	HookUp Communications	905 847 8000	info@hookup.net
519	MGL Systems Computer Technologies Inc.	519 651 2713	info@mgl.ca
519	UUNET Canada, Inc.	416 368 6621	info@uunet.ca
520	Opus One	602 324 0494	sales@opus1.com
520	RTD Systems & Networking, Inc.	602 318 0696	info@rtd.com
602	Crossroads Communications	602 813 9040	crossroads@xroads.com
602	Internet Direct, Inc.	602 274 0100	info@indirect.com
602	Internet Express	719 592 1240	info@usa.net
602	NETCOM On-Line Communications Services	408 554 8649	info@netcom.com

602	New Mexico Technet, Inc.	505 345 6555	granoff@technet.nm.org
602	Opus One	602 324 0494	sales@opus1.com
602	Primenet	602 870 1010	info@primenet.com
602	RTD Systems & Networking, Inc.	602 318 0696	info@rtd.com
603	Agate Internet Services	207 947 8248	ais@agate.net
603	The Destek Group, Inc.	603 635 3857	inquire@destek.net
603	MV Communications, Inc.	603 429 2223	info@mv.mv.com
603	NETIS Public Access Internet	603 437 1811	epoole@leotech.mv.com
604	AMT Solutions Group, Inc. Island Net	604 727 6030	info@islandnet.com
604	Mind Link!	604 534 5663	info@mindlink.bc.ca
604	Okanagan Internet Junction	604 549 1036	info@junction.net
604	Sunshine Net, Inc.	604 886 4120	admin@sunshine.net
604	UUNET Canada, Inc.	416 368 6621	info@uunet.ca
606	IgLou Internet Services	800 436 4456	info@iglou.com
606	Internet Access Cincinnati	513 887 8877	info@iac.net
608	FullFeed Communications	608 246 4239	info@fullfeed.com
609	Digital Express Group	301 220 2020	info@digex.net
609	New Jersey Computer Connection	609 896 2799	info@pluto.njcc.com
610	FishNet	610 337 9994	info@pond.com
610	Microserve Information Systems	717 779 4430	info@microserve.com
610	SSNet, Inc.	302 378 1386	info@ssnet.com
610	Oasis Telecommunications, Inc.	610 439 8560	staff@oasis.ot.com
610	YOU TOOLS Corporation	610 954 5910	info@youtools.com
612	Millennium Communications, Inc.	612 338 5509	info@millcomm.com
612	Minnesota Regional Network	612 342 2570	info@mr.net
612	StarNet Communications, Inc.	612 941 9177	info@winternet.com
613	Information Gateway Services	613 592 5619	info@igs.net
613	HookUp Communications	905 847 8000	info@hookup.net
613	o://info.web	613 225 3354	kevin@magi.com
613	UUNET Canada, Inc.	416 368 6621	info@uunet.ca
614	Branch Information Services	313 741 4442	branch-info@branch.com
614	Internet Access Cincinnati	513 887 8877	info@iac.net
614	OARnet (corporate clients only)	614 728 8100	info@oar.net
615	ERC, Inc. / The Edge	615 455 9915	staff@edge.ercnet.com
615	GoldSword Systems	615 691 6498	info@goldsword.com
615	ISDN-Net Inc	615 377 7672	jdunlap@rex.isdn.net
615	Telalink Corporation	615 321 9100	sales@telalink.net
616	Branch Information Services	313 741 4442	branch-info@branch.com
616	Msen, Inc.	313 998 4562	info@msen.com
617	Alternet (UUNET Technologies, Inc.)	703 204 8000	info@alter.net
617	The Internet Access Company (TIAC)	617 276 7200	info@tiac.net
617	intuitive information, inc.	508 341 1100	info@iii.net
617	NETCOM On-Line Communications Services	408 554 8649	info@netcom.com
617	North Shore Access	617 593 3110	info@shore.net
617	UltraNet Communications, Inc.	508 229 8400	info@ultra.net.com
617	The World	617 739 0202	info@world.std.com
617	Wilder Systems, Inc.	617 933 8810	info@id.wing.net
619	CTS Network Services (CTSNET)	619 637 3637	info@cts.com
619	The Cyberspace Station	619 634 2894	info@cyber.net
619	Electriciti	619 338 9000	info@powergrid.electriciti.com
619	NETCOM On-Line Communications Services	408 554 8649	info@netcom.com
701	Red River Net	701 232 2227	info@rrnet.com
702	@wizard.com	702 871 4461	info@wizard.com
702	Great Basin Internet Services	702 829 2244	info@greatbasin.com
702	NETCOM On-Line Communications Services	408 554 8649	info@netcom.com
702	Sierra-Net	702 831 3353	giles@sierra.net
703	Alternet (UUNET Technologies, Inc.)	703 204 8000	info@alter.net
703	CAPCON Library Network	202 331 5771	info@capcon.net
703	Charm.Net	410 558 3900	info@charm.net
703	Clark Internet Services, Inc. ClarkNet	410 995 0691	info@clark.net
703	DataBank, Inc.	913 842 6699	info@databank.com
703	Digital Express Group	301 220 2020	info@digex.net
703	Genuine Computing Resources	703 878 4680	info@gcr.com
703	NETCOM On-Line Communications Services	408 554 8649	info@netcom.com

703	Quantum Networking Solutions, Inc.	805 538 2028	info@qnet.com
703	US Net, Incorporated	301 572 5926	info@us.net
704	SunBelt.Net	803 328 1500	info@sunbelt.net
704	Vnet Internet Access	704 334 3282	info@vnet.net
705	Mindemoya Computing	705 523 0243	info@mcd.on.ca
706	Internet Atlanta	404 410 9000	info@atlanta.com
706	MindSpring	404 888 0725	info@mindspring.com
707	West Coast Online	707 586 3060	info@calon.com
707	Zocalo Engineering	510 540 8000	info@zocalo.net
708	American Information Systems, Inc.	708 413 8400	info@ais.net
708	CICNet, Inc.	313 998 6103	info@cic.net
708	InterAccess Co.	800 967 1580	info@interaccess.com
708	MCSNet	312 248 8649	info@mcs.net
708	Open Business Systems, Inc.	708 250 0260	info@obs.net
708	Ripco Communications, Inc.	312 477 6210	info@ripco.com
708	Tezcatlipoca, Inc.	312 850 0181	info@tezcat.com
708	WorldWide Access	708 367 1870	info@wwa.com
713	Alternet (UUNET Technologies, Inc.)	703 204 8000	info@alter.net
713	The Black Box	713 480 2684	info@blkbox.com
713	Internet Connect Services, Inc.	512 572 9987	info@icsi.net
713	NeoSoft, Inc.	713 684 5969	info@neosoft.com
713	USiS	713 682 1666	admin@usis.com
714	Cloverleaf Communications	714 895 3075	sales@cloverleaf.com
714	Delta Internet Services	714 778 0370	info@deltanet.com
714	DigiLink Network Services	310 542 7421	info@digilink.net
714	EarthLink Network, Inc.	213 644 9500	info@earthlink.net
714	Electriciti	619 338 9000	info@powergrid.electriciti.com
714	KAIWAN Corporation	714 638 2139	info@kaiwan.com
714	Lightside, Inc.	818 858 9261	info@lightside.com
714	NETCOM On-Line Communications Services	408 554 8649	info@netcom.com
715	FullFeed Communications	608 246 4239	info@fullfeed.com
716	E-Znet, Inc.	716 262 2485	
716	ServiceTech Inc. Cyber-Link	716 546 6908	dam@cyber1.servtech.com
717	Microserve Information Systems	717 779 4430	info@microserve.com
717	Oasis Telecommunications, Inc.	610 439 8560	staff@oasis.ot.com
717	YOU TOOLS Corporation	610 954 5910	info@youtools.com
718	Blythe Systems	212 226 7171	infodesk@blythe.org
718	Creative Data Consultants (SILLY.COM)	718 229 0489	info@silly.com
718	escape.com - Kazan Corp	212 888 8780	info@escape.com
718	I-2000 Inc.	516 867 6379	info@i-2000.com
212	Ingress Communications Inc.	212 679 2838	info@ingress.com
718	InterCom Online	212 714 7183	info@intercom.com
718	Interport Communications Corp.	212 989 1128	info@interport.net
718	Mordor International BBS	201 433 4222	ritz@mordor.com
718	Phantom Access Technologies, Inc.	212 989 2418	bruce@phantom.com
718	The Pipeline Network	212 267 2626	info-info@pipeline.com
719	Colorado SuperNet, Inc.	303 296 8202	info@csn.org
719	Internet Express	719 592 1240	info@usa.net
719	Old Colorado City Communications	719 528 5849	thefox@oldcolo.com
719	Rocky Mountain Internet	800 900 7644	info@rmii.com
801	Infonaut Communication Services	801 370 3068	info@infonaut.com
801	Internet Direct, Inc.	801 578 0300	info@indirect.com
801	XMission	801 539 0852	support@xmission.com
803	Global Vision Inc.	803 241 0901	info@globalvision.net
803	SIMS, Inc.	803 762 4956	info@sims.net
803	SunBelt.Net	803 328 1500	info@sunbelt.net
804	Widomaker Communication Service	804 253 7621	bloyall@widowmaker.com
805	EarthLink Network, Inc.	213 644 9500	info@earthlink.net
805	Quantum Networking Solutions, Inc.	805 538 2028	info@qnet.com
805	Regional Alliance for Info Networking	805 967 7246	info@rain.org
808	Hawaii OnLine	808 533 6981	support@aloha.net
808	LavaNet, Inc.	808 545 5282	info@lava.net
808	Pacific Information Exchange, Inc.	808 596 7494	info@pixi.com
810	Branch Information Services	313 741 4442	branch-info@branch.com

810	ICNET / Innovative Concepts	313	998	0090	info@ic.net
810	Msen, Inc.	313	998	4562	info@msen.com
810	RustNet, Inc.	810	650	6812	info@rust.net
812	IgLou Internet Services	800	436	4456	info@iglou.com
813	CyberGate, Inc.	305	428	4283	sales@gate.net
813	Intelligence Network Online, Inc.	813	442	0114 x22	info@intnet.net
813	PacketWorks, Inc.	813	446	8826	info@packet.net
815	American Information Systems, Inc.	708	413	8400	info@ais.net
815	InterAccess Co.	800	967	1580	info@interaccess.com
816	Primenet	602	870	1010	info@primenet.com
817	Association for Computing Machinery	817	776	6876	account-info@acm.org
817	DFW Internet Services, Inc.	817	332	5116	info@dfw.net
817	Texas Metronet, Inc.	214	705	2900	info@metronet.com
818	Abode Computer Service	818	287	5115	eric@abode.ttank.com
818	Delta Internet Services	714	778	0370	info@deltanet.com
818	DigiLink Network Services	310	542	7421	info@digilink.net
818	EarthLink Network, Inc.	213	644	9500	info@earthlink.net
818	KAIWAN Corporation	714	638	2139	info@kaiwan.com
818	Lightside, Inc.	818	858	9261	info@lightside.com
818	NETCOM On-Line Communications Services	408	554	8649	info@netcom.com
818	Primenet	602	870	1010	info@primenet.com
818	Regional Alliance for Info Networking	805	967	7246	info@rain.org
818	ViaNet Communications	415	903	2242	info@via.net
819	o://info.web	613	225	3354	kevin@magi.com
901	ISDN-Net Inc	615	377	7672	jdunlap@rex.isdn.net
901	Magibox Incorporated	901	757	7835	info@magibox.net
903	Cloverleaf Technologies	903	832	1367	helpdesk@clover.cleaf.com
904	CyberGate, Inc.	305	428	4283	sales@gate.net
904	SymNet	904	385	1061	info@symnet.net
905	HookUp Communications	905	847	8000	info@hookup.net
905	InterLog Internet Services	416	975	2655	internet@interlog.com
905	Vaxxine Computer Systems Inc.	905	562	3500	admin@vaxxine.com
906	Branch Information Services	313	741	4442	branch-info@branch.com
906	Msen, Inc.	313	998	4562	info@msen.com
908	Castle Network, Inc.	908	548	8881	request@castle.net
908	Digital Express Group	301	220	2020	info@digex.net
908	INTAC Access Corporation	800	504	6822	info@intac.com
908	Internet Online Services	201	928	1000 x226	help@ios.com
909	Delta Internet Services	714	778	0370	info@deltanet.com
909	KAIWAN Corporation	714	638	2139	info@kaiwan.com
909	Lightside, Inc.	818	858	9261	info@lightside.com
910	Vnet Internet Access	704	334	3282	info@vnet.net
912	Internet Atlanta	404	410	9000	info@atlanta.com
913	DataBank, Inc.	913	842	6699	info@databank.com
914	Cloud 9 Internet	914	682	0626	info@cloud9.net
914	Computer Solutions by Hawkinson	914	229	9853	info@mhv.net
914	Creative Data Consultants (SILLY.COM)	718	229	0489	info@silly.com
914	DataBank, Inc.	913	842	6699	info@databank.com
914	INTAC Access Corporation	800	504	6822	info@intac.com
914	I-2000 Inc.	516	867	6379	info@i-2000.com
914	InteleCom Data Systems, Inc.	401	885	6855	info@ids.net
914	Phantom Access Technologies, Inc.	212	989	2418	bruce@phantom.com
914	The Pipeline Network	212	267	2626	info-info@pipeline.com
914	TZ-Link	914	353	5443	drew@j51.com
914	WestNet Internet Services	914	967	7816	info@westnet.com
915	Internet Access of El Paso	915	533	1525	rosteen@itsnet.com
915	New Mexico Technet, Inc.	505	345	6555	granoff@technet.nm.org
916	Great Basin Internet Services	702	829	2244	info@greatbasin.com
916	NETCOM On-Line Communications Services	408	554	8649	info@netcom.com
916	Sierra-Net	702	831	3353	giles@sierra.net
916	West Coast Online	707	586	3060	info@calon.com
916	Zocalo Engineering	510	540	8000	info@zocalo.net
918	Galaxy Star Systems	918	835	3655	info@galstar.com
918	Internet Oklahoma	918	583	1161	info@ionet.net
919	NETCOM On-Line Communications Services	408	554	8649	info@netcom.com
919	Vnet Internet Access	704	334	3282	info@vnet.net

FOREIGN

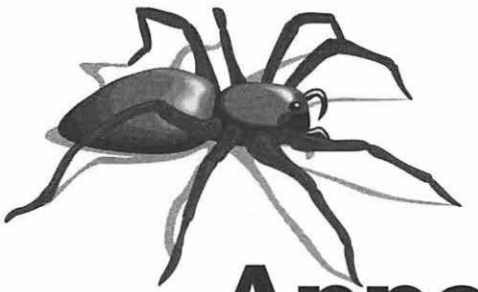
A listing of Internet service providers in countries other than the U.S. and Canada, sorted by country. Fields are country, service provider name, voice phone number, and eMail address for more information.

Australia	APANA	+61 42 965015	wollongong@apana.org.au
Australia	Apanix Public Access	+61 8 373 5575	admin@apanix.apana.org.au
Australia	arrakis.apana.org.au	+61 8 296 6200	greg@arrakis.apana.org.au
Australia	AusNet Services Pty Ltd	+61 2 241 5888	sales@world.net
Australia	Byron Public Access	+61 18 823 541	admin@byron.apana.org.au
Australia	DIALix Services	+61 2 948 6995	justin@sydney.dialix.oz.au
Australia	FidoNet Zone 3 Gateway	+61 3 793 2728	info@csource.pronet.com
Australia	Hunter Network Association	+61 49 621783	mbrown@hna.com.au
Australia	iiNet Technologies	+61 9 3071183	iinet@iinet.com.au
Australia	Kralizec Dialup Unix System	+61 2 837 1397	nick@kralizec.zeta.org.au
Australia	Informed Technology	+61 9 245 2279	info@it.com.au
Australia	The Message eXchange Pty Ltd	+61 2 550 5014	info@tmx.com.au
Australia	Microplex Pty. Ltd.	+61 2 888 3685	info@mpx.com.au
Australia	Pegasus Networks Pty Ltd	+61 7 257 1111	fwhitmee@peg.apc.org
Australia	PPIT Pty. Ltd. (059 051 320)	+61 3 747 9823	info@ppit.com.au
Australia	Winthrop Technology	+61 9 380 3564	wthelp@yarrow.wt.uwa.edu.au
Austria	EUnet EDV	+43 1 3174969	info@austria.eu.net
Austria	Hochschuelerschaft...	+43 1 586 1868	sysop@link-atu.comlink.apc.org
Austria	PING EDV	+43 1 3194336	info@ping.at
Bashkiria	UD JV 'DiasPro'	+7 3472 387454	iskander@diaspro.bashkiria.su
Belarus	Open Contack, Ltd.	+7 172 272127	admin@brc.minsk.by
Belgium	EUnet Belgium NV	+32 16 236099	info@belgium.eu.net
Belgium	Infoboard Telematics SA	+32 2 475 25 31	ocaeymaex@infoboard.be
Belgium	INnet NV/SA	+32 14 319937	info@inbe.net
Belgium	KnoopPunt VZW	+32 9 2333 686	support@knooppunt.be
Bulgaria	EUnet Bulgaria	+359 52 259135	info@bulgaria.eu.net
Crimea	Crimea Communication Centre	+7 652 257214	sem@snail.crimea.ua
Denmark	DKnet / EUnet Denmark	+45 3917 9900	info@dknet.dk
England	Compulink (CIX Ltd)	+44 181 390 8446	cixadmin@cix.compulink.co.uk
England	CONNECT - PC User Group	+44 181 863 1191	info@ibmpcug.co.uk
England	Demon Internet Services Ltd.	+44 81 349 0063	internet@demon.co.uk
England	The Direct Connection	+44 81 313 0100	helpdesk@dircon.co.uk
England	EUnet GB	+44 1227 266466	sales@britain.eu.net
England	ExNet Systems Ltd.	+44 81 244 0077	info@exnet.com
England	GreenNet	+44 71 713 1941	support@gn.apc.org
England	Lunatech Research	+44 1734 791900	info@luna.co.uk
England	Sound & Visions BBS	+44 1932 253131	info@span.com
England	Specialix	+44 932 3522251	keith@specialix.co.uk
England	WinNET (UK)	+44 181 863 1191	info@win-uk.ne
Finland	Clinet Ltd	+358 0 437 5209	clinet@clinet.fi
Finland	EUnet Finland Ltd.	+358 0 400 2060	helpdesk@eunet.fi
France	French Data Network	+33 1 47975873	info@fdn.org
France	OLEANE	+33 1 43283232	info-internet@oleane.net
Georgia	Mimosi Hard	+7 8832 232857	kisho@sanet.ge
Germany	EUnet Germany GmbH	+49 231 972 2222	info@germany.eu.net
Germany	Individual Network e.V.	+49 441 980 8556	in-info@individual.net
Germany	INS Inter Networking Systems	+49 2305 356505	info@ins.net
Germany	MUC.DE e.V.	+49 89 324 683 0	postmaster@muc.de
Germany	PFM News & Mail Xlink POP	+49 171 331 0862	info@pfm.pfm-mainz.de
Germany	SpaceNet GmbH	+49 89 324 683 0	info@space.net
Greece	Ariadne	+30 1 651 3392	dialup@leon.nrcps.ariadne-t.gr
Greece	Foundation of Research	+30 81 221171	forthnet-pr@forthnet.gr
Hong Kong	Hong Kong SuperNet	+852 358 7924	trouble@hk.super.net
Iceland	SURIS / ISnet	+354 1 694747	isnet-info@isnet.is

Ireland	Ieunet Limited	+353 1 679 0832	info@ieunet.ie
Ireland	Ireland On-Line	+353 91 592727	info@iol.ie
Israel	Elronet	+972 313534	info@elron.net
Israel	NetVision LTD.	+972 550330	info@netvision.net.il
Italy	ITnet S.p.A.	+39 10 6563324	info@it.net
Japan	Global OnLine, Japan	+81 3 5330 9380	hahne@acm.org
Japan	Internet Initiative Japan	+81 3 3580 3781	info@iij.ad.jp
Japan	M.R.T., Inc.	+81 3 3255 8880	sysop@janis-tok.com
Japan	People World Ltd.	+81 3 5661 4130	18005044@people.or.jp
Japan	TWICS	+81 3 3351 5977	info@twics.com
Japan	Typhoon Inc.	+81 3 3757 2118	info@typhoon.co.jp
Kazakhstan	Bogas Soft Laboratory Co.	+7 322 262 4990	pasha@sl.semsk.su
Kuwait	Gulfnet Kuwait	+965 242 6729	john@gulfa.ods.gulfnet.kw
Latvia	LvNet-Teleport	+371 2 551133	vit@riga.lv
Latvia	Versia Ltd.	+371 2 417000	postmaster@vernet.lv
Lisboa	Esoterica	716 2395	info@esoterica.com
Luxemburg	EUnet Luxemburg	+352 47 02 61 361	info@luxemburg.eu.net
Netherlands	The Delft Connection	+31 15560079	info@void.tdcnet.nl
Netherlands	Hobbynet	+31 365361683	henk@hgatenl.hobby.nl
Netherlands	Internet Access Foundation	+31 5982 2720	mail-server@iafnl.iaf.nl
Netherlands	NEST	+31 206265566	info@nest.nl
Netherlands	NetLand	+31 206943664	info@netland.nl
Netherlands	NLnet (EUnet)	+31 206639366	info@nl.net
Netherlands	Psyline	+31 80445801	postmaster@psyline.nl
Netherlands	Simplex Networking	+31 206932433	skelmir@simplex.nl
Netherlands	Stichting XS4ALL	+31 206225222	helpdesk@xs4all.nl
New Zealand	Actrix Networks Limited	+64 4 389 6356	john@actrix.gen.nz
New Zealand	Efficient Software Limited	+64 3 4738274	bart@dunedin.es.co.nz
Norway	Oslonett A/S	+47 22 46 10 99	oslonett@oslonett.no
Romania	EUnet Romania SRL	+40 1 312 6886	info@romania.eu.net
Russia	GlasNet	+7 95 262 7079	support@glas.apc.org
Russia	InterCommunications Ltd.	+7 8632 620562	postmaster@icomm.rnd.su
Russia	N&K Company	+7 86622 72167	serge@nik.nalchik.su
Russia	NEVAlink Ltd.	+7 812 592 3737	serg@arcom.spb.su
Russia	Relcom CO	+7 95 194 25 40	postmaster@ussr.eu.net
Russia	SvjazInform	+7 351 265 3600	pol@rich.chel.su
Slovakia	EUnet Slovakia	+42 7 725 306	info@slovakia.eu.net
Slovenia	NIL, System Integration	+386 61 1405 183	info@slovenia.eu.net
South Africa	Aztec	+27 21 419 2690	info@aztec.co.za
South Africa	Internet Africa	+27 0800 020003	info@iaccess.za
South Africa	The Internet Solution	+27 11 789 6071	info@is.co.za
Switzerland	SWITCH	+41 1 268 1515	postmaster@switch.ch
Switzerland	XGP Switzerland	+41 61 8115635	service@xgp.spn.com
Tataretan	KAMAZ Incorporated	+7 8439 53 03 34	postmaster@kamaz.kazan.su
Ukraine	ConCom, Ltd.	+7 0572 27 69 13	igor@ktts.kharkov.ua
Ukraine	Electronni Visti	+7 44 2713457	info%elvisti.kiev.ua@kiae.su
Ukraine	PACO Links Int'l Ltd.	+7 48 2200057	info@vista.odessa.ua
Ukraine	UkrCom-Kherson Ltd	+7 5522 64098	postmaster@ukrcom.kherson.ua

This list was prepared by:

Celestin Company
1152 Hastings Avenue
Port Townsend, WA 98368
United States of America
voice: 360-385-3767
fax: 360-385-3586
eMail: **celestin@olympus.net**



Appendix C

NETSCAPE EXTENSIONS TO HTML

There's been much ado about Netscape Communications' "extensions" to HTML. It's interesting to watch the back and forth between a highly successful and innovative commercial concern and the traditional Internet standards people. It's not just with HTML that Netscape is pushing the envelope and creating a stir either. The whole approach that Netscape Communications is taking with its secure HTTP server and how it differs from the secure HTTP approach being adopted by others is another story unto itself.

We'll try to keep you up to date on the WebMaster Mac WWW server, but please realize, it's a daunting task to try to keep up with it all.

At the time of this writing, Netscape had just released version 1.1b3 of its client and it was reported in many magazines and periodicals that three out of four Web surfers were using Netscape. Go to the Netscape Communications' WWW server sometime for the whole story. You need to keep up with what they're up to. They are among the most innovative folks on the Net. If you want to get a handle on how they approach HTML and HTTP standards there's a great URL to check out at:

http://www.netscape.com/info/open-standards.html

Below are the extensions to HTML that Netscape first broke ground with around the time that HTML 2.0 was being set in stone. HTML 3.0 is in the works now and once again Netscape is pushing the envelope with some innovative uses of HTML. The latest round of extensions and client features as of this writing includes the ability to do tables, backgrounds, and animation. You can find the latest examples of those at **http://home.netscape.com/home/demo/1.1b1/.**

According to the folks at Netscape Communications, "extensions to HTML take the form of additional tags and attributes added to the HTML specification and are specifically designed not to break existing WWW browsers." The following info is right off their WWW server at **http://www.netscape.com/home/services_docs/html-extensions.html** and is reprinted with the permision of Netscape Communications Corporation. Here is the page off their server with all the tags.

<ISINDEX>

To the ISINDEX element we have added the PROMPT tag. ISINDEX indicates that a document is a searchable index. PROMPT has been added so the document author can specify what message they want to appear before the text input field of the index. The default is of course that unfortunate message: This is a searchable index. Enter search keywords:

<HR>

The HR element specifies that a horizontal rule of some sort (the default being a shaded engraved line) be drawn across the page. To this element we have added 4 new tags to allow the document author some ability to describe how the horizontal rule should look.

<HR SIZE=number>

The SIZE tag lets the author give an indication of how thick they wish the horizontal rule to be.

<HR WIDTH=number|percent>

The default horizontal rule is always as wide as the page. With the WIDTH tag, the author can specify an exact width in pixels, or a relative width measured in percent of document width.

<HR ALIGN=left|right|center>

Now that horizontal rules do not have to be the width of the page we need to allow the author to specify whether they should be pushed up against the left margin, the right margin, or centered in the page.

<HR NOSHADE>

Finally, for those times when you really want a solid bar, the NOSHADE tag lets you specify that you do not want any fancy shading of your horizontal rule.

<UL>

Your basic bulleted list has a default progression of bullet types that changes as you move through indented levels, from a solid disk to a circle to a square. We have added a TYPE tag to the UL element so no matter what your indent level you can specify whether you want a TYPE=disc, TYPE=circle, or TYPE=square as your bullet.

<OL>

Your average ordered list counts 1, 2, 3, ... etc. We have also added the TYPE tag to this element to allow authors to specify whether they want their list items marked with capital letters (TYPE=A), small letters (TYPE=a), large roman numerals (TYPE=I), small roman numerals (TYPE=i), or the default numbers (TYPE=1).

For lists that wish to start at values other than 1 we have the new tag START. START is always specified in the default numbers, and will be converted based on TYPE before display. Thus START=5 would display either an 'E', 'e', 'V', 'v', or '5' based on the TYPE tag.

<LI>

To give even more flexibility to lists, we thought it would be nice if the author could change the list type, and for ordered lists the list count index as they progressed. To this end we added the TYPE tag to the LI element as well. It takes the same values as either UL or OL depending on the type of list you are in, and it changes the list type for that item and all subsequent items. For ordered lists we have also added the VALUE element so you can change the count for that list item and all subsequent items.

<IMG>

The IMG tag is probably the most extended tag:

<IMG ALIGN=left|right|top|texttop|middle|absmiddle|baseline| bottom|absbottom>

The additions to your ALIGN options need a lot of explanation. First, the values "left" and "right". Images with those alignments are an entirely new floating image type. An ALIGN=left image will float down and over to the left margin (into the next available space there), and subsequent text will wrap around the right-hand side of that image. Likewise, for ALIGN=right the image aligns with the right margin, and the text wraps around the left.

The rest of the align options are my way of trying to correct for the horrible errors I made when first implementing the IMG tag, without destroying the look of existing documents. ALIGN=top does just what it always did, which is align itself with the top of the tallest item in the line. ALIGN=texttop does what many people thought top should do, which is align itself with the top of the tallest text in the line (this is usually but not always the same as ALIGN=top). ALIGN=middle does just what it always did, it aligns the baseline of the current line with the middle of the image. ALIGN=absmiddle does what middle should have done, which is align the middle of the current line with the middle of the image. ALIGN=baseline aligns the bottom of the image with the baseline of the current line. ALIGN=bottom does just what it always did (which is identical to ALIGN=baseline but baseline is a better name). ALIGN=absbottom does what bottom should have done, which is align the bottom of the image with the bottom of the current line.

<IMG WIDTH=value HEIGHT=value>

The WIDTH and HEIGHT tags were added to IMG mainly to speed up display of the document. If the author specifies these, the viewer of their document will not have to wait for the image to be loaded over the network and its size calculated.

<IMG BORDER=value>

This lets the document author control the thickness of the border around an image displayed. Warning: setting BORDER=0 on images that are also part of anchors may confuse your users as they are used to a colored border indicating an image is an anchor.

<IMG VSPACE=value HSPACE=value>

For the floating images it is likely that the author does not want them pressing up against the text wrapped around the image. VSPACE controls the vertical space above and below the image, while HSPACE controls the horizontal space to the left and right of the image.

With the addition of floating images, we needed to expand the BR tag. Normal BR still just inserts a line break. We have added a CLEAR tag to BR, so CLEAR=left will break the line, and move vertically down until you have a clear left margin (no floating images). CLEAR=right does the same for the right margin, and CLEAR=all moves down until both margins are clear of images.

NEW ELEMENTS

<NOBR>

The NOBR element stands for NO BReak. This means all the text between the start and end of the NOBR elements cannot have line breaks inserted between them. While NOBR is essential for those odd character sequences you really don't want broken, please be careful; long text strings inside of NOBR elements can look rather odd.

<WBR>

The WBR element stands for Word BReak. This is for the very rare case when you have a NOBR section and you know exactly where you want it to break; also, any time you want to give the Netscape Navigator help by telling it where a word is allowed to be broken. The WBR element does not force a line break (BR does that); it simply lets the Netscape Navigator know where a line break is allowed to be inserted if needed.

<FONT SIZE=value>

Surprise! You can change the font size. Valid values range from 1–7. The default font size is 3. The value given to size can optionally have a '+' or '-' character in front of it to specify that it is relative to the document basefont. The default basefont is 3, and can be changed with the BASEFONT element.

<BASEFONT SIZE=value>

This changes the size of the BASEFONT that all relative font changes are based on. It defaults to 3, and has a valid range of 1–7.

<CENTER>

You aren't dreaming, yes, you can center your text. All lines of text between the begin and end of CENTER are centered between the current left and right margins. A new tag has been introduced rather than using the proposed <P Align="center"> because using <P Align="center"> breaks many existing browsers when the <P> tag is used as a container. The <P Align="center"> tag is also less general and does not support all cases where centering may be desired.

BEHAVIORAL CHANGES

Font attributes are now properly cumulative. Text inside something like **<tt>Text here</tt>** will be italic fixed bold text of size 6.

The Netscape Navigator should now properly deal with the awful HTML comment sequence. This should be:

<!-- Comment here -->

These comments can include other elements, and thus be used to quickly comment out large chunks of markup.

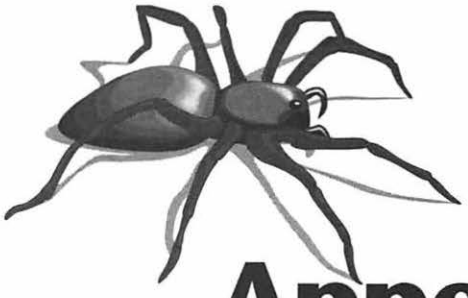
Line breaking is a little more under control now. Unless specified with a formatting element, lines can only be broken where empty space occurs in the original document. This means any spaces, tabs, or new lines. You should never again have the sequence <A HREF=url>Anchor here</A>. broken between the highlighted anchor and the period.

New Entities

In addition to the usual and escaped entities:

® -> Registered Trademark -> ®

© -> Copyright -> ©



Appendix D

Entities & ISO Latin-1 Character Entities

Here's some more HTML you may need from time to time to do the unusual. We thought you'd like to have it all in a couple of pages. Have at it!

Entities

The following entity names are always prefixed by ampersand (&) and followed by a semicolon, as shown.

<

The less than sign, <

>

The greater than sign, >

&

The ampersand sign, &

"

The double quote sign, "

** **

A nonbreaking space

ISO LATIN-1 CHARACTER ENTITIES

This list is derived from **ISO 8879:1986//ENTITIES.**

Æ

capital AE diphthong (ligature), Æ

Á

capital A, acute accent, Á

Â

capital A, circumflex accent , Â

À

capital A, grave accent, À

Å

capital A, ring, Å

Ã

capital A, tilde, Ã

Ä

capital A, dieresis or umlaut mark, Ä

Ç

capital C, cedilla, Ç

Ð

capital Eth, Icelandic

É

capital E, acute accent, É

Ê

capital E, circumflex accent, Ê

È

capital E, grave accent, È

Ë

capital E, dieresis or umlaut mark, Ë

Í

capital I, acute accent, Í

Î

capital I, circumflex accent, Î

Ì

capital I, grave accent, Ì

&luml;

capital I, dieresis or umlaut mark, Ï

Ñ

capital N, tilde, Ñ

Ó

capital O, acute accent, Ó

Ô

capital O, circumflex accent, Ô

Ò

capital O, grave accent, Ò

Ø

capital O, slash, Ø

Õ

capital O, tilde, Õ

Ö

capital O, dieresis or umlaut mark, Ö

Þ

capital THORN, Icelandic

Ú
capital U, acute accent, Ú

Û
capital U, circumflex accent, Û

Ù
capital U, grave accent, Ù

Ü
capital U, dieresis or umlaut mark, Ü

Ý
capital Y, acute accent

á
small a, acute accent, á

â
small a, circumflex accent, â

æ
small ae diphthong (ligature), æ

à
small a, grave accent, à

å
small a, ring, å

ã

small a, tilde, ã

ä

small a, dieresis or umlaut mark, ä

ç

small c, cedilla, ç

é

small e, acute accent, é

ê

small e, circumflex accent, ê

è

small e, grave accent, è

ð

small eth, Icelandic

ë

small e, dieresis or umlaut mark, ë

í

small i, acute accent, í

î

small i, circumflex accent, î

ì

small i, grave accent, ì

ï

small i, dieresis or umlaut mark, ï

ñ

small n, tilde, ñ

ó

small o, acute accent, ó

ô

small o, circumflex accent, ô

ò

small o, grave accent, ò

ø

small o, slash, ø

õ

small o, tilde, õ

ö

small o, dieresis or umlaut mark, ö

ß

small sharp s, German (sz ligature)

þ

small thorn, Icelandic

ú

small u, acute accent, ú

û

small u, circumflex accent, û

ù

small u, grave accent, ù

ü

small u, dieresis or umlaut mark, ü

ý

small y, acute accent

ÿ

small y, dieresis or umlaut mark, ÿ

Appendix E

WebMaster Applications and Tools

The list of apps and tools out there for Mac WebMasters grows each week. It's a full-time job waiting for somebody who can find an employer. The following list is as of May 1, 1995. You can find this list and links to all the latest programs on the WebMaster Mac WWW server at **http://www.webmastermac.com**. The ones with an * before their names are included on the CD-ROM that comes with this book.

* WebSTAR™

WebSTAR is the application that turns your Macintosh into a World Wide Web server. STARNINE is the company that markets the WebSTAR family of products and there are a couple of ways they have packaged things. WebSTAR PS (formerly known as MacHTTP) is included on the CD-ROM that came with this book. For the latest info and pricing on the entire WebSTAR family of products, point your browser to **http://www.starnine.com**.

Cost:	To be announced
Developer:	Chuck Shotton
eMail:	**cshotton@biap.com**

GRAPHICS TOOLS

clip2gif 0.4

clip2gif is a freeware utility that converts PICTs to GIF or JPEG files. It includes manual and batch conversion of PICTs, direct conversion from the clipboard, and is scriptable. Output options include output size, pixel depth, and transparency, gray shades only, and JPEG compression.

Cost:	Freeware
Developer:	Yves Piguet
eMail:	**piguet@ia.epfl.ch**

*GraphicConverter 2.1

GraphicConverter can import more than 50 different graphic formats and export 19, including transparent and interlaced GIFs, and can do this to individual files or batches. It contains a few tools for image manipulation and comes with copious online documentation. The program's architecture provides for plug-ins and the download .sit includes samples of the source code. Very well written and easily worth the shareware fee.

Cost:	$35
Developer:	Thorsten Lemke
eMail:	**thorsten_lemke@pe.maus.de**

GIFConverter 2.3.7

GIFConverter is also capable of importing several graphic formats. It's a simpler program than Graphic Converter but also very well written. Supports transparent and interlaced GIFs.

Cost:	$40
Developer:	Kevin Mitchell
eMail:	**74017.2573@compuserve.com**

* Transparency

The info window for Transparency states "A quick and dirty utility for creating transparent GIF images." That's what it is, that's what it does. Anyone doing graphics for the Web should have this; it's the quickest way to make one or two GIFs transparent.

Cost:	Freeware
Developer:	Aaron Giles
eMail:	**giles@med.cornell.edu**

MOVIE AND VIDEO TOOLS

AVI->QuickTime

An application and extension for converting AVI formatted files to QuickTime movies.

Cost:	Freeware
Developer:	We don't have a clue. Let us know if you do.

flattenMooV

A utility to flatten movies for use on other platforms. A flattened file is one in which all media references are contained in one file. It uses QuickTime FlattenMovie to move the mooV resource into the data fork of the MooV file.

Cost:	Freeware
Developer:	Robert Hennessy
eMail:	**70363.2164@compuserve.com**

PhotoCapture 1.1

A scriptable utility that allows you to capture video source images and save them to PICT files. Using a combination of PhotoCapture, clip2gif, and AppleScript will allow you to have up-to-the-minute images on your Web pages.

Cost:	$15
Developer:	Scott Gruby
eMail:	**Scott_Gruby@hmc.edu**

SOUND TOOLS

SoundSmith

A great application for converting and capturing sound files.

Cost:	Distribution is limited to Ziff Communications online services and electronic publishing projects
Developer:	Jeff Moore
eMail:	**JCM10@aol.com**

SoundEdit 16

The best commercial sound tool for creating professional-quality audio and converting sound and video to Windows formats and back again.

Cost:	$275.00–$300.00

HOTLIST UTILITIES

Hotlist2HTML 0.7.2

Quick and basic program that does just what it says. It will convert a hotlist from NCSA's Mosaic or EINet's MacWeb into an HTML document, which can in turn be imported into Netcom's Netscape as a Bookmark. (Requires System 7 or greater.)

Cost:	Freeware
Developer:	Lutz Weimann
eMail:	**weimann@zib-berlin.de**

HotList Sorter 1.0b1

Sorts a Mosaic or MacWeb hotlist.

Cost:	Freeware
Developer:	Dave Ransen
eMail:	**dransen@gate.net**

SERVER UTILITIES

Annotate

Allows clients to add and remove their own comments and remarks on a WebSTAR Web site. Useful for discussions a lá USENET.

Cost:	Freeware
Developer:	Phil Harvey
eMail:	**phil@nsun.phy.queensu.ca**

AutoWeb 1.0b

A set of scripts that allows a group of Macintosh users on a network to send raw text to a server and have it automatically formatted for the Web. Allows Internet publishing by large groups of people who don't want to learn HTML. The documentation uses the analogy of automated newsroom systems. Groupware for the Web!

Cost:	$100 site license fee for a public server. (Comes with a free script editor, but to run the scripts, you need Frontier Runtime 3.0, which is $35 per machine.)
Developer:	UserLand Software (Dave Winer)
eMail:	dwiner@well.com

Cron 1.0d16

The Cron clock daemon executes commands at specified dates and times according to the instructions in the **crontab** file. It basically allows to you to set any function on a Macintosh on a timer, such as launching WebStat. (Can it launch the coffe maker?) Absolutely essential.

Cost:	Freeware
Developer:	Chris W. Johnson
eMail:	**chrisj@mail.utexas.edu**

FTPd 2.3

A Macintosh implementation of the UNIX FTP and Gopher protocol. Allows any Mac on a network to be used as an FTP server. Supports MacBinary and BinHex transfers, and the "MACB" FTP command. FTPd can be run either as a foreground app or as background only.

Cost:	$10, single user; $500 business and university license; $2000 worldwide license; $5000 worldwide source code license
Developer:	Peter Lewis
eMail:	**peter@mail.peter.com.au**

* TR-WWW (Total Research World Wide Web)

TR-WWW is a search engine that will allow browsers on your site to search text documents, extract information, generate reports, and integrate it into a database. Runs native on 68k and PowerMacs.

Cost:	Educational/Non-Profit/Internal use: $50 single license, $500 site license. Commercial/For Profit use: $300 single license, $3000 site license. Annual renewal fee of 30% includes updates and upgrades
Developer:	Adrian Vanzal
eMail:	**adrian@medlan.med.monash.edu.au**

* WebStat

WebStat reads your Web server's .LOG file and produces a summary file. The default format for the summary file is HTML, but it can be reconfigured. It can be launched automatically with the cron utility, for example, and you can retrieve your .LOG files from remote computers.

Cost:	Freeware
Developer:	Phil Harvey
eMail:	**phil@nsun.phy.queensu.ca**

TEXT/HTML EDITORS

HTML Pro 1.5

An interesting approach to HTML editing using two windows, one for the source code and one for a fairly accurate WYSIWYG (HTML) display of the document. We couldn't get the HTML window's grabber bar to show up on my monitor, though. It was alway up and to the left, so we could never see the entire window. Includes support for just about every Scandanavian character without having to type the escape codes.

Cost:	Freeware
Developer:	Niklas Frykholm
eMail:	**nisfrm95@student.umu.se**

HTML.edit 1.5b1

An HTML document editor with extensive documentation. A stand-alone application written in HyperCard.

Cost:	Freeware
Developer:	Murray M. Altheim
eMail:	**murray.altheim@hq.nasa.gov**

HTML SuperText 2.0.1

A well-written editor that uses a color and font style scheme to show links and codes.

Cost:	Freeware
Developer:	Robert C. Best III
eMail:	**robert.best@potsdam.edu**

*HTML Grinder v1.2

Grinder is a batch processor for HTML documents that allows you to do things like find and replace, build indices, and build sequential links. You can expand the program with plug-ins (called wheels), which vary in cost. Grinder itself is free and comes with the Find and Replace wheel fully functioning, and four or five other wheels in demo mode. Handy for moving a site from one machine to another, or if one of your favorite links moves to a different machine.

Cost:	Grinder is free, the plug-ins range in cost (all revert to demo-mode when copied to a new disk)
Developer:	Michael Herrick/Matterform Media
eMail:	**matterform@nets.com**

*BBEdit 3.1

An extremely robust text editor that is scriptable and includes features like Macintosh Drag and Drop, the ability to find differences between several documents, XTND translators, and a host of others. BBEdit is native on the PowerMac. While it is not an HTML editor per se, due to its system of extensions, HTML editing tools can be added to an already extremely useful program. A freeware version called BBEdit Lite is available on most major online services. It is fully functioning as a text editor, but has had several of the bells and whistles removed.

Cost:	$119 (quantity and student discounts available)
Developer:	Bare Bones Software
eMail:	**bbedit@world.std.com**

*BBEdit HTML Extensions 8

A set of extensions for BBEdit and BBEdit Lite that include the most frequently used HTML tags.

Cost:	Freeware
Developer:	Carles Bellver
eMail:	**bellverc@si.uji.es**

*BBEdit HTML Tools 1.3b2

Another set of extensions for BBEdit and BBEdit Lite. Not only includes the most popular tags, but has keyboard equivalents for all of them. A very nice function if you're already used to formatting from the keyboard, as in Word and QuarkXPress. Also has Balloon Help and allows the creation of custom markup routines. You can register with the author to receive updates automatically. Killer.

Cost:	Freeware
Developer:	Lindsay Davies
eMail:	**LD11@unix.york.ac.uk**

Hotlist to HTML Convert

A very simple HyperCard stack for converting a Mosaic or MacWeb hotlist to an HTML document.

Cost:	Free for non-commercial use; commercial use requires licensing from developer.
Developer:	Larry Aronson
eMail:	**rinaldi1@applelink.apple.com**

MAP TOOLS

Hyper MapEdit 1.0

A HyperCard stack for creating ISMAP map files for use with the NCSA httpd Web server. Requires HyperCard 2.0.

Cost:	Freeware
Developer:	Nathan Gasser
eMail:	**gasser@eniac.seas.upenn.edu**

MacMapMaker 1.0

A freestanding application written in HyperCard for mapping images.

Cost:	Freeware
Developer:	Dan Crockett
eMail:	**crockett@ncsa.uiuc.edu**

* WebMap 1.0.1

Probably the best of several utilities for creating CERN- and NCSA-compliant .map files from PICTs and GIFs.

Cost:	Freeware
Developer:	Rowland Smith
eMail:	**rowland@city.net**

imagemap.cgi 1.6

A Perl cgi utility for creating and using imagemap files. Requires ResEdit.

Cost:	Freeware
Developer:	Scott Atwood
eMail:	**atwood@cs.stanford.edu**

Mac-ImageMap 1.3

A cgi utility for using imagemap files. Requires MacHTTP 2.0 or higher, System 7, and WebMap.

Cost:	Freeware
Developer:	Lutz Weimann
eMail:	**weimann@zib-berlin.de**

* MapServe

A cgi application for serving imagemap files. Requires MacHTTP 2.0 or higher. Runs native on 68k or PowerMac machines.

Cost:	Freeware
Developer:	Kelly Campbell
eMail:	**camk@ksu.ksu.edu**

APPLESCRIPT ADDITIONS

*ACME Script Widgets

ACME Script Widgets is a fantastic collection of AppleScript Scripting Additions that includes Tokenize and Join List.

Cost:	$10, single user; $100, site license
Developer:	Wayne Walrath
eMail:	**acme@kagi.com**

*TCP/IP Scripting Addition

The TCP/IP Scripting Addition package allows Open Scripting Architecture (OSA) scripts — like AppleScript scripts — to execute commands related to TCP/IP.

Cost:	$20, single user; $200, site license for business and education; $500 per year if distributed as part of a commercial product.
Developer:	Wayne Walrath
eMail:	**acme@kagi.com**

CGIs: COMMON GATEWAY INTERFACE APPLICATIONS

* AppleSearch.acgi 1.5

AppleSearch.acgi provides a mechanism for users to make their AppleSearch information sources available to others on the World Wide Web. AppleSearch.acgi is implemented as a gateway (CGI) application for use with WebSTAR.

Cost:	Freeware (for use only with a licensed copy of WebSTAR)
Developers:	Chuck Shotton/Apple Computer

*ButlerLink/Web

ButlerLink/Web is from Everyware Development Corp. and is the ideal tool for allowing your Web users to search, enter, and modify data in your Butler SQL databases. Butler SQL is a client-server relational database management system that runs on any Macintosh or Power Macintosh computer. ButlerLink/Web is the tool you use to automatically create the Web pages that utilize Butler SQL as the backend database server.

Cost:	Freeware
Developer:	Mark Wickens
eMail:	**mark@everyware.com**

*DALgate.acgi

Connects WebSTAR to a DAL server, such as the Butler relational DBMS. It can return the result on-screen or by sending mail with Eudora. Users can make simple queries, insert, and delete data.

Cost:	Freeware
Developer:	Michael Bjorn
eMail:	**michael@wiz.sk.tsukuba.ac.jp**

*email.acgi

Email.cgi is a CGI script to be used on WebSTAR servers for the purposes of sending eMail from WWW browsers that are not mail-to-capable.

Cost:	Freeware
Developer:	Eric Lease Morgan
eMail:	**eric_morgan@ncsu.edu**

*Chuck Shotton's FileMaker Pro cgi

FMPro.cgi connects your WebSTAR server to FileMaker Pro and lets you add, delete, and find records in a FileMaker Pro database from a Web page.

Cost:	Freeware
Developer:	Chuck Shotton
eMail:	**cshotton@biap.com**

*Russell Owen's FileMaker.acgi

Russell's FileMaker.acgi also lets you connect your WebSTAR server to FileMaker Pro. Of the two, his is the better one and is one of the best and most popular CGIs for WebSTAR going.

Cost:	Freeware
Developer:	Russell Owen
eMail:	**owen@astro.washington.edu**

*Fortune Cookie.acgi

The Fortune CGI implements a random fortune cookie generator for WebSTAR. It's a fun little CGI that everyone should have.

Cost:	Freeware
Developer:	Rob Hafernik
eMail:	**shokwave@io.com**

*GIFserv

GIFserv is a neat little CGI that lets you serve up a series of .GIF files. With it you can do tiny animations and other neat tricks.

Cost:	Freeware
Developer:	Rick Cardona
eMail:	**rick_cardona@capmac.org**

*Mail Tools Gateway

Mail Tools is a CGI web gateway for use with your WebSTAR server. The gateway makes it easy for people to turn the mail "vacation" feature on and off and to turn mail "forwarding" on and off.

Cost:	Freeware
Developer:	John Norstad
eMail:	**j-norstad@nwu.edu**

*NetCloak

NetCloak is a WebSTAR add-on that allows you to create dynamic HTML documents. With NetCloak, you can show and hide portions of your HTML documents based on a variety of different criteria, as well as add dynamic information to your pages.

Cost:	$50, single user; $250, commercial site
Developer:	John O'Fallon
eMail:	**john@maxum.com**

*NetForms

NetForms is another add-on application that runs on your Mac WebSTAR WWW server. It allows information entered by users of your server to be automatically converted to formatted HTML documents, which can then be read by other Web clients. In other words, NetForms allows users to contribute to the information published on your Web server, instead of just retrieving it. You definitely should take a look at NetCloak and NetForms!!

Cost:	$50, single user; $250, commercial site
Developer:	John O'Fallon
eMail:	**john@maxum.com**

Appendix F

MAC VS UNIX AND THE PC

(The Mac OS Advantage: Apple's Workgroup Server Solution for WWW Publishing)

Sooner or later you're going to come up against some UNIX or Windows NT geek who scoffs at the Mac as a WWW server. Man, are they in for a rude awakening! Someday soon they are going to wake up to the fact that the Mac OS is the way to go.

In the meantime, you need some ammo to blow them away. With Apple's permission we've included the white paper written by Chuck Shotton and Louis Slothauber. It makes a compelling case for using the Mac OS as a WWW server. Apple has put together a fantastic WWW server built upon the Apple Workgroup Server hardware platform. If you are serious about setting up a high-performance WWW site that can handle a lot of traffic you should seriously consider the Web server bundle that Apple has put together. It's a great package that includes a CD-ROM with a boatload of cool software.

PUBLISHING ON THE WORLD WIDE WEB: THE MAC OS ADVANTAGE

Abstract: We describe why a Workgroup Server from Apple running the Mac OS and MacHTTP is the preferred hardware and software platform on which to build a World Wide Web site. We do this by contrasting the Mac OS–based Workgroup Servers from Apple with the alternatives (workstations running UNIX and PCs running Microsoft Windows). Our preference is founded on an analysis of each platform's price/performance value, security, reliability, flexibility, and ability to provide custom solutions.

WHAT IS THE WORLD WIDE WEB?

The World Wide Web (WWW) is a low-cost, low-overhead technology that enables people to publish electronically almost any information on the Internet—today's forerunner to the information superhighway. Catalogs, research papers, order forms, databases, discussion forums, multimedia demos, software archives, books, and periodicals—all can be published electronically and made available immediately to anyone, anywhere in the world. All it takes is a single computer, connected to the Internet—the possibilities are endless.

Organizations of any size can use the WWW to project a global presence by publishing their contact information, background, and product information. Businesses can rapidly publish catalogs and price lists without the costs and time required for paper publishing. They can even take orders and perform credit card transactions on line. Large corporations can use the WWW to securely connect the data resources of their various sites at a fraction of the conventional cost. Schools and universities can post curriculum information, research findings, campus activities, and resources to their campus community or to the world at large. Electronic publishing via the WWW has been called the 1990s equivalent of the desktop publishing revolution, and now is the time to get onboard. Why? Because the Internet has a global audience consisting of more than 25,000 connected networks, and more than 31 million users (and potential customers). The WWW is the fastest growing segment of the Internet, with a 5 to 10 percent monthly growth rate. The tools and infrastructure are in place and mature. Costs are low and the potential return is high, and growing.

CLIENT AND SERVER PROGRAMS

Two kinds of software are needed to make the WWW work: client programs and server programs. A client program provides you with access to the published content of the WWW, while a server program allows you to do the publishing. A number of client programs (or browsers) are available today, and new ones are always under development. Many client programs are free to the public (such as Mosaic, MacWeb, and

Netscape), and they will run on multiple hardware platforms. Just about anyone with access to a computer can tap into the WWW. On the server side, however, the ability to publish information is limited by the availability of software. Relatively few server programs exist, and each is designed specifically for one of three hardware platforms: UNIX-based workstations, Windows software–based PCs, and Mac OS–based systems.

Server Programs for UNIX-based Workstations

UNIX-based workstations are typically high-performance, high-cost machines. Because of the complexity of the UNIX operating system, these workstations require a significant amount of time to set up, and they need to be configured by a qualified UNIX system administrator. Usually this administrator is a full-time staff member whose sole job is to maintain—and modify—the system. UNIX-based workstations range in price from $10,000 to $30,000 (U.S.). When the cost of a full-time administrator is added, these workstations become quite expensive. The first WWW server programs were developed for UNIX-based workstations, and many commercial UNIX server programs are available today. Like the workstations on which they run, these WWW server programs typically require a great deal of effort to set up and maintain. In addition, they can only interact directly with text stream–based applications (such as UNIX shell scripts). The server program cannot interact with other applications (such as an SQL database, Lotus 1-2-3, and so on) running on the same workstation unless a custom C program is written to perform each requested interaction.

Server Programs for PCs Running Windows

Server programs for PCs running Windows suffer from all of the afflictions of server programs for UNIX-based workstations, because that is what they are: UNIX server programs that have been ported to Windows. However, because they do not have access to the capabilities provided by UNIX, these server programs are even more limited than the UNIX server programs. Server programs for PCs running Windows severely limit the number of simultaneous client connections, making them unsuitable for high-volume WWW sites. They have only a vestigial abil-

ity—via Perl scripts—to interact with most applications for Windows. And they have only minimal security support. Worst of all, because of known memory allocation problems, none of these server programs can remain in continuous use for more than a few days before crashing the host machine. They are not recommended for production use, and therefore we do not consider them further in this article.

MacHTTP, the Server Program for the Mac OS Platform

MacHTTP, the WWW server program for the Mac OS platform, provides the full functionality of commercial UNIX server programs and more, without any of the disadvantages. MacHTTP was designed to work in the Mac OS–based server environment, and is completely integrated with Macintosh® System 7.5; it is not a ported product. The software places a relatively small load on your computer, in terms of memory and processor requirements. MacHTTP runs in native mode on both 680x0-based Apple Macintosh systems and PowerPC processor–based systems.

Like most applications for Macintosh systems, MacHTTP can be installed, configured, and running in about five minutes by anyone; no system administration experience is required. MacHTTP can even be installed and set up remotely, over a network.

MacHTTP is easy to use and set up, even from a remote Macintosh system. It also provides an easy-to-understand interface for controlling access to your WWW server. MacHTTP version 2.0, the most recent version, offers many new features, including:

Common Gateway Interface (CGI) support for integration with external applications, such as databases or e-mail applications.

Security enhancements that support username and password security for all files it serves, in addition to MacHTTP access controls by domain name and IP address.

User interface enhancements, such as improved statistics reporting and saved preferences for window settings and menu selections.

Increased AppleEvent support to allow complete remote administration of MacHTTP from another Macintosh or any authorized WWW client.

MACHTTP SPECIFICATIONS

Native versions are available for both 680x0-based Macintosh computers and PowerPC processor–based systems

- Requires less than 1 megabyte of RAM
- Implements the HTTP/1.0 Standard
- Multithreaded transfers
- Compatible with other MacTCP® applications
- Completely scriptable and recordable

The Mac OS Advantage over the UNIX Platform

Once you have decided to publish information on the WWW, the next decision you need to make is which hardware and software to use. This decision should be based on performance, reliability, cost, security, customer support, and flexibility.

Performance

At first glance, you might think that UNIX-based workstations would be the clear winners in the performance arena over any Mac OS–based system. After all, a single UNIX workstation can outperform a Workgroup Server from Apple, in terms of simultaneous connections supported. However, comparing the performance of a Workgroup Server and a UNIX-based workstation is like comparing apples and oranges. To compare apples to apples, you must compare price and performance together. For example, compare an $11,000 (U.S.) investment in Workgroup Servers with the same investment in UNIX-based workstations. For that amount today, you can buy either three Workgroup Servers or one low-end UNIX-based workstation. The multiple Workgroup Servers, working together, will outperform the single workstation every time. A single Workgroup Server can serve 3,000 to 5,000 (or more) connections per hour, depending on the data being served and the network bandwidth. This number of connections is far greater than the number that the average WWW site receives in an entire day.

Reliability

For many potential WWW applications (such as on-line catalogs) downtime is not just an inconvenience—it costs money. While a Workgroup Server is, in general, as reliable as the typical UNIX-based workstation, you can buy several Workgroup Servers for less than the cost of a single UNIX-based workstation. If your one and only UNIX-based workstation bites the dust, you're off the air until it gets fixed. But if one of your several Workgroup Servers goes down, the rest are unaffected. The others can continue to publish your information and service your customers. Mac OS–based systems running as dedicated WWW servers have demonstrated the ability to run unattended for months at a time, while UNIX-based workstations performing the same functions are forced into periodic reboots by operating system bugs, kernel panics, and administrative downtime. A list of some of the most persistently available Mac OS–based servers on the Internet is available at the WWW address http://www.ape.com/. Many of these servers have been running since September 1994 with no downtime.

Cost

Workgroup Servers from Apple start at less than half the price of the lowest cost UNIX-based workstations, and they're easy to use and maintain. When you compare the price of a UNIX-based workstation plus the additional cost of a full-time UNIX system administrator with the price of a Workgroup Server, the Apple solution clearly provides more "bang for the buck."

With MacHTTP and a Workgroup Server from Apple, configuring your WWW server is simply a matter of double-clicking the MacHTTP icon. There's no need for a full-time system administrator to wade through complex installation requirements.

Security

A Workgroup Server running MacHTTP is much more secure than any UNIX-based workstation, because of the many documented security holes in the UNIX operating system. To provide even minimal security for a UNIX-based workstation often requires expensive or proprietary firewall software, in addition to an experienced UNIX system administrator.

On the other hand, a Workgroup Server connected to the Internet is not subject to unauthorized access (a big UNIX security problem). Furthermore, MacHTTP does not allow random access to the entire document tree. Only those files and documents that you want to publicize are available to the Internet community.

Support

Customer support for most UNIX-based WWW server programs is limited to a few on-line documentation files. With some commercial server programs, you can ask questions by e-mail, and they might be answered a few days later. In addition, a few Internet newsgroups are devoted to the topic, so other users can commiserate. In contrast, complete on-line support is provided for MacHTTP by BIAP Systems, including on-line documentation, answers to frequently asked questions, examples, tutorials, and personal support to registered users and clients. E-mail questions are usually answered within an hour, and always within 24 hours. And there's a large on-line community of MacHTTP users who share tips and information in a dedicated e-mail forum.

Flexibility

While UNIX-based server programs can interact only with text stream–based applications, MacHTTP has no such limitation. A number of interface utilities already exist to link MacHTTP to databases and text search engines, and to process credit card sales through the First Virtual Holdings system. In addition, MacHTTP can use AppleScript® (the built-in scripting language of the Mac OS) to interact with any other application designed for Macintosh systems (such as spreadsheets, word processors, and databases), including custom applications. No formal C programming experience is required to take advantage of these capabilities.

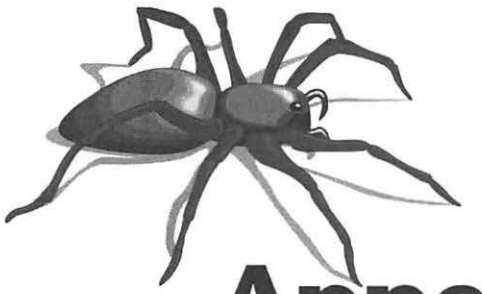
CONCLUSION

Now is the time to become a part of the World Wide Web, and the best way to do so is with Apple's Workgroup Servers and MacHTTP software. The desktop publishing revolution that started with the Macintosh computer will continue to evolve on the WWW, and Workgroup Servers from Apple provide the platform of choice.

Chuck Shotton
cshotton@biap.com
Chuck Shotton has been engineering software systems for the past 12 years. He's been involved in projects for defense and space programs, as well as for academic and medical institutions, and he is currently the president of BIAP Systems. Chuck is the author of MacHTTP, the WWW server for Mac OS–based systems.

Louis Slothouber, Ph.D.
lpslot@biap.com
Louis Slothouber specializes in artificial intelligence and programming language compilers with applications in intelligent agent systems. He received his Ph.D. from William and Mary in 1989, and has been collaborating with BIAP Systems since 1994.

Used with the permission of Apple Computer, Inc.



Appendix G

Announcing Your WWW Site and Then Some...

This is the last appendix. It may also be the most useful. This is where you'll find out how to announce your new WWW site and where'll your find the location of almost every Mac-related resource on the Net.

Extra special thanks (yet again!) to Carl de Cordova and Jon Wiederspan for helping us pull this all together. Thanks to them, this appendix contains just about every Mac and WebMaster Mac-related newsgroup, mailing list, WWW site, FTP site, FAQ, and journal in the universe. Or at least most of them.

Announcing Your Site

The best place to go to announce your site is Jon Wiederspan's WWW server at:

http://www.uwtc.washington.edu/Computing/WWW/AnnouncingYourSite.html

He's got a comprehensive list of pointers including the one to the Submit It! WWW Server at:

http://www.cen.uiuc.edu/~banister/submit-it/

If you go to the Submit It! WWW server and register your site there, you'll cover all the bases in one shot.

USEFUL WWW-RELATED NEWSGROUPS, MAILING LISTS AND WWW SITES FOR MAC WEBMASTERS

You can find the list of all the useful WWW newsgroups and mailing lists at:

http://www.uwtc.washington.edu/Computing/WWW/WebListsAndGroups.html.

Senior Wiederspan has done a nice job of pulling this all together.

Useful WWW Newsgroups

comp.infosystems.www.users provides a forum for discussion of WWW client software (such as Mosaic, Cello, and Lynx). New use questions, client setup questions, client bug reports, resource discovery questions on how to locate information on the WWW that can't be found in the FAQ, and comparisons between various client packages are among the acceptable topics for this group.

comp.infosystems.www.providers provides a forum for the discussion of WWW server software and the use of server software to provide information to users. General server design, setup questions, server bug reports, security issues, HTML page design, and other concerns of information providers are among the likely topics for this group.

comp.infosystems.www.misc provides a general forum for discussing WWW issues that are not covered by the other **comp.infosystems.www** groups.

comp.infosystems.announce is for announcement of new information services (e.g., new WWW sites) and new software products (new server software, new clients, new document convertors, etc.).

alt.hypertext has less traffic than the **comp.infosystems.www** groups but is also less "professional."

comp.infosystems.gopher provides a forum for discussing Gopher, but sometimes covers WWW issues.

comp.infosystems.wais covers WAIS topics, including integration of WAIS with WWW.

comp.text.sgml covers SGML, including HTML.

Useful WWW Mailing Lists

WWW-talk
General discussion about the World Wide Web.
Server: **listserv@info.cern.ch**
Send eMail with one line in the body containing:
subscribe www-announce your_name

www-html
Technical discussions of the HyperText Markup Language (HTML) and HTMLPlus designs.
Server: **listserv@info.cern.ch**
Send eMail with one line in the body containing:
subscribe www-html your_name

www-talk
Technical discussion for those developing WWW software or with that deep an interest.
Server: **listserv@info.cern.ch**
Send eMail with one line in the body containing:
subscribe www-talk your_name

www-rdb
Discussion of gatewaying relational databases into WWW.
Server: **listserv@info.cern.ch**
Send eMail with one line in the body containing:
subscribe www-rdb your_name

www-proxy
Technical discussion about WWW proxies, caching, and future directions.
Server: **listserv@info.cern.ch**
Send eMail with one line in the body containing:
subscribe www-proxy your_name

web-support
A Mailbase list that can be used for discussions about WWW issues.
Server: **mailbase@mailbase.ac.uk**
Send eMail with the one line in the body stating:
join web-support your_name

uctlig-infs
A Mailbase list that can be used for discussions about information systems, including WWW.
Server: **mailbase@mailbase.ac.uk**
Send eMail with the one line in the body stating:
join uctlig-infs your_name

unite
A Mailbase list that can be used for discussions about a User Network Interface To Everything.
UNITE is available as an archive.
Server: **mailbase@mailbase.ac.uk**
Send eMail with the one line in the body stating:
join unite your_name

Mac WWW Server Database and Consultants Directory

For a nice database of Mac WWW servers check out: **http://www.barnet.com:80/ape/machttp_talk/machttpservers.**

Finally, if you are in a bind and need help, check out the Mac WebMaster Consultants Directory at: **http://www.macweb.com/.**

THE BEST MAC-RELATED WWW SITE ON THE NET

You can always go to Apple's site at **http://www.info.apple.com** and take it from there. In fact, one of the best pages for Mac WebMasters to check out is Carl de Cordova's Web Dev Pointers page at:

http://atlantis.austin.apple.com/people.pages/carldec/web.dev.pointers.html

The hands down for the best all 'round Mac-related WWW site easily goes to the Well Connected Mac WWW Server. It is the best site on the Net for every Mac user — WebMmaster or not — and is maintained by Elliotte Rusty Harold of the Department of Mathematics, New Jersey Institute of Technology. You can find it at:

http://rever.nmsu.edu/~elharo/faq/Macintosh.html

There's an incredible index of Mac-related WWW sites at:

http://rever.nmsu.edu/~elharo/faq/web.html

And you'll find every mailing list under the sun at:

http://rever.nmsu.edu/~elharo/faq/mailinglists.html

For info on Mac-related newsgroups, surf to:

http://rever.nmsu.edu/~elharo/faq/newsgroups.html

For info about cool Mac FTP sites, check out:

http://rever.nmsu.edu/~elharo/faq/software.html

And for all the Mac FAQs you could possible desire, cruise to:

http://rever.nmsu.edu/~elharo/faq/faqs.html -

Finally, you'll find info about online Mac-related journals and periodicals at:

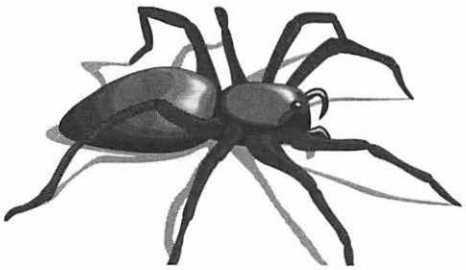
http://rever.nmsu.edu/~elharo/faq/periodicals.html -

Remember that you can always find your way to the Well Connected Mac and all the other places listed above from the WebMaster Mac WWWserver at **http://webmastermac.com**!

FINAL WORDS

As they say in LoonyTunes, "That's All Folks." We hope you've had as good a time with it as we have.

— Bob LeVitus & Jeff Evans



INDEX

B

C

D

E

F

G

H

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Name _____

Company _____

Address _____

City _____ State _____ ZIP Code _____

Phone _____ Fax _____

PPP Account Number: _____

One Free Month of UUNET's AlterDial Service

This *free* month of UUNET's AlterDial® service is a limited-time introductory offer, which allows WebMaster purchasers to receive AlterDial service free for one month, including up to fifteen (15) hours of connect time. Connect time in excess of the fifteen free hours will be billed at UUNET's current rates. UUNET's standard $25 start-up fee will still apply.

To establish service, you must either sign and return an AlterDial subscription form or autoregister for the service using our special software. For more information, or to receive the software, please call UUNET Technologies, Inc., at **+1 800 265 0499**. You must return this original coupon after receipt of your first bill to obtain the credit. Photocopies and facsimiles of this coupon are not acceptable. Your credit card may be billed; however, we will issue you a credit as soon as we receive this coupon.

Billing is automatic. You must cancel the service before the start of any additional months if you do not wish to continue service. Use of this coupon does not waive any of the standard terms and conditions found on the reverse side of the AlterDial subscription form.

This offer is restricted to **new** UUNET customers only. Only one coupon per customer will be accepted. This offer may be revised or rescinded at any time, and is valid only in the continental United States. Monthly service fees and hourly connection charges are subject to change per the standard AlterDial Terms and Conditions. This coupon expires July 31, 1996.

Name _____

Company _____

Address _____

City _____ State _____ ZIP Code _____

Phone _____ Fax _____

This coupon entitles WebMaster purchasers to a $100 discount off the normal start-up fee for UUNET's Web server hosting services.

$100 off UUNET's Web Server Hosting Start-up Fee

UUNET's Web server hosting services provide you with an easy and affordable way to establish and maintain your company's presence on the World Wide Web. Both our Standard and Premium services offer a low price for the first month that we host your server, and include usage reports showing activity for your server "pages." At your request, UUNET can also develop your server content, providing a fully-integrated service.

To establish service, you must complete a Web Hosting Subscription Form and a Web Hosting Configuration Form. When you receive your first invoice, you must attach this original coupon and return it to us to receive the $100 credit. Photocopies and facsimiles of this coupon are not acceptable. For more information or to sign up, please call UUNET Technologies at **+1 800 258 4039**.

This offer is restricted to **new** UUNET customers only. Only one coupon per customer will be accepted. This offer may be revised or rescinded at any time, and is valid only in the continental United States. This coupon expires July 31, 1996.

Supercharge your Web Site with WebSTAR™ and Save!

Save 20% on your purchase of WebSTAR™, the ultimate upgrade for WebSTAR™PS (formerly MacHTTP)

Also save on the Security and Commerce Toolkits for WebSTAR

WebSTAR™ is the ultimate upgrade for WebSTAR PS. Just like its shareware cousin, WebSTAR helps you publish hypertext documents to millions of Web users around the world, right from your Macintosh®. You can also use WebSTAR to put any Macintosh file on the Web, including GIF and JPEG images and even QuickTime™ movies. But with WebSTAR, you get these fantastic new features:

- WebSTAR is four times faster than WebSTAR PS
- Support thousands of connections per hour
- Greatly improved user interface
- Remote administration from anywhere on your network
- Control multiple servers from one Mac
- Completely extensible and AppleScriptable
- Publish information from any source
- Fully customizable log formatting to track server usage

Security and Electronic Commerce Today

The Security and Commerce Toolkits for WebSTAR are available separately. The Security Toolkit provides authentication and encryption using the Secure Sockets Layer to ensure that Web connections are completely private. The Commerce Toolkit adds support for electronic commerce to WebSTAR. See **http://www.starnine.com** for more information.

So order your copy of WebSTAR with the order form on the following page today! And supercharge your Web site...with WebSTAR!

WebSTAR™ Order Form

Name _____

Company _____

Address _____

City _____ **State/Province** _____ **Postal Code** _____

Country _____ **Phone** _____ **Fax** _____

Technical contact e-mail _____

Business contact e-mail _____

Special Offer Pricing for *WebMaster Mac* Readers

$635 Corporate ($795 SRP)
$235 Educational ($295 SRP)

Number of Corporate Licenses _____ x $635 = _____

Number of Educational Licenses _____ x $235 = _____

Postage and Handling (see details below)*: _____

SUBTOTAL: _____

Tax (California residents only): _____

TOTAL: _____

*Continental US = $8.00
Hawaii, Alaska, and Canada = $25.00
Other countries should contact Starnine Sales at 510-649-4949 or send e-mail to sales@starnine.com

Method of Payment (Visa ☐, Mastercard ☐, check ☐)

Credit Card Number _____ **Expiration Date** _____

Name as it appears on the card _____

Signature _____

Fax this order form to 510-548-0393 or mail to:

StarNine Technologies, Inc.
ATTN: Sales Dept.
2550 Ninth Street, Suite 112
Berkeley, CA 9471

If ordering by check, please make check payable to "StarNine."
To contact StarNine by phone, please call 1-800-525-2580.

LIMITED WARRANTY AND DISCLAIMER OF LIABILITY

ACADEMIC PRESS, INC. ("AP") AND ANYONE ELSE WHO HAS BEEN INVOLVED IN THE CREATION OR PRODUCTION OF THE ACCOMPANYING CODE ("THE PRODUCT") CANNOT AND DO NOT WARRANT THE PERFORMANCE OR RESULTS THAT MAY BE OBTAINED BY USING THE PRODUCT. THE PRODUCT IS SOLD "AS IS" WITHOUT WARRANTY OF ANY KIND (EXCEPT AS HEREAFTER DESCRIBED), EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, ANY WARRANTY OF PERFORMANCE OR ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE. AP WARRANTS ONLY THAT THE MAGNETIC DISC(S) ON WHICH THE CODE IS RECORDED IS FREE FROM DEFECTS IN MATERIAL AND FAULTY WORKMANSHIP UNDER THE NORMAL USE AND SERVICE FOR A PERIOD OF NINETY (90) DAYS FROM THE DATE THE PRODUCT IS DELIVERED. THE PURCHASER'S SOLE AND EXCLUSIVE REMEDY IN THE EVENT OF A DEFECT IS EXPRESSLY LIMITED TO EITHER REPLACEMENT OF THE CD-ROM DISC(S) OR REFUND OF THE PURCHASE PRICE, AT AP'S SOLE DISCRETION.

IN NO EVENT, WHETHER AS A RESULT OF BREACH OF CONTRACT, WARRANTY OR TORT (INCLUDING NEGLIGENCE), WILL AP OR ANYONE WHO HAS BEEN INVOLVED IN THE CREATION OR PRODUCTION OF THE PRODUCT BE LIABLE TO PURCHASER FOR ANY DAMAGES, INCLUDING ANY LOST PROFITS, LOST SAVINGS OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PRODUCT OR ANY MODIFICATIONS THEREOF, OR DUE TO THE CONTENTS OF THE CODE, EVEN IF AP HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY OTHER PARTY.

Any request for replacement of a defective CD-ROM disc must be postage prepaid and must be accompanied by the original defective disc, your mailing address and telephone number, and proof of date of purchase and purchase price. Send such requests, stating the nature of the problem, to Academic Press Customer Service, 6277 Sea Harbor Drive, Orlando, FL 32887, 1-800-321-5068. APP shall have no obligation to refund the purchase price or to replace a disc based on claims of defects in the nature or operation of the Product.

Some states do not allow limitation on how long an implied warranty lasts, nor exclusions or limitations of incidental or consequential damage, so the above limitations and exclusions may not apply to you. This Warranty gives you specific legal rights, and you may also have other rights which vary from jurisdiction to jurisdiction.

THE RE-EXPORT OF UNITED STATES ORIGIN SOFTWARE IS SUBJECT TO THE UNITED STATES LAWS UNDER THE EXPORT ADMINISTRATION ACT OF 1969 AS AMENDED. ANY FURTHER SALE OF THE PRODUCT SHALL BE IN COMPLIANCE WITH THE UNITED STATES DEPARTMENT OF COMMERCE ADMINISTRATION REGULATIONS. COMPLIANCE WITH SUCH REGULATIONS IS YOUR RESPONSIBILITY AND NOT THE RESPONSIBILITY OF AP.

About the CD-ROM

The *WebMaster Macintosh* WWW Site at **http://www.webmastermac.com** maintains current links to everything listed below.

Contents

Shareware Descriptions

WebSTAR™PS is the WWW server for the Macintosh developed by Chuck Shotton.

Jon Wiederspan's Tutorials are the definitive lessons that help you learn how to create CGI applications to extend the capabilities of your WebSTAR server.

The BBEdit Bare Bones Software Goodies includes BBEdit Lite 3.0 and BBEdit 3.1 Demo. BBEdit is considered by many Mac WebMasters to be the best HTML authoring tool.

BBEdit HTML extensions.8 lets you convert plain text files into HTML documents and insert HTML tags inside of them.

BBEdit HTML Tools v1.3b2 is another excellent set of HTML tools and extensions to BBEdit.

HTML Grinder 2.0 accesses special plug-in tools called "wheels" to modify your HTML pages. The Find and Replace Tool is especially useful.

GraphicConvertor 2.1 is an application that converts pictures to different formats. It contains many useful features for picture manipulation.

MapServe is the best tool to use to allow WebSTAR to serve clickable imagemaps.

WebMap 1.01. reads GIF and PICT files, and allows you to draw out the various graphic shapes that represent the hot areas on an html image map.

Transparency 1.0 is a utility for creating transparent GIF images.

AppleSearch.acgi 1.5 connects your WebSTAR server to AppleSearch and allows you to search and retrieve full text documents over the World Wide Web.

The ButlerLink/Web Package includes a ButlerLink/Web and a sample Butler SQL database. ButlerLink/Web is the tool you use to automatically create the web pages that utilize Butler SQL as the backend database server and allows your web users to search, enter, and modify data in your Butler SQL databases.

Chuck Shotton's FMPro CGI connects WebSTAR to FileMaker Pro and allows addtion, search, and deletion of FileMaker Pro database records over the World Wide Web.

Russel Owen's FMPro CGI does the same thing as Chuck's FMPro CGI—only better!

NetCloak is a WebSTAR add-on developed by John O'Fallon that allows you to create dynamic HTML documents. With NetCloak, you can show and hide portions of your HTML documents based on a variety of different criteria, as well as add dynamic information to your pages.

NetForms is another add-on application from John O'Fallon. It allows information entered by users of your server to be automatically converted to formatted HTML documents, which can then be read by other Web clients. NetForms allows users to contribute to the information published on your Web server, instead of just retrieving it.

Email.cgi allows users to send email from WWW browsers that are not mail-to-capable.

The Fortune Cookie CGI is a fun little CGI that implements a random fortune cookie generator for WebSTAR.

DALgate connects WebSTAR to a DAL server (such as the Butler relational DBMS) and returns results by sending mail with Eudora.

Mail Tools Gateway is a CGI that makes it easy for people to turn your email "vacation" feature on and off and to turn mail "forwarding" on and off.

TR-WWW 1.3 is a Macintosh-based search engine for use with the WebSTAR server software. It allows users to search and subsequently browse document collections using a forms-based interface.

GIFserv is a fun little CGI written by Rick Cardona that serves up animation with Netscape.

ACME Script Widgets is an important set of AppleScript Scripting Additions including Tokenize put together by Wayne Walrath.

TCP/IP Scripting Addition allows Open Scripting Architecture (OSA) scripts—like AppleScript scripts—to execute commands related to TCP/IP.

WebStat 2.3.4 is a great program used to summarize WebSTAR transmission statistics.

MacTCP Switcher is a very simple little program that makes it easy to save and quickly restore multiple MacTCP configurations.

PhotoCapture is a scriptable utility that allows you to capture images from a video source and save images to PICT files. Useful for helping serve up your QuickCam PICTs on your Web site.

WARNING:
Seal may not be broken prior to purchase.
If the seal on this pouch is broken,
product cannot be returned.

Macintosh / Internet / World Wide Web

"Tired of being a Web weenie? Well, fear no more as you read this book. It turns 98 pound Web weaklings into fire-breathing, cgi-toting, HTML-living WebMasters!!!" -Don Crabb, syndicated columnist and Macintosh crustacean-at-large

WebMaster Macintosh®

Bob LeVitus and Jeff Evans

AP PROFESSIONAL

This book is a fast and easy guide for turning your Macintosh into a World Wide Web site. With this book and the enclosed CD-ROM, you can create your own Web page in a matter of hours.

- A brief overview of the origin and rapid growth of the World Wide Web
- The nuts and bolts details you need to connect with your Internet provider
- Everything you need to know to write great HTML code for Web page design
- All the setup and configuration information you need for your Web server, including important details about security, clickable maps and buttons, client-server architecture, and CGIs (common gateway interfaces) and how they connect to common database applications
- Interviews with WebMasters such as Chuck Shotton, developer of WebSTAR™ (formerly MacHTTP)
- A WebMaster Web site that will keep you up to date with all the latest tools, applications, and Web resources at your fingertips
- A wealth of WebMaster information, such as useful Internet mailing lists, newsgroups, key Web site locations, and pointers to every Internet resource a WebMaster would ever need
- All the tips you need to maximize your Web site's performance, as well as how to announce it to the world

And if that's not enough... CD-ROM ENCLOSED

A ***CD-ROM*** is included with all the shareware you need to set up your Web site, including WebSTAR™PS server software, HTML editors, graphics tools, CGIs galore, scripting additions, and other essential goodies. ***PLUS*** money saving coupons!

About the Authors

Bob LeVitus has been a contributing editor/columnist for *MacUser Magazine* since 1989. He has written 12 popular computer books, including *Stupid Mac Tricks, Dr. Macintosh Second Edition*, and most recently *Macintosh System 7.5 for Dummies*. Bob is known not only for his expertise, but also his entertaining writing style. In his newest capacity, Bob is the Director of Evangelism for Power Computing Corporation, promoting their new line of Mac™ OS compatibles.

Jeff Evans is the President of VersaCom, a Texas-based on-line communications and Internet marketing firm. In his day-to-day responsibilities, Jeff implements his own system designs to the specifications of his on-line clients. In his designs he integrates a wide variety of state-of-the-art technologies including Internet connectivity, WWW services, SQL database solutions, and FirstClass (a network & communication software package).

System Requirements: Macintosh or compatible running system 7 or later
CD-ROM drive

Skill Level: Beginner to Intermediate, *but you'll be a WebMaster before you know it!*

ISBN 0-12-445574-3

90056
9 780124 455740