TEACHING A NEW DOG OLD TRICKS

A Macintosh-Based World Wide Web Starter Kit Featuring MacHTTP and Other Tools

by

ERIC LEASE MORGAN

version 1.0 (September 26, 1995)

TABLE OF CONTENTS

Acknowledgements i
About and how to get this book iii
Distribution iii
Hardware and software used to create this book iv
Overview and demonstration page 1
Demonstrations 2
Teaching a New Dog Old Tricks 3
See Also 4
Client/server model of computing 7
Uniform resource locators (URLs) 9
Scheme 9
Examples 10
"Common Syntax" 10
Examples 11
Path 11
FTP path statements 12
HTTP path statements 12
WAIS path statements 13
Gopher path statements 13
See Also 14
Macintosh-based WWW Browsers 15
Summary 16
America Online Web Browser 16
eWorld World Wide Web Browser 17
MacWeb 18
The popup menu 18
Stylizing text 19
Unique features 19
Hostlist management 19
Summary 20
NCSA Mosaic 20
Custom Menu 20
Annotate 20
Kiosk mode 21
Customizing the HTML 21
Adding MIME type information 21
Summary 22
Netscape 22
Licensing and cost 22
Speed 23
Hotlist management 23
Security 23
Usenet news 24
"Netscapisms" 24
Summary 24

TCP/Connect II 25
Configuration 25
Cacheing 25
Hotlists 26
Helper Applications 26
Summary 26
Hotlist and URL utilities 27
Clay Basket 28
Cyber Link 29
CyberFinder 30
Duke of URL 31
GrabNet 32
Hotlist Grabber 32
Hotlist Sorter 33
Hotlist2HTML 34
Internet Site 34
URLKey 36
WWW Helper Applications 37
See Also 37
Acrobat Reader 38
Drop Stuff w/Expander Enhancer 39
Fast Player 39
Ghostscript (MacGS Viewer) 40
GIF Converter 41
GraphicConverter 42
JPEGView 43
MacBinary II+ 44
MacGzip 45
Sound Machine 45
SoundApp 46
Sparkle 47
Stuffit Expander 48
Tar for the Macintosh 49
Extending WWW Browsers with AppleEvents and AppleScript 51
See Also 51
Flypaper 52
Example #1 - Dear Santa 52
Example #2 - Ask MacAlcuin 53
Scripting WWW Browsers 54
Example #1, Sending AppleEvents from HyperCard 54
Example #2, Sending AppleScript from HyperCard 55
Possible Uses 56
WebRunner 57
Writing ascpt Scripts 57
WebRunner Problem #1 58
WebRunner Problem #2 58
Macintosh-based HTTP servers 59
FTPd 59
httpd4Mac 60

MacHTTP 60
WebSTAR 61
Bringing Up a MacHTTP Server 63
Initially Testing Your Server 63
Editing Your First HTML Document 64
MacHTTP Can Handle the Load 64
See Also 65
MacHTTP configuration 67
Directives, the Meat of machttp.conf 67
Server enhancements and evaluation tools 79
Annotate 80
Count WWWebula 80
Email.cgi 81
folder_list2 82
Index CGI 82
Kelly's Error.acgi 83
MacHTTP AppleGuide 84
Randomgif.acgi 85
Redirect.acgi 85
ServerStat Lite 86
WebStar/MacHTTP Indexer 87
WebStat 88
HTML in seven lessons 91
See Also 92
Basic HTML tags 94
Reserved characters 94
Structure tags 95
Creating paragraphs 96
Line breaks 96
Headings 96
Horizontal rulers 97
Preformatted text 98
Comment tags 99
Creating hypertext links 100
Links to other documents 100
Relative URLs 102
Links to specific parts of the same document 103
Links to specific parts of a different document 104
Menu and Directory 109
Definition list 109
Incorporating graphics with the image tag 110
Basic IMG tag 111
Adding alignment 112
Making your images hot 114
Using the ISMAP element 114
Macintosh-based HTML editors 117
Summary 118
See Also 119
Alpha 119

Arachnid 120
BBEdit HTML extensions 121
BBEdit HTML Tools 122
High Tea 123
HTML Editor for the Macintosh 124
HTML Pro 125
HTML SuperText 126
HTML Tag Macros 127
HTML Web Weaver 128
HTML-HyperEditor 129
HTML.edit 130
HTMLWriter 131
Simple HTML Editor (SHE) 131
Site Writer Pro 132
Webtor 133
HTML Converters and Utilities 135
ColorHEX 135
Colour HeXer 136
FORMSGENERATOR 136
Hotlist2HTML 137
HTML Grinder 137
HTML Markdown 138
HTML Markup 139
HTML Sketcher 140
HTML Special Effects 140
HTML TableTool 141
HTML+ XTND Translator 142
rtftohtml 143
WebDoor Publisher 144
XTND HTML Translator 145
Graphic Utilities 147
See Also 147
clip2gif 148
FlattenMooV 149
Transparency 150
3 Essential qualities of information systems 153
Readability 154
Guidelines 154
Use a consistent layout 154
White space is good 154
Visually organize your pages 154
Keep your pages short 155
Include elements of contrast 155
Use all stylistic elements in moderation 155
See Also 155
Browsability 156
Advantages 156
Disadvantages 156
Classification changes over time 156

A philosophy of classification 157
"Sour milk" 157
A cultural example 157
Back to the real world 158
Guidelines 158
Know your audience 158
Provide "about" texts 158
Use the vocabulary of your readership 158
Create a hierarchal system of ideas 159
Create a system that is both flexible and exhaustive 159
Classify by format last 159
See Also 159
Searchability 160
Advantages 160
Disadvantages 160
Guidelines 161
Include help texts 161
Map located items to similar items 161
Provide simple as well as "power user" search mechanisms 161
Searching Your Server 163
AppleSearch 163
TR-WWW 165
WWW Scripting 167
See Also 168
"Hello, World!" 170
Making your scripts HTTP-friendly 171
Global variables in .script scripts 173
Adding rudimentary input to your .script files 175
Getting input via ISINDEX 176
But don't use .script files 179
Running scripts via AppleEvents 179
Global variables and ISINDEX revisited 181
Trapping and handling errors 185
Overriding the "Stay Open" option 189
Using FORMs to get input 191
Decoding and reading the input in your scripts 199
OSAX for WWW scripting 205
ACME Script Widgets 205
Decode URL 206
Encode URL 206
Parse CGI OSAX 207
Script Tools 207
TCP/IP Scripting Addition 208
Image Mapping 209
Implementation 210
Step #1: Create your graphic 210
Step #2: Create an imagemap configuration file 210
Step #3: Setup and configure your imagemapping CGI program 211
Step #4: Incorporate the graphic into an HTML file 211

Step #5: Test 212
Imagemapping software 212
See Also 212
Hyper Mapedit 212
WebMap 213
Imagemap-pl 214
Mac-ImageMap 214
MapServe 215
The end... of the beginning 217
Appendix A: Special characters and entities 219
Appendix B: Running servers without the Internet 223
Running your server on a AppleTalk-only network 226
See Also 226
Appendix C: Selected Readings 227
Browsability 227
CGI Scripting 227
Editing HTML 230
Extending browsers 232
Graphic utilities 232
Helper Applications 233
HTML Editors 233
Image mapping 234
No network computing 235
Readability 236
Uniform Resource Locators 236
World Wide Web 236
Appendix D: Software Index 239
CGI scripting OSAX and tools 239
Configuration file editors for imagemaps 240
Converters and utilities for HTML documents 241
Extending WWW Browsers 243
Graphic utilities 244
Helper applications 244
HTML editors 247
HTTP servers 249
Imagemap CGI Programs 250
Managing Bookmarks and URLs 250
Searching 252
enhancements 252
World Wide Web browsers 254
Appendix E: Macintosh-related Internet resources 257
Apple Corporation servers 257
Companies 258
FAQs 259
Guides 260
Hardware 262
Programming 262
Publications and publishers 263
Reviews 264

Software 265
Usenet newsgroups 266
User groups 267

ACKNOWLEDGEMENTS

I would like to take time out to acknowledge the people who have made this work possible.

This book would not have happened without the support of my wife, Mary Grey. She has demonstrated unusual tolerance while I type away on my computer trying to finish up "just one more thing."

I would also like to acknowledge Steve Cisler and Monica Ertel, both of the Apple Computer Library. It was Steve who first proposed I write this book and Monica who supported it through the Apple Library of Tomorrow program. Steve and Monica have helped me with my career more than they know.

A special thank you goes to David Cherry. David is a friend from my childhood and the creator of the Tricks illustrations. As I was considering the addition of graphics for the book, naturally I wanted pictures of dogs performing tricks. I searched my memory and the first person I thought of was David. By happenstance, I was able to get in touch with him at just the right time and he agreed to offer his talents. I think the illustrations add a lot to the manuscript's "readability."

Last, but not least, I acknowledge the following people who have contributed ideas and suggestions during the creation of this "book":

Jonathan Ryan Day, Brian A. Sullivan, Carles Bellver, David Habermann, Chuck Shotton, Nathan Gasser, Scott [Guthery?], Colleen Thompson, Jon Wiederspan, Chris Tacy, Jeffrey A. See, Jim Mitchell, Tom Sankey, Hartmut Joh. Otto, Guy Gatien, Paul Kent, Jon Stevens, Bob Cope, Diana Osborne, Henk Schotel, Alan Oppenheimer, Joel Haspel, Kurt S. Harlan, Brian Beckett, John Edward Hill, Bessie Carrington, Lisa Schattner, David Cramer, Patrick Coin, Jim Harrison,

Yoshi Ishikawa, Larry Reynolds, Mark Beatty, Bao Chu-Chang, Isabelle Saint-Saens, Daniel Scherly, Dave Kosiur, Katy Agnor, Larry W. Virden, Chang Sheng Kuoh, Gregory S. Combs, Jouni Miettunen, Bill Melotti, Volker Stahlschmidt.

Thank you, one and all!

About and How to Get This Book

This section describes this book in general.

Teaching a New Dog Old Tricks is an instruction manual describing how to create and manage Macintosh-based World Wide Web servers.

After providing a bit of background about the World Wide Web, this book describes how to:

- Effectively use and enhance your World Wide Web browsers
- Install and maintain MacHTTP
- Write hypertext markup language documents
- Enhance your server with imagemaps and common gateway scripts
- Organize and provide searching services for your server

This book is intended for the person who wants to disseminate information on the Internet including persons in government, education, and industry.

Distribution

This book is intended to be distributed as freely as possible and in as many formats as possible. To that end, you can download and/or access this book in any one or more of the following ways:

Live version (currently at version 1.0)
: This is the full, online, hypertext version. At sometime in the future this may be mirrored somewhere at Apple Computer as well as <URL:http://tricks.lib.ncsu.edu/>, but for right now, it can be found only at <URL:http://152.1.24.177/teaching/manuscript/>

Text-only (currently at version 1.0)
This format is a simple ASCII representation of the text at
<URL:http://152.1.24.177/teaching/archives/tricks-text-only.txt>

MacWrite Pro (currently at version 1.0)
This is a MacWrite Pro version of the text at
<URL:http://152.1.24.177/teaching/archives/tricks-macwrite.hqx>

HTML (currently at version 1.0)
This is the set of HTML documents and graphic images of the book. Using this archive you could set up a mirror of this book. Try:
<URL:http://152.1.24.177/teaching/archives/tricks-html.hqx>

PDF for printing (currently at version 1.0)
This PDF (portable document format) version is intended for printing.
<URL:http://152.1.24.177/teaching/archives/tricks-pdf-print.pdf>.

PDF for viewing (currently at version 1.0)
This PDF version is intended to be read on our computer screen.
<URL:http://152.1.24.177/teaching/tricks-pdf-screen.pdf>

Hardware and Software Used to Create This Book

Normally there is no reason to tell people what computer platform a server is running on, but since this book is about the technology, it may be a good idea here.

Much of the development of this book was done on a PowerBook 540c with 12 MB of RAM and 500 MB of hard disk space. Its configuration also included RAM Doubler. The computer actually hosting this book (http://152.1.24.177/) is a Quadra 610 with 8 MB of RAM and a 200 MB hard disk. It too uses RAM Doubler.

The software used to create this book is not too extensive. It included:

- Simple HTML Editor (SHE) as well as BBEdit 3.5 with the BBEdit HTML Tools extensions to write the text
- Clip2gif to convert any PICT files into GIF images
- HTML Grinder to sequentially link each of the HTML files together as well as create the table of contents
- WebMap to create the imagemap configuration files
- software database, see also database, and simple Internet database (SID) are three locally developed HyperCard stacks used to create and maintain the software reviews, write lists of suggested readings, and create lists of Internet resources, respectively

The software used to serve the book includes:

- WebSTAR to act as the server

- AppleSearch and TR-WWW to provide searching services
- MapServe to serve the imagemap
- Error.acgi as a replacement for the default error.html file
- toc.cgi, a locally developed AppleScript, to use as a pop-up menu table of contents
- email.cgi, a locally developed AppleScript, to provide a feedback mechanism

Lastly, while the creation of this book as been an exhausting process, the process has been quite enjoyable and thought provoking. If you have any suggestions or complaints about this book, then please don't hesitate to drop me a line. I love to get email.

Eric Lease Morgan
eric_morgan@ncsu.edu
http://www.lib.ncsu.edu/staff/morgan/

OVERVIEW AND DEMONSTRATION PAGE

The WorldWideWeb (W3) is the universe of network-accessible information, an embodiment of human knowledge. It is an initiative started at CERN, now with many participants. It has a body of software, and a set of protocols and conventions. W3 uses hypertext and multimedia techniques to make the web easy for anyone to roam, browse, and contribute to. --Tim Berners-Lee 1993

Essentially, World Wide Web (WWW) servers are information dissemination tools. They allow you to save information or data on your hard disk and allow others to access and read that information. The information you can disseminate can be simple ASCII text, formatted hypertext markup language (HTML) documents, graphics, sounds, and/or movies.

In 1989, Tim Berners-Lee of CERN (a particle physics laboratory in Geneva, Switzerland) began work on the World-Wide Web. The Web was initially intended as a way to share information between members of the high-energy physics community. By 1991, the Web had become operational. The World Wide Web is a hypertext system, a concept originally described by Vannevar Bush. The term "hypertext" was coined by Theodore H. Nelson. In a hypertext system, a document is presented to a reader that has "links" to other documents that relate to the original document and provide further information about it.

For example, by clicking on the link embedded in this sentence you will be..... you will be transported back again.

Not only does the hypertext feature work within documents, but it works between documents as well. For example, by clicking on Table Of Contents a new document will be presented to you, the table of contents of this book. If you every get lost, you can always use your WWW

browsing software to go back to where you came because there is always a "go back" button or menu choice.

Scholarly journal articles represent an excellent application of this technology. For example, scholarly articles usually include multiple footnotes. With an article in hypertext form, the reader could select a footnote number in the body of the article and be "transported" to the appropriate citation in the notes section. The citation, in turn, could be linked to the cited article, and the process could go on indefinitely. The reader could also backtrack and follow links back to where he or she started.

Here are just a couple examples of electronic journal/magazine articles employing hypertext features:

- Vannevar Bush, "As We May Think," *Atlantic Monthly* 176 (July 1945): 101-108.
- Mac Net Journal

The Hypertext Transfer Protocol (HTTP) that allows this technology to happen is older than the gopher protocol. The original CERN Web server ran under the NeXTStep operating system, and, since few people owned NeXT computers, HTTP did not become very popular. Similarly, the client side of the HTTP equation included a terminal-based system few people thought was aesthetically appealing. All this was happening just as the gopher protocol was becoming more popular. Since gopher server and client software was available for many different computing platforms, the gopher protocol's popularity grew while HTTP's languished.

It wasn't until early 1993 that the Web really started to become popular. At that time, Bob McCool and Marc Andreessen, who worked for the National Center for Supercomputing Applications (NCSA), wrote both Web client and server applications. Since the server application (httpd) was available for many flavors of UNIX, not just NeXTStep, the server could be easily used by many sites. Since the client application (NCSA Mosaic for the X Window System) supported graphics, WAIS (see WAIS, Inc., CNIDR's freeWAIS, and Ulrich Pfeifer's freeWAIS-sf), gopher, and FTP access, it was head and shoulders above the original CERN client in terms of aesthetic appeal as well as functionality. Later, a more functional terminal-based client (Lynx) was developed by Lou Montulli, who was then at the University of Kansas. Lynx made the Web accessible to the lowest common denominator devices, VT100-based terminals. When NCSA later released Macintosh and Microsoft Windows versions of Mosaic, the Web became even more popular. Since then, other Web client and server applications have been developed, but the real momentum was created by the developers at NCSA.

DEMONSTRATIONS

Since its beginnings distributing text-only documents, WWW servers have been put to other uses. For example, they can be used to create browsable and searchable collections of information. Usually these collections are Internet resources. For example, WWW servers can be used to disseminate lists of:

- Macintosh-related Internet resources
- Academic/scholarly resources
- Entertainment items

Besides disseminating text, WWW servers can disseminate more formatted data like movies.

Sounds are an alternative file format that can be distributed through World Wide Web servers and WWW browser applications:

- "Computers can do that?"
- "Welcome to Macintosh."
- "Good afternoon Mr. Danger."
- "Well I'm sorry."
- "Look what you've done!"
- `"Yes I am."`

With the incorporation of special formatting (like the hypertext markup language tag ISINDEX) and/or common gateway interface (CGI) scripts, World Wide Web servers can be used for the front-ends to simple or complex programs and scripts:

- Display the date and time of the remote server
- Display random messages
- Send email
- Play the "Name Game"

TEACHING A NEW DOG OLD TRICKS

To repeat, the World Wide Web is the colloquial term used to describe the hypertext transfer protocol (HTTP). The main purpose of HTTP is to disseminate and communicate information.

People have always been trying to discover new ways to disseminate and communicate their ideas. The technologies used to accomplish this end have included things like gesturing, speaking, art, writing, printing, telegraph, telephone, radio, television, and email. HTTP is simply another technology accomplishing the same goal, the goal of sharing and spreading information from one person to another.

Thus, this book is about "teaching a new dog old tricks", where the "new dog" is your Macintosh implementing the hypertext transfer protocol through a WWW server and the "old tricks" are the dissemination and communication of information, your ideas.

... transported to the bottom of the page. By clicking on the link in this sentence...

SEE ALSO

"World Wide Web" - [This URL will take you to a terminal-based WWW browser.] <URL:telnet://telnet.w3.org:23/>

Aaron Anderson, "Mac Net Journal" <URL:http://www.dgr.com/web_mnj/>

Alan Richmond, "WWW Development" <URL:http://www.charm.net/~web/Vlib/>

Bob Alberti, et al., "Internet Gopher protocol" <URL:gopher://boombox.micro.umn.edu/11/gopher/gopher_protocol>

CERN, "[Summary of HTTP Error Codes]" <URL:http://info.cern.ch/hypertext/WWW/Protocols/HTTP/HTRESP.html>

CERN European Laboratory for Particle Physics , "CERN Welcome" - CERN is one of the world's largest scientific laboratories and an outstanding example of international collaboration of its many member states. (The acronym CERN comes from the earlier French title: "Conseil Europeen pour la Recherche Nucleaire") <URL:http://www.cern.ch/>

CNIDR, "freewais Page" <URL:http://cnidr.org/cnidr_projects/freewais.html>

Distributed Computing Group within Academic Computing Services of The University of Kansas, "About Lynx" <URL:http://kufacts.cc.ukans.edu/about_lynx/about_lynx.html>

Internet Engineering Task Force (IETF), "HTTP: A protocol for networked information" - HTTP is a protocol with the lightness and speed necessary for a distributed collaborative hypermedia information system. It is a generic stateless object-oriented protocol, which may be used for many similar tasks such as name servers, and distributed object-oriented systems, by extending the commands, or "methods", used. A feature if HTTP is the negotiation of data representation, allowing systems to be built independently of the development of new advanced representations. <URL:http://www.w3.org/hypertext/WWW/Protocols/HTTP/HTTP2.html>

Jon Wiederspan, "Macintosh WWW Information" - "This is everything I have been able to gather about software, information sources, and online documentation that will help you put up a WWW site on your Macintosh computer. Someday soon I'll have some spiffy graphics, but that's not why you came, right? So here's the straight stuff for now." <URL:http://www.uwtc.washington.edu/Computing/WWW/Mac/Directory.html>

Karen MacArthur, "World Wide Web Initiative: The Project" - [This site hosts many standard concerning the World Wide Web in general.] <URL:http://www.w3.org/>

Mary Ann Pike, et al., *Special Edition Using the Internet with Your Mac* (Que: Indianapolis, IN 1995)

NCSA, "NCSA Home Page" <URL:http://www.ncsa.uiuc.edu/>

NCSA, "NCSA Mosaic Home Page"
<URL:http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/Docs/help-about.html>

NCSA, "NCSA Mosaic for the Macintosh Home Page"
<URL:http://www.ncsa.uiuc.edu/SDG/Software/MacMosaic/MacMosaicHome.html>

NCSA, "NCSA Mosaic for Microsoft Windows Home Page"
<URL:http://www.ncsa.uiuc.edu/SDG/Software/WinMosaic/HomePage.html>

NCSA HTTPd Development Team, "NCSA HTTPd Overview"
<URL:http://hoohoo.ncsa.uiuc.edu/docs/Overview.html>

Software Development Group (SDG) at the National Center for Supercomputing Applications, "SDG Introduction" <URL:http://www.ncsa.uiuc.edu/SDG/SDGIntro.html>

Thomas Boutell, "World Wide Web FAQ" - "The World Wide Web Frequently Asked Questions (FAQ) is intended to answer the most common questions about the web."
<URL:http://sunsite.unc.edu/boutell/faq/www_faq.html>

Tim Berners-Lee, Roy T. Fielding, and Henrik Frystyk Nielsen, "Hypertext Transfer Protocol" - "The Hypertext Transfer Protocol (HTTP) has been in use by the World-Wide Web global information initiative since 1990. HTTP is an application-level protocol with the lightness and speed necessary for distributed, collaborative, hyper media information systems. It is a generic, stateless, object-oriented protocol which can be used for many tasks, such as name servers and distributed object management systems, through extension of its request methods (commands). A feature of HTTP is the typing and negotiation of data representation, allowing systems to be built independently of the data being transferred."
<URL:http://www.w3.org/hypertext/WWW/Protocols/Overview.html>

Ulrich Pfeifer, "FreeWAIS-sf" <URL:http://ls6-www.informatik.uni-dortmund.de/freeWAIS-sf/>

University of Kansas, "KUfact Online Information System"
<URL:http://kufacts.cc.ukans.edu/cwis/kufacts_start.html>

University of Minnesota Computer & Information Services Gopher Consultant service, "Information about gopher"
<URL:gopher://gopher.tc.umn.edu/11/Information%20About%20Gopher>

Vannevar Bush, "As We May Think" *Atlantic Monthly* 176 (July 1945): 101-108
<URL:http://www.csi.uottawa.ca/~dduchier/misc/vbush/as-we-may-think.html>

WAIS, Inc., "WAIS, Inc." <URL:http://www.wais.com/>

CLIENT/SERVER MODEL OF COMPUTING

This chapter discusses the client/server model of computing, a fundamental aspect of Internet networking.

To truly understand how much of the Internet operates, including the Web, it is important to understand the concept of client/server computing. The client/server model is a form of distributed computing where one program (the client) communicates with another program (the server) for the purpose of exchanging information.

The client's responsibility is usually to:

- Handle the user interface.
- Translate the user's request into the desired protocol.
- Send the request to the server.
- Wait for the server's response.
- Translate the response into "human-readable" results.
- Present the results to the user.

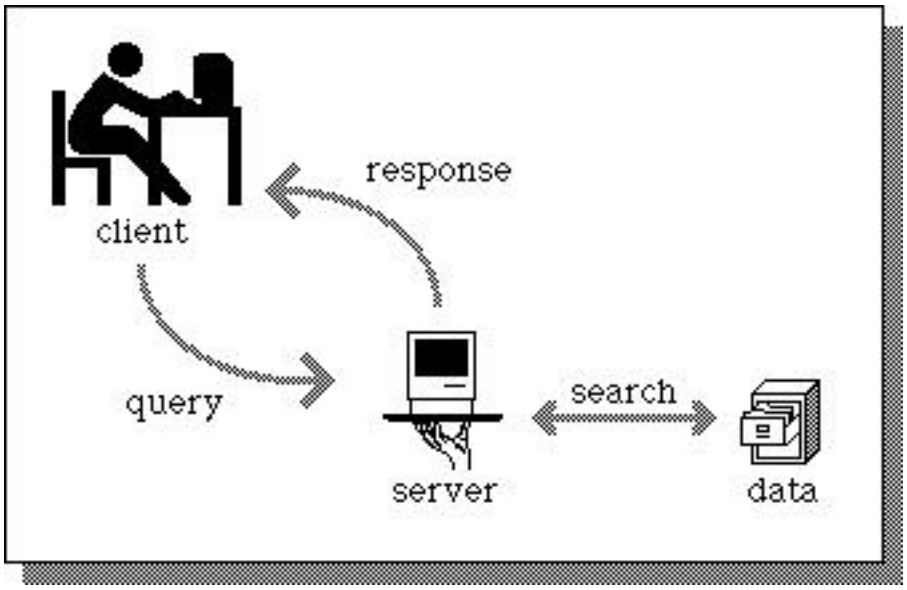
The server's functions include:

- Listen for a client's query.
- Process that query.
- Return the results back to the client.

A typical client/server interaction goes like this:

1. The user runs client software to create a query.
2. The client connects to the server.
3. The client sends the query to the server.
4. The server analyzes the query.
5. The server computes the results of the query.

6. The server sends the results to the client.
7. The client presents the results to the user.
8. Repeat as necessary.



A typical client/server interaction

This client/server interaction is a lot like going to a French restaurant. At the restaurant, you (the user) are presented with a menu of choices by the waiter (the client). After making your selections, the waiter takes note of your choices, translates them into French, and presents them to the French chef (the server) in the kitchen. After the chef prepares your meal, the waiter returns with your diner (the results). Hopefully, the waiter returns with the items you selected, but not always; sometimes things get "lost in the translation."

Flexible user interface development is the most obvious advantage of client/server computing. It is possible to create an interface that is independent of the server hosting the data. Therefore, the user interface of a client/server application can be written on a Macintosh and the server can be written on a mainframe. Clients could be also written for DOS- or UNIX-based computers. This allows information to be stored in a central server and disseminated to different types of remote computers. Since the user interface is the responsibility of the client, the server has more computing resources to spend on analyzing queries and disseminating information. This is another major advantage of client/server computing; it tends to use the strengths of divergent computing platforms to create more powerful applications. Although its computing and storage capabilities are dwarfed by those of the mainframe, there is no reason why a Macintosh could not be used as a server for less demanding applications.

In short, client/server computing provides a mechanism for disparate computers to cooperate on a single computing task.

UNIFORM RESOURCE LOCATORS (URLS)

Uniform resources locators (URLs) are a way of unambiguously describing the locations of Internet resources.

The uniform resource locator (URL) is a fundamental part of the Web. It is utilized to unambiguously describe and identify both the protocol used by and the location of Internet resources. In general, a URL has the following form:

```
scheme://"common syntax"/path
```

Using URLs as a standard, Internet client programs like Web browsers can interpret URLs and retrieve the desired information. URLs describe the protocols and locations of Internet resources without regard to the particular Internet client software the user is employing to access them.

Each of part of a URL (scheme, "common syntax", and path) are described in the sections below.

SCHEME

"Scheme" denotes the type of Internet resource, and it is always followed by a colon (:). The most common schemes include:

- ftp - file transfer protocol
- file - a local file
- gopher - gopher protocol
- http - hypertext transfer protocol
- mailto - electronic mail address

- news - Usenet news
- telnet and tn3270 - interactive sessions
- wais - wide area information servers

Other, less commons schemes include:

- cid - content identifiers for MIME body part
- mid - message identifiers for electronic mail
- nntp - Usenet news for local NNTP access only
- prospero - access using the prospero protocols
- rlogin - another type of interactive session

In the future we may even see scheme like z3950 or whois to denote Z39.50 query/retrieval services or whois databases, respectively.

Examples

- http:
- file:
- wais:
- ftp:
- gopher:
- news:
- nntp:
- telnet:

"Common Syntax"

A pair of slashes (//) and a trailing slash (/) are used to surround "a common syntax" for those schemes "which refer to Internet protocols." This common syntax is dependent on the scheme. The "common scheme" can be divided into 3 parts:

User names and passwords
: It includes options for user names and passwords. Names and passwords are delimited with a colon (:). This is useful for those FTP services using the word "guest" as a part of the log on procedure as opposed to "anonymous". Similarly, this is useful for instructing the end-user on how to log into a remote telnet sessions. The user name and password combination are then delimited from the balance of the "common syntax" by the at-sign (@).

Host
: The Internet name or IP (Internet Protocol) address are the next part of the "common syntax". Examples include "sumex-aim.stanford.edu", "ftp.lib.ncsu.edu", or "152.1.24.177".

Port

Internet communications take place over "ports". Ports represent a section of a communications band available from the Internet protocols. Think of them like telephone number extensions. Many times you are given a telephone number with an extension (1 800 555 1212 x1234). You know this means to call the 800 number and ask for extension 1234. Internet ports work the same way where you communicate with a remote machine (152.1.24.177) and ask for a connection to port 80, or some other port. In most cases, the port number is assumed given a particular scheme. For example, the telnet protocol assumes you want port 23. Gopher assumes port 70. HTTP assumes port 80. WAIS (and Z39.50) assume port 210. But sometimes a information-service provider may not have the authority or does not want to use the standard port for their information service. In these cases, a port number must be specified and it is proceeded with a colon to separate it from the rest of the "common syntax." A good example includes the Geographic Name Server that resides at martini.eecs.umich.edu on port 3000 so it would be denoted as martini.eecs.umich.edu:3000

EXAMPLES

- scheme://library.ncsu.edu/ - connect to the host library.ncsu.edu and use the default port defined by the scheme
- scheme://library@library.ncsu.edu/ - connect to library.ncsu.edu with the username library and use the default port defined by the scheme
- scheme://anonymous@ftp.lib.ncsu.edu/ - connect to library.ncsu.edu with the username anonymous and use the default port defined by the scheme
- scheme://martini.eecs.umich.edu:3000/ - connect to martini.eecs.umich.edu on port 3000
- scheme://gopher.lib.ncsu.edu:70/ - connect to gopher.lib.ncsu.edu on port 70
- scheme://www.lib.ncsu.edu:80/ - connect to www.lib.ncsu.edu on port 80
- scheme:/// - make no Internet connection; use the local machine
- scheme://80/ - invalid; host names are a prerequisite when using ports
- scheme://anonymous/ - invalid; again, a host name needs to be included
- scheme://root:asecret@my.machine:23/ - "jus' plain 'ole stupid"

PATH

The last part of a URL is the path. It includes all the text after the "common syntax." Simplistically speaking, think of the path statement as the lists of folders and then a file name. Like the "common syntax", the structure of the path section is dependent on the URL's scheme. The structure of ftp, http, wais, and gopher path statements are described below. Telnet- nor tn3270-based URLs do not use path statements.

FTP PATH STATEMENTS

The path statements of FTP URLs are the easiest to understand. They form the basis of all other path statements. The path statement of an FTP URL takes the form of a directory (folder) structure with an optional file name appended to the end. Like this:

```
ftp://hostname.edu/folder/subfolder/sub-subfolder/filename.txt
```

In other words, this URL specifies the location of a file named "filename.txt" residing in the directory "sub-subfolder", which is in the directory "subfolder", which is in turn in the directory "folder."
Here is a real world example:

```
ftp://ftp.lib.ncsu.edu/pub/stacks/alawon/alawon-v1n04
```

This URL denotes the following actions:

1. FTP to ftp.lib.ncsu.edu
2. Log on as anonymous
3. Change the directory to /pub/stacks/alawon/
4. Get the file alawon-v1n04

You do not have to specify the filename of a FTP-based URL. You only have to specify the directory path. In doing so, you must end your URL with a trailing slash (/) as in:

```
ftp://ftp.lib.ncsu.edu/pub/software/mac/
```

If you do so, then the URL tells your URL-capable application to simply list all the files in the designated directory. Incidentally, you do not have to specify a path statement in an FTP-based URL either. By omitting the path statement, your Internet application should retrieve a list of the filenames of the root directory of the remote FTP archive.

HTTP PATH STATEMENTS

Here is an example of a URL for an HTML document. It has the exact same structure as the FTP-based URL path statements:

```
http://www.lib.ncsu.edu/stacks/alawon-index.html
```

This URL opens up a HTTP connection to www.lib.ncsu.edu, changes the directory to stacks, and retrieves the file alawon-index.html.

Sometimes the path statements of HTTP-based URLs contain path and/or search arguments. The United States dollar sign ($) and the question mark (?) are used to denote these elements, respectively, as illustrated below:

- http://hostname.edu/hello-world-03.script?TryThisAtHome
- http://hostname.edu/hello-world-03.script$ScriptingIsFun
- http://hostname.edu/hello-world-03.script$communication?WWWServersAreAbout...

It will be a rare instance for you to manually enter a URL containing path and/or search arguments. These elements are generated automatically by WWW browser applications when sending the input for WWW-based scripts like common gateway scripts or imagemapping programs. (For more information about the search and path arguments, see the section entitled "Adding rudimentary input to your .script files" in the chapter WWW Scripting.)

WAIS PATH STATEMENTS

WAIS searches can be specified using URLs. Unfortunately, at the present time, only NCSA Mosaic for the X Window System directly implements the WAIS protocol. WAIS URLs have the following form:

```
wais://host:port/database?query
```

"Port" is assumed to be 210 (the standard WAIS/Z39.50 port), "database" is the source file to search, "?" delimits the database from the query, and "query" is the your search strategy. Here is an example of a URL for a WAIS search:

```
wais://vega.lib.ncsu.edu/alawon.src?nren
```

GOPHER PATH STATEMENTS

Gopher servers and files can be specified with URLs as well. Since gopher resource specifications require "Type" identifiers, and since the paths to gopher resources often include spaces, gopher URLs usually deviate from the norm. Here is an example of a URL for a gopher subdirectory:

```
gopher://gopher.lib.ncsu.edu/11/library/
```

Notice the pair of 1's after the Internet name of the computer. These 1's specify the resource as a directory. On the other hand, the following URL specifies a specific text file within that directory:

```
gopher://gopher.lib.ncsu.edu/00/library/about
```

The "00" denotes a text file. Constructing URLs is more difficult when the path and/or file names of the Internet resources contain special characters like spaces or colons. In these cases, escape codes must be used to denote the special characters. For example:

```
gopher://gopher.lib.ncsu.edu/0ftp%3amrcnext.cso.uiuc.edu%40/pub/etext/e
text91/aesop11.txt
```

This long URL first asks a gopher server (gopher.lib.ncsu.edu) to FTP a file (aesop11.txt) from an anonymous FTP server (mrcnext.cso.uiuc.edu). Notice the "%3a" and "%40" in the URL. They are used to denote a colon (":") and at sign ("@"), respectfully. Furthermore, notice the zero proceeding the "ftp." This is used to identify the remote file as a text file.

As you can see, gopher URLs are particularly difficult to decipher. The easiest way to construct a URL for a gopher item it to access the gopher server via a Web client, traverse the gopher menus until you locate the resource, and then copy the displayed URL from the appropriate part of your client's screen.

SEE ALSO

Daniel W. Connolly, "WWW Names and Addresses, URIs, URLs, URNs, URCs" - "Addressing is one of the fundamental technologies in the web. URLs, or Uniform Resource Locators, are the technology for addressing documents on the web. It is an extensible technology: there are a number of existing addressing schemes, and more may be incorporated over time." <URL:http://www.w3.org/hypertext/WWW/Addressing/Addressing.html>

National Center for Supercomputing Applications, "A Beginner's Guide to URLs" <URL:http://www.ncsa.uiuc.edu/demoweb/url-primer.html>

URI working group of the Internet Engineering Task Force, "Uniform Resource Locators" - Try also, <URL:http://www11.w3.org/hypertext/WWW/Addressing/URL/Overview.html>. <URL:http://www11.w3.org/hypertext/WWW/Addressing/URL/URL_TOC.html>

MACINTOSH-BASED WWW BROWSERS

This section reviews Macintosh-based World Wide Web browsers.

The applications described here are some of the best windows to the Internet. They are the sorts of applications that have changed the Internet into a "world wide web." In a nutshell, these applications do only three things. Send requests for information, wait for the replies, and interpret any returned results.

What makes them truly powerful applications is their ability to understand and interpret many different Internet protocols. At the very least, they can "speak" gopher, FTP, HTTP, and NNTP. Their ability to communicate on all these levels has reduced your need to own and use many different Internet programs. Instead, you can use one program for most of your Internet needs. Since these programs and others like them, can understand many different "languages" and tie together a "host" of different Internet servers, Internet end-users can "weave" their way around divergent Internet server applications with one interface. Thus, the programs described here are what have made turned the Internet into a "world wide web."

In brief, the browser supported by AOL and eWorld is the easiest to get up an running. MacWeb represents a browser adhering to standards. TCP/Connect II is much more than a WWW browser. Mosaic is important because it represents the beginnings of the World Wide Web. Netscape is the most full-featured browser, but as it grows, it becomes a bigger and bigger application demanding more and more computing resources to use effectively.

SUMMARY

The table below summarizes the features of each of the browsers reviewed here.

	AOL	eWorld	MacWeb	Mosaic	Netscape	TCP/Connect
Annotation				√		
Menu editing				√		
Tables				sort of	√	
Font preferences			√	√	sort of	
Hierarchical hotlists					√	
Enhanced HTML					√	sort of
Balloon Help						
AppleScriptable	√	√		sort of	√	√
Mail document					√	
Copy URL from window			√		√	

AMERICA ONLINE WEB BROWSER

Version: 1.0
Author: Intercon
Requirements: subscription to America Online
Cost: $8.95 per month + $2.95/hour after the first 5 hours + 20/minute for telecommunications charges (if applicable)
Pros: easiest of the browsers to use
Cons: provides limited functionality
Remote location: <URL:http://www.blue.aol.com/>
Tricks location: <URL:>


This browser, licensed from Intercon, is the simplest WWW browser available, but with this simplicity comes a few limitations.

To get started with America Online's WWW browser, all you need is a Macintosh an a modem. Preferably, you should have at least modem capable of 14,000 baud. The next thing you need is their software. This is as easy as calling them up on the phone and requesting it at 1 (800) 827-6364. Alternatively, browse the magazine racks of your local bookstores. America Online software seems to always be bundled with at least one of the current Macintosh-related magazines.

After connecting to America Online (AOL), select "Switch to Browser" from the Windows menu and AOL's WWW browser, a separate application, will launch. The first time you use the browser you may want to choose "Configure" from its Edit menu. This is will be the only configuration you will have to do, and it only allows you to change the URL of your home page, the default colors of hypertext links, and whether or not you want graphics automatically

loaded. Now you can click the Home button to go to your home URL, or you can select "Open URL..." from the Services menu. You are now surfing the Web with AOL.

Using this browser is extremely easy, and one the the nicest things it does is cache pages you've visited and reloads them when you select those URLs again. Mosaic does this also and it reduces online time as well as the time you wait for pages to load.

On the downside, it does not support many of the things from the more mature browsers available. For example, it does not support the reading of USENET newsgroups. On the other hand, newsgroups are a part of AOL proper. You can't change the display font; headers are in Helvetica and the body text is in Palatino. Like MacWeb, you can't select and copy any text from the screen, but you can save the screen's text to a ASCII file or HTML file. Last, it does not support telnet URLs and HTML+ forms are not completely implemented.

America Online's WWW browser may be the way to go if you are on a very limited budget and you plan to spend less than about 10 or 15 hours per month using World Wide Web-type services. Otherwise, it may behoove you to subscribe to some sort of Internet provider services and get a direct Internet connection instead.

EWORLD WORLD WIDE WEB BROWSER

Version: 1.0.1
Author: Intercon
Requirements: subscription to eWorld
Cost: $8.95/month + $2.95/hour after the first 4 hours
Pros: easiest of the browsers to use
Cons: provides limited functionality
Remote location: <URL:http://www.eworld.com/>
Tricks location: <URL:>

This WWW browser is exactly like America Online's browser because eWorld (Apple Computer) has licensed the America Online communications software from America Online. See the description of America Online's WWW browser for more information. To get the eWorld software, call eWorld at 1 (800) 775-4556.

MacWeb

Version: 1.00A3.2
Author: Microelectronics and Computer Technology Corporation
Requirements: System 7
Cost: free
Pros: most strictly adheres to standards
Cons: can't select text from the window
Remote location: <URL:ftp://ftp.einet.net/einet/mac/macweb/macweb.latest.sea.hqx>
Tricks location: <URL:http://152.1.24.177/teaching/archives/macweb.latest.sea.hqx>

MacWeb was the second Macintosh-based WWW browser to become available. MacWeb is distributed via the Enterprise Integration Network (EINet).

The advantages of MacWeb are that it is fast, has an elegant and easily customizable interface, supports the automatic creation of HTML documents from its hotlists, and indirectly supports the WAIS protocol by launching WAIS client, MacWAIS. One of its biggest drawbacks is its inability to let you select and copy text from the screen. (This is known problem and is being addressed.)

The Popup Menu

MacWeb pioneered the ability to manipulate hypertext links by clicking on links while holding the mouse button down thus allowing a pop-up menu to appear. The graphic below illustrates this implementation.

The popup menu in MacWeb.

From this menu you can get retrieve the linked item, see information about the linked item, or get the URL of the linked item. The third to last option (Copy URL to Clip) is especially usefu

for copying URLs and pasting them into email messages without having to retrieve the linked item first.

STYLIZING TEXT

MacWeb offers an extraordinary number of options for stylizing how HTML is rendered on your screen. Through the Edit/Styles... menu option you can change the font, style, size, and colors of any HTML tag. The dialog box to make these customization is not quite as easy to understand as hoped, but fortunately it includes an option allowing you to return all your customizations back to their original settings. As a information provider, this feature can prove to be quite useful if you know exactly who your audience is and whether or not they are using MacWeb as their browser. If this is the case, then you can recommend to your audience they configure their Styles menu in a particular way and you can design HTML accordingly. Alternatively, you can distribute your MacWeb Preferences file (found in the Preferences folder of the system folder) to everybody in your organization and rest assured they will be viewing your HTML documents in the manner you designed them. On the other hand, this flies in the face of the advantages of client/server computing.

UNIQUE FEATURES

A unique feature of MacWeb is it ability to display more information about remote HTML files and servers than the other browsers reviewed here. This was alluded to in the section describing MacWeb's popup menu. For example, the View MIME Info and View Suffix Info of the popup menu allow you to easily edit and configure what will happen when you select links with specific suffixes as well as whether or not downloaded file will be viewed and with what application. Thus you can directly edit your MIME types table as soon as you discover new documents you want to retrieve.

MacWeb allows you to view the source of an HTML document in any number of ways from the Option/View Source menu. This menu allows you to see the HTML source at it generated by MacWeb, as it is distributed from the remote server, included with the HTTP header that accompanies the source, or only just the header. Generally speaking, the header information is only useful to people who are delving into the heart of HTTP, but it MacWeb's ability to display this information makes the process easier.

HOSTLIST MANAGEMENT

Hotlist management is a straight forward process with MacWeb. When you want to save document's URL to your hotlist you simply select the Hotlist/Add This Document menu option. While you can not create hierarchal hotlists wit MacWeb, you can create multiple hotlist files for different topics. Furthermore, you can export your hotlist files to simple HTML documents allowing you to more easily share your hotlists with others.

Summary

MacWeb is a solid but basic browser. It has been applauded for its adherence to standards. It could best be described as a straight forward browser without a lot of frills.

NCSA Mosaic

Version: 2.0.0 beta 9
Author: National Center for Supercomputing Applications (NCSA)
Requirements: 4MB RAM
Cost: free
Pros: good for public spaces, annotation support, most customizable
Cons: buggy
Remote location:
<URL:ftp://ftp.ncsa.uiuc.edu/Mosaic/Mac/NCSAMosaic200B12.68K.hqx>
Tricks location:
<URL:http://152.1.24.177/teaching/archives/NCSAMosaic200B9.68K.hqx>


NCSA Mosaic for the Macintosh (or more commonly called MacMosaic) is a part of the family of browsers that raised the public's awareness of the World Wide Web. Unlike MacWeb or Netscape (described later), MacMosaic seems to trying to create a browser application for use in public spaces (exemplified by it Custom Menu, Annotate, and Kiosk menu options) and not necessarily for an individual's desktop.

Custom Menu

Using the Navigate/Custom Menu menu option, you can create new menu items in the menubar. Once new items are created, you can import previously created hotlists into those items or add and edit your own URLs. No other Macintosh WWW browser offers this feature to date. This can be extraordinary useful when MacMosaic is put into a public space. By creating new menus you can preselect tried and true Internet resources for your reader's use. These resources can then be consistently and continuously made available from the menubar. On an individual level, you too can use this feature to make available extended hotlists of resources. In effect, more than any other browser, you can begin to create your on personalized Internet resources library.

Annotate

Annotation is another unique feature of MacMosaic. In the "early days" of the World Wide Web there was much talk about annotation servers. The purpose of these servers was to save people's comments about Internet accessible documents. Once an annotation was created it was to be accessible to anybody else viewing same page. Thus allowing for some sort of

continual dialog surrounding Internet documents. While annotation never really made it out off of the drawing board, MacMosaic still allows you to annotate Internet documents, although only locally, with text or audio.

To annotate a document you simply choose Text... or Audio from the Annotate menu. If you choose the text option, then you are given the opportunity to enter comments. After you are finished writing your comment a link is appended to the bottom of the page. Clicking on the link then brings up the annotation. MacMosaic keeps track of the URL's you view, and every time you return to a URL you have visited before and annotated your annotations appear. You can now even annotate the annotation.

Audio annotations work in a very similar manner. By selecting Audio... from the Annotation you are presented with the standard Macintosh sound recording dialog box. You can then record your annotation and it is "attached" to the document. Later you then have the opportunity to hear your annotation.

KIOSK MODE

By selecting the Options/Kiosk menu option you turn MacMosaic into a minimal level browser; kiosk mode eliminates the Annotate menu, as well as any Custom menus you may have created, and reduces the number of options under the Options menu. It also removed the Open commands from the File menu eliminating the possibility of viewing documents other than the ones on your server. Thus, MacMosaic can be put into a public space for public use and reducing the number of options for end-users. (Fewer options usually leads to less confusion.) Unfortunately, the only way to get out of kiosk mode is to delete MacMosaic's Preference file all together and start all over again.

CUSTOMIZING THE HTML

Just like MacWeb, you can customize just about every aspect of the HTML rendered on your screen. You can do this with the Options/Styles... menu. If you first create a preferences file specifying how you want your HTML rendered, and then you can distribute that preferences file to your audience, then your HTML will appear on their screens in the manner you intended.

ADDING MIME TYPE INFORMATION

Adding MIME types and helper applications to MacMosaic is done through the Options/Preferences... menu. Once you select Helpers from the icons on the left, you are presented with two fields. The field on the left list file name extension and the MIME types associated with the extensions. You can add new MIME types using the button below this field. The field on the left lists MIME types and the applications with which they are associated. Adding a help application then requires you to select the MIME type and then

choose the application from a standard Macintosh dialog box. The procedure is relatively straight forward.

SUMMARY

MacMosaic has a lot of potential, especially for use in public areas. Unfortunately, because it frequently hangs and crashes your computer the application does not seem quite mature. Granted, the "current" release is version 1.03 and not 2.0.0 beta 9, but version 1.03 does not support FORMs, a necessary part of the World Wide Web today. On the other hand, this browser and its siblings are what brought HTTP to the attention of the Internet community. For that alone it deserves special recognition.

NETSCAPE

Version: 1.1N
Author: Netscape Communications Corporation [Aleksandar Totic]
Requirements: 4MB RAM, System 7, 68020 or better microprocessor
Cost: $39.00 to non-academic and government institutions
Pros: hierarchical hotlists, fast, supports mailto and "enhanced" HTML
Cons: opens multiple, simultaneous connections to servers, hotlist editing is confusing
Remote location: <URL:ftp://ftp.mcom.com/netscape/mac/>
Tricks location: <URL:http://152.1.24.177/teaching/archives/netscape-1.1N.hqx>


Netscape, frequently called "Mozilla" is seen by much of the Internet community as the best, all-around Internet application. In many respects, this is not disputed. Since it has been created by the same people who created the original Mosaic applications, it has benefited from experience and improved with age.

LICENSING AND COST

Unlike most of the applications listed in this book, Netscape is not free. Nor is it shareware.

> Netscape Communications Corporation ("Netscape") hereby grants you a non-exclusive license to use its accompanying software product ("Software") free of charge if (a) you are a student, faculty member or staff member of an educational institution (K-12, junior college or college) or an employee of a charitable non-profit organization; or (b) your use of the Software is for the purpose of evaluating whether to purchase an ongoing license to the Software.

For anybody not included in the above statement (like government institutions), Netscape costs $39.00. Reference manuals are extra and can be purchased separately.

Speed

One of first things you notice after using Netscape is the speed in which it seems to load documents. This occurs because, unlike MacMosaic and MacWeb, Netscape open up multiple, simultaneous connections to remote hosts. This is especially evident when the documents in question contains text as well as graphics. While Netscape is retrieving the text it open up another connection to the server and begins retrieving the graphics at the same time. This when this feature was first implemented many server administrators did not like it because Netscape seems to flood the servers with too many connections, but as time has gone on the server applications improved and the situation is under control. Incidentally, you can modify Netscape's behavior in this regard using the Options/Preference/Cache and Network menu option.

Hotlist Management

Netscape supports hierarchal hotlists. This means you can classify your hotlist items or organize them by subjects. The dialog box allowing you to do this is a bit confusing, but with a bit of practice it is not difficult. Like MacWeb, the importing and exporting of hotlists is easy and makes sharing your hotlist with others a trivial matter.

Security

Another unique feature of Netscape is security. Netscape Communications, the company who wrote Netscape, believes, as well as many others, than there is a huge potential for commerce to take place over the Internet. (This may the the understatement of the year.) But before any of these commercial transactions can take place, there have to be methods for confidential communications over the Internet. Thus, security is seen as a major obstacle and implementing technologies like the ones developed by RSA Data Security, Inc. into the Netscape browsers and servers is seen as one solution to this problem.

In any event, Netscape implements many different security protocols (RSA, MD2, MD5, RC4), and when Netscape communicates with a server applications that understand these protocols, then you can rest assured the data being transmitted over the Internet network is remaining confidential.

USENET NEWS

Netscape is a much better Usenet news reader than either MacWeb or MacMosaic. While MacWeb and MacMosaic will read and display Usenet news articles, neither one of these applications remembers which articles you have read, allows you to post new articles, or displays the articles in outline form. On the other hand, Netscape does all these thing as well as allowing you to subscribe and unsubscribe to news groups and emailing responses to the original poster of an article as well as to the group itself.

"NETSCAPISMS"

In an effort to make the World Wide Web more attractive, Netscape supports a number of "Netscapisms" or enhancements to the hypertext markup language (HTML). These enhancements allow editors of HTML to include features like centering, font sizes, tables, inline sounds, and graphic sizes and alignments into their documents. These enhancements, while not necessarily a part of the HTML standard, should not "break" browsers like MacWeb or MacMosaic, but they will cause the documents to be rendered completely differently on those platforms.

More about these enhancements to HTML (or "Netscapisms") are outline more completely in the chapter "HTML Explained".

SUMMARY

Presently, Netscape is the premier WWW browser application for Macintosh computers. It is a solid program that works as documented with few, if any, difficulties. It does require a lot of RAM (4-8MB), and it is a rather large application. In many ways, Netscape is turning out to be much like present day word processors that try to be everything to everybody. On the other hand, it has eliminated the use of a Usenet newsreader on at least one person's desk, and it has almost eliminated the use of an FTP program as well.

TCP/CONNECT II

Version: 2.2
Author: InterCon
Requirements: Systems 7, 5MB RAM, Macintosh SE30 or better
Cost:
Pros: no helper applications needed, all-in-one Internet tool
Cons: does not support news nor mailto URLs
Remote location: <URL:http://www.intercon.com/>
Tricks location: <URL:>


TCP/Connect is much more than a World Wide Web browser; TCP/Connect is an all-in-one Internet application supporting just about every known terminal emulation and Internet service like gopher, FTP, email, finger, ping, WHOIS, HTTP, etc. It can even be used as an FTP server and spelling checker. A nickname for TCP/Connect may just as well be "The Kitchen Sink." Because the subject of this book is the World Wide Web, the paragraphs below will only focus on the Web browser aspects of TCP/Connect.

CONFIGURATION

Configuring TCP/Connect's Web browser is a matter of choosing Configure... from the Edit menu, scrolling down the long list of configurable options until you get to Web, and finally selecting your options. There aren't a lot of choices. You can enter your default home page, choose the colors of HTML text and background, and choose whether you want small, medium or large sized text. You don't get a choice of fonts nor specific sizes. You can also configure here whether or not you want URLs displayed in your windows by default. Incidentally, this same "feature" makes it difficult to use TCP/Connect as a preview application when writing HTML files.

CACHEING

Like Mosaic, TCP/Connect relies on caching retrieved documents to enhance response time. In other words, TCP/Connect saves the HTML and graphic files you access in a separate folder. When you later access the same URLs, TCP/Connect will first display the cached version of the page. If you believe the page has had its content changed, then you can select the Reload button to get a new version. This feature has its good and bad points. Its good in that it makes things appear very fast and greatly reduces network traffic. On the other hand, you may not "remember" to Reload a document when you should and consequently, you may be missing something important. The only way you can "take this dilemma by the horns" is to set the "Expire read links after" option in the configuration section to a low number.

HOTLISTS

TCP/Connect hotlists are rudimentary HTML files. You can open multiple hotlist files simultaneously. You can move items from one hotlist to another, but you can not change the order of the hotlist items. Nor are the hotlists hierarchical in nature.

HELPER APPLICATIONS

In keeping with TCP/Connect's "kitchen sink" philosophy, many of the specialized file types you will find in your Internet travels do not require special helper applications to be displayed; TCP/Connect displays graphic files, QuickTime movies, and sounds all inline. This makes TCP/Connect particularly easy to get up and running.

SUMMARY

This program tries to be everything to everybody. In many respects, it accomplishes its goal. Unfortunately, there are a number of things it could do better. First, it could allow you to copy text from the screen. It could allow you to open up Usenet newsgroup (news) URLs. Since it includes an email program it could support the mailto URL as well. TCP/Connect does not support tables, and there are too few options for setting your font display. (They could at least give you a choice of fonts.) On the other hand, it is AppleScriptable. Just about the only Internet protocol it does not support is WAIS and consequently it can be just about the only Internet program you need.

HOTLIST AND URL UTILITIES

Capturing and managing your URLs and hotlist can often seem like a juggling act. This section reviews utilities helping you keep control of these items.

As you browse and search the Internet, you are going to find resources you would like to visit again. Each of the browsers support a hotlist function allowing you to keep a list of the items you like. The tools described here will help you manage your collection of hotlinks. Some of these tools (Hotlist Sorter and Hotlist2HTML) directly manipulate your hotlist files. URLKey is a nice utility used to open up URLs from within any application that supports the clipboard.

The other tools represent alternatives to hot lists and allow you to save URLs as files and open them up directly from the Finder. Be forewarned about some of these utilities. Applications like CyberFinder and CyberLink create separate, distinct files for every URL you save. These files are small, usually less than a few bytes in size. But because of the way your hard disk may be formatted, these tiny files will occupy much more than a few bytes of disk space. Consequently, it is very possible to fill up our hard disk with many tiny files when these files are not being stored effectively. This is why applications like Clay Basket, GrabNet, or Internet Site are more efficient since they store all your URLs into one or more files, making better use of your hard disk space.

CLAY BASKET

Version: 1.0b3
Author: Dave Winer
Requirements: Netscape or Netscape bookmark files
Cost: free
Pros: can quite possibly condense large collections bookmarks into small, easy-to-read sections
Cons: sparse if not nonexistent documentation
Remote location:
<URL:http://www.hotwired.com/Staff/userland/clay/news_187.html>
Tricks location: <URL:http://152.1.24.177/teachings/archives/clay-basket.hqx>


This application (who knows how it got its name) provides the means of reorganizing your Netscape bookmarks. It also provides the means to "listen" to Netscape and record the URLs you visit. The result of this reorganization and listening process is an outline of URLs. Simple formatting can then be applied to this outline and the entire set of URLs can be saved to a file.

The simplest way to begin learning how to use Clay Basket is to first export a list a Netscape bookmark. Next, launch Clay Basket and open the file you just created. Using the Font menu you can not change the font of you bookmarks. This function is extended through the "Look and Feel" menu option where you can change the colors of the windows and you are given options whether or not the URLs and dates associated with the bookmarks are displayed. After you change the aesthetic appearance of the bookmarks, you can also rearrange the items in the list and create a very nice hierarchal outline. Furthermore, you can create new headings and move your URLs under them with copy, paste, and drag procedures, as well as edit the titles and URLs of each of the items.

Another way to create a Clay Basket document is to turn on the "Recording" feature. Once initiated, Clay Basket will "listen" to Netscape and create lists of URLs you visit. This list can then be edited using the same procedures described above.

The main advantage of Clay Basket is two-fold. First, it makes, through its expandable/collapsible hierarchal lists, a very large collection of hotlist items easily browsable. For example, an entire hypertext information system could be digested into one Clay Basket document. This "digest" could then be distributed displaying the overall structure of the system to would-be end-users. Furthermore, this structure would be presented to the end-user in quite possibly small, easy-to-ready sections.

The only drawback, at the present time, to Clay Basket is that it contains no documentation and therefore takes a bit of experimentation before you can figure out what it is suppose to do.

CYBER LINK

Version: 1.0
Author: Seth Snyder
Requirements: WWW browser
Cost: $20 (shareware)
Pros: reduces data-entry error by directly extracting URLs
Cons: not well integrated with the Finder
Remote location: <URL:http://www.rtz.com/Cyber_Link_Info.html>
Tricks location: <URL:http://152.1.24.177/teaching/archives/cyber-link-10.hqx>


Cyber Link creates files containing a single URL. These files can then be opened from the Finder and thus automatically opening the URL with your WWW browser. The interface is simple enough, just choose New from the Cyber Link's File menu and you are prompted for a new file name. Cyber Link then extracts the URL of the page presently displayed on your WWW browser and saves the file.

Additionally, Cyber Link also contains two menu options automatically opening a URL for company or school names. By selecting either one of these options you are prompted for a name (apple, ncsu, bethany, sony, etc.). Cyber Link then creates a URL for you in the form of "http://www.company.com" or "http://www.school.edu" and sends the URL to your browser. Obviously, this only works if the remote host is accessible via such a URL.

Last, Cyber Link is designed to work with an application called Virtual Meeting, a technology enabling multiple people to "meetings" through WWW-browsers. More specifically, Cyber Link includes and option to send URLs to the WWW browsers other the other people in the meeting.

Cyber Link does what it is suppose to do, but creating new Cyber Link files a bit tedious and not as well integrated into the Finder. On the other hand, unlike some of the other application in this category, Cyber Link does not require you to enter any information about the URL in question since it reads this information directly from the WWW browser itself.

CyberFinder

Version: 1.0
Author: Leonard Rosenthol and Victor Tan
Requirements: Systems 7 and the scriptable Finder, Internet Config 1.1
Cost: free (for personal use)
Pros: saves URLs as files
Cons: files are too small to be saved efficiently on your hard disk
Remote location:
<URL:http://hyperarchive.lcs.mit.edu/HyperArchive/Archive/comm/tcp/cyber-finder.hqx>
Tricks location: <URL:http://152.1.24.177/teaching/archives/cyber-finder.hqx>

 CyberFinder allows you to create files on your hard disk containing URLs. These files, once opened will launch the helper application associated with the URL, as specified by another application called Internet Config.

To get CyperFinder to work for you, simply put the CyberFinder extension in the Extensions Folder of your system folder and restart. (You may want to make sure your desktop file is up-to-date as well.) Then, while the Finder is the front-most application, press command-shift-N. This results in a dialog box waiting for you to enter a URL. The URL then gets saved in the front-most Finder window with an icon representing the Internet protocol. If everything is working correctly, you can now be able to double-click and open the new file and the URL in the file will get sent to the associated helper application. Obviously (or maybe not so obviously), the ability for this to work properly depends on the helper application being able to understand the standard AppleEvent GURL.

CyberFinder works as advertised, but you should keep in mind the size of the CyberFinder files. They are tiny, about 1K. On the other hand, depending on how your hard disk was initialized, they may take up as much 17K or more. (The same problem is associated with aliases.) Consequently, many of these little files will eat up your hard disk even though they are not very big by themselves.

DUKE OF URL

Version: 1.0
Author: Kapor Enterprises
Requirements: AppleScript, System 7, Netscape 1.1N
Cost: free (postcardware)
Pros: good for studying the AppleScripting of Netscape
Cons: not robust
Remote location: <URL:http://www.kei.com/duke-of-url/>
Tricks location: <URL:http://152.1.24.177/teaching/archives/duke-of-url-10-as.hqx>

The Duke of URL, like Cyber Link, extracts URLs from the current window of Netscape and creates a new file that, when opened, will open Netscape and return you to that URL.

The Duke's wizardry is accomplished through short and simple AppleScripts. These scripts basically extract the URL from the topmost window of Netscape 1.1N as well as the URL's title. They then duplicate a template file from your Preferences folder (distributed with the Duke of URL) and save the extracted URL to the duplicated file. Last, the Duke of URL renames the duplicated file with the title of the URL. Preformed many times, the result is a set of AppleScript applications that, when activated, will optionally launch Netscape and open the stored URLs.

The Duke works, but not without a couple of caveats. First, the name "Netscape 1.1N" is hard coded into the AppleScripts. Consequently, if you have not saved your Netscape application as "Netscape 1.1N", then the Duke will always ask you for the location of your browser. This can be avoided by either renaming Netscape or editing the Duke of URL AppleScript as well as its template file. Second, the Duke of URL saves its extracted URLs as AppleScript "properties". As you may or may not know, AppleScript properties are special variables that retain their values once they have been initialized. Unfortunately, AppleScript properties loose their values when you edit the scripts wherein they are contained. Consequently, if you try to edit any of the files created by the Duke of URL, then those scripts will "forget" the URLs they originally preserved.

The Duke of URL is not a bad set of AppleScripts. If you use it exactly the way it was intended, then it will preform admirably. More importantly, these scripts, because they are editable, are more useful for the knowledge they can demonstrate in scripting Netscape through AppleScripts.

GRABNET

Version: 2.0
Author: ForeFront Group, Inc.
Requirements: Netscape 1.1 for full feature set
Cost: $19.95 + shipping and handling
Pros: efficiently organizes URLs into on set of files
Cons: slightly confusing interface
Remote location: <URL:http://www.ffg.com/internet.html>
Tricks location: <URL:>


GrabNet is a URL collecting utility that works very closely with your WWW browser. In its main window can be placed folders. Within these folders you can save URLs. Once the URLs are in your "catalog", you can click on them and open the URLs automatically. Consequently, you can use GrabNet to save and organize a large number of URLs within one single application.

URLs can be entered manually, from your browser's menubar, or via a button from the GrabNet palette. Once you have collected your URLs, you can edit them, give them comments, or view them in a wide variety of ways. GrabNet also provides the ability to export your GrabNet catalogs in the form of HTML documents and/or GrabNet files. These features provide the means of sharing your URLs with others.

GrabNet's interface takes just a bit of time getting used to, but after a bit of practice and after reading the documentation and AppleGuide help, GrabNet may become one of our most valued WWW sidekicks.

HOTLIST GRABBER

Version: 1.0.1
Author: Terry Bingham
Requirements: NCSAMosaic for Macintosh (MacMosaic)
Cost: free
Pros: provides a necessary addition for Mosaic hotlists
Cons: none
Remote location: <URL:ftp://ftp.ncsa.uiuc.edu/Mosaic/Mac/Related/hotlist-grabber-101.hqx>
Tricks location: <URL:http://152.1.24.177/teaching/archives/hotlist-grabber-101.hqx>


Hotlist Grabber is a simple little utility converting NCSA Mosaic hotlist files into text files and vice versa. The URL's of Mosaic hotlists are saved in the resource fork of Macintosh

files, not the data fork were most text-type information is saved. This makes Mosaic hotlists inaccessible and unreadable by text editors like BBEdit, TeachText, or SimpleText. Furthermore, it makes them almost impossible to share your hotlist files with friends through normal delivery mechanisms like email or USENET news.

Hotlist Grabber helps overcome this problem by converting NCSA Mosaic hotlist files into text files. It can also take lists of URLs and convert them into Mosaic hotlist files providing the lists contain valid information.

Using Hotlist Grabber is simple enough. To convert a hotlist into a simple text file, just select "List -> Text" from the File menu, select your hotlist, and give the output file a name. Converting a list of URLs into a hotlist file is just as easy; select "Text -> List."

The only way that Hotlist Gabber could be improved would be if it were AppleScriptable, but that may be overkill.

HOTLIST SORTER

Version: 1.0b3
Author: Dr. Dave
Requirements: hotlists from MacWeb or MacMosaic
Cost: free
Pros: easy and straight-forward
Cons: none
Remote location: <URL:ftp://ftp.ncsa.uiuc.edu/Mosaic/Mac/Related/hotlist-sorter-10b3.hqx>
Tricks location: <URL:http://152.1.24.177/teaching/archives/hotlist-sorter-10b3.hqx>


None of the browsers reviewed here support the ability to sort your hotlists. If you use MacMosaic or MacWeb, then you can use this utility, Hotlist Sorter, to sort your bookmarks in alphabetical order according to title.

HotList Sorter's spartan interface simply allows you to choose a Mosaic or MacWeb hotlist file. Once chosen it prompts you for the name of its output file. The result is a new hotlist sorted in alphabetical order. Easy and straight-forward.

HOTLIST2HTML

Version: 0.71
Author: Lutz Weimann
Requirements: MacWeb or MacMosaic hotlist files
Cost: free
Pros: by default, sorts hotlists
Cons: none
Remote location: <URL:ftp://ftp.ncsa.uiuc.edu/Mosaic/Mac/Related/hotlist-to-html-071.hqx>
Tricks location: <URL:http://152.1.24.177/teaching/archives/hotlist-to-html-071.hqx>

 Where Hotlist Grabber creates text files from MacWeb or Mosaic hotlists, Hotlist2HTML goes one step further and provides, by default, the option to sort the resulting text file as well.

Like Hotlist Grabber, Hotlist2HTML simply allows you to select a hotlist file and then supply an output name. then extracts the URLs from the hotlist file's resource fork and creates a text file of your URL's. By default, this utility sorts your URL's in the text file alphabetically by name. You can override this feature by holding down the shift key while selecting your hotlist file. Hotlist2HTML does not support the conversion of text files (lists of URLs) into native Mosaic nor MacWeb hotlist files.

INTERNET SITE

Version: 1.2
Author: Martin Simoneau
Requirements: PowerTalk
Cost: free
Pros: most powerful Internet resource manager available
Cons: PowerTalk is not widely used
Remote location: <URL:http://www.er.uqam.ca/nobel/simoneau/aoce/InternetSite1-2.sea.hqx>
Tricks location: <URL:http://152.1.24.177/teaching/archives/InternetSite1-2.sea.hqx>

 Internet Site is an extension to Apple Computer's PowerTalk technology. Through PowerTalk, Internet Site allows you to save URLs of Internet resources as PowerTalk Personal Catalog items.

As you may or may not know, PowerTalk is a set Macintosh operating system extensions optionally installed under System 7 Pro, System 7.5, or later. PowerTalk can integrate things

like email and remote AppleShare servers more completely into the Macintosh's user interface. Using Internet Site, Internet resources can be more completely integrated into the interface as well.

After installing PowerTalk and after dragging the two files comprising the Internet Site "application" (Internet Site and Site Collection) into the Extensions Folder of your System Folder, two new menu items appear under your PowerTalk's Catalog menu: New Site and New Site Collection. By selecting "New Site" a new item is added to your presently opened Personal Catalog. After opening the new item you are given the opportunity to change its name, give it a verbose description, select an Internet "scheme" (mailto, gopher, http, wais, etc.), enter information used to complete a URL, and finally, a graphic. Lastly, your have the option to clicking the "Go" button. Doing so launches an Internet application associated with the URL. Consequently, you can create dozens of these Sites saving them as files on your hard disk or sharing them with other PowerTalk users.

To create associations between Internet "schemes" and applications on your computer, you can select "AppleScript" from the pop-up menu within an Internet Site document. There you can change what application gets sent the "GURL" AppleEvent as described in the previous chapter "Extending WWW Browsers Through AppleEvents and AppleScripts". Additionally, you can change the AppleEvents sent by Internet Site to something completely different from GURL, and you can change the parameters that get sent to the application as well. These features are not for the light-hearted, especially since few people have experience with PowerTalk in the first place.

Since PowerTalk's Catalogs are not hierarchal, Internet Site comes with another template called "Site Collection". This template allows you to drop Internet Site documents onto Site Collection files, thus organizing your collection of Internet resources.

Internet Site is the most powerful URL manager "application" described here. It takes advantage of some of the more robust features of the Macintosh operating system and applies them to the Internet resources. Unfortunately, this application may not find a wide following since the use of the PowerTalk technology is barely being used.

URLKEY

Version: 1.0.7
Author: Kevin Altis, Rowland Smith, and Nancy Tindle
Requirements: a WWW browser supporting the GetURL AppleEvent
Cost: "pointerware"
Pros: very easy to use
Cons: none
Remote location: <URL:http://www.city.net/cnx/software/urlkey.html>
Tricks location: <URL:http://152.1.24.177/teaching/archives/URLKey_1.0.7.sea.hqx>


This handy-dandy control panel document makes opening a URL from within any application as easy as Command-C and Command-Shift-U. In other words, by selecting a URL from any document on your Macintosh and copying it to the clipboard, you can open the copied URL with your WWW browser by next pressing Comman-Shift-U. Doing so opens up your preferred WWW browser and points it to the copied URL. If you happen to copy more of the URL than just the URL itself, URLKey will parse out the unneeded information for you. Very nice. To set up URLKey all you have to do is put it in your Control Panels folder, reboot, and configure it so it knows where your WWW browser is. URLKey even has a place to insert new types of URLs when they make themselves available. Incidentally, this application is "pointerware". If you use it, then you are suppose to point to CityNet as your form of payment.

WWW Helper Applications

This chapter describes, reviews, and recommends helper applications for your WWW browsers.

Helper applications are programs supplementing the features of your WWW browsers. For example, if you download a QuickTime movie from a remote server, your WWW browser will not be able to run nor display this movie (yet). To overcome this limitation, your WWW browser can automatically run a second application and tell that application to open and load the downloaded file.

Frequently, the types of files that you will download with your WWW browser will include:

- Compressed archives containing multiple files
- Multimedia files like pictures, sounds, and movies

The sections below review a number of helper applications in terms of there strengths and weaknesses. The quinessencial helper application for your Macintosh is StuffIt Expander. Without this application you will have a difficult time getting any others since StuffIt Expander will be able to uncompress most files you will find on the Internet. After StuffIt Expander you may want SoundApp to play your sound files, JPEGView to display your graphics, and FastPlayer to run your movies. The other helper applications are less critical.

See Also

Scott D. Nelson, "WWW Viewer Test Page" - This pages helps you test your WWW browser for helper applications. By clicking on the "test" buttons a file will be downloaded to your machine and then your browser will try to open it. If it doesn't open the file, then this page

ists helper applications that you can use to "fix" the problem. <URL:http://www-dsed.llnl.gov/documents/WWWtest.html>

Acrobat Reader

Version: 2.0
Author: Adobe Systems Incorporated
Requirements: 2MB RAM, 4MB hard disk, Systems 7, 68020 microprocessor or better
Cost: free
Pros: allows searching of indexed PDF files
Cons: requires a lot of disk space
Remote location: <URL:ftp://ftp.ncsa.uiuc.edu/Mosaic/Mac/Helpers/acrobat-reader-20.hqx>
Tricks location: <URL:http://152.1.24.177/teaching/archives/acrobat-reader-20.hqx>

Acrobat Reader is a program used to read Adobe Acrobat Portable Document Format (PDF) files. These files are essentially pictures of highly formatted documents saved in a proprietary format. They enable people to create and distribute structured documents in a way that does not rely on the end-user's computer hardware and software. The Acrobat Reader application, freely distributable, handles these problems. As the Internet grows and more people are concerned about the structure of their documents (as opposed to the content) things like PDF files may become more prevalent. Acrobat Reader will be one program enabling you to read these documents.

MIME type: application/pdf
Filename extensions: .pdf

DROP STUFF W/EXPANDER ENHANCER

Version: 3.5.2
Author: Leonard Rosenthol
Requirements: System 7
Cost: $30 (shareware)
Pros: uncompresses most file formats
Cons: none
Remote location: <URL:ftp://ftp.ncsa.uiuc.edu/Mosaic/Mac/Helpers/drop-stuff-with-ee-352.hqx>
Tricks location: <URL:http://152.1.24.177/teaching/archives/drop-stuff-with-ee-352.hqx>

 Essencially, DropStuff w/Expander Enhancer is an archiving utility. It allows you to "drag and drop" files and folders on to it and have the automatically become StuffIt archives and optionally BinHex'ed as well. These functions are more for information providers rather than information consumers.

The real use of this program for information consumers becomes apparent when DropStuff is used in conjunction with StuffIt Expander. DropStuff comes with an extension called StuffIt Engine. This file allows StuffIt Expander to automatically uncompress PC ZIP archives (.zip), PC ARC archives (.arc), AppleLink Packages (.pkg), GNU ZIP files (.zip), Unix compressed archives (.Z), and uuencoded files (.uu). The only prevalent compression method not supported by StuffIt Expander in conjunction with DropStuff is are Unix tape archives (.tar).

MIME type: application/x-compressed
Filename extensions: .z, .zip, .gz, or .tgz

FAST PLAYER

Version: 1.1.0
Author: Kevin McMurtrie
Requirements: QuickTime
Cost: free
Pros: small, compact, RAM efficient, save as "flattened" movie
Cons: none
Remote location: <URL:ftp://ftp.ncsa.uiuc.edu/Mosaic/Mac/Helpers/fast-player-110.hqx>
Tricks location: <URL:http://152.1.24.177/teaching/archives/fast-player-110.hqx>

 Fast Player is QuickTime movie player, and unlike SimplePlayer, its free. Fast Player

can load many movies at one time and plays them smoothly. You can set up Fast Player so it will make your computer's desktop color black providing a more aesthetically appealing backdrop for the movies being played. It also includes an option to save a movie as a "flat" movie so the movie can be exported to non-Macintosh platforms.

MIME type: video/quicktime
Filename extensions: .qt or .mov

GHOSTSCRIPT (MACGS VIEWER)

Version: 1.0
Author: Ghostscript by L. Peter Deutsch, MacGS Viewer by Mark Lentczner
Requirements: 4MB free RAM
Cost: free
Pros: effectively renders Postscript files
Cons: does not support printing
Remote location: <URL:http://www.glyphic.com/glyphic/projects/macgs.html>
Tricks location: <URL:http://152.1.24.177/teaching/archives/ghostscript.hqx>


During your 'Net travels you may come across a few Postscript files. Ghostscript is a program that will help you read, render, and print these files on your Macintosh.

Postscript is a programing language. This language's primary purpose is to interpret commands and describe what gets put on a piece of paper. Once a page has been described, the result is intended to be sent to a printer for actual output. Since Postscript is pretty much a "universal" page description language, Postscript files (which may represent text or graphics), can be printed on just about any computing platform. They are also very difficult to edit by hand. Consequently, Postscript files, precursors to PDF files, are/were a good way to distributed "immutable" documents while preserving page layout.

For a long time, the problem with Postscript files was they had to be printed in order to be used. Then came Ghostscript. This program interprets Postscript commands and implements them in a window instead of a piece of paper. Thus you can read Postscript files without a printer.

To get the Macintosh version of GhostScript (MacGS Viewer) to work for you, you must first get the necessary files. Be forewarned. The distribution is large and comes in many parts. Read the documentation carefully. For people who are using non-PowerPC Macintoshes, you can get the distribution from the Tricks archive and everything is set up for you.

After getting the necessary files, make sure your WWW browser is set up to receive Postscript files. Make sure there is a MIME Type of application/postscript matching the file extensions .ai, .eps, and/or .ps. You may want to configure your browser to automatically launch the Ghostscript as well. All you have to do now is download the Postscript files and wait for them

to be rendered by Ghostscript (MacGS Viewer). After they are rendered you can copy the text from the window (in the form of a picture), put the image into some other application, and then print. You will only be able to process one page at a time. If you want to simply print the downloaded Postscript file, then it is recommended you get your hands on BBEdit which comes with an extension just for this purpose.

MIME type: application/postscript
Filename extensions: .ai, .eps, .ps

GIF CONVERTER

Version: 2.3.7
Author: Kevin A. Mitchell
Requirements: 1500K RAM
Cost: $40 (shareware)
Pros: properly reads most common graphic file formats
Cons: limiting compared to other applications
Remote location: <URL:ftp://ftp.ncsa.uiuc.edu/Mosaic/Mac/Helpers/gif-converter-237.hqx>
Tricks location: <URL:http://152.1.24.177/teaching/archives/gif-converter-237.hqx>

GIFConverter reads a number of graphic formats most commonly found on Macintosh computers (GIF, JPEG, TIFF, EPSF). While it can write to these formats, it supports only rudimentary editing capabilities like selecting, rotating, and cropping. GIFConverter works, but there are more full-featured applications available at a lower cost.

MIME type: image/gif
Filename extensions: .gif

MIME type: image/tiff
Filename extensions: .tif or .tiff

MIME type: image/jpeg
Filename extensions: .jpg or .jpeg

GRAPHICCONVERTER

Version: 2.0.7
Author: Thorsten Lemke
Requirements: System 7, 2MB RAM
Cost: $35 (shareware)
Pros: reads and writes multiple graphic formats
Cons: none
Remote location: <URL:ftp://ftp.ncsa.uiuc.edu/Mosaic/Mac/Helpers/graphic-converter-207.hqx>
Tricks location: <URL:http://152.1.24.177/teaching/archives/graphic-converter-207.hqx>


GraphicConverter is an application that can read, write, and edit just about any type of graphic file you can think of. Consequently, it makes for a good, all-around graphics helper application. More specifically, it can read all of the following formats as well as many others: PICT, MacPaint, TIFF, JPEG/JFIF, GIF, BMP, and XBM. The most common formats found on the Internet at this time are GIF, TIFF, and JPEG. An added benefit of GraphicConverter is that you can use it to create new graphics or edit the ones you find on the 'Net.

MIME type: image/gif
Filename extensions: .gif

MIME type: image/tiff
Filename extensions: .tif or .tiff

MIME type: image/jpeg
Filename extensions: .jpg or .jpeg

JPEGView

Version: 3.31
Author: Aaron Giles
Requirements: System 7
Cost: free (postcardware)
Pros: simple and straight-forward, AppleScriptable
Cons: limited editing capabilities
Remote location: <URL:ftp://ftp.ncsa.uiuc.edu/Mosaic/Mac/Helpers/jpeg-view-331.hqx>
Tricks location: <URL:http://152.1.24.177/teaching/archives/jpeg-view-331.hqx>


JPEGView is another graphics file viewer. It and and write PICT, BMP, TIFF, GIF, and JPEG file formats. Unlike GIFConverter or GraphicConverter, JPEGView includes no editing features. Its strongest features are first its price; JPEGView costs the price of a colorful postcard sent to the author. Second, JPEGView is AppleScriptable meaning you can create AppleScripts that drive and manipulate JPEGView's behavior (like converting a file from one format into another).

MIME type: image/gif
Filename extensions: .gif

MIME type: image/tiff
Filename extensions: .tif or .tiff

MIME type: image/jpeg
Filename extensions: .jpg or .jpeg

MACBINARY II+

Version: v1.0.1
Author: Peter Lewis
Requirements: System 7
Cost: free
Pros: very simply encodes or decodes MacBinary files
Cons: none
Remote location:
<URL:http://hyperarchive.lcs.mit.edu/HyperArchive/Archive/cmp/mac-binary-ii-plus-101.hqx>
Tricks location: <URL:http://152.1.24.177/teaching/archives/mac-binary-ii-plus-101.hqx>

This handy utility translates files saved in a binary format into a format usable on Macintosh computers. As you may or may not know, Macintosh files can be divided into two parts: the data fork and the resource fork. The data fork is usually text and the resource fork is usually programming code. These "forks" loose their identity on computers not supporting these formats (non-Macintosh computers). Also, resource forks are not sent by the Macintosh-based WWW servers. MacBinary is an application "melding" the the data and resource forks together making them savable and servable on such computers. This translation process is more efficient than BinHexing a document since the resulting file sizes are smaller, but since BinHex'ed documents are ASCII text files they can be sent through email. Documents encoded with the MacBinary can not be sent through email.

In any event, you may want the MacBinary utility to decode (or encode) any documents you may receive through the WWW that have the extension .bin. On the other hand, StuffIt Expander can take care of MacBinary files as well.

MIME type: application/x-macbinary
Filename extensions: .bin

MacGZip

Version: 1.2.4
Author: SPDsoft
Requirements: System 7
Cost: free
Pros: works as advertised
Cons: compression looses resource fork
Remote location:
<URL:http://persephone.cps.unizar.es/general/gente/spd/gzip/gzip.html>
Tricks location: <URL:http://152.1.24.177/teaching/archives/mac-gzip-03b3.hqx>

Another compressing and uncompressing utility, MacGZip specializes in GNU ZIP files. If you have your browser's MIME types configured as below and you specify that MacGZip is the application to launch after receiving a .gz or .zip file, then MacGZip will uncompress the file for you. StuffIt Expander will provide the same functionality if you have also installed DropStuff w/ Expander Enhancer.

The added benefit of MacGZip is its ability to create GNU ZIP files as well as uncompress them. So if you have a need to create such files, then MacGZip is a utility for you.

MIME type: application/x-compressed
Filename extensions: .gz or .zip

Sound Machine

Version: 2.1
Author: Rod Kennedy
Requirements: System 7, 68020 microprocessor (or better)
Cost: free
Pros: supports recording and looping
Cons: not conversion features, does not support .wav files
Remote location: <URL:ftp://ftp.ncsa.uiuc.edu/Mosaic/Mac/Helpers/sound-machine-21.hqx>
Tricks location: <URL:http://152.1.24.177/teaching/archives/sound-machine-21.hqx>


This program plays AIFF, u-law, Macintosh sound files, as well as few others. It does this with a nice progress bar keeping you informed of its progress as well as other information about the sound. Unlike SoundApp, SoundMachine allows you to loop your sounds thus playing them over and over again. It also supports the recording of sounds and saving them with various compression techniques. SoundMachine does not support the conversion of

sound files into other formats. If you have the Macintosh system extension named Speech Manager installed on your computer, then SoundMachine will speak your commands. It makes you laugh.

MIME type: audio/basic
Filename extensions: .au or .snd

MIME type: audio/aiff
Filename extensions: .aif or .aiff

MIME type: audio/x-aiff
Filename extensions: .aif or .aiff

MIME type: audio/x-quicktime
Filename extensions: .qt, .mov, or .moov

MIME type: audio/basic
Filename extensions: .au or .snd

SoundApp

Version: 1.5.1
Author: Norman Frankle
Requirements: QuickTime for some features
Cost: free
Pros: support multiple formats, optionally quits after playing sounds
Cons: does not support looping
Remote location:
<URL:http://hyperarchive.lcs.mit.edu/HyperArchive/Archive/snd/util/sound-app-151.hqx>
Tricks location: <URL:http://152.1.24.177/teaching/archives/sound-app-151.hqx>

This sound utility plays the most common sound file formats found on Macintosh computers and the Internet including: AIFF, Windows WAVE, and u-law, as well as quite a number of others. One of its nicest features, especially when it is used as a helper application, it that it can be configured to automatically quit after playing a sound. This makes it a rather transparent application seamlessly integrating itself into your workstation. SoundApp is also (slightly) AppleScriptable which opens up other possibilities. Last, SoundApp can convert the sounds it reads into a number of other formats making it a useful tool for people distributing sounds to others on the Internet. Unfortunately, SoundApp provides no looping feature enabling you to play a sound over and over again. Also, it does not include the ability to record sounds.

MIME type: audio/basic
Filename extensions: .au or .snd

MIME type: audio/aiff
Filename extensions: .aif or .aiff

MIME type: audio/x-aiff
Filename extensions: .aif or .aiff

MIME type: audio/x-quicktime
Filename extensions: .qt, .mov, or .moov

MIME type: audio/basic
Filename extensions: .au or .snd

MIME type: audio/x-wav
Filename extensions: .wav

SPARKLE

Version: 2.4
Author: Maynard Handley
Requirements: System 7.5, QuickTime
Cost: free
Pros: mulit-format aware, many playback options, converts MPEG and QuickTime movies
Cons: not the best PICT viewer
Remote location: <URL:ftp://ftp.ncsa.uiuc.edu/Mosaic/Mac/Helpers/sparkle-24.hqx>
Tricks location: <URL:http://152.1.24.177/teaching/archives/sparkle-24.hqx>

Sparkle is a mulit-format, multimedia player. It can play QuickTime movies as well as MPEG movies or display PICT files. (MPEG movies represent a growing de-facto standard for multimedia files.) While it can not save files as "flattened" movies like Fast Player, it can save QuickTime movies as MPEG movies. Sparkle has more options for playing movies like backwards, back and forth, and various speeds. It also contains a number of options for image quality not found in the other movie players.

MIME type: video/quicktime
Filename extensions: .qt or .mov

MIME type: video/mpeg
Filename extensions: .mpg, .mpeg., or .mpe

MIME type: image/pict
Filename extensions: .pic, .pict

STUFFIT EXPANDER

Version: 3.5.2
Author: Leonard Rosenthol
Requirements: System 7
Cost: free
Pros: supports widest variety of compression and encoding formats
Cons: none
Remote location: <URL:ftp://ftp.ncsa.uiuc.edu/Mosaic/Mac/Helpers/stuffit-expander-352.hqx>
Tricks location: <URL:http://152.1.24.177/teaching/archives/stuffit-expander-352.hqx>


Because many files distributed over the Internet in some sort of compressed archive format and you need a program to uncompress and convert those files into something more useful, StuffIt Expander is the quinessencial helper application.

In order to facilitate faster more efficient transfers of files, many files on the Internet are compressed into single files. These files are called archives. These archives take up less disk space than the some of the files individually at the expense of being able to directly execute or use these file; the archives must be uncompressed before they can be used.

To facilitate the special format of Macintosh file structures, these archives are usually translated into one of two formats known as BinHex or MacBinary. Both of these formats allow the transfer of files intended for Macintosh computers to be done with ASCII text. Thus, these files, like the compressed files, need to be translated from their ASCII formats to their original Macintosh formats.

In short, many files on the Internet are first compressed into archives and then converted into ASCII files. You must have a program to reverse this process in order to use these files. StuffIt Expander is one such application.

Using StuffIt Expander couldn't be easier. Simply retrieve the archive and install it. Next, configure your WWW browser to automatically launch StuffIt Expander. (This has probably already been done.) Then whenever you select a file whose file name ends in .sit, ,cpt, .bin, .sea, or .hqx StuffIt Expander will "automagically" convert and uncompress the file making it ready for you to use. What is really great is that it is free.

When you have installed an application called DropStuff (more specifically the Stuffit Engine included with DropStuff), StuffIt Expander will convert and uncompress even more file types including: PC ZIP archives (.zip), ARC archives (.arc), AppleLink Packages (.pkg), GNU

ZIP files (.zip), Unix compressed files (.Z), and uuencoded files (.uu).

MIME type: application/mac-binhex40
Filename extensions: .hqx

MIME type: application/x-stuffit
Filename extensions: .sit

MIME type: application/x-macbinary
Filename extensions: .bin

If you have installed DropStuff, then these MIME types will work as well.

MIME type: application/x-compressed
Filename extensions: .z, .zip, .gz, or .tgz

TAR FOR THE MACINTOSH

Version: 3.0
Author:
Requirements: System 6.0.5 or higher
Cost: free
Pros: successfully extracts .tar archives
Cons: must manually select Extract menu
Remote location: <URL:>
Tricks location: <URL:http://152.1.24.177/teaching/archives/tar.hqx>


A common way of archiving files on Unix machines is tar. Tar stands for "tape archive" and it is a method of reducing a number of files into a single. Tar for the Macintosh can read and write this file format, and if you need to uncompress a .tar file, then this helper application will help you out. It is the only Macintosh application that currently does this, so it can be particularly handy. On the other hand, you may not find many .tar archives on the Internet. Be forewarned, Tar for the Macintosh does not automatically uncompress the tape archives; to uncompress the archives you must select Extract from the File menu.

MIME type: application/x-tar
Filename extensions: .tar

EXTENDING WWW BROWSERS WITH APPLEEVENTS AND APPLESCRIPT

This chapter outlines how you extend your WWW browser Applications through AppleEvents and AppleScripts.

A few Macintosh-based WWW browsers support the ability to send and receive AppleEvents. AppleEvents are a relatively new feature of the Macintosh operating system allowing applications to be given commands by other programs.

One of the most universal and useful commands available is the GetURL command. Using this command you and tell a WWW browser to open a URL and then display the results in a window or even save the results to a file. For more information about the GetURL AppleEvent and other commands, read "Macintosh Remote Control Documentation" and "Software Development Interface".

The applications and demonstration program below show you how you can begin extending your WWW browser in these areas and begin to incorporate their use even greater into your work environment.

SEE ALSO

Dave Winer, "Aretha Website" - "Frontier is a scripting system for the Macintosh. Lots of features, lots of verbs. It used to be a commercial product, but now it's free. Why? Because I want Frontier to have a shot at becoming a standard. I think it'll be fun!" <URL:http://www.hotwired.com/staff/userland/aretha/>

Martin Fenner, Fred Terry, and PreFab Software, Inc., "ScriptWeb" - "This virtual site is a collaborative effort to provide a single source of information for Macintosh scripting, primarily for AppleScript and Frontier." <URL:http://www.gz.com/scriptweb/>

Netscape Communications, Inc., "Netscape API for the Macintosh" - "Netscape uses AppleEvents to interact with other Macintosh applications. It is scriptable, and partially recordable. Most of the events (and all Netscape-specific ones) are documented in the Netscape's AppleEvent dictionary. You see the dictionary with the Script Editor, a scripting utility available from Apple." <URL:http://home.netscape.com/newsref/std/mac-remote-control.html>

Spyglass, Inc. , "Software Development Interface" - "This document describes a cross-platform API which can be used to extend the capabilities of Web browsers by integrating them with other applications. The API is specified as a set of platform-independent generic verbs which can be issued either to or from Web browsers. Platform-specific implementations of this spec are given for the AppleEvents and DDE transports. Future transports to be provided include OLE2 and TCP/IP (for UNIX)." <URL:http://www.spyglass.com:4040/newtechnology/integration/iapi.htm>

FLYPAPER

Version: 0.1.2
Author: Eric Iverson
Requirements: Netscape
Cost: free
Pros: very secure since scripts must already be on your mac. Very fast since scripts are/can be compiled. Creating flyp html codes is simple.
Cons: you must have the script already on your mac (could be a good thing).
Remote location: <URL:http://www.pass.wayne.edu/~eric/flypaper>
Tricks location: <URL:http://152.1.24.177/teaching/archives/latest_flypaper.sit.hqx>


Like WebRunner, FlyPaper is a protocol helper application. It is a solution to the problems introduced by WebRunner. It does this by first creating for itself a new protocol (flyp) to be used in URLs. Second, it requires that the script to be executed must reside on the end-user's computer. Thus, the script is not embedded in any HTML file, but it requires the end-user to retrieve script before hand.

EXAMPLE #1 - DEAR SANTA

Here is an example FlyPaper application. It requires you to have AppleScript installed on our computer as well as the Scriptable Text Editor. All the program does is ask you for your name

and what you want for Christmas, and then it writes a simple letter to Santa Clause, saves it, and opens it using Netscape.

Here's what you need to do:

1. Download the FlyPaper application (above)
2. Download Dear Santa making sure you save it in the same folder as FlyPaper
3. Run Dear Santa

Notice that the URL for running Dear Santa is simply:

```
flyp:dear-santa
```

This means your WWW browser will look for the FlyPaper application and run the associated script.

Example #2 - Ask MacAlcuin

A slightly more practical application exemplifying these techniques is Ask MacAlcuin. This program asks you for a subject you would like to use to search the Internet. It then creates an HTML file encoding URLs of various Internet search engines like an index of Usenet FAQs and Yahoo. This script, Ask MacAlcuin, requires the Scriptable Text Editor as well as the Encode URL OSAX (included with the Ask MacAlcuin).

Here what to do:

1. Download FlyPaper (above)
2. Download Ask MacAlcuin
3. Save the file ask-macalcuin in the same folder as FlyPaper
4. Save the Encode URL OSAX in the Scripting Additions Folder of the Extensions Folder of your System Folder
5. Run Ask MacAlcuin

SCRIPTING WWW BROWSERS

Version: 0.01
Author: Eric Lease Morgan
Requirements: HyperCard 2.2
Cost: free
Pros: demonstrates sending AppleEvents and AppleScripts to WWW browsers
Cons: incomplete
Remote location: <URL:http://152.1.24.177/teaching/archives/scripting-WWW-browsers.hqx>
Tricks location: <URL:http://152.1.24.177/teaching/archives/scripting-WWW-browsers.hqx>

The HyperCard stack, Scripting WWW Browsers version 0.01, exemplifies how you can send AppleEvents and AppleScripts to WWW browsers. It is a rudimentary stack intended to demonstrate the points made in the section below.

EXAMPLE #1, SENDING APPLEEVENTS FROM HYPERCARD

Using an XCMD called SendAppleEvent (written by Donald Olson) it is possible to send AppleEvents to remote applications. It is available and documented in the stack at <URL:http://152.1.24.177/teaching/archives/SendAppleEvent-XCMD.hqx>.

The SendAppleEvent XCMD has the following syntax:

```
SendAppleEvent eventClass, eventID, targetProgram, replyRequest,
interactionRequest, timeOutValue, data, keyword, data, keyword, data,
keyword, data, keyword, data, keyword
```

where:

eventClass
An AEEventClass

eventID
An AEEventID

targetProgram
Path to target application in the form "zone:mac:application" if local, only "application" is required

replyRequest
noReply, queueReply, waitReply

interactionRequest
    canInteract, neverInteract, alwaysInteract

timeOutValue
    in ticks

data
    TEXT

keyword
    data Type (Descriptor type) If you are only sending one piece of data and it is to be the direct object, the first key is optional.

Using this command you can create a button sending an AppleEvent to MacWeb as shown below:

```
on mouseUp
  -- make sure MacWeb is open
  open "MacWeb"

  -- send the apple event necessary to to open a URL
  sendappleevent "wwwc", "ourl", "MacWeb", "noReply", & ¬
  "neverInteract","300","http://www.lib.ncsu.edu/","kURL", & ¬
  data2,key2,data3,key3,data4,key4,data5,key5
end mouseUp
```

(Unfortunately, this exact example does not work with MacMosaic nor Netscape.)

As you can see from the example, sending AppleEvents from HyperCard to applications can be a cryptic process! This is exactly why AppleScript was invented. AppleScript is a way of putting a more-human language on top of these cryptic AppleEvents.

Example #2, Sending AppleScript from HyperCard

Using HyperCard 2.2 or later, you can put AppleScripts into your HyperCard stacks. AppleScripts are much easier to work with than raw AppleEvents.

A developing standard AppleScript command, currently supported by MacMosiac and Netscape, is GetURL. Given a URL, this command enables you to tell your WWW browser to take the URL, retrieve the item is is associated with, and display it or save it to a file. Here is the syntax of the GetURL AppleScript command:

```
GetURL  string  -- The url
  [to  file specification]  -- file the URL should be loaded into
  [inside  reference]  -- Window the URL should be loaded to
  [from  string]  -- Refererer, to be sent with the HTTP request
```

Consequently, a button in your HyperCard stack (or a handler in an AppleScript) containing the following code will tell MacMosic open a connection to http://www.lib.ncsu.edu/:

```
tell application "MacMosaic"
  activate
  OpenURL "http://www.lib.ncsu.edu/"
end tell
```

Similarly, this code will tell Netscape to open up a connection to http://www.lib.ncsu.edu/staff/morgan/ and save the output to a file named myfile.html on your Macintosh's desktop:

```
tell application "Netscape"
  activate
  set outputFile to (path to desktop as string) & "myfile.html"
  GetURL "http://www.lib.ncsu.edu/staff/morgan/" to file outputFile
end tell
```

It should be noted that Netscape includes many more AppleScript command then just GetURL. For example, it includes a command called OpenURL which is a more full-featured version of GetURL. It also includes options to dynamically register MIME types, get the information about its open windows, and navigate back and forth between recently visited URLs.

Possible Uses

There are a number of possibilities for this sort of thing. For example you could create a database of URLs and include in the database the ability to view any of the URLs by selecting a function from the database application.

A simpler idea is to create an AppleScript with the following code:

```
set theURL to the clipboard
tell application "Netscape"
  activate
  GetURL theURL
end tell
```

This code simply takes the contents of the clipboard and puts it into a variable called "theURL". It then tells a WWW browser (Netscape) to open the URL. Saved as an AppleScript application that never shows its startup screen, you could save it in the Automated Tasks folder of your Apple Menu Folder. Once there you could select and copy URLs from any application (email, word processor, etc.) and open your newly created script. The script will then activate your WWW browser and open the URL.

Another, more elaborate application could be the creation of a simple expert system used to search the Internet. This application would ask you questions. Based on the answers it would ask you other questions. At the end of the question and answer process, the script would

dynamically create an HTML page presenting the results of the question and answer process as well as listing a number of hot links to remote Internet resources that may have the information you seek.

WebRunner

Version: 0.1.3
Author: Eric Iverson
Requirements: Netscape
Cost: free
Pros: scripts are completely self-contained within the html
Cons: a security risk. nothing prevents you from clicking on a link to a harmful script in somebody else's pages. Takes time to compile the script. Creating ascpt html codes is a complex process.
Remote location: <URL:http://www.pass.wayne.edu/~eric/webrunner>
Tricks location:
<URL:http://152.1.24.177/teaching/archives/latest_webrunner.sit.hqx>


If a copy of WebRunner is located on your computer, and you select a URL having the scheme "ascpt", then the data contained in the URL will be sent to your computer and run. Unlike CGI scripts run by the remote computer hosting the server, ascpt URLs are executed locally as the following link exemplifies. (You must have a copy of WebRunner on your computer before the script will work, and, until other Macintosh-based browsers mature, only the Netscape browser produces the desired effect.)

By the way, Alcuin was a Medieval librarian and advisor to Charlemagne.

Writing ascpt Scripts

The process of writing ascpt scripts is the same as writing any AppleScript scripts, except you must encode them before they can be added to your the URLs of your HTML documents. Unencoded, here is the AppleScript for the "Resurrect Alcuin" script above:

```
tell application "Netscape 1.1N"
     display dialog "MacAlcuin says, 'Hello, World. I'm alive!'" & ¬
       buttons {"OK"} default button {"OK"}
end tell
```

As you can see, this script simply tells the Netscape application to display a dialog box.

After you write your scripts making sure they run correctly and include plenty of error trapping, save the script as an ASCII text file. Next, drag the text file on to the WebRunner

application. WebRunner will encode the script for you changing all the spaces, returns, colons, etc. to their proper escaped sequences. WebRunner will then copy your encoded script to the clipboard. From there, paste your script into a URL of an HTML document. This leads you to two big problems with WebRunner.

WebRunner Problem #1

There are two problems with WebRunner (ascpt) scripts. First, the AppleScript to be executed must be completely contained within a single URL with all HTML escape sequences encoded (return characters, colons, backslashes, greater than and less than signs, etc.). Watch out though! WebRunner does not automatically encode characters like less than (<), greater than (>), or ampersand (&), and if your script contains these characters, then you must encode them also, by hand.

Consequently, your HTML files are even more difficult to debug, read, and maintain. For example, below is the URL that was used to create the "Resurrect Alcuin" script above:

```
<A
HREF="ascpt:tell%20application%20%22Netscape%201.1N%22%0D%09display%20d
ialog %20%22MacAlcuin%20says,%20%27Hello,
%20World.%20I%27m%20alive!%27%22%20 buttons%20{%22OK%22}%20default%20
button%20{%22OK%22}%0Dend%20tell%0D">Resurrect Alcuin</A>
```

Yuck! Furthermore, these problematic URLs only get worse as the lengths of your scripts increase.

WebRunner Problem #2

More importantly, ascpt scripts represent a potential security risk. For example, a hapless WWW surfer like yourself may select an ascpt script that could do some potential harm to your computer:

A non-functioning, potentially dangerous script

MACINTOSH-BASED HTTP SERVERS

This chapter provides an overview of Macintosh-based HTTP servers.

The applications reviewed here are the other half of the client/server equation necessary for the hypertext transfer protocol to work. Put another way, the programs listed here will be the programs you can used to disseminate and communicate your ideas to the world.

FTPD

Version: 3.0.0
Author: Peter N. Lewis
Requirements: System 7, MacTCP 1.1, and File Sharing enabled
Cost: $10 (shareware)
Pros: simple to set up, relies on File Sharing
Cons: rudimentary
Remote location: <URL:>
Tricks location: <URL:http://152.1.24.177/teaching/archives/FTPd-300.hqx>

This little program is a multi-protocol server application. FTPd is primarily an FTP server, but also works as a gopher and HTTP server as well. The HTTP server aspects of FTPd are brand new to this release and the features it includes are rudimentary. It supports no CGI scripting and no customization of error messages. Customization of file types (MIME) is handled by a separate application also written by Peter Lewis called Internet Config. This "other" HTTP server is not as robust as MacHTTP or WebSTAR, but in the future it may give them a run for their money. Especially considering the $10 fee.

HTTPD4MAC

Version: v123a
Author: Bill Melotti
Requirements: System 7 and a network connection
Cost: free
Pros: fast, unobtrusive
Cons: CGI scripts not supported, basic configuration
Remote location: <URL:http://130.246.18.52/>
Tricks location: <URL:http://152.1.24.177/teaching/archives/httpd4mac-123a.hqx>

This particular HTTP server is expressly intended to be a free application so anybody on the Internet with a Macintosh can become a WWW server. httpd4Mac is a "faceless background application" meaning that once it is launched it displays no menubar and you can not make it active from the Application Menu. Upon launching httpd4Mac looks for a preferences file in its own folder and then the systems folder Preferences Folder. If it does not find a preference file, then httpd4Mac creates one of its own and you can use it to configure the application. There are really few configurations possible: MIME types, port, logging, domain name resolution, etc. Upon reading the instructions in the preferences file, configuring the application is straight forward. Like FTPd, httpd4Mac is rudimentary HTTP server. It does not support CGI scripts and your ability to customize error output is restricted. Furthermore, httpd4Mac supports only one default file name, home.html. If you are moving your data from one server to this one, then you may have to change the names of your default files. httpd4Mac is good program for simple needs. If you want to create a low volume server that runs concurrently on your desktop with other applications, then consider httpd4Mac.

MACHTTP

Version: 2.2
Author: Chuck Shotton
Requirements: System 7, MacTCP, AppleScript (recommended)
Cost: $80 (shareware)
Pros: complete WWW server, very easy to setup and administrate
Cons: not mulitthreaded
Remote location: <URL:http://www.biap.com/>
Tricks location: <URL:http://152.1.24.177/teaching/archives/machttp-2.2.sit.hqx>

MacHTTP was the first Macintosh-based WWW server to become available. It is extremely easy to set up and configure. In fact, it is so easy it works "straight out of the box." MacHTTP requires System 7 to support advanced features like AppleScript. MacHTTP runs on Macintosh II-type computers (e.g., Macintosh IIci, SE/30, LC, Centris, and Quadra

computers). It does not run on low-end Macintoshes based on the Motorola 68000 microprocessor (e.g., Macintosh Plus, SE, and PowerBook 100 computers).

MacHTTP supports common gateway scripts (CGIs) as well as specialized CGI scripts called "asynchronous" CGIs or ACGIs. ACGIs are written exactly like CGI program except they are handled asynchronously with the server application. This allows the server to process other requests for files or information and allows some processing time for the ACGI application.

MacHTTP takes extreme advantage of AppleEvents. In this regard is it completely AppleScriptable and recordable; all of its menu options are available via documented AppleEvents.

MacHTTP is very well documented and supported. It comes with extensive documentation and tutorial files well worth your reading. The scripts demonstrating how to take advantage of MacHTTP's AppleScript interface and custom AppleEvents have been the model for many Macintosh-based CGI/ACGI scripts.

MacHTTP has been superseded by WebSTAR, a product marketed by StarNine. In short, WebStar is really MacHTTP on steroids since WebSTAR supports a number of features not included in MacHTTP. For example, WebSTAR sports the Macintosh Thread Manager meaning WebSTAR can perform internal multitasking allowing it to process more requests more efficiently. Additionally, WebSTAR supports pre- and post-processing of requests so particular server-side functions can take place before or after requests have been handled. Last, WebSTAR supports a feature called "actions" enhancing the server's ability to handle scripts.

While MacHTTP is not longer being improved upon, it is still an excellent program. If you can not afford WebSTAR, then MacHTTP is definitely the way to go.

WEBSTAR

Version: 1.2
Author: StarNine (Chuck Shotton)
Requirements: Macintosh Plus or greater, 4 MB RAM, AppleScript, MacTCP
Cost: $295 - $599 depending on what type of institution you represent
Pros: fast, easy to set up and administrate
Cons: none
Remote location: <URL:http://www.starnine.com/webstar/>
Tricks location: <URL:>

WebSTAR is the commercial version of MacHTTP. Written by Chuck Shotton, WebSTAR is in the marketing hands of StarNine. WebSTAR offers significant improvements over MacHTTP. The most notable is the increase in speed. This is accomplished through the use of the a Macintosh extension called the Thread Manager. The Thread Manager allows applications to more effectively pass processing control to other applications. Thus, using the

Thread Manager, WebSTAR is able to use your computer more efficiently and consequently run faster. Not incidentally, the increase in speed is quite noticeable over MacHTTP.

WebSTAR comes with a new administration program. This program enables you to setup and configure WebSTAR without sitting directly at the computer hosting the server. Consequently, it is possible to put your WebSTAR computer in a more secure environment while you administrate it from your office. Using this administration program you can specify what information you want to capture in your log files, add and remove passwords, set up realms as well as allow/deny settings, and configure file suffix settings.

New with WebSTAR are preprocessing, postprocessing, and ACTIONs. These functions allow you to perform user-defined functions before, after, or during the delivery of your data. Consequently, using these functions, you can customize your server environment to a greater extent.

If you are willing to spend the extra money to buy WebSTAR over MacHTTP, then your money will be well spent since this application is more robust than it predecessor and just as well supported.

BRINGING UP A MACHTTP SERVER

This chapter instructs you how to bring up the Macitosh-based WWW server called MacHTTP.

Bringing up a MacHTTP is a very simple process. Follow these instructions:

1. Acquire the MacHTTP distribution. The authoritative address is http://www.biap.com/.
2. Debin-hex the archive with StuffIt Expander or some other debin-hexing program like StuffIt Lite. If you use a WWW browser to download the archive, then it will most likely be debin-hexed for you.
3. Uncompress the resulting archive with something like StuffIt Lite. Again, it may get uncompressed for you if you used a WWW browser to download it.
4. Make sure the MacTCP control panel is configured correctly. (It most likely is if you just acquired MacHTTP from the Internet. Otherwise you would not have gotten this far.)
5. After uncompressing the archive, a folder should have been created called "MacHTTP 2.2". Open it. Open the folder named "MacHTTP Software & Docs". Double-click on icon named "MacHTTP 2.2".

Congratulations! You have just brought up MacHTTP.

INITIALLY TESTING YOUR SERVER

To test your server you need a WWW browser like Netscape, MacWeb, or MacMosaic. Use your WWW browser to open a URL to your server. Here's how:

1. Open the control panel document called MacTCP and note your IP address. It will look something like "152.1.24.177".

2. Use your WWW browser's "Open URL" menu option and enter the URL specifying your server. The URL will look something like "http://152.1.24.177/".
3. As a result you should see a screen titled "Welcome to this MacHTTP Server!"

Congratulations! Your server is on the World Wide Web.

Editing Your First HTML Document

To demonstrate how easy it is to put your own documents on the World Wide Web, you will now edit a hypertext markup language (HTML) document and view the changes.

1. In the same folder as your MacHTTP application there is a file named "default.html". Open it. It was created with Simple Text, the editor that came with your Macintosh, and therefore it should open right up.
2. In the first paragraph change the text "Welcome to this MacHTTP Server!" to something like "Hello World! I'm on the Web".
3. Save and close the file.
4. Return to your WWW browser and reopen the connection to your server like you did in the previous section
5. You should now see "Hello World! I'm on the Web" instead of the text that was previously there.

Congratulations! You have just edited your first HTML document.

MacHTTP Can Handle the Load

MacHTTP is a robust hypertext transfer protocol (HTTP) server on par with any Unix-based HTTP server. This is not necessarily true because MacHTTP is so well written, even though it is. This it is also true because HTTP is a simple protocol that does not require very much overhead.

Because MacHTTP is so effective it can handle as many as 10,000 connections per hour, as many connections made to many university servers per day. To be this effective it is a wise idea to dedicate the computer running MacHTTP to be just that, a dedicated computer. While MacHTTP can run on the same computer as an individual's desktop computer, it is not recommended. Because MacHTTP is a server application the computer running MacHTTP should be treaded as a server and not somebody's everyday computer. MacHTTP was not necessarily designed to be run on the same computer as somebody's desktop computer. Respect this design feature and MacHTTP will give you many, many happy months of trouble free services in return.

See Also

Bill Doerrfeld, "MacHTTP-Talk Digests"
<URL:http://www.blueworld.com/bluespace/bluedata/MacHTTP-Talk_Search.html>

Brad Schrick, "Brad's WebSTAR/MacHTTP Pages" - [This is the most comprehensive list of MacHTTP/WebSTAR servers available.] <URL:http://brad.net/webstar/>

Chuck Shotton, "How to serve Mac documents" - Information on serving documents from Mac-specific applications like MS Word, Excel, or MacDraw Pro. (This info is somewhat outdated by the addition of MIME types in MacHTTP 1.2.4 and later.)
<URL:http://www.biap.com/machttp/howto_docs.html>

Chuck Shotton, "MacHTTP Technical Reference" - This document is intended to provide a reference to MacHTTP features and capabilities. [This document is also available as a part of the complete MacHTTP 2.2 distribution.]
<URL:http://www.biap.com/documentation/technical_ref.html>

Chuck Shotton, "MacHTTP Questions and Answers"
<URL:http://www.biap.com/documentation/QandA.html>

Chuck Shotton, "MacHTTP Security" - MacHTTP has two techniques for providing access controls. MacHTTP can restrict incoming connections based on the IP address or domain name of the remote client. Also, MacHTTP implements the "Basic" authentication scheme supported by most WWW clients and servers. This method allows you to assign access controls to files or folders that require a remote user to enter a valid username and password before they are allowed to access the information.
<URL:http://www.biap.com/tutorials/Security.html>

Chuck Shotton, "Performance Tuning for MacHTTP" - There are lots of different ways to get MacHTTP to "go real fast." This document discusses a few of the factors that affect MacHTTP's performance and how you can adjust them.
<URL:http://www.biap.com/tutorials/Performance.html>

Eric Bickford and Brad Schrick, "Mac Webmasters Consultant Directory" - "This Web site lists more then 150 consultants available throughout the world who are experts in electronic publishing on the Internet's World Wide Web using Apple Macintosh and compatible computers." <URL:http://www.macweb.com/consultants/>

Grant Neufeld, "Macintosh World Wide Web Frequently Asked Questions"
<URL:http://arpp1.carleton.ca/machttp/doc/>

Ian Andrew Bell, "Macintosh WWW Tools Compendium"
<URL:http://www.arpp.sfu.ca/tools/>

Robert Lentz, "Macintosh Web Programs and Utilties"
<URL:http://www.astro.nwu.edu/lentz/mac/net/mac-web.html>

StarNine, "StarNine Mailing List Maintenance"
<URL:http://emod.starnine.com/starnine/ListSTAR-Administration/Address-List-Editor.html>

StarNine, "WebSTAR by StarNine Technologies" - Based on Chuck Shotton's MacHTTP, WebSTAR(TM) 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(TM) movies. And yet, using WebSTAR is as easy as AppleShare(r). Plus, it's faster than many Web servers running on UNIX. <URL:http://www.starnine.com/webstar/webstar.html>

Stephen Collin, "Web 66" - <URL:http://web66.coled.umn.edu/>

Stephen E. Collins, "Classroom Internet Server Cookbook" - This is cookbook gives the recipes for setting up an Internet server in a classroom.
<URL:http://web66.coled.umn.edu/Cookbook/contents.html>

Verity Inc., "Recent MacHTTP/WebSTAR messages sorted by date"
<URL:http://asearch.mccmedia.com/MacHTTP_Talk/>

Verity Inc., "Virtual Library Search [Macintosh]" - This page allow you to search the archives of the MacHTTP Mailing List.
<URL:http://asearch.mccmedia.com/MacHTTP_Talk/search.html>

MACHTTP CONFIGURATION

This chapter instructs you how to configure MacHTTP.

The next step in understanding MacHTTP is understanding machttp.conf, the server's configuration file. The configuration file defines MacHTTP's setup. It is because the default values contained in machttp.conf are valid for 99% of MacHTTP server administrators, MacHTTP is so easy to get up and running.

The configuration file, machttp.conf, is an ASCII text file residing in the same folder as MacHTTP. You will be able to edit it with any text editor or word processing program. Which ever type of program you use, make sure your edited configuration file is saved as an ASCII text file. If you are using TeachText or SimpleText, then the file will automatically get saved as a text file. If you are using a word processor like MacWrite or WordPerfect, then you must specify in the Save dialog box that you want to save your edits as a text file. This is very important. Otherwise, MacHTTP will not be able to read its configurations and it will not run.

The format of the machttp.conf file is very simple:

Each setting in machttp.conf is called a directive. Directives are case and stylistically insensitive; you can use upper- and/or lower-case letters as well as stylistic formatting in writing your directives. Any line not beginning with a directive is ignored; by convention, the hash mark (#) is used to denote comments not interpreted by MacHTTP. There can be only one directive per line. Each directive is associated with 0 or more values.

DIRECTIVES, THE MEAT OF MACHTTP.CONF

Below each of the directives are defined and examples are provided. Following that there are some sample machttp.conf files with comments and explanations.

VERSION

This is the version number of your copy of MacHTTP. After launching MacHTTP, you can read the version number in title bar. Unless you are using an older version of MacHTTP, this value should be 2.2. If the value of VERSION does not correspond with the version number of MacHTTP, then MacHTTP will complain at startup but will most likely keep running.

Examples:

- VERSION 2.2
- VERSION 2.01
- VERSION 2.0.3tm

INDEX

The INDEX setting specifies the name of the default file name to search for and return to the client application when a specific file is not requested. This specification must be a simple file name and not include a path. In other words, when a client applications specifies a URL only including a folder name like http://152.1.24.177/, then MacHTTP will search and return the file in the specified folder defined by the INDEX directive.

This directive must specify a simple file name; the INDEX directive can not specify a path and therefore it should not contain any colons, the folder delimiter of path statements in the Macintosh operating system.

Examples:

- INDEX default.html
- INDEX index.html
- INDEX home.html

Thus, if a client application requests the URL http:/photos/ from your computer, then MacHTTP will automatically look in the photos folder for the file specified by the INDEX setting. If the file is found, then it is returned to the client. If the file is not found, then the contents of the file specified by ERROR definition are returned.

ERROR

This directive specifies the name of an HTML file to return to the client application when it requests a folder of file that is not found on your server.

Examples:

- ERROR :error.html
- ERROR :file-not-found.html
- ERROR :not-available.html

Thus, if a client application requests the URL http:/photos/chicago.gif, and the file chicago.gif is not found in the folder named photos, then MacHTTP will return to the client application the contents of the file specified in the ERROR directive. To help the readers of your server, it is a good idea to include in your ERROR file instructions on how to contact you as well as pointers to significant parts of your server.

NOACCESS

The NOACCESS directive is an HTML file returned to a client application when it tries to read from a folder or file for which it does not have permission. MacHTTP support various security features defined by the RELM and DENY directives. When somebody tries to access your server who is defined by the DENY directive, or when somebody tries to access your server who is not authorized by the RELM directive, MacHTTP will return the contents of the file specified by the NOACCESS directive.

Examples:

- NOACCESS :noaccess.html
- NOACCESS :access-not-allowed.html
- NOACCESS :permission-denied.html

Like the file specified in the ERROR directive, it is a good idea to include in the NOACCESS file your name and how you can be contacted in case there seems to be a mistake as well as pointers to main aspects of your server.

LOG

The LOG directive specifies the file where a listing of who used your server and what they read is written. If no LOG directive is specified, then MacHTTP will not write to a log file.

Examples:

- LOG :machttp.log
- LOG :logfile.log
- LOG :access.log

TIMEOUT

TIMEOUT is an integer specifying how long MacHTTP will wait to close a connection with a client before it considers the client application inactive. The value should be higher for slower Internet connections.

Examples:

- TIMEOUT 60
- TIMEOUT 30
- TIMEOUT 90

MAXUSERS

MAXUSERS is an integer between 3 and 48, inclusive. In other words, 3 <= MAXUSERS <= 48. It specifies the maximum number of client requests MacHTTP will handle at any given time.

Examples:

- MAXUSERS 10
- MAXUSERS 3
- MAXUSERS 48
- MAXUSERS 20

The larger the value of MAXUSERS the larger amount of memory you will have to allocate to MacHTTP.

MAXLISTENS

MAXLISTENS is an integer between 3 and 48, inclusive, and not greater than the value of MAXUSERS. Expressed algebraically, MAXLISTENS <= (3 <= MAXUSERS <= 48). If your server tends to be busy (constantly receiving connections) and end-users report "Unable to connect" error messages, then this number should be closer to 48. On the other hand, the higher the value of MAXLISTENS, the more RAM must be allotted to MacHTTP. The default value is 5.

Examples:

- MAXLISTENS 10
- MAXLISTENS 3
- MAXLISTENS 48
- MAXLISTENS 5

PORT

PORT is an integer specifying the TCP/IP port where MacHTTP will "listen" for HTTP requests. Ports are like telephone number extensions where a remote computer "dials a telephone number" (computer name or IP address) and then requests services from a particular "extension" (port). Port 80 is the standard port of HTTP requests.

Examples:

- PORT 80

It is not a good idea to change this value to anything but 80. Doing so may either create conflicts with other TCP/IP services (gopher, FTP, mail, etc.) you may have running on your server, or it will make it difficult for people to get to your server because you are not using the standard port. In short, specify PORT as 80.

PIG_DELAY

This is an integer between 0 and 120, inclusive (0 <= PIG_DELAY <= 120). When multiple programs are running on your MacHTTP server, each program allots a period of time when it will give up processing and allow other program to do some processing. This directive defines the amount of time MacHTTP will actually spend processing connections in "ticks" (60ths of a second). The number of programs you have running on your server is inversely proportional to the size of PIG_DELAY. The more programs you have running, the smaller PIG_DELAY should be. The default value is 30 (30 ticks = .5 seconds).

Examples:

- PIG_DELAY 30
- PIG_DELAY 45
- PIG_DELAY 60

DUMP_BUF_SIZE

This is an integer between 256 and 10240, inclusive (256 <= DUMP_BUF_SIZE <= 10240). When transferring files to clients, MacHTTP will divide the file into pieces

specified by DUMP_BUF_SIZE. If your Internet connection is slow (your server is connected to your Internet provider through a modem), then make the number smaller. If you connection is fast (your server is directly connected to the local area network), then you can make the number larger.

Examples:

- DUMP_BUF_SIZEORT 4096
- DUMP_BUF_SIZEORT 10240
- DUMP_BUF_SIZEORT 256

Keep in mind, the smaller the number, the faster MacHTTP will get around to handling multiple connections, but if the number is larger, then file transfers may be speedier. Unfortunately, there seems to be no mathematical way to determine the best value for this directive. At the present time, only experimentation will prove most effective.

NO_DNS

DNS is techno-speak for "domain name services". Domain names are the alpha-numeric names given to computers on the Internet. All computers on the Internet have an Internet Protocol (IP) address, but to make the identities of Internet computer easier to remember, domain name services were created. You have seen these "domain names" before. They look like "trick.lib.ncsu.edu", "www.yahoo.com", or "sumex-aim.stanford.edu".

When a client application connects with MacHTTP, the client applications send along its IP number. When MacHTTP updates its log file (defined with the LOG directive) it can update the log with the client computer's IP address, or MacHTTP can look up the IP address and translate it into a name. Thus, you have the option to have MacHTTP do domain name service lookups or not.

Examples:

- NO_DNS
- #NO_DNS

Incidentally, if you include NO_DNS in your machttp.conf file, then the performance of MacHTTP will be improved because it will have less work to do. On the other hand, your log files may not be as informative as they could be.

ALLOW

This directive, as well as the following directive (DENY), provide rudimentary security features based on the client application's IP number or domain name. These directives "allow" or "deny" access to your entire server based on the identity of the client application's computer.

Follow these guidelines:

The machttp.conf file can contain limitless ALLOW and DENY directives. If no ALLOW or DENY directives are included in your machttp.conf file, then the default behavior is to "allow" every client access. If there is at least one ALLOW or DENY directive, then the default behavior is to "deny" every client access unless they are

defined by an ALLOW directive. IP numbers are read by MacHTTP from left to right. To ALLOW or DENY exact IP addresses, terminate the addresses with a period (.). To ALLOW or DENY a range of IP addresses, do not terminate the addresses with a period (.). Domain names are read by MacHTTP from right to left.

Examples:

- ALLOW .com
- ALLOW .net
- ALLOW .org

In the first example above, every client would be denied access to the server except clients whose domain name ended in .com, .net, or .org. Educational institutions, as well as most of the world would not be allowed to access the server.

Examples:

- ALLOW 152.1

In the second example, only client applications whose IP addresses began with 152.1 would be allowed to use the server. All other clients would be denied access. This example is a good one if you want to restrict access to only the people in your particular institution or department.

Examples:

- DENY 152.1.24.177.
- ALLOW 1
- ALLOW 2
- ALLOW 3
- ALLOW 4
- ALLOW 5
- ALLOW 6
- ALLOW 7
- ALLOW 8
- ALLOW 9

In this example, the computer whose IP address is 152.1.24.177 would not be allowed access, but every other computer would be allowed access.

Examples:

- DENY .com
- ALLOW 1
- ALLOW 2
- ALLOW 3
- ALLOW 4
- ALLOW 5
- ALLOW 6
- ALLOW 7
- ALLOW 8
- ALLOW 9

This set of DENY/ALLOW directives simply denies access to the server if and only if the client application's computer has an domain name ending in .com.

DENY
(See ALLOW above.)

REALM
Where ALLOW and DENY restrict access to your entire server based on Internet names or numbers, this directive (REALM) restricts access to sections of your server based on user names and passwords. The REALM directive is not required, and if you do not want to restrict access to sections of your server, then do not include any REALM directives. You can have any number of REALM directives in your machttp.conf file.

The format of the REALM directive is REALM <match_string> <descriptive_name> where:

Match_string is an alpha-numberic expression that corresponds to some section (folder) of your server's hierarchy to which you want to provide restricted access Descriptive_name is a human readable name to be associated with the restricted access of your server. The descriptive_name is a realm.

In both cases, it is strongly recommended these qualifiers contain no spaces.

Once you have defined one or more realms for your server you must assign user-names and passwords to individuals who are to be granted access to those realms. This is done through the Passwords menu option of MacHTTP's Edit menu or via an AppleScript.

To assign user-names and passwords, first edit your machttp.conf file to include one or more realms. Next, launch MacHTTP. Third, choose Passwords from the Edit menu. A dialog box will be presented to you. At the bottom of the dialog box is the list of realms (descriptive_names) you have previously defined. Select a realm. Next, begin entering user-names and passwords. When you are finished, simply close the dialog box. MacHTTP will save the user-names and passwords in a file called "MacHTTP Settings" for future reference.

Examples:
- REALM in-house employees
- REALM very-private department
- REALM registered-users customers

In each of the examples above, the match_strings (in-house, very-private, and registered-users) are intended to correspond to sections of your server's hierarchy. Every time a client application makes a connection with your server, MacHTTP checks to see whether or not the requested URL contains any match_strings. If it does, then MacHTTP will query the end-user for a user name and password.

Suppose a client application requested the URL http://www.shareware.com/registered-users/. This URL contains one of the match_strings from the example (registered-

users). Consequently, MacHTTP would query the end-user for a user-name and password. If the user-name and password were defined in the "MacHTTP Settings" file, then the end-user would be granted access to this folder of your server. Client applications remember the user-name and password during any one session. Thus, subsequent requests for documents within a realm (http://www.shareware.com/registered-users/updates.html, for example) are handled automatically.

DEFAULT

The DEFAULT directive defines the default transfer type and MIME type for files that have no filename extension.

All filenames within your MacHTTP hierarchy are expected to have suffixes. These suffixes are suppose to unambiguously describe the format of the named file. The process of assigning filename suffixes with file formats is called "suffix mapping." When a filename does not have a suffix (example suffixes include: .txt, .html, .au, .cgi, .acgi) the DEFAULT directive takes effect. The syntax is: DEFAULT <default transfer type> <default MIME type>.

When MacHTTP is requested to deliver a file to a client, there are only two possible transfer types: BINARY and TEXT. BINARY transfers are used for files that can only be read by computers. TEXT transfers are for files that can be read by humans as well as computers. MIME types are universal file-type and encoding-type pairs describing files. MIME types originated out of the need to send files as enclosures through divergent email systems. (More about transfer and MIME types are discussed in the TEXT directive section.)

Examples:

- DEFAULT TEXT text/html
- DEFAULT TEXT text/plain
- DEFAULT TEXT application/mac-binhex40
- DEFAULT BINARY audio/x-aiff

The first example is the one that will be used by the vast majority of MacHTTP administrators. It means if a file does not have a suffix, then the file will be transferred to the client application using the TEXT transfer mode and the file is expected to be ASCII text file formatted in HTML.

The second example assumes files without suffixes are simple text files without any formatting.

The third examples assume files without suffixes are BinHex'd files and the client application should deBinHex the file once it is received.

The last example assumes files without suffixes are audio files of type aiff and should be transferred to the client application in BINARY mode.

TEXT

The TEXT directive, as well as the BINARY, CGI, ACGI, SCRIPT directives, are

used to tell MacHTTP how to handle requests for files with specific filename suffixes. The syntax for each of these directives is similar:

<transfer type> <suffix> <file type> <creator> <MIME type>

where:

• <transfer type> is either TEXT, BINARY, CGI, ACGI, or SCRIPT

• <suffix> is a case-insensitive file name extension like .txt, .cgi, .acgi, or .aiff

• <file type> is a four-letter, case-sensitive code used by the Macintosh operating system to describe the file types like APPL, TEXT, AIFF, or MooV. (These codes can be examined for any particular file by using ResEdit's "Get File/Folder Info..." menu command of the File menu.)

• <creator> is a four-letter, case-sensitive code used by the Macintosh operating system to identify the application that created the file. (These codes can be examined for any particular file by using ResEdit's "Get File/Folder Info..." menu command of the File menu.)

• <MIME type> corresponds to the same universal file-type and encoding-type pairs described in the DEFAULT directive description.

An asterisk (*) can be supplemented for the suffix, file type, or creator parameters.

When a client application requests a particular file from MacHTTP, MacHTTP can only do one of two things with the file: transfer it or run it. To determine which of these actions to take, MacHTTP relies on filename extensions or Macintosh file type/creator codes. You, as the MacHTTP administrator, tell MacHTTP how to handle various filename extensions or Macintosh file type/creator codes through the TEXT, BINARY, CGI, ACGI, and SCRIPT directives.

When MacHTTP receives are request for a file, it examines the file's suffix. If the suffix is mapped to a TEXT or BINARY directive, then the file is transmitted with the associated MIME types in text or binary mode, respectively.

On the other hand, if the suffix is mapped to a CGI, ACGI, or SCRIPT directive, then the requested file is executed and any results the cgi, acgi, or script program return are sent to the client application.

If the requested file does not have an extension, then the files Macintosh file type and creator codes are examined by MacHTTP. If those codes are mapped to the TEXT or BINARY directives, then the file is transmitted along with the associated MIME type. If the codes are mapped to the CGI, ACGI, or SCRIPT directives, then the requested file is run and any results are returned to the client.

If the requests file does not have an extension and the Macintosh file type/creator codes are not mapped to any directive, then the DEFAULT directive is used to transfer the file to the client.

(Whew!)

Examples:

- *TEXT .html TEXT * text/html
- *TEXT .text TEXT * text/plain
- *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
- TEXT .HTML TEXT * text/html
- TEXT .TEXT TEXT * text/plain
- TEXT .TXT TEXT * text/plain
- TEXT .RTF RTFf MSWD text/richtext
- APPL .EXE APPL * text/html
- SCRIPT .SCRIPT TEXT * text/html
- SCRIPT * TEXT ToyS text/html
- TEXT .HQX TEXT BNHQ application/mac-binhex40
- BINARY .SIT SITD * application/x-stuffit
- BINARY .ZIP pZIP * application/zip
- BINARY .au ULAW * audio/basic
- BINARY .AIFF * * audio/x-aiff
- BINARY .GIF GIFf * image/gif
- BINARY .PICT PICT * image/pict
- BINARY .JPG JPEG * image/jpeg
- BINARY .JPEG JPEG * image/jpeg
- BINARY .XBM XBMm * image/x-xbm
- BINARY .TIF TIFF * image/tiff
- BINARY .EPS EPSF * application/postscript
- BINARY .MOV MooV * video/quicktime
- BINARY .MOOV MooV * video/quicktime
- BINARY .MPG MPEG * video/mpeg
- BINARY .WORD WDBN MSWD application/msword
- BINARY .XL XLS3 * application/excel
- BINARY .XLS XLS3 * application/excel
- BINARY .XLS XLS4 * application/excel

BINARY

(See the TEXT directive.)

CGI

(See the TEXT directive.)

ACGI
(See the TEXT directive.)

APPL
(Do not use this directive. It is had been superseded by the CGI and ACGI directives, and it is provided for backwards compatibility only. See the TEXT directive.)

SERVER ENHANCEMENTS AND EVALUATION TOOLS

This chapter reviews various CGI program and utilities enhancing your ability to maintain a WWW server.

After you have your WWW server up an running, you will want to keep it that way. Not only will you want to keep it running smoothly, but you will want to add features to your service that the server application can not provide. The programs listed here will help you achieve these goals.

Most of the programs below are implementations of common gateway interface (CGI) scripts. Some of them create indexes to your server's directory structure. Others are substitutes for the error.html file returned when a file is not found on your server. Email.cgi provides the ability to get feedback about your server (or just about anything else) via email. Others are statistic programs or hit counters, counting the number of times particular pages where read by remote users.

ANNOTATE

Version: 1.5
Author: Phil Harvey
Requirements: MacHTTP 1.3 or later
Cost: free
Pros: easy to set up and customize
Cons: not fast
Remote location: <URL:>
Tricks location: <URL:http://152.1.24.177/teaching/archives/Annotate.cgi.sit.Hqx>

This CGI program allows your readers to add their comments to a list of comments. To use it, just put the annotate.cgi program on your server, edit the supplied annotate.html and annotatePrompt.html files, and finally, point to annotate.html. Its fun and provides a sort of functionality originally designed for the WWW but never seemed to make it very far. Give it a try.

COUNT WWWEBULA

Version: 1.0a9
Author: Gregory S. Combs
Requirements: MacHTTP or later
Cost: free
Pros: easy to set up
Cons: may not work on all machines
Remote location:
<URL:http://www.edb.utexas.edu/greg/ftp/programming/counter/index.cgi>
Tricks location:
<URL:http://152.1.24.177/teaching/archives/CountWWWebula1.0a9.hqx>

This program counts the number of times a particular page has been accessed. By inserting a specialized IMG tag into your HTML documents (like <img src=http://machine.edu/path-to-cgis/counter.acgi?user/searchKey>) Count WWWebula dynamically updates a GIF file and send it back to the client application with the HTML. Fun! The version reviewed here is still an alpha version and many people have reported problems running this program on Power Macintoshes and/or under WebSTAR. Give it a try.

EMAIL.CGI

Version: 3.0.2
Author: Eric Lease Morgan
Requirements: TCP Scripting Additions, Parse CGI OSAX
Cost: free
Pros: requires very little configuration
Cons: does not support international characters nor MIME support
Remote location: <URL:http://www.lib.ncsu.edu/staff/morgan/email-cgi.html>
Tricks location: <URL:http://152.1.24.177/teaching/archives/email-cgi.hqx>

This program was originally designed to supplement WWW browsers that did not support the mailto URL. In this regard, it displayed a simple FORM allowing the reader to supply their name, email address, and a message that would be sent to the server administrator.

This CGI program has matured allowing the server administrator to create just about any FORM with just about any FORM elements and have the output of that FORM be sent to an administrator-defined email address. Consequently, you can use email.cgi to create order forms for products, conduct surveys, create mailing lists, or get feedback on your service.

To get email.cgi to work you must first install the TCP Scripting Addition as well as the Parse CGI OSAX. (It is a good idea to restart your Macintosh after installing these scripting additions.) Then you must create a FORM. You are allowed to create just about any FORM you want as long as it contains fields for a Simple Mail Transfer Protocol (SMTP) server, recipient name and email address, as well as sender name and email address. Email.cgi requires no knowledge of AppleScripting to get up and running.

On the down side, email.cgi is not completely customizable. For example, it generates some output text listing what was received and sent on to the email address. This text can not be defined by the server administrator unless they know how to AppleScript. Another limitation is email.cgi's inability to send documents to multiple recipients.

Email.cgi does what it is suppose to do, but it can be done a bit better. On the other hand, since the script is distributed in an editable state and liberally documented, email.cgi may be a good script to use as a model for other CGI scripts written in AppleScript.

FOLDER-LIST2

Version: 1.0
Author: Brian Monson
Requirements: Tokenize and ScriptTools OSAXs
Cost: free
Pros: works as advertised
Cons: none
Remote location: <URL:http://www.phy.utulsa.edu/Physics/files/folderList2.hqx>
Tricks location: <URL:http://152.1.24.177/teaching/archives/folderList2.hqx>


Folder_list2 is a program creating a list of files from the same folder in which it is saved and returns that list back to the reader. It is a enhancement of one of Jon Wiederspan's CGI tutorial files. For example, you can create an alternative table of contents page or default.html file through the use of this ACGI program. It's output is not overwhelming, but if you have a folder whose contents is constantly changing, then folder_list2 may be a useful script for you.

Alternatively, if you drag a folder on to this script, then it will output the unordered list to you. You may want to then use this list as the beginnings of an HTML page. Its only limitation is that it can not index a folder within a folder.

INDEX CGI

Version: 1.0
Author: Felipe Campos
Requirements: AppleScript
Cost: free
Pros: indexes sub folders
Cons: knowledge of AppleScript necessary for configuration
Remote location: <URL:ftp://edb518ea.edb.utexas.edu/ftp/acgi/index.cgi.hqx>
Tricks location: <URL:http://152.1.24.177/teaching/archives/index.cgi.hqx>


Index CGI is a more robust version of folder_list2. Its purpose is to dynamically create a list of files contained in a folder and return that list as an HTML document. More than folder_list2, Index CGI (index.cgi) is intended to be the file automatically opened when a reader does not specify a particular file from your server. In other words, index.cgi is intended to be the file specified by the INDEX directive of your machttp.conf file.

Unlike folder_list2, Index CGI will index sub folders in your directory structure. Unfortunately, it automatically appends "index.cgi" to the URLs representing folders.

Consequently, if you do not have index.cgi defined as your INDEX attribute, then the links to the folders will not work. Furthermore, you must edit Index CGI to create some of the HTML that it returns. Otherwise, Index CGI will return "broken" images. Give it a try.

Kelly's Error.acgi

Version: 0.6b4
Author: Kelly A. Campbell
Requirements: none
Cost: free
Pros: handles errors well
Cons: none
Remote location: <URL:http://www.spub.ksu.edu/other/machttp_tools/error/>
Tricks location: <URL:http://152.1.24.177/teaching/archives/Error.acgi.0.6b4.sit.hqx>

Kelly's Error.acgi is a replacement for the default error.html file used in your machttp.conf file. By configuring your WWW server to point to error.acgi in the machttp.conf file or the WebSTAR Admin application, many errors caused by invalid URLs can be corrected.

Error.acgi works by intercepting errors returned from your WWW server. It takes the requested URL and validates its path arguments and resolved file aliases. If the only problem with the URL is the lack of a trailing stash, then Error.acgi will append the necessary slash, send a redirection command to the requesting client application, and the correct document will be returned. For example, the following URL is invalid for this distribution because it does not have a trailing slash, but Error.acgi will intercept this error and correct the situation:

```
/teaching/manuscript
```

(Note: if you are not reading this document through a WWW server, then the example above will not work.)

If the requested URL is incorrect altogether, then Error.acgi will return either your standard error.html file, or a file you create by editing the Error.acgi with ResEdit. What is really useful about this server enhancement is that it will return the offending URL as well as the referring URL (if existent) of the referring HTML document. For example, the following URL is also invalid for this server:

```
/teaching/tricks/
```

Error.acgi is a utility well worth your exploration.

MacHTTP AppleGuide

Version: 1.1
Author: Bill Dyer
Requirements: System 7.5, AppleGuide, and MacHTTP 2.0
Cost: free
Pros: documents what and how-to-use MacHTTP
Cons: none
Remote location:
<URL:ftp://192.253.114.3/pub/MacHTTP_stuff/MacHTTPGuide_v1.1.sit.bin>
Tricks location:
<URL:http://152.1.24.177/teaching/archives/MacHTTPGuide_v1.1.sit.bin>


MacHTTP Apple Guide provides online, context sensitive help for MacHTTP through Apple's Apple Guide technology. As you may or may not know, Apple Guide is like Balloon Help on steroids; Apple Guide provides the interface guidelines and resources for developers to create online help modules for applications. Bill Dyer has created such a module.

To install MacHTTP Apple Guide you must simply put it in the same folder as MacHTTP. After launching MacHTTP, you will be able to choose "MacHTTP Guide" from the Guide menu (formally the Balloon Help menu). In standard Apple Guide fashion, you now have the option of reading the help text through its classified or alphabetical index. You can also search the guide for specific topics. As expected, this guide briefly describes how to setup, configure, and interpret MacHTTP. If defines terms and leads you step-by-step through all the procedures to get MacHTTP up and running. For example, it more than adequately describes how to set up security realms by showing you how to all the necessary dialog boxes and fields you must complete. In some cases, the guide relies on documents on the Internet to explain details. It accomplishes this by launching Netscape and opening up a URL to specific pages.

MacHTTP Apple Guide is a professional, well-deserved addition to MacHTTP.

RANDOMGIF.ACGI

Version: 1.0
Author: Felipe Campos
Requirements: AppleScript
Cost:
Pros: demonstrates innovative scripting
Cons: serves little utility
Remote location: <URL:http://edb518ea.edb.utexas.edu/felipe.html>
Tricks location: <URL:http://152.1.24.177/teaching/archives/randomgif.acgi.hqx>


This little script displays random GIF images from a directory in your server's directory structure. Written in AppleScript, you must edit it to fit your particular site. The program works and effectively creates a sort of slide show. It is a good demonstration of what AppleScripting can do with files when combined with your Web server. Other than that, in its present form, it serves little utility. Give it a try.

REDIRECT.ACGI

Version: 1.0
Author: Net.Dreams
Requirements: OSAX Acme Script Widgets and Decode URL
Cost: $10 (shareware)
Pros: successfully redirects invalid URLs
Cons: does not work with Error.acgi
Remote location: <URL:http://www.netdreams.com/net.dreams/software.html>
Tricks location: <URL:http://152.1.24.177/teaching/archives/redirect.hqx>


Redirect is a utility to help preserve the integrity of your server even when you move files in your server's directory structure. To do this, it first maintains a list of invalid URLs on your server and maps them to valid URLs. Second, it traps requests for these invalid URLs, and if they exist in Redirect's list, it then sends a redirection command back to the requesting client application with the correct URL. If the URL does not exist in Redirect's list, then the default error file is returned instead.

To get Redirect to work you first must install the Tokenize, Replace, Offsets Of, and Decode URL OSAX. Next, you must reconfigure your machttp.conf file instruction MacHTTP what file to return when an error exists. In other words, you must redefine the ERROR directive to the location of "redirect.acgi." Third, configure Redirect by accessing it through your server with the redir__admin path arguments. This configuration process allows you to create your table of invalid/valid URLs. It also allows you to specify your default error.html file. (As an

experiment, Kelly's Error.acgi program was defined for this purpose, but Error.acgi did not work as expected since Redirect did not send it AppleEvents; Redirect simply calls files without any arguments.) The last step in using Redirect is to create a REALM directive protecting redirect.acgi eliminating the possibilities for other people configuring your redirection tables.

Redirect is a good program for you to use if you change the names and/or locations of the files on your server.

ServerStat Lite

Version: 1.0.4
Author: J. Eric Bush
Requirements: MacHTTP, WebSTAR, or gopher log files
Cost: $20 (shareware)
Pros: successfully analyses log files
Cons: does not create tab-delimited output
Remote location: <URL:http://www.ericse.ohio-state.edu/kitchen-sink/ss.html>
Tricks location: <URL:http://152.1.24.177/teaching/archives/server-stat-lite.hqx>


This well documented and supported piece of shareware analyses your server's log files. In fact, it supports three log file formats: gopher, MacHTTP, and WebSTAR. Like its predecessor, WebStat, ServerStat reports statistics on:

- Day of the year
- Hour of the day
- Day of the week
- Client domain
- Client reversed subdomain
- Archive section

Unlike WebStat, ServerStat provides you with many more options for reporting on your statistics. For example, it allows you to set date ranges. You can use "filters" to include or exclude parts of your log when processing. For example, you can use this feature to exclude the server administrator's IP address. To improve performance, you can turn on or off an option called "fastmode". If turned on, ServerStat simply doesn't share your computer's microprocessor with other applications until it is done with it work. To generate these various options, all you have to do is select them from the menu. Once selected, you can then save these configurations for later use.

Similar to WebStat, this program does not export any data in a format suitable for importing into spreadsheet programs or database applications. While the output can either be an HTML file or a simple text file, these files will need some massaging in order to put them into spreadsheets or database programs successfully.

WebSTAR/MacHTTP Indexer

Version: 2.2
Author: David Habermann
Requirements: System 7.1.3 with the scriptable Finder, Script Tools
Cost: free
Pros: indexes your file structure completely
Cons: requires careful configuration
Remote location: <URL:http://www.dow.com/~haberman/wsi/wsi.html>
Tricks location:
<URL:http://152.1.24.177/teaching/archives/webstarindexer2.2.sit.hqx>

This is another utility used to create an index of your WWW server's file structure. It has been written in both AppleScript as well as Aretha. Only the AppleScript version is discussed here.

WebStar/MacHTTP Indexer does require a bit of configuring to run. First, within the Index All Folders script, it is a good idea to define NoIndexFolders to contain the folders you do no want indexed. For example, you may not want your root folder indexed as well as a folder that contains restricted information. Thus, you may want to define NoIndexFolders as {"", "restricted"} assuming you have a folder called "restricted". You may also want to define DebugSys so the indexer creates a report of its activities and you may want to define SaveLog so old logs get deleted automatically. Finally, since WebStar/MacHTTP Indexer is really a set of two programs you will want to define Using WebSTAR in the WebSTAR Indexer script to be true or false depending on whether or not your server is MacHTTP or WebSTAR.

After running the Index All Folders script, a new file will be created in the root directory of your server. This file will be named "index.html" and will contain a list of all the files in your server's hierarchy. If you copied the supplied icons folder to your root, then each file will be accompanied by a representative icon as well. Depending on how many documents you have on your server, the creation of index.html could take quite a long time.

Like similar scripts in its class, this program is intended to eliminate the "file not found" errors that sometimes occur; the index.html file is intended to be displayed by the server when the server can not find a requested file. Unfortunately, WebStar/MacHTTP Indexer assumes the default file of every folder is "index.html". If you have defined your default differently, then WebStar/MacHTTP Indexer points to incorrect files when folders are selected. Three solutions are suggested for the problem:

1. Change your default file setting to "index.html"
2. Use the supplied error.cgi program to point to the proper file
3. Don't index any folders

A better solution would be a configuration allowing you to tell WebStar/MacHTTP Indexer what your default file was.

In summary, WebStar/MacHTTP Indexer does what it says it does, but you get what you pay for. If you can script AppleScript, then you could modify this script to better suit your needs. WebStar/MacHTTP Indexer would then be a robust program. Otherwise, you might want to try some other solution.

WEBSTAT

Version: 2.3.4
Author: Phil Harvey
Requirements: MacHTTP, WebSTAR, or gopher log files
Cost: free
Pros: easily analyses your log files
Cons: rather un-Macintosh-like and doesn't generate tab-delimited files
Remote location: <URL:>
Tricks location: <URL:http://152.1.24.177/teaching/archives/webstat.sit.hqx>

 This program helps you analyze your server's log files. It is designed to be stored in the same folder as your log file and run on a periodic basis. If you have not changed any of its configurations, then all you have to do is launch it and WebStat will do the rest.

The output of Webstat, by default, is a file named WebStat.html. This default, as well as others, can be changed by editing the WebStat.config file. The final report (WebStat.html) lists statistics by:

- Day of the year
- Hour of the day
- Day of the week
- Client domain
- Client reversed subdomain
- Archive section

Each of these mini-reports are sorted according to the number of files transmitted as well as the number of bytes sent. Percentages of the whole are then calculated as well. Consequently, you can use WebStat to determine what dates, hours, weeks are more popular for you server. You can also determine who is using your service and what pages they are reading.

If you don't want particular parts of the report, then you must edit WebStat's WebStat.format file. This is easy. All you have to do is delete or add the parts you want. The process is rather un-Macintosh-like, but its not difficult. One very nice thing about WebStat is that it will try to resolve IP numbers into domain names. This process will give you more meaningful reports.

There is at least one way this program could be improved. Its output could be put into some sort of tab-delimited format making it easy to import into a spreadsheet program. If this were

done, then further analysis may be more easily accomplished like determining which of your files are most popular and/or which of your files are not being read at all.

HTML IN SEVEN LESSONS

This chapter provides and overview of the hypertext markup (HTML) language.

The hypertext markup language (HTML) is used to format text documents delivered by Web servers. Using HTML you can format documents in such a was to communicate your ideas both in words as well as pictures.

Entire tomes have been written about HTML. Subjectively speaking, HTML has to be one of the World Wide Web's more popular subjects. Since information about HTML is so common on the Internet, this chapter will only provide an outline of the topic. You are encouraged to read the formal, evolving standard (Tim Berners-Lee and Dan Connolly, ed., "HyperText Markup Language (HTML)"), participate in the Usenet newsgroup comp.infosystems.www.html for the latest information, and peruse the See Also references at the end of this section for more information.

Since the Web employs the client/server model, there is little control over the fonts and styles of formatted text at the client end. Therefore, HTML provides logical rather than stylistic formatting capabilities. Put another way, HTML files are simple ASCII files containing rudimentary "tags" describing the content of a document. HTML's purpose is not to imitate or mimic a word processor; HTML was originally based on the standardized generalized markup language (SGML), and SGML is used to provide a logical structure to files. This logical structure is intended to allow readers (as well as computers) to locate and interpret types of information in a document. For example, the logical structure may easily bring out types of information like:

- Author-defined important topics and headings
- Email addresses
- Citations
- Quotations
- Examples

Unlike SGML, HTML provides a mechanism for connecting (linking) multiple documents together. This linking technique, first articulated by Vannevar Bush in 1945, represents the "hypertext" feature of the hypertext markup language.

As the use of HTML is becoming more widespread, there is a greater and greater demand for HTML tags describing a document's stylistic elements as well as document's content. This has resulted in tags used to center text on a page, make images bigger or smaller, define how lists of items are rendered, etc. This demand for formatting tags stems from people who believe that the presentation of an idea is just as important (if not more important) as the idea itself; "The medium is the message."

This demand for formatting tags has also generated "non-standard" HTML codes only rendered by a few WWW browsers. Netscape is the most notable of these browsers. These enhanced HTML codes are not necessarily bad. In fact, used correctly, these codes can be used to create quite stunning electronic documents.

The pages following will only elaborate on the basic HTML codes, but your decision to use enhanced, WWW browser-specific HTML codes should be based on your intended audience. If you know your readership is using a particular WWW browser, then by all means, put your best foot forward and use those specialized tags. This may be the case when you are writing for internal corporate readers who are expected to use the supported software.

On the other hand, if your readership is using a wide variety of WWW browser applications including text-based browsers or unknown browsers, then it behooves you to compose your documents in such a way that you don't alienate some of your intended audience with unreadable documents, and don't compose your documents so that their meaning is lost if the reader is not using a particular browser.

SEE ALSO

"Guides to Writing HTML Documents" - "These guides are about writing style, which HTML constructs to use when, when to divide up documents into multiple parts, etc. Also see information on the HTML language itself, learning HTML, editors, and converters." <URL:http://union.ncsa.uiuc.edu/HyperNews/get/www/html/guides.html>

"ANSI/ISO 8859-1 Coded Character Set" - "This list, sorted numerically, is derived from ANSI/ISO 8859-1 8-bit single-byte coded graphic character set." <URL:http://www.utirc.utoronto.ca/HTML_2.0/html-spec_11.html#SEC90>

Bill Spurlock, "HTML Writers Guild Website" - The Guild provides information resources for members in the form of FAQs (Frequently Asked Questions) and answers to those questions. FAQs are especially helpful to new Guild members and beginners in the art of Web authoring. <URL:http://www.mindspring.com/guild/>

David Siegel, "David Siegel's High Five" - This site demonstrates what can be done with HTML and Netscapisms. <URL:http://www.best.com/~dsiegel/high_five/high_five.html>

Eamonn Sullivan, "Crash course on writing documents for the Web" - "wrote this to help people in PC Week Labs to write documents for our web server. It's designed for someone who wants to put a page on the web but could care less about most of the technical details and doesn't want to read a book. I hope it's helpful." <URL:http://www.ziff.com:8002/~eamonn/crash_course.html>

Eric Tilton, "Composing Good HTML" - "As the Web continues to explode in its own inimitable fashion, it is becoming more and more important to write HTML that conforms to certain guidelines. Specifically, with the current diversity of clients for the Web (and we can only expect to see more!), it's become important to write HTML that will look good on any client, and not just on the specific client which the author may have access to." <URL:http://www.cs.cmu.edu/~tilt/cgh/>

Hakon W Lie, "HTML Style sheets" - "This page contains pointers to information about style sheets in the context of HTML. A mailing list, www-style@w3.org, has been started to host discussions on this topic. Feel free to add/delete yourself or browse the archive." <URL:http://www.w3.org/hypertext/WWW/Style/>

HTML Writers Guild, "Principles of Good HTML Design" - This is a list of ``general principles" of quality HTML design. It is intended to educate HTML authors to the elements of good and bad HTML style. It does not seek to ``control" Guild members, but rather to encourage them to adopt these practices in their everyday HTML construction. <URL:http://ugweb.cs.ualberta.ca/~gerald/guild/style.html>

Iam Graham, "HTML Documentation" <URL:http://www.utirc.utoronto.ca/HTMLdocs/NewHTML/htmlindex.html>

Jeff Barry, "The Hypertext Markup Language (HTML) and the World-Wide-Web: Raising ASCII Text to a New Level of Usability." *The Public Access Computer Systems Review*, 5 no. 5 (1994):5-62 <URL:ftp://ftp.lib.ncsu.edu/pub/stacks/pacsr/pr-v5n05-barry-hypertext>

John Price-Wilkin, "Using the World Wide-Wide Web to Deliver Complex Electronic Documents: Implications for Libraries." *The Public Access Computer Systems Review*, 5 no. 3 (1994): 5-21 <URL:ftp://ftp.lib.ncsu.edu/pub/stacks/pacsr/pr-v5n03-price-wilkin-using>

Larry Aronson, *HTML Manual of Style* (Ziff-Davis Press: Emerville, CA 1994) - This is one of the first HTML books to appear on the market.
Laura Lemay, Teach Yourself Web Publishing with HTML in a Week (SAMS Publishing: Indianapolis, IN 1995) - This easy-to-read book is thorough and complete.
Martin Ramsch, "iso8859-1 table" - [This is a list of the codes used to generate special characters in HTML.] <URL:http://www.uni-passau.de/~ramsch/iso8859-1.html>

National Center for Supercomputing Applications, "A Beginner's Guide to URLs" <URL:http://www.ncsa.uiuc.edu/demoweb/url-primer.html>

NCSA, "HTML Primer " <URL:http://www.ncsa.uiuc.edu/demoweb/html-primer.html>

Robert Lentz, "Macintosh Web Programs and Utilties" <URL:http://www.astro.nwu.edu/lentz/mac/net/mac-web.html>

Tim Berners-Lee and Dan Connolly, ed., "HyperText Markup Language (HTML)" - "HTML is a markup language for hypertext which is understood by all WWW clients. Here we discuss the HTML language, i.e. its syntax and semantics, including information on the history of the language, status of the standard, and development issues."
<URL:http://www.w3.org/hypertext/WWW/MarkUp/MarkUp.html>

Tom Magliery, "Mag's Big List of HTML Editors"
<URL:http://sdg.ncsa.uiuc.edu/~mag/work/HTMLEditors/>

Tom Savola with Alan Westenbroek and Joseph Heck, *Special Edition Using HTML* (Que: Indianapolis, IN 1995) - This book, while rather expensive described in detail HTML as well as providing many pieces of software and documentation on its CD-ROM.
Wm. Dennis Horn, "HTML Documents: A Mosaic Tutorial"
<URL:http://fire.clarkson.edu/doc/html/htut.html>

Yale Center for Advanced Instructional Media, "Yale C/AIM WWW Style Manual" - This is one of the more scholarly treatments of the subject.
<URL:http://info.med.yale.edu/caim/StyleManual_Top.HTML>

BASIC HTML TAGS

This section describes the rudimentary HTML tags.

Creating HTML documents is a lot like using the old word processing program WordStar. (Remember WordStar?) For example, to print a word in boldface type using WordStar, the user would first select text from the screen. Then the user would enter a code like "^b." This code would be inserted before and after the selected text. When the document was printed, WordStar would interpret the "^b" and print boldface letters until another "^b" was encountered.

HTML works in a similar fashion. The author goes through his or her document surrounding text with special codes denoting content. As HTML editing software matures, there is less and less of a need to know HTML tags, but until this comes to complete fruition, it is a good idea to have a knowledge of the HTML codes themselves.

RESERVED CHARACTERS

HTML tags are always delimited with the less than (<) and greater than signs (>). Therefore when you want to use these symbols in your HTML document, you need a special way of displaying those characters. This is why the less than and greater than symbols are reserved characters.

If you want to insert the less than or greater than symbols into your text, then you need to insert an "entity" representing this symbols instead. The entity for less than (<) is < and the

entity for greater than (>) is >. Consequently, when you insert "<" or ">" into your text, then the less than or greater than symbols will be displayed.

Similarly, since the ampersand (&) is used as an "escape" character for entities, there need to be a way of "escaping the escape." Thus, the ampersand can be displayed using the entity "&". The double-quote is another symbol that is used in HTML tags and needs to be escaped as well. Consequently, if you want to insert quote marks into your HTML documents, then you should use the entity "&quot" instead. (Fortunately for the people who create HTML by hand, many WWW browser do not get confused when quote marks appear in the HTML code.)

There are a whole slew of additional entities you can use in your HTML documents. Many of these entities represent non-English characters. You are encouraged to read Appendix A for a list.

Structure Tags

The basic structure of an HTML document looks like this:

```
<HTML>
<HEAD>
<TITLE>My first HTML document</TITLE>
</HEAD>
<BODY>
Hello, World!
</BODY>
</HTML>
```

All tags are case-insenstive. Consequently, each of the tags here (<HTML>, <HEAD>, <TITLE>, </TITLE>, <BODY>, </BODY>, and </HTML>), as well as any other tag, can be written in lowercase letters or even a combination of upper- and lower-case letters. Furthermore, all carriage return sequences are ignored by WWW browsers (with one exception described later). Thus, the text above could have been written like this and will be rendered exactly the same way:

```
<html><head><title>My First HTML Document</title></head><body>Hello,
World!</body></html>
```

The most important thing about this example is the tags themselves. They are used to outline the basic HTML document. The <HTML> and </HTML> tags define the document as an HTML document; the <HEAD> and </HEAD> tags denote the leading matter of a document; the <TITLE> and </TITLE> tags specify the document's title; and the <BODY> and </BODY> tags specify the location of your content. Notice how the second tag of each tag pair is identical to the first tag except the second tag includes a backward slash ("/"); the backward slash denotes the completion of a logical formatting option.

Creating Paragraphs

Since WWW browsers (in most cases) ignore blank lines, HTML provides a tag to denote paragraphs, <P>.

The paragraph tag inserts a blank line between the text proceeding the tag and after the tag. Do not expect to insert multiple paragraph marks into our documents and expect to get multiple blank lines. That is the purpose of the next tag.

Line Breaks

The use of the line break tag (<BR>) is one way to force a carriage-return and line feed effect into your documents. It is very useful when you want to make sure a sequence of lines always begin at the left-hand side of page and you don't want blank lines between each item in the sequence. This was the technique used in the very first example above. The HTML looks like this:

```
<HTML><BR>
<HEAD><BR>
<TITLE>My first HTML document</TITLE><BR>
</HEAD><BR>
<BODY><BR>
Hello, World!<BR>
</BODY><BR>
</HTML><BR>
```

If you want to create a sequence of blank lines in your documents, then insert a sequence of line breaks as in

.

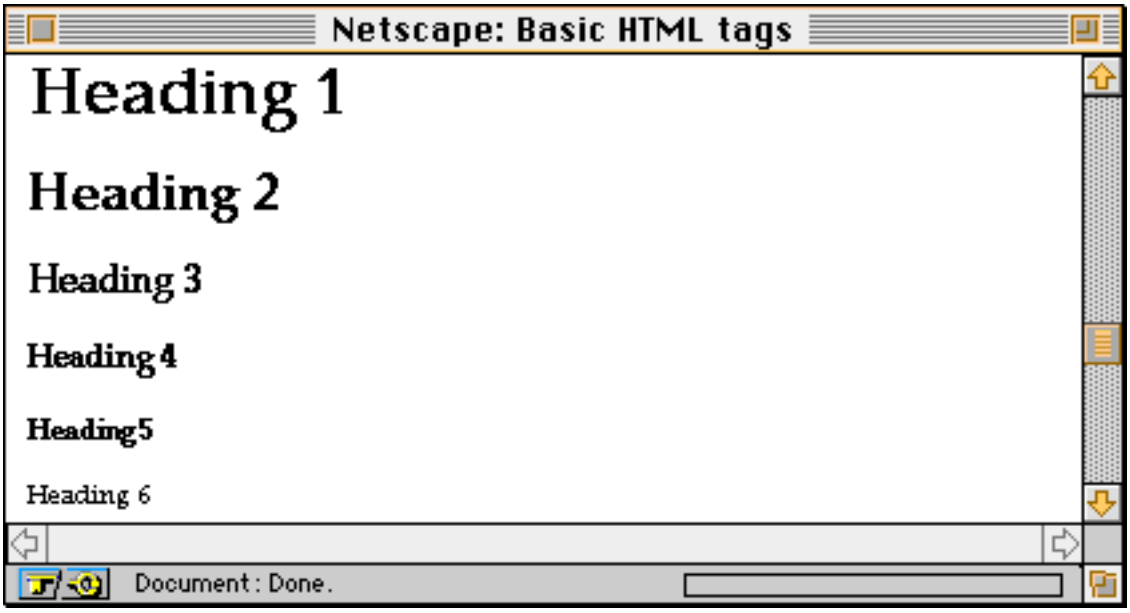
Headings

Headings are a means of marking off sections in your document. There are six levels of headings numbering 1 through 6. The first heading (<H1></H1>) is meant to be used to denote the broadest heading. Whereas the sixth heading (<H6></H6>) is used to denote the most specific.

Like most of the HTML tags, these tags are intended to surround lines of text, usually a single line. Here are six lines using the headings tags:

```
<H1>Heading 1</H1>
<H2>Heading 2</H2>
<H3>Heading 3</H3>
<H4>Heading 4</H4>
<H5>Heading 5</H5>
<H6>Heading 6</H6>
```

Here is how those codes get rendered:



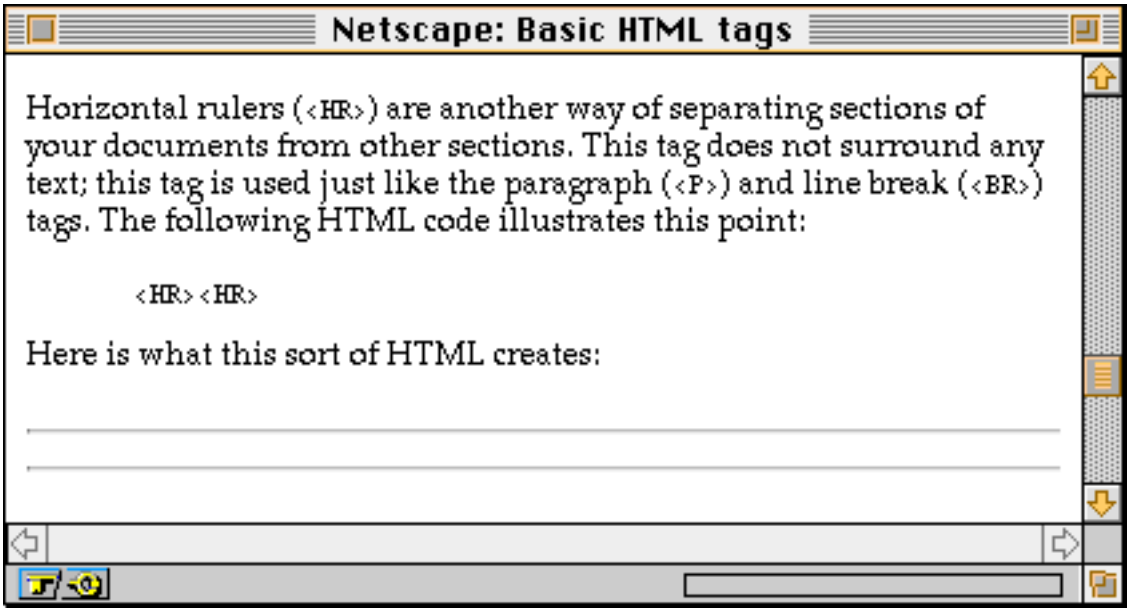
Just because heading tags render text in large fonts, do not use heading tags to denote emphasis or important concepts. Use heading tags to create headings.

Horizontal Rulers

Horizontal rulers (<HR>) are another way of separating sections of your documents from other sections. This tag does not surround any text; this tag is used just like the paragraph (<P>) and line break (<BR>) tags. The following HTML code illustrates this point:

```
<HR><HR>
```

Here is what this sort of HTML creates:



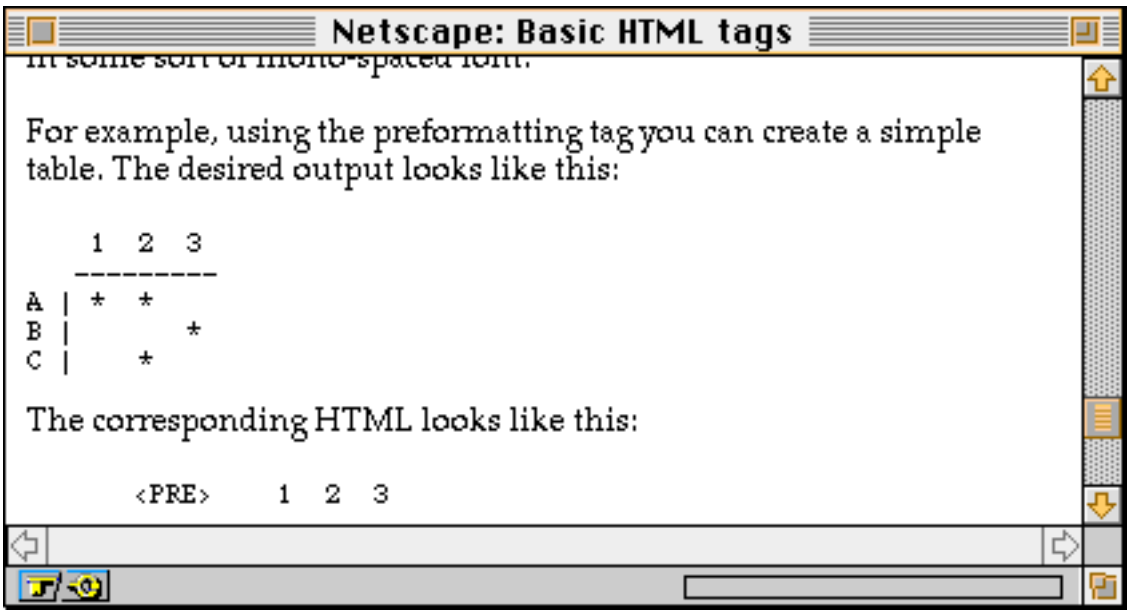
Judicious use of the <HR> tag makes your HTML documents easier to read.

Preformatted Text

There are times when you will want to display some sort of table or chart in your HTML documents. One way to satisfy this need is with <PRE>...</PRE>, the preformatted tag.

By surrounding text with the preformatted tag in your HTML, the carriage returns and spaces within the text will be retained. Unfortunately (or fortunately), the text will most likely be displayed in some sort of mono-spaced font.

For example, using the preformatting tag you can create a simple table. The desired output looks like this:



The corresponding HTML looks like this:

```
<PRE>    1  2  3
   ---------
A | *  *
B |       *
C |    *</PRE>
```

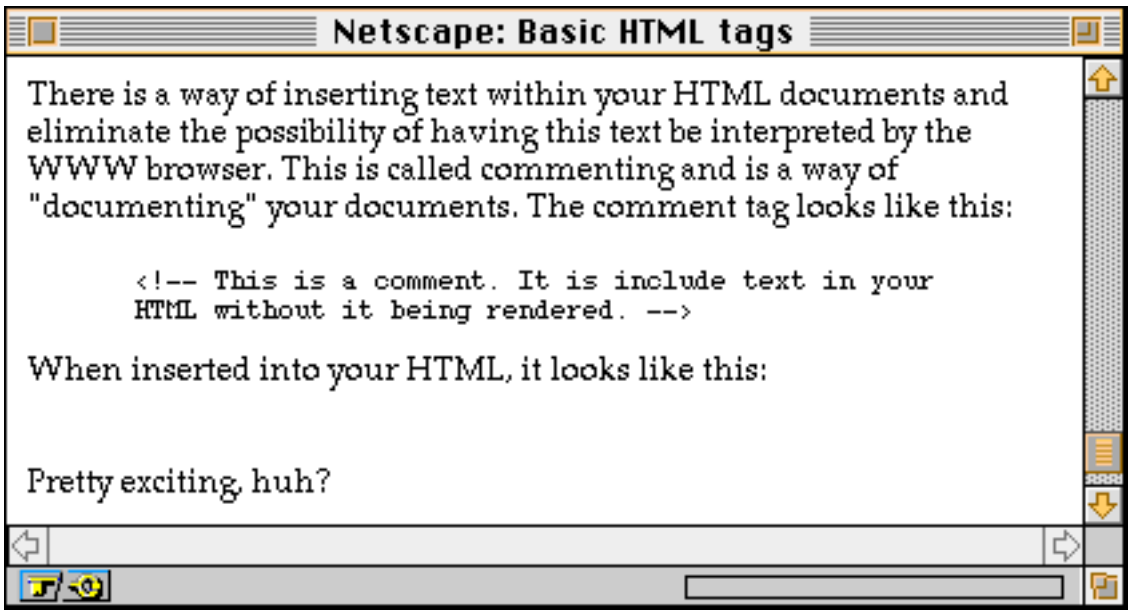
Remember, when you use the <PRE> tag, your WWW browser retains the spaces and line breaks of the formatted text.

COMMENT TAGS

There is a way of inserting text within your HTML documents and eliminate the possibility of having this text be interpreted by the WWW browser. This is called commenting and is a way of "documenting" your documents. The comment tag looks like this:

```
<!-- This is a comment. It is include text in your HTML without it
being rendered. -->
```

When inserted into your HTML, it looks like this:



Pretty exciting, huh?

CREATING HYPERTEXT LINKS

This section describes how to put hypertext links into your HTML documents.

The real utility of HTML is not its ability to format text. Rather, its real strength lies in its ability to transport a user from one section of text to another (or to a completely new document) by clicking on (or selecting) highlighted words or graphics.

The hypertext features of HTML are implemented with tags called "links." There are three types of this tags: links to other documents, links to specific parts of the same document, or links to specific parts of other documents.

LINKS TO OTHER DOCUMENTS

To create a link to another document, you must associate text (or graphics) with a URL. This is done with the anchor tag. In this case, the anchor tag takes the following form:

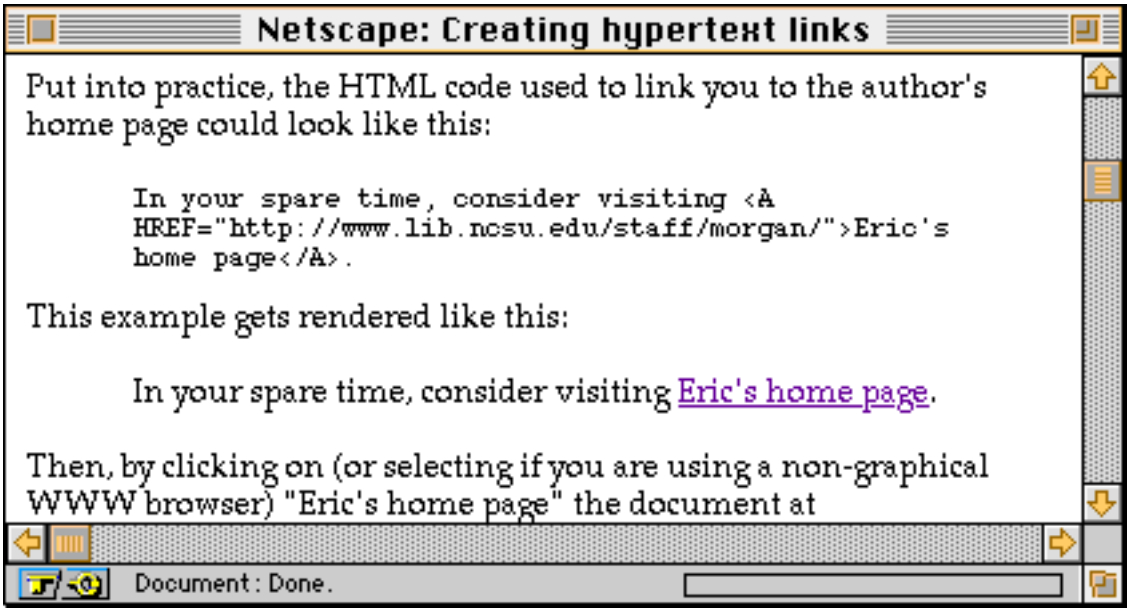
```
<A HREF="URL">text</A>
```

The "A" denotes an "anchor" and the "HREF" denotes a hypertext reference. Note the URL is enclosed in quote marks. While many browsers do not require the quote marks, future browsers may not be so lenient and it is in your best interest to include them.

Put into practice, the HTML code used to link you to the author's home page could look like this:

```
In your spare time, consider visiting <A
HREF="http://www.lib.ncsu.edu/staff/morgan/">Eric's home page</A>.
```

This example gets rendered like this:



Then, by clicking on (or selecting if you are using a non-graphical WWW browser) "Eric's home page" the document at http://www.lib.ncsu.edu/staff/morgan/ is presented to you.

One of the nicest things about the World Wide Web is that you can "link" all sorts of Internet resources to your documents. To do this all you have to do is include URLs in your anchors denoting these other resources. Thus, you can link gopher servers, FTP sites, WWW servers, as well as other Internet resources all within your documents. This is what makes the World Wide Web a "web" in the first place. For example:

In your spare time, consider visiting the <A HREF="gopher://gopher.lib.ncsu.edu/">gopher server</A>, the <A HREF="http://www.lib.ncsu.edu/">WWW server</A>, and <A HREF="ftp://ftp.lib.ncsu.edu/">FTP server</A> of the NCSU Libraries.

When creating hypertext links, there is at least one stylistic thing to remember. Avoid the "click here" syndrome. HTML was designed to be read as well as printed. Your documents should be able to stand on their own and make just as much sense in their hypertext form as their printed form. Creating HTML with "click here" is considered bad form.

Relative URLs

Including the complete URL of every link in your documents can be tedious. More importantly, including the complete URL of every link in your documents makes them difficult to move from folder to folder, or even server to server. This is why the concept of "relative links" was invented.

Relative links are partial URLs. They include just enough information necessary for the reader's WWW browser to make the connection to the linked document. This usually means you do not have to include the scheme and host name in the URL and sometimes the path and file names as well. (Of course if you eliminated all of these items, then you wouldn't have a URL at all.) The URL information not included in the anchor is assumed by the client application.

Suppose there was an HTML document on a WWW server. The server's address is library.ncsu.edu. Furthermore, this particular document was named instructions.html and was saved in the folder (relative to the WWW server application) scripts:searching. The complete URL for this particular document would be:

```
http://library.ncsu.edu/scripts/searching/instructions.html
```

Put into an HTML document as an anchor, this URL may look like this:

```
Searching <A
HREF="http://library.ncsu.edu/scripts/searching/instructions.html">inst
ructions</A> are available online.
```

Now assume this HTML code were in a document saved on the computer library.ncsu.edu. If so, then the HTML document could have been written using a relative link:

```
Searching <A
HREF="/scripts/searching/instructions.html">instructions</A> are
available online.
```

This works because the WWW browser "remembers" the scheme and hostname of the document presently being displayed, and when a link is accessed not containing the scheme or hostname (a relative link), then the scheme and hostname of the present document are assumed.

Let's extend the assumption. Assume the HTML document being written was saved in the "scripts" folder. If so, then the URL of the anchor could have been:

```
./searching/instructions.html
```

In this case the period (.) in the URL denotes the present folder. Put another way, the period represents the folder containing the currently displayed document.

If the presently displayed document resides a folder scripts:email, then the instructions document could be referenced with the following relative URL:

```
../scripts/instructions.html
```

In this case the double periods (..) tell the WWW browser to go back one folder level and then down into the scripts directory for the document instructions.html. The use of the double period (unlike the single period) can be used a number of times in the same URL. Thus, you can tell the WWW browser to move back a number of folder levels from the present folder and then back down to a specified file. For example,

```
../../../scripts/instructions.html
```

Finally, if the document you want to link to is in the same directory as the presently displayed document, then specifying that link in your anchor simply requires the file's name. For example, suppose you were reading a document residing in the scripts folder, then the simplest URL would be just the file name. The HTML markup could look like this:

```
Searching <A HREF="instructions.html">instructions</A> are available
online.
```

The real advantage of relative URLs, besides the increased readability of your native markup, is the transportability of your HTML documents. If you include full and complete URLs in all your HTML, then moving those same documents to a computer with a different hostname will require you to change all our URLs to the new hostname. Similarly, if your always include the full path name (folder structure) in your URLs, then when (and if) you have to move your documents to a different folder, you will have to edit your HTML documents again. All of this can be avoided if you use relative links as much as possible.

LINKS TO SPECIFIC PARTS OF THE SAME DOCUMENT

You can also create hypertext links to other parts of the same document being displayed. This is done with "named" anchors. Long, single page documents usually have some sort of table of contents at the beginning. These tables of contents are done using named anchors.

To used named anchors, you must first mark up an item in your document to link to. This is done with a variant of the anchor HTML code and may look like this:

```
<A NAME="3">Section 3</A>
```

Here the NAME defines the name of the anchor. It can be any text you like, but the first word of the text must be unique within the document and it is case-sensitive. The text to be linked to is surrounded by the <A>...</A> tags.

The next step is to insert hypertext references into your HTML document directing the reader to the named anchors. URLs containing number signs (#) are used for this purpose. For example, a table of contents could be written like this:

```
<H1>Table of contents</H1>
<A HREF="#1">Section 1</A><BR>
<A HREF="#2">Section 2</A><BR>
<A HREF="#3">Section 3</A><BR>
.
.
.
<A NAME="1">Section 1</A><BR>
<A NAME="2">Section 2</A><BR>
<A NAME="3">Section 3</A><BR>
```

Consequently, when the reader selects "Section 1", the WWW browser will interpret the anchor as a named reference (#1) and move to the location in the document marked with the number 1 by the NAME anchor.

Links to specific parts of a different document

Combining the hypertext reference and the named anchor, you can even create links to specific parts of other documents. Creating the named anchor is done just as before. Surround the text to link to with the anchor tag using the NAME attribute.

Creating the hypertext link to the named anchor then requires you to specify the file to link to as well as the NAME to link to. Consequently, a URL of this sort may look like any of the following:

- http://library.ncsu.edu/scripts/searching/instructions.html#step01
- /scripts/searching/instructions.html#step01
- ./searching/instructions.html#step01
- ../searching/instructions.html#step01
- ../../searching/instructions.html#step01

Obviously, relative links can play an essential role in the use of named anchors to specific parts of other documents.

Logical tags

This section describes the logical tags denoting informational elements of your HTML documents.

HTML offers a number of tags intended to logically describe the content of your documents. The use of these tags is recommended over the use of the purely stylistic tags even though the output may be exactly the same. This is recommended because in the future you may be able to search documents based on their content markup. For example, sometime in the future, you may be able to use your WWW browser to search a document for all the "addresses" equal to

"eric_morgan@ncsu.edu" or even the "citations" containing the words "review of."

The most used logical tags include:

Address (<ADDRESS>...</ADDRESS>)
This tag is intended to surround email addresses.

Blockquote (<BLOCKQUOTE>...</BLOCKQUOTE>)
Use the tag when you are reproducing a long quote of another document.

Citation (<CITE>...</CITE>)
This tag is used to enclose citations like the titles of magazines, journals and/or books.

Code (<CODE>...</CODE>)
Code is for "computer code"; use this tag to markup computer programs.

Definition (<DFN>...</DFN>)
Use this tag to delimit the definitions of terms.

Emphasis (<EM>...</EM>)
Use this tag when a particular idea is important.

Sample (<SAMP>...</SAMP>)
This tag is used for examples.

Strong emphasis (<STRONG>...</STRONG>)
The strong emphasis is a tag used to denote an idea even more important than emphasis.

Variable (<VAR>...</VAR>)
Variable is another programming convention. It is used to markup variables from computer programs.

Stylistic tags

This section describes the stylistic tags for your HTML documents.

Stylist tags are used to add word processing-like formatting to your documents. Instead of using these tags, you are encouraged to use the logical tags from the previous section. The greatest disadvantage of using these tags is they may not be able to be rendered very well by text-based WWW browsers. For example, in a text-based browser, all fonts are displayed as mono-spaced fonts (typewriter), and it is not possible to display fonts in italics.

Bold (<B>...</B>)
Use this tag when you want the text bigger and blacker than normal.

Italic (<I>...</I>)
This tag is used to slant your fonts to the right.

Typewriter (<TT>...</TT>)
Typewriter is used to display the encoded text in a mono-spaced font.

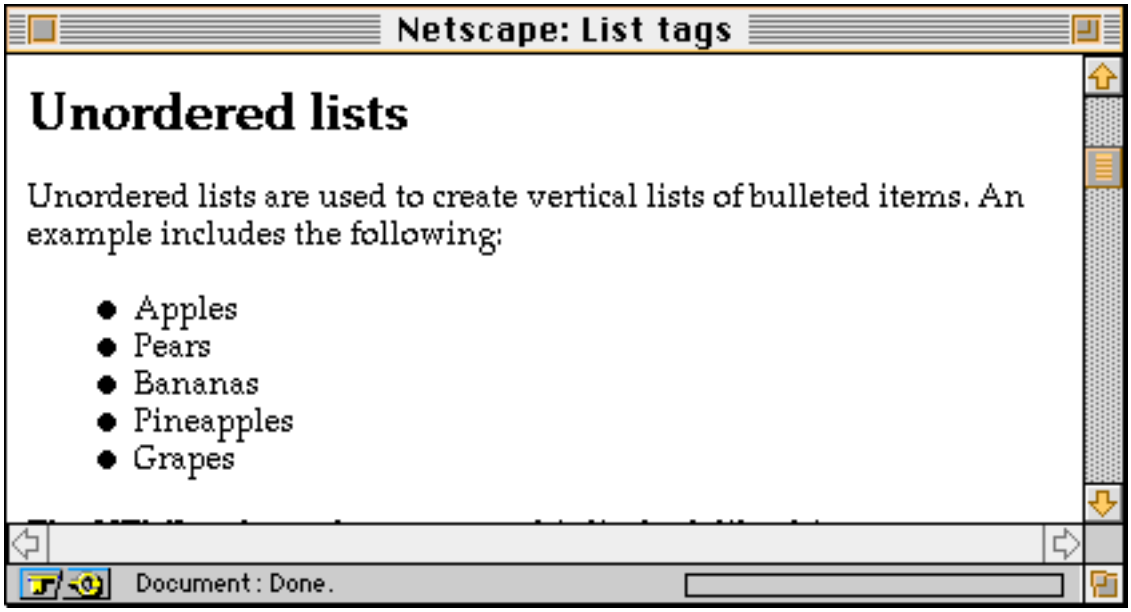
LIST TAGS

This section describes how to incorporate lists into your HTML documents.

Frequently your text will contains lists of items. In recent years, it seems to become popular to write lists of items as bulleted lists. The HTML codes below allow you to do this. Keep in mind the following HTML codes can nested together producing lists within lists. Also, some of these list tags are "logical" in nature and should be used whenever possible.

UNORDERED LISTS

Unordered lists are used to create vertical lists of bulleted items. An example includes the following:



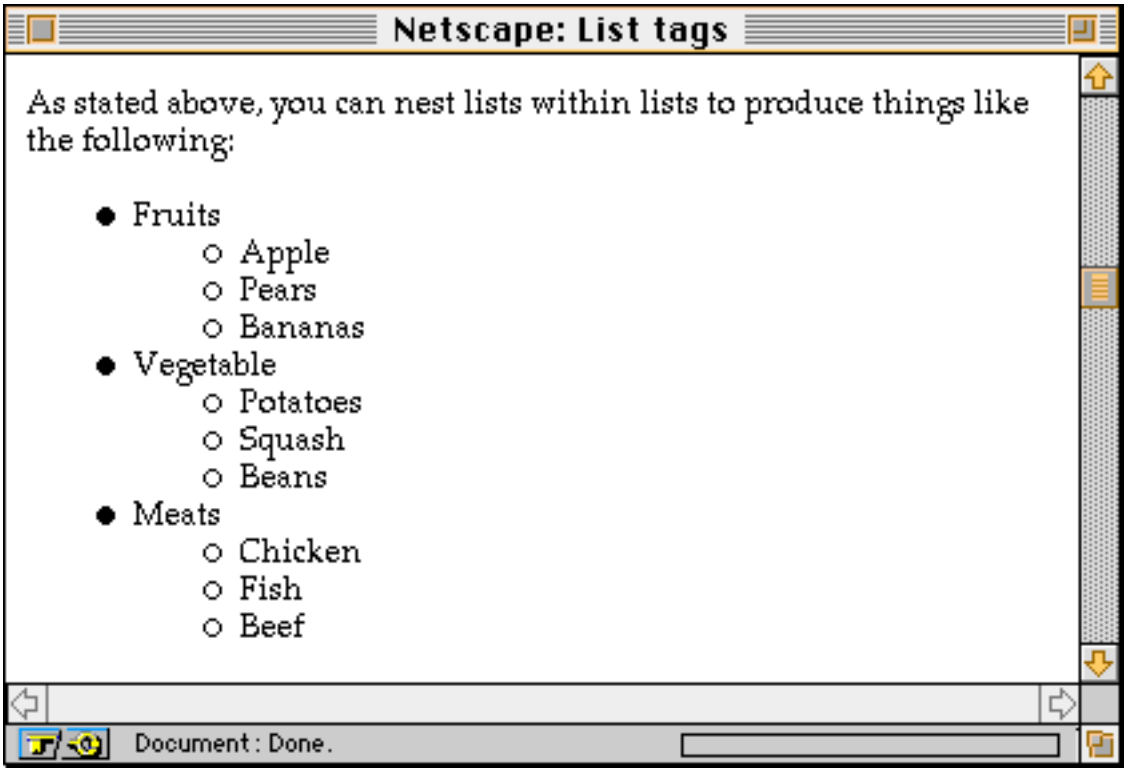
The HTML code used to generate this list look like this:

```
<UL>
<LI>Apples
```

```
<LI>Pears
<LI>Bananas
<LI>Pineapples
<LI>Grapes
</UL>
```

In this example, the <UL>...</UL> is used to mark off the beginning and ending of the list. Within the list, every item is delimited with the "list item" code, <LI>. The HTML "standard" does not provide a means to define the style and shape of the bullets in lists. This function is left up the client application. On the other hand, if you use a few Netscape-specific tags, then you can define the style and shape of the bullets. Of course, these stylistic modification will only work if your reader is using a Netscape browser.

As stated above, you can nest lists within lists to produce things like the following:



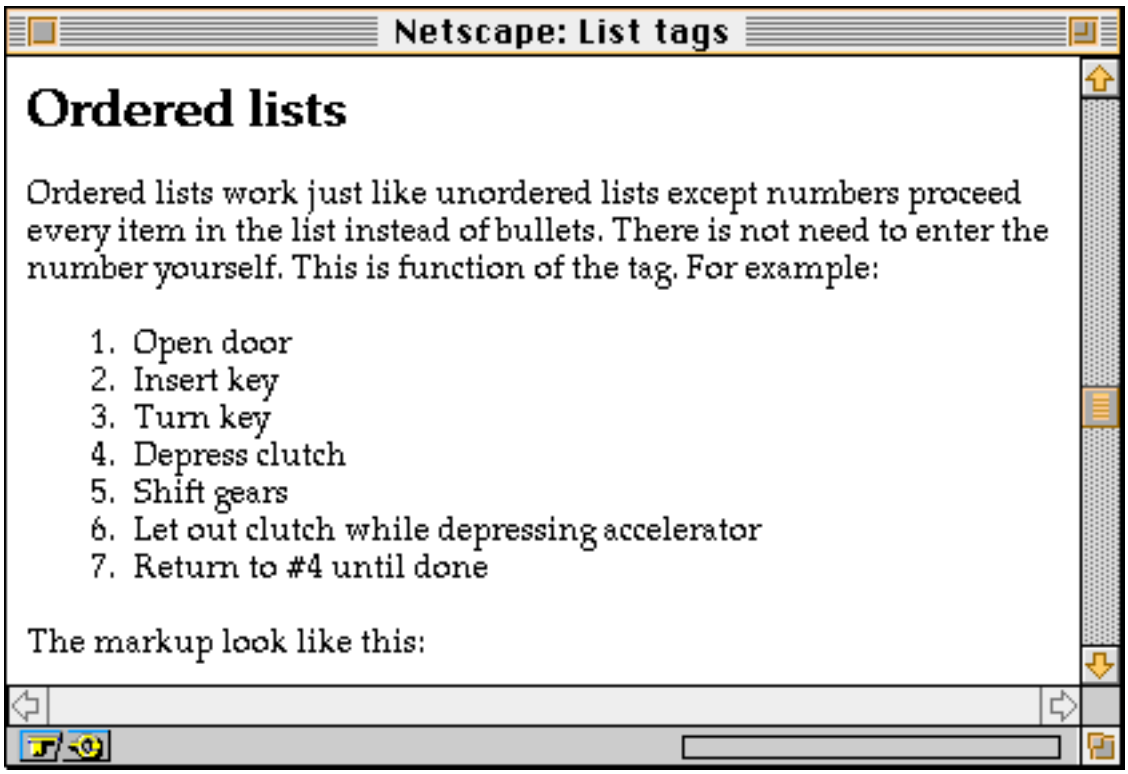
This effect is produced with this HTML markup:

```
<UL>
<LI>Fruits
<UL>
<LI>Apple
<LI>Pears
<LI>Bananas
</UL>
```

```
<LI>Vegetable
<UL>
<LI>Potatoes
<LI>Squash
<LI>Beans
</UL>
<LI>Meats
<UL>
<LI>Chicken
<LI>Fish
<LI>Beef
</UL>
</UL>
```

Ordered Lists

Ordered lists work just like unordered lists except numbers proceed every item in the list instead of bullets. There is not need to enter the number yourself. This is function of the tag. For example:



The markup look like this:

```
<OL>
<LI>Open door
<LI>Insert key
<LI>Turn key
<LI>Depress clutch
<LI>Shift gears
<LI>Let out clutch while depressing accelerator
<LI>Return to #4 until done
</OL>
```

Again, Netscape offers enhancements to this tag enabling you to specify the type of numeration you want. You can even specify what number (or letter) will be the first one in the list. For example, you can specify:

- Upper case letters
- Lower case letters
- Large Roman numerals
- Small Roman numerals
- Arabic numbers
- Upper case letters staring with "E"
- Arabic numbers starting with "11"

Menu and Directory

The menu (<MENU>...</MENU>) and directory (<DIR>...</DIR>) tags are the logical list tags. They are used to denote a menu of items or the file listing from a directory. Browsers tend to render these tags just like unordered list tags.

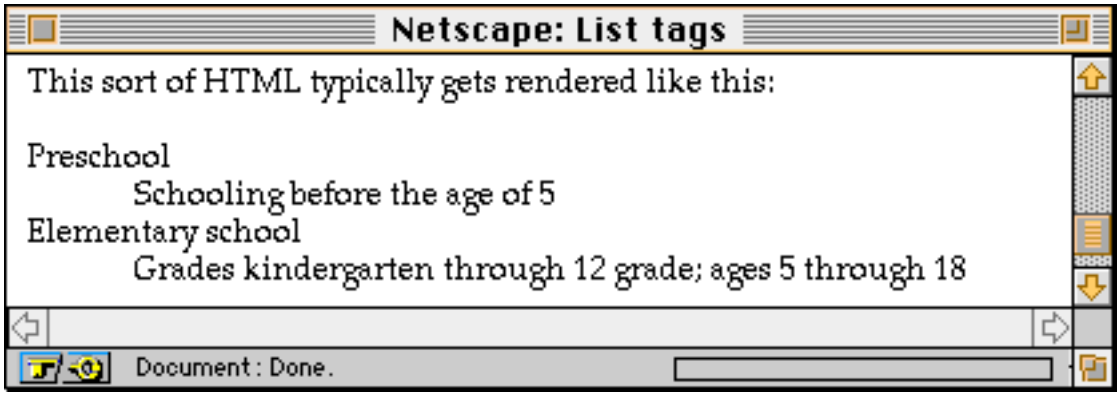
Definition List

The definition list tags is used to list term or phrases as well as the definitions of those terms and phrases. It is also the tag that deviates from the norm since it contains no <LI> indicators.

A typical glossary list in HTML may look like this:

```
<DL>
<DT>Preschool
<DD>Schooling before the age of 5
<DT>Elementary school
<DD>Grades kindergarten through 12 grade; ages 5 through 18
</DL>
```

This sort of HTML typically gets rendered like this:



The <DL>...</DL> tags are used to denote the beginning and ending of the list, respectively. The <DT> is for "definition term" and delimits the term to be defined. The <DD> is "definition definition" and is the explanation or... definition of the term. It is important to make sure the number of <DT>'s is equal to the number of <DD>'s or your list will not be generated correctly.

INCORPORATING GRAPHICS WITH THE IMAGE TAG

This section describes how to incorporate images into your HTML.

Next to the ability to create hypertext links, the ability to incorporate graphics into documents has helped make the World Wide Web so popular.

The first thing you need before you can put images into our HTML documents are the images themselves. There are a few things you need to keep in mind about these images. First, use them only when necessary. Ask yourself whether or not the images add value to our document. If they don't, then consider doing without them. Too many people have relatively slow modem access to the Internet. The time they spend downloading your images better be worth it to them or they will not continue to read your documents.

Second, when you do incorporate images into your documents, keep them small. Keep them small in terms of screen size; do not create images that are larger than most people's monitors (approximately 600 x 480 pixels). The files should be small in terms of their file size as well. Images larger than 30 or 40 kilobytes start to get too big. Additionally, your images, especially if they are intended for on-screen viewing only, do not need to have high dots-per-inch (dpi) ratios. Seventy-two (72) dpi is usually a more than satisfactory resolution. You can reduce the size of your images by reducing the number of colors the image contains. Consider using no more than 256 colors or shades of gray.

Third, there are a wide variety of image formats you can save your graphics in, but the graphics interchange format (GIF) is the format understood by all graphical WWW browsers.

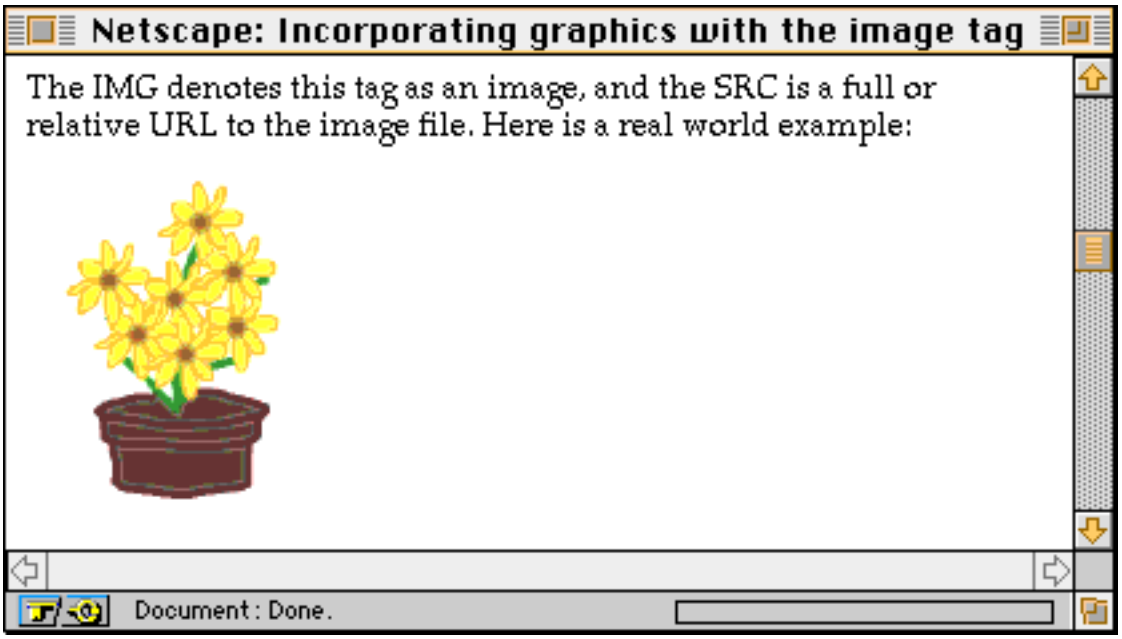
Save or convert your files into GIF files. Furthermore, to make your images more attractive, consider using a utility like clip2gif or Transparency to make your images transparent. (See the section on graphic utilities for more information about this process.) Also consider saving your images using the GIF "interlaced" option. This produces a "Venetian blind" effect when the image is rendered and gives the reader an idea of the image's shape and content even though the entire image has not been received yet.

BASIC IMG TAG

After you have created your images, you are ready to insert image tags into our HTML document. The image tag has the following basic form:

```
<IMG SRC="url-to-images.gif">
```

The IMG denotes this tag as an image, and the SRC is a full or relative URL to the image file. Here is a real world example:



This images was generated with this HTML code:

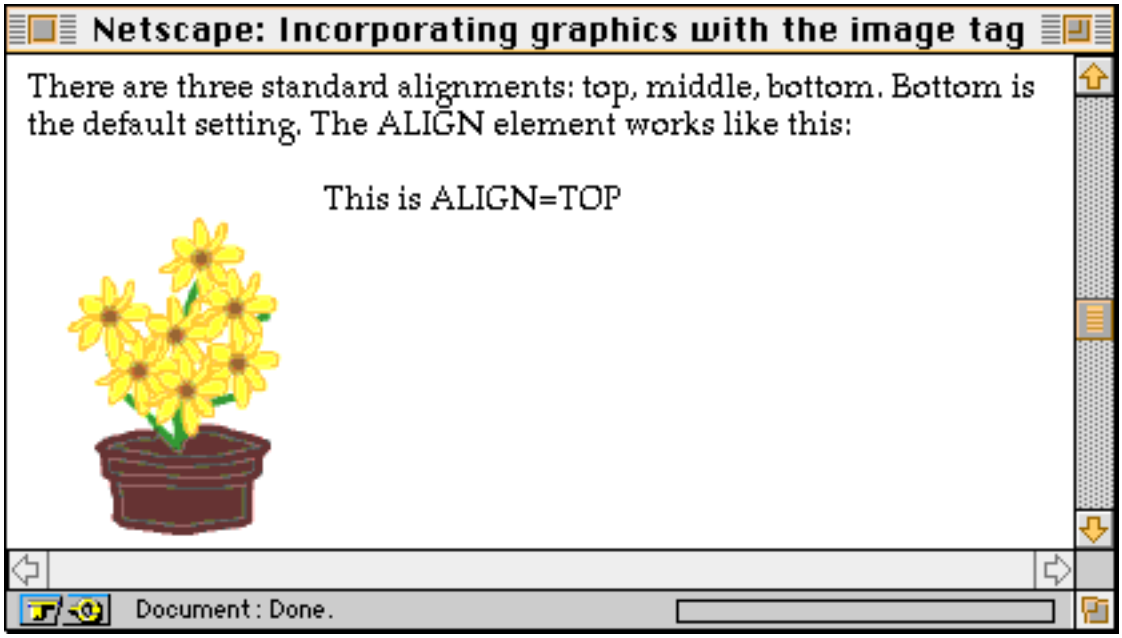
```
<IMG SRC="../graphics/flowers-for-html.gif">
```

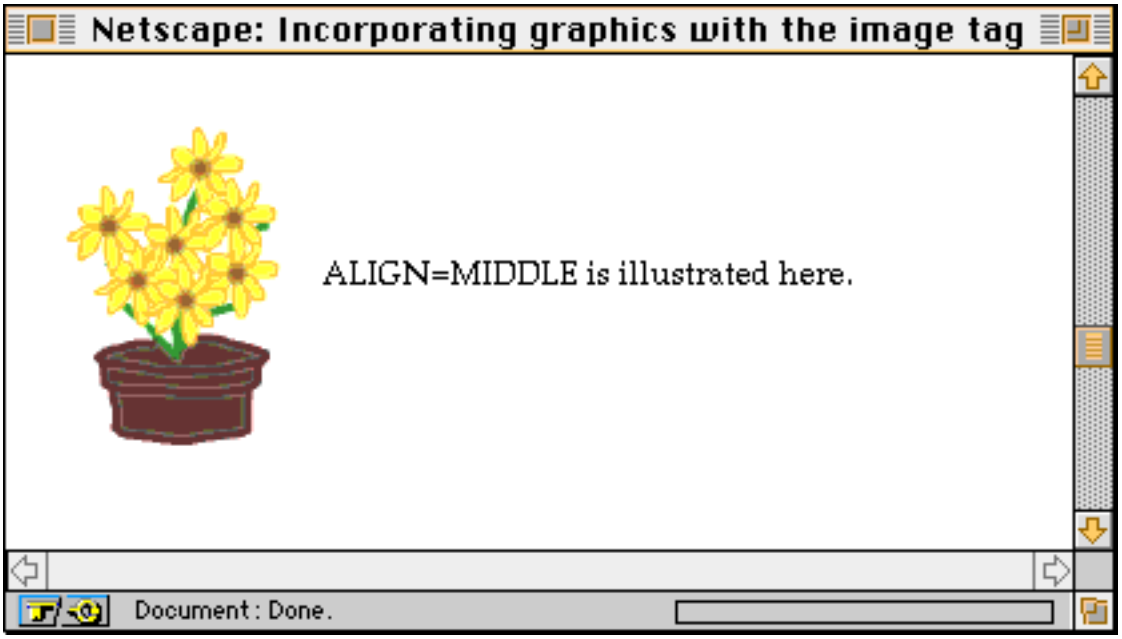
A not-so-obvious limitation of the image tag is that it, by default, always left justifies your image. More specifically, browsers always renders the image just to the right of the previous character. Consequently, if you put your images within paragraphs, then you might get slightly unpredictable results.

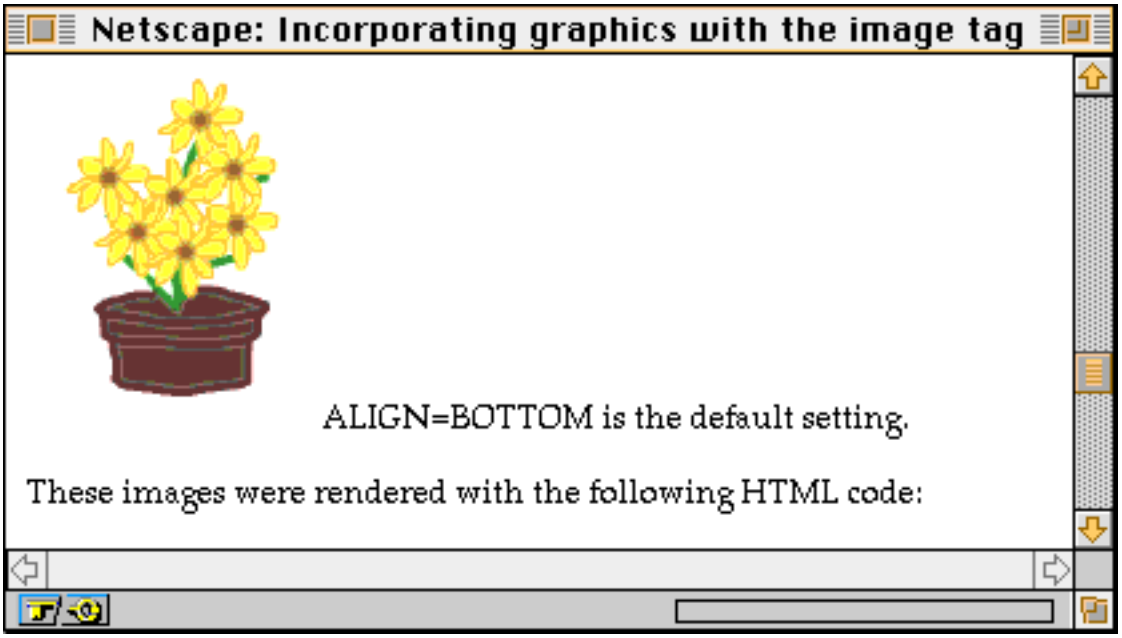
Adding Alignment

As demonstrated above, you can insert images into your documents. Unfortunately, you can not wrap any text around your images using these tags. Enhanced Netscape tags do allow you to do this, but the "standard" HTML codes do not. The closest thing you can get to wrapping text is through the use of the ALIGN element of the image tag.

There are three standard alignments: top, middle, bottom. Bottom is the default setting. The ALIGN element works like this:







These images were rendered with the following HTML code:

```
<IMG SRC="../graphics/flowers-for-html.gif" ALIGN="TOP"> This is
ALIGN=TOP<P>
<IMG SRC="../graphics/flowers-for-html.gif" ALIGN="MIDDLE">
ALIGN=MIDDLE is illustrated here.<P>
```

```
<IMG SRC="../graphics/flowers-for-html.gif" ALIGN="BOTTOM">
ALIGN=BOTTOM is the default setting.<P>
```

Making your images hot

Just like text, you can make your images hot by marking them up with hypertext links. This is how many WWW pages incorporate navigational aids into their documents. For example, the following tag links like you to a page where you can send a comment to the author:

This feat was accomplished through the following HTML:

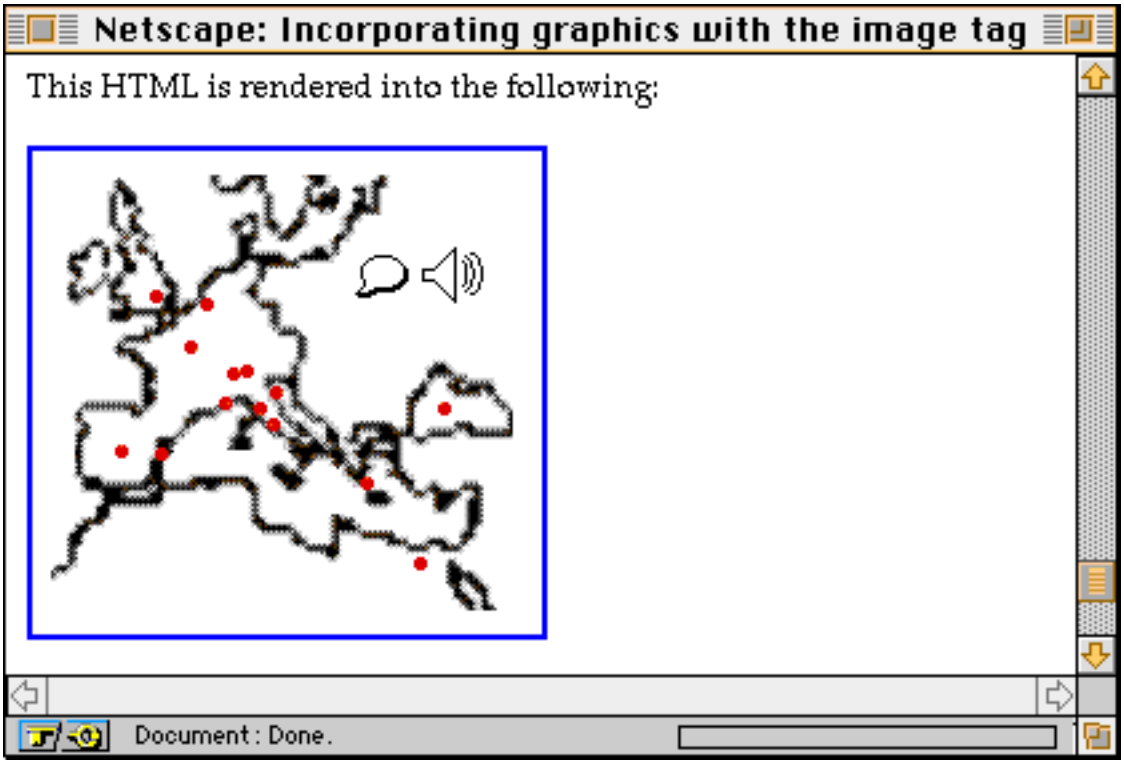
```
<A HREF="../scripts/email-morgan.html"><IMG
SRC="../graphics/mailbox.gif" ALIGN="BOTTOM"></A> Please feel free to
send a comment.
```

Using the ISMAP element

The ISMAP element of the image tag is used to tell the WWW browser that this image is an imagemap. This is done by simply adding the ISMAP element to the tag and making the entire image a link. Here is an example:

```
<A HREF=../scripts/mapserve.acgi$water-map.map><IMG ALIGN=Bottom
SRC=../graphics/water-map.gif ISMAP></A>
```

This HTML is rendered into the following:



Consequently, when the reader clicks on the image, the WWW browser knows to send the clicked point to the program denoted in the hypertext reference. (Please see the chapter on imagemapping for more details.)

MACINTOSH-BASED HTML EDITORS

The chapter reviews hypertext markup language (HTML) editors.

Creating HTML documents by hand can be a laborious process; it is easy to forget all the various tags and formatting rules. Consequently, there are a growing number of software tools available to make the HTML document creation process easier.

At the present time, there are four ways to compose HTML files. The first way is to compose your document using any word processor or text editor. When the composition of your text is complete, you can save the text as an ASCII file, and then import it into HTML editor to begin the mark-up process. The advantage of this method lies in the word processor's spell-checking and robust search/replace functions. This method also makes you focus on the content of your HTML document rather than its style.

The second method is the method most people use where they use their HTML editor to compose and format their documents all at the same time. This method is considered more direct.

The third method is to use a "converter" program. These programs take the output of one file format (like Rich Text Format, Word, WordPerfect, et cetera), and convert the file into HTML.

The fourth method, and the one holding the most promise for commercial developers, allows end-users to their existing word processor and use its Save As feature to create HTML documents.

All four of these HTML-creation methods are described in the sections below.

(Note: These reviews have been done with your best interest at heart, but you should keep in mind that the author of this book is also the author of an HTML editor.)

SUMMARY

The tables below summarize features of the reviewed editors.

	Alpha	Arachnid	BBedit extensions	BBedit Tools
Preview		√		√
Headers/footers				sort of
Extended characters	√			√
Editable	√	√		√
Open URL				√
Netscape ex-HTML		sort of		
Online help				√
Character translation			√	√
Command key options		√	sort of	√

	High Tea	HTML Editor	HTML Pro	HTML SuperText
Preview	√	√	√	√
Headers/footers			√	
Extended characters		√	√	√
Editable	s	√	√	√
Open URL				
Netscape ex-HTML			sort of	sort of
Online help	√	√	sort of	√
Character translation		√	√	
Command key options	√	s	√	√

	HTML Macros	Web Weaver	HyperEditor	HTML.edit
Preview	s	√	√	√
Headers/footers	s			√
Extended characters		√	√	√
Editable	√	√	√	sort of
Open URL				
Netscape ex-HTML		sort of	sort of	
Online help		√		√
Character translation			√	√
Command key options	√	√	√	sort of

	HTML Writer	SHE	Site Writer Pro	Webtor
Preview	√	√	√	√
Headers/footers		√	√	
Extended characters		√		√
Editable			√	
Open URL		sort of		
Netscape ex-HTML		mostly	√	
Online help		√	√	
Character translation				√
Command key options	√	√	√	√

See Also

"Computers:World Wide Web: HTML Editors:Macintosh" <URL:http://www.yahoo.com/Computers/World_Wide_Web/HTML_Editors/Macintosh/>

"Macintosh WWW Development Resources" <URL:http://www.uwtc.washington.edu/Computing/WWW/Macintosh.html>

IETF (Internet Engineering Task Force), "IETF HyperText Markup Language (HTML) Working Group" <URL:http://www.ics.uci.edu/pub/ietf/html/>

Yahoo, "Computers: World Wide Web: HTML Editors:Macintosh" <URL:http://www.yahoo.com/Computers/World_Wide_Web/HTML_Editors/Macintosh/>

Alpha

Version: 6.0b7
Author: Scott Brim
Requirements: 1536K of RAM
Cost: free
Pros: Multi-lingual, extremely customizable, fast, all markup options are associated with command key sequences
Cons: confusing interface, undo does not always Completely work, doesn't support preview, requires the knowledge of TCL to take advantage of it real power
Remote location: <URL:http://www.cs.umd.edu/~keleher/alpha.html>
Tricks location: <URL:http://152.1.24.177/teaching/archives/Alpha-html-0.20.sit.hqx>


Alpha is not so much an HTML editor as it is an editor for programmers. Like BBEdit, Alpha is an extensible editor. It gets it extensibility through a interpreted scripting language called TCL (Tool Command Language or "tickle"). Scott Brim has written such an extension enabling Alpha to become an HTML editor. It is based very much on scripts written for emacs, an editor on many Unix computers.

To use the extension, you first must "install" it. This is as easy as selecting a menu item because the extension is included with the Alpha distribution. Next, you must put Alpha into HTML editing "mode" by selecting another special menu called a "mark". Now you can begin creating HTML documents just like any other editor: importing text or entering it from the keyboard, selecting markup options, and saving.

Since the HTML extension is written in TCL, anybody with knowledge of TCL can customize the extension and improve upon it. For example, the HTML extension does not support FORMs creation, nor does it support enhanced Netscape HTML. If you don't like the command keys associated with various markup options, then you can change them.

Every markup option is associated with a command key sequence, but they are non-standard for the Macintosh. For example, option-command-T creates <title></title> tags, and option-command-3 creates an H3 header. This extension is even smart enough to insert the escape sequence for the greater than symbol when you type "<" The downside of all this is the extension does not display these command key options in the menu; you must read the TCL script to discover them or guess.

Alpha and the HTML extension it currently supports is a power-user, programmers tool. Alpha itself is intended to make the writing of programming code easier exemplified by the many programming modes it includes like C, C++, MacPerl, csh, Pascal, et cetera. Alpha presents itself as an excellent tool for the programmer fluent in many computer languages. It is not so well oriented for the casual HTML editing user.

ARACHNID

Version: beta 1.5.4
Author: Robert McBurney
Requirements: 2MB of RAM
Cost: free
Pros: semi-WYSIWYG
Cons: not zippy
Remote location: <URL:http://sec-look.uiowa.edu/about/projects/arachnid-page.html>
Tricks location: <URL:http://152.1.24.177/teaching/archives/arachnid-1.5.4b-full.sea.hqx>

 Arachnid is a semi-WYSIWYG HTML editor, but it is so slow a person grows old just using it.

The concept behind Arachnid is good one. Create your documents outside of the editor saving them as ASCII text or RTF files. Second, you create an Arachnid "project" that will form a family of HTML documents. Third, you then import the saved ASCII or RTF files and mark them up. If the copy was saved as RTF, then Arachnid successfully imports it and displays it in a window. Arachnid is the only editor with this feature. The editor even displays the text with its original formatting.

To create hypertext links you first select the text (or Arachnid "objects") you want to be hot and then URL Link tool icon from the floating palette onto the selected text. You are then prompted for a URL and a name for the link. The text then gets highlighted as if it appeared in a WWW browser and a database of URLs is updated.

Creating hypertext links to sound, movie, or picture files works in a similar manner. First you select some text and then drag the appropriate icon from the palette on to the selection. You are then prompted for a file through a standard Macintosh dialog box. (Only files that are in the

same folder as your current Arachnid project are valid.) Once the file is selected, Arachnid creates the hypertext link for you.

Arachnid has three modes of operation: edit, pointer, browse. Edit mode is intended for entering text and marking up your document. Pointer mode allows you to move objects (paragraphs of text, picture, etc.) around on the screen. Browse mode imitates a WWW browser without any external hypertext links.

Despite these features, Arachnid is not very easy to use. It is a slow application taking a lot of time to switch between modes as well as simple entering of text. To actually create any HTML documents you must "Export" your text. A simple file with no formatting containing 2K characters took 2 minutes of processing on a Quadra 660AV. Importing text takes just as long. Furthermore, Arachnid does not even generate valid HTML; the resulting documents contain no BODY tags. Last, the Arachnid palette containing many of the tool you need for editing consume a large amount of screen real estate; it could be smaller.

Arachnid works as advertised, but it does it in such a slow and round-about manner that you may want to use some other HTML editor reviewed here.

BBEdit HTML Extensions

Version: release 9
Author: Charles Bellver
Requirements: BBEdit
Cost: free
Pros: simple and straight-forward
Cons: not feature-rich
Remote location: <URL:http://www.uji.es/bbedit-html-extensions.html>
Tricks location: <URL:http://152.1.24.177/teaching/archives/bbedit-html-ext.hqx>

The BBEdit HTML Extensions, like the BBEdit HTML Tools (described below) are extensions to BBEdit. BBEdit is a full-featured text editor supporting enhanced find/replace functions, line wrapping and unwrapping, and various other "text tweaking" features. BBEdit comes in two versions. The enhanced version costing about $100 includes these features as well as AppleScriptability, automatic line wrapping, and a host of Internet tools. (For more information try <URL:ftp://ftp.std.com/pub/bbedit/bbedit-lite-30.hqx>.)

HTML Extensions has a nice template feature providing a straight-foward method of locating prewritten headers and footers. Furthermore, if you first open a previously saved document and then use the template function, HTML Extensions will incorporate the text into the BODY of your HTML.

Creating links and anchors in our HTML is straight-forward as well. Simply select the text intended to be "hot" and enter your NAME or URL. HTML Extensions does not make you go

through an extensive dialog box before you can make some text a link or an anchor. Similarly, the creation of FORMs is not a long, drawn out process where you have to navigate many dialog boxes before anything gets done.

HTML Extensions is not as full-featured as some of the other editors available, and it does not support any "Netscapisms". This not a bad thing since Netscape has created their own flavor of HTML, and if you were to use HTML Extensions, then you could rest assured of creating valid HTML documents. At the same time there seems to be some blatant omissions. For example there seems to be no way to enter the tag for horizontal rulers (<hr>) or line breaks (<br>). This is not an oversight by the author. Rather these tags have been omitted because they are so easy to enter by hand or incorporate into BBEdit's glossary function.

BBEdit HTML Tools

Version: 1.2.2
Author: Lindsay Davies
Requirements: BBEdit
Cost: free
Pros: supports templates and Level 2 HTML, includes HTML validation function
Cons: too many dialog boxes or keystrokes need to get simple work done
Remote location: <URL:http://www.york.ac.uk/~ld11/BBEditTools.html>
Tricks location: <URL:http://152.1.24.177/teaching/archives/BBEdit-HTML-Tools.hqx>

BBEdit HTML Tools, like the BBEdit HTML Extensions, requires... BBEdit <URL:ftp://ftp.std.com/pub/bbedit/bbedit-lite-30.hqx>. Not only do these extensions provide the "usual suspects" of HTML markup options, but they also come with features like:

- HTML validation
- Graphic and character-mode previews
- Full-featured template files and standard HTML document elements
- Remove tags

Starting with the "usual suspects", BBEdit HTML Tools provides the capability to create hot links through a specialized dialog box. From this dialog box you select the elements of a URL from popup menus and buttons. In turn the URL is created for you; you can not simply enter a URL. The same concept is used in the creation of FORMS.

Creating lists is a pleasure with HTML Tools because it can indent the items in your lists making them easier to distinguish later on.

HTML Tools makes liberal use of Balloon Help.

HTML Tools includes a HTML validation routine. Simply select "Check markup" from the

"Utilities" menu and HTML tools creates list of errors in your HTML. (Upon trying this out HTML Tools produced a list of errors including the message "Unknown or malformed tag", and for the life of me I couldn't figure out what I had done wrong.)

HTML Tools includes a preview option, but unlike other browsers it you have the option of previewing your document graphically with a Macintosh-based browser as well as throughout BBEdit itself as if BBEdit were a vt100 terminal. Using this feature you can get an idea of what your HTML will look like for people using Lynx, a vt100, character-cell browser.

The ability to create, update, and use template files is a real strength of these BBEdit extensions. To use the template you first create on by hand, save it, and then use the "Preferences" button of the "Utilities" menu to specify your template file. Then, the next time you choose "New" from the "File" menu your template file will be used. To enhance the template functionality, HTML Tools allows you to include a number of placeholders in your template. These placeholders include variables for: title, user name, time, server, whether or not the document is an ISINDEX, as well as others. By placing the variables associated with these placeholders into your template file, the the placeholders are updated dynamically. Furthermore, you can create "include" placeholders. These placeholders can be used to insert entire text like standard headers and footer.

On the down side of HTML Tools is that none of the options are immediately available. In other words, to get anything done with HTML Tools you must either select more than one menu option and go through at least one dialog box, or you must enter quite a number of keyboard commands before anything will happen. For example, to get a preview of your document requires first saving the document, second invoking the preview dialog box, and last choosing the dialog mode. Similarly, to insert a paragraph mark (<p>) you must choose a menu item then go through a dialog box.

Because of HTML Tools validation routines and template files, these extension are well worth examination, especially if you already use BBEdit as your text editor.

HIGH TEA

Version: 1.9
Author: Dr. R. (Stan) Stanier
Requirements: 1500K RAM
Cost: free for all non-profit making purposes
Pros: simple, does tables very well
Cons: limited window size, could use more dialog boxes for easier selection of input options
Remote location: <URL:http://www.mistral.co.uk/cbuzz>
Tricks location: <URL:http://152.1.24.177/teaching/archives/high-tea.hqx>

 This is a happy little editor. For some reason it makes you smile. The best thing High

Tea has going for it is its method for creating tables. As the documentation states, it works from the inside to the outside meaning you first create you table (with a fixed sized font), then format elements of the table, headers of the table, rows of the table, and finally the table's borders. The process works very, very well.

By setting your preferences, High Tea acts as a launcher when you want to edit graphic files or imagemaps. Thus you can select items from the toolbar and automatically edit the last graphic or imagemap file as well as any other graphic or imagemap file included in your document.

High Tea only suffers from a couple tiny problems. First, because it is seemed to be a HyperCard application, its window size is fixed, and the size is small. This wouldn't be so bad if the toolbar didn't take up so much of the screen. Second, and more importantly, High Tea could be improved if it were to collapse the number of simple dialog boxes it uses for input into a few multi-input dialog boxes. For example, to create an HTML table tag requires answering four questions. These four questions could easily be integrated into one dialog box. Furthermore, this dialog box could "remember" its previous settings making it easy to create mark up subsequent tags.

HTML Editor for the Macintosh

Version: 1.0
Author: Rick Giles
Requirements: a Macintosh SE/30, Mac II, or other Macintosh with a 68020-compatible CPU, System 7 or higher, 2 Megabytes RAM (larger partition may be needed if you are running in greater than 16 bit color
Cost: $25 (shareware)
Pros: semi-WYSIWYG, simple interface
Cons: few command key options, does not support forms nor tables
Remote location:
<URL:http://dragon.acadiau.ca/~giles/HTML_Editor/Documentation.html>
Tricks location: <URL:http://152.1.24.177/teaching/archives/html-editor-10.hqx>


This editor, one of the first Macintosh editors to make its appearance, could best be described as solid. Not only that, but it is a close to a WYSIWYG editor available.

The interface to HTML Editor is remarkable simple consisting of a menu bar with few options, a windows containing your text and a row of buttons, and optionally, the palette window. The menu bar consists of the usual File and Edit menus. The Search menu provides full find/replace features. A Windows menu provides access to the presently open windows. A unique menu to HTML Editor, the URL menu, is used to create and export a database of URLs included in your document(s). To use it you first choose New and enter URLs into a simple dialog box. After that you can edit or export them to a file. Unfortunately, there seems to be no smooth way to integrate this simple database of URLs into your HTML editing procedure.

To add markup to your document you simple select or click any one of the button from the button bar. Each button, as well as every menu item, is well documented with Balloon Help. Creating hot link using the anchor button can be accomplished in two ways. First you can simply enter your desired URL. Second, you can go through a dialog box that will construct the URL for you. If you have selected the a root directory option from the Preferences menu, then all you will have to do to link to files on your own computer is select them from a standard dialog box. HTML Editor creates the URL for you.

When you add tags to your document, the tagged items are formatted in a pseudo-WYSIWYG way. Headers are made big and bold. Items marked as CITE are italicized, etc. The tags themselves are displayed, by default, in an unobtrusive shade of gray. To enhance the WYSIWYG effect you can hide the tags. Unfortunately you can not continue editing in this mode and must show the tags to continue.

There are a number of things that could be improved. For example, there could be more command key options for the usual formatting tags; it is sometime laborious to interrupt typing to mark up an item. Second, the IMG tag is not complete and only offers ALIGN TOP as an option. There is not simple way to enter the HTML, HEAD, and BODY tags. Since they are not there it will be easy for novices to create poorly formatted documents. Furthermore, there are no markup options for FORMS or tables. Lastly, and this is minor point, when you enter tags like "extended Netscape" tags and then you select Hide Tags, HTML Editor does not hide these tags. On the other hand, you can blame the editor to much since it was written before these tags hit the scene.

HTML PRO

Version: 1.05
Author: Niklas Frykholm
Requirements: 500K of RAM
Cost: $5 (shareware)
Pros: semi-WYSIWYG, handles extended characters very well
Cons: undo does not work, only one file open at a time
Remote location: <URL:http://www.ts.umu.se:80/~r2d2/shareware/shareware.html>
Tricks location: <URL:http://152.1.24.177/teaching/archives/html-pro-105.hqx>


HTML Pro is a semi-WYSIWYG HTML editor with an interface that is a joy to use, but can only open one file a time. Each file is displayed in two windows, the source window and the HTML window. The source window displays raw HTML created by the program.The HTML window displays (almost) what the rendered HTML will look like. Either window can be used during the editing process.

Like the balance of HTML editors reviewed here, HTML Pro allows you to markup your documents by first entering your text and choosing a markup option. It doesn't include the use of any palettes or button bars. Rather, all the HTML tags are logically organized into one

Styles menu with six hierarchal menus. Once a markup option is selected, the HTML window changes the text accordingly. You can't choose what the rendering will look like, but it does give you a pretty good idea.

HTML Pro handles the translation of extended characters very well. For example, to display something like an accented é all you have to do is type your character the way you normally would using any other word processor. First press the option-e and then the e key. Option-a for example, automatically generates the necessary escape codes for å. When importing pre written text files, HTML Pro examines your text for extended characters and automatically translates them. Also, when importing text files, HTML Pro translates your paragraphs into <p> marks as well.

Items called macros are used by HTML Pro to allow you to create HTML tags not included with the editor. To use them you select New MacroÉ from the Macros menu. You then give a title to your macro, enter a beginning tag and an ending tag. You can use this, as the documentation states, to create things like tags. Unfortunately, like a number of the other editors reviewed here, this featured does not allow for tags that may require a number of parameters like the image tag.

HTML SUPERTEXT

Version: 2.1beta4
Author: Robert C. Best III
Requirements: Systems 7, 800K of RAM
Cost:
Pros: extensive customizability and online help
Cons: all formatting tags in one long menu
Remote location:
<URL:http://www.potsdam.edu/HTML_SuperText/About_HTML_S.html>
Tricks location: <URL:>


As the standard for HTML is enhanced, HTML SuperText users will be able to keep up with many of the changes because of HTML SuperText's extensive customizability. For example, end-user's can modify SuperText's Tags menu to include new HTML tags. This is done by simply filling in a dialog box. Unfortunately, these tags must have a specific form like <citation></citation> where there is a beginning and ending tag. Other tags, like the ones used to create radio buttons, are not available in SuperText. Similarly, you can create special characters the same way, characters like the copyright symbol (&copy) or extended characters like a tilde "a" (&atilde).

All HTML markup options are available in one menu of SuperText. At first glance, could be seen as a poor way to organize the options. But with SuperText's ability to create "Custom Tag Floaters", end-users can create their own floating palettes of tags to meet their own working habits.

SuperText supports a "Revert to last saved" version of your file, feature not found in other editors reviewed here. SuperText edit also displays HTML tags in end-user defined fonts, colors, and styles thus helping to distinguish between text and tags.

SuperText also gives you the option of including Macintosh-based formatting in your text (underlining, bold, shadow) as well as font choices and colors. This feature enables you to get a hint of what your rendered HTML might look like, but at the same time, combined with a preview feature, these features seem extraneous.

SuperText supports the creation forms by leading you through a dialog box. The dialog box is a bit confusing, but the creating forms is confusing itself. SuperText uses a similar format for creating links to other documents. By selecting items and completing fields in dialog boxes, SuperText creates for you valid URLs.

SuperText supports a vast array of online help options from Balloon Help to precreated HTML files. In fact, SuperText has more online help than any other editor reviewed here.

HTML TAG MACROS

Version: 1.0 (?)
Author: Jim Tummins
Requirements: WordPerfect 3.1
Cost: free
Pros: represents a good concept
Cons: not quite ready yet
Remote location: <URL:>
Tricks location: <URL:http://152.1.24.177/teaching/archives/html-tag-macros.hqx>


HTML Tag Macros are a set of WordPerfect 3.1 macros for creating HTML documents. If your usual word processor were WordPerfect, and if these macros were more mature, then they would be represent a useful set of tools. Unfortunately, there are too many errors in the way they create HTML codes, especially if you don't know what to look for.

The concept is great. Open a stationary document containing the HTML Tag Macros. Insert or enter the text you intend to mark up. Select text and choose from the rather complete set of HTML markup tags from the Macros menu of the Tools menu. Answer the questions asked by the resulting dialog boxes, and the macros do the rest. When you are finished, remember to save your document as a text file and preview it with your browser. To make your document a bit more WYSIWYG, you can use the supplied Styles options to make your marked up text more true to form.

The reality is that these macros are not quite ready yet, at least with the version reviewed here. BLOCKQUOTE inserts a > in the wrong place. The FORM and SELECT tags do not include

a > symbol at all. The documentation does not remind you to turn off "smart quotes" and consequently, you may create markup that produces unknown characters. They are slow.

Again, the concept of these macros is good. Since they are available in a editable form, you can enhance them yourself. As they stand now they exhibit quite a lot of work, but they are not quite ready for prime time.

HTML Web Weaver

Version: 2.5.2
Author: Robert C. Best III
Requirements: 700K RAM, System 7, 1MB of hard disk space
Cost: $14-25 (shareware)
Pros: semi-WYSIWYG
Cons: overly-simplistic menu organization
Remote location: <URL:http://www.student.potsdam.edu/web.weaver/about.html>
Tricks location: <URL:http://152.1.24.177/teaching/archives/html-web-weaver.hqx>


HTML Web Weaver was written and is being maintained by the author of HTML SuperText, and Web Weaver picks up where SuperText left off.

One of the nicest things about this editor is the way it visually distinguishes between HTML tags and text. By default, tags are displayed in a green 7 point Geneva font and text is displayed in a 12 point, black Monaco. This makes it easy see what is text and what is HTML. These text distinguishing characteristics are customizable through a preference setting.

Web Weaver allows you to create FORMs through a dialog box and the elements of the FORMs are manipulatable while they are contained within the dialog box. Once they are rendered as HTML, then you must edit them manually. It provides the same functionality for the various types of lists.

The creation of links and anchors are also done through dialog boxes. This is both a blessing and a curse. Many times you want to past the entire URL of an item into a link and this can't be done through the dialog box. On the other hand, since it remembers past links, you can easily select parts of previously used URLs to create new links.

Like its forbearer, HTML Web Weaver allows you to create new, customized HTML tags. It is easy enough to do, but unfortunately, all your tags end up on one very long menu making it difficult to get to them. To reduce the effects of this problem you can create custom floating palettes containing the tags you use most frequently, but close the palettes because you won't be able to get them back again without recreating them. HTML Web Weaver also allows you to change your HTML document's font, size, and style. These features seem rather superfluous considering the text is intended to be rendered through a WWW browser.

HTML-HyperEditor

Version: 1.0
Author: Lars-Olof Albertson
Requirements: HyperCard 2.2
Cost: free
Pros: formats RTF files nicely, creates tables of contents
Cons: makes for a cluttered screen, creation of hot links not intuitive
Remote location: <URL:http://balder.syo.lu.se/Editor/HTML-HyperEditor.html>
Tricks location: <URL:http://152.1.24.177/teaching/archives/HTML-HyperEditor1.0.S.sea.hqx>

This editor addresses the issue of non-English character sets, specifically the translation of diacritics. More so than some of the other editors here, HTML-HyperEditer is designed to markup copy after it has been created outside the editor and imported. For example, it contains features to import RTF files, simple ASCII files, as well as HTML files.

If you are importing an RTF file, HTML-HyperEditor will convert the quote marks ("), greater than signs (>), and less than signs (<) to their appropriate escape sequences. After reading the English version of the documentation, the characters like &Aacute were expected to be translated as well. This never happened. Again, if you imported an RTF file, the editor formats the text in such a way that it is WYSIWYG. When you select Format from the HTML menu, HTML-HyperEditor very cleanly goes through your document and surrounds headings with the good-guess heading tags, paragraph and line break codes, as well as inserting the necessary HTML, HEAD, and BODY tags.

HTML-HyperEditor supports a user configurable menu for customized tags. This editor can create simple tables based on a selection of tab-delimited text.

One of the nicest and unique features is its ability to create a tables of contents to your document automatically. It does this by examining your document for header tags and creating a list. Very nice.

It was a frustrating experience trying to create an HTTP or gopher link. After selecting something to be hot and choosing HTTP from the Link menu, an error was always produced -- "Can't understand arguments of 'put'". Also, since this application is designed with HyperCard, the size of the window is static. This application was designed for at least a 12 inch monitor. Combine this with the very large palettes included with the application, and it makes for a very cluttered screen.

HTML.EDIT

Version: 1.5b
Author: Murray M. Altheim
Requirements: 1300K of RAM but prefers 2MB
Cost: free
Pros: supports multiple headers and footers, stringently conforms to HTML standards, full-featured
Cons: cluttered interface
Remote location: <URL:http://nctn.hq.nasa.gov/tools/htmledit/HTMLEdit.html>
Tricks location: <URL:http://152.1.24.177/teaching/archives/HTMLEdit.hqx>

HTML.Edit, a full-featured editor, is a HyperCard 2.2 stack saved as an application. Its interface could best be described as baroque with its cute finger-pointing cursor and flashy splash screen. These same characteristics make using HTML.Edit just a bit confusing. But to HTML.edit's credit, it is difficult to implement so many features and at the same time implement a simple interface.

For example, HTML.edit supports 3 sets of headers and footers. It hilights and indexes any HTML tags in your document. It translates Macintosh text files containing high ASCII codes into HTML compatible characters, and you can even add your own translations like the copyright or trademark symbols. Like HTML Editor, HTML.edit keeps a database of entered URLs. These URLs are then available during the editing process. It supports extended characters (entities) from a clickable palette of choices.

On the down side, HTML.edit is no speed demon. It takes little advantage of command keys nor menubar choices. Every markup option is contained in a floating palette with hierarchical menus. If you remove close the palette by mistake, then you must show it again before you can continue editing. Lastly, since it is a HyperCard application, it suffers from the 30,000 character limit of any one editing field. This is not so bad because such large files are frowned upon in the WWW community.

HTMLWriter

Version: 0.9d4
Author: Jon Wiederspan
Requirements: 2MB RAM
Cost: free
Pros: distributed in an uncompiled form
Cons: incomplete
Remote location:
<URL:http://www.uwtc.washington.edu/JonWiederspan/HTMLEditor.html>
Tricks location: <URL:http://152.1.24.177/teaching/archives/HTMLWriter0.9d4.hqx>


Based on the version number, the number of times it has been mentioned in discussions of HTML editors, and its interface, this editor does not seem ready for prime time, nor is it intended for wide use. Despite its lack of polish and instability, it supports all the standard HTML tags through menu options as well as palettes. It can create list of URLs from your documents as well as lists of anchors and images.

Rather than being a fully functional editor, this application represents a development opportunity for somebody who want to raise the stakes in the world of Macintosh-based HTML editors. HTMLWriter is written in SuperCard and distributed in an uncompiled form. You may want to take what the author of HTMLWriter has begun and complete it. Of course you will want to contact him before you do such a thing.

Simple HTML Editor (SHE)

Version: 2.9
Author: Eric Lease Morgan
Requirements: HyperCard 2.1 or HyperCard Player, System 7 to use Balloon Help, AppleScript extensions to use automatic opener
Cost: free
Pros: requires very little RAM (750K), simple interface, facilitates Netscape HTML extensions, supports preview (requires MacWeb or Netscape), includes FORM creation
Cons: Not WYSIWYG, can open only one file at a time, windows are not resizable, no choice of display font, UNDO does not always work correctly, file size is limited to 30,000 characters
Remote location: <URL:http://www.lib.ncsu.edu/staff/morgan/simple.html>
Tricks location: <URL:http://152.1.24.177/teaching/archives/SHE.hqx>


Simple HTML Editor (SHE) was one of the first HTML editors to be made available to

the Internet community. This free, HyperCard stack has been continually enhanced since its release in December of 1993. It is one of the few (if not only) HTML editors to support Netscape enhanced HTML.

SHE creates a standard HTML template upon the creation of a new document. It supports the insertion of headers and footers. It also supports the creation of FORMS by querying the end-user for the necessary information.

Since SHE creates generic text files, double-clicking these text files from the desktop do not automatically launch the editor. To overcome this problem, SHE comes with an AppleScript droplet. By dragging and dropping a text file onto the SHE droplet, SHE is automatically launched and the text file is imported.

SHE is not without its problems. First of all, since it is written in HyperCard, text fields can be no larger than 30,000 characters. This can be a limitation. Similarly, SHE can only open one file at a time. Its windows are not resizable and the display font can not be changed.

SHE's original purpose was two-fold. First there was (and still is) a need for simple methods for creating HTML documents. SHE does this. Second, SHE was intended to demonstrate to the Internet community was could be done. Since SHE is written in HyperCard and it is distributed in an uncompiled form, the internal workings of SHE are laid bare to any would-be hacker or well-intentioned programmer. Consequently, SHE has played a significant role as a prototype for many HTML editing project across the globe. While SHE may not be the best HTML editor nor the best editor for your purposes, it has been used to accomplish the goals originally set out for it.

"SHE, the grandmother of all Macintosh-based HTML editors."

Site Writer Pro

Version: Beta 2.5
Author: Rik Jones
Requirements: 14" monitor
Cost: $20 (shareware)
Pros: anticipates HTML 3.0, headers and footers, provides mechanisms for background colors
Cons: slow, interface needs refining
Remote location: <URL:http://www.rlc.dcccd.edu/Human/SWPro.htm>
Tricks location: <URL:http://152.1.24.177/teaching/archives/Site-Writer-Pro.hqx>


Site Writer Pro is colorful editor to say the least, and it wins the prize for the most floating palettes. In fact, there are so many palettes in Site Writer Pro that it has a palette for the palettes. Needless to say, the palettes contain the vast majority of the functionality of the editor.

Based in HyperCard 2.2, this editor takes up most of a 14" monitor's screen. Consequently, if you have anything less than this, then Site Writer is not for you. Once up an running, it is good idea to set your preferences right away so Site Writer knows how to operate. Otherwise some of the functions won't work as expected, notably the "Test" (preview) function.

Like HTML.edit, Site Writer Pro is really a database of pages. So you first begin by creating a new document in the database. You are taken to the new document and the screen is divided into three sections: head, body, footer. Using the Edit menu's "Add All HTML Tags..." menu, you can insert the necessary tags for a valid HTML document. One unique feature of Site Writer Pro is its ability to insert background graphics and colors a la the Netscape browser. Unfortunately, the selection of colors is not done graphically and consequently you have to insert the hexadecimal color values by hand.

The next step is to begin entering your text and marking it up. Entering your text is straight-forward. Marking up your text can be done from a menu or through a one of the numerous palettes. This brings up another unique feature of this editor. Site Writer Pro has tried to anticipate the implementation of HTML 3.0 with the inclusion of may advanced HTML codes like complex headings and paragraphs, mail handler tags, subscripts and superscripts, etc. Furthermore, WWW-browser specific tags (like Netscape's BLINK) are denoted in the menus with special symbols. Site Writer Pro also supports numerous header and footers for your documents.

Site Writer Pro represents a lot of effort. It has made good use of its development platform. On the other hand, because of all the colorizing of the stack, this editor is not any speed demon. The interface takes a bit of getting used to, and while the palettes are nice, they are rather large and seem to get in the way more than anything else.

WEBTOR

Version: 0.9.1 pre a1
Author: Jochen Schales
Requirements: System 7
Cost: free
Pros: create "good" HTML
Cons: incomplete
Remote location: <URL:http://www.igd.fhg.de/~neuss/webtor/webtor.html>
Tricks location: <URL:http://152.1.24.177/teaching/archives/webtor0.9a.sea.hqx>

This is an editor to keep you eye on. While not quite ready for prime-time, this is the closest editor seen that is WYSIWYG. Not only that, but it creates perfectly valid and standard HTML documents. As the version number states, this is alpha software, but as soon as it overcomes some of it obvious bugs, this could become and editor of choice.

HTML CONVERTERS AND UTILITIES

This chapter reviews a few HTML converters and utilities.

The programs listed here are not so much HTML editors as they are utility programs for creating or updating HTML documents.

COLORHEX

Version: B0.93
Author: Dorian Dowse and Gavin Sade
Requirements: color monitor, 2MB free RAM
Cost: free (for personal use)
Pros: very simply creates hexadecimal codes from selected colors
Cons: not zippy and RAM hungry
Remote location: <URL:http://firehorse.com/colorhex/>
Tricks location: <URL:http://152.1.24.177/teaching/archives/ColorHEXb0.93.sit.hqx>


This very specialized utility allows you to create Netscape background, text, link, and visited link colors in a WYSIWYG manner. In other words, it translates color selections into hexadecimal codes.

By clicking the buttons at the top of its screen you select the item to be colored. Then, you can click on either a color wheel or a chart of colors. Doing so changes the sample text to the color you have chosen. After making selections for each of the items, you then have the option of creating a file describing your selections or copying your selections to the clipboard. From there you can paste these color codes into your editor or use the saved file as a template. This application is not zippy, but it sure gets the job done.

COLOUR HEXER

Version: 1.0
Author: Graeme Smith and Matthew Hudson
Requirements: HyperCard or HyperCard Player
Cost: free (?)
Pros: very simple
Cons: none
Remote location: <URL:http://www.arts.unimelb.edu.au/Horwood/hcf.html>
Tricks location: <URL:http://152.1.24.177/teaching/archives/colour-hexer.hqx>


This little HyperCard stack converts red/green/blue (RGB) codes into hexadecimal notation suitable for insertion into HTML documents. It does this by allowing you to point your mouse to either a color wheel, a swatch of color, or any point on your monitor. When you click the mouse button, Colour HeXer examines the point where you clicked and extracts point's color value, and translates it into a hexadecimal number. This is an especially useful utility if you plan to use any of the color-based Netscape attributes like BACKGROUND or to colorize your text. It means no more guessing at color values.

There are not many ways this tool could be improved except if it offered the option to copy to the clipboard the resulting hexadecimal code. In any event, it is well worth its price, $0.00.

FORMSGENERATOR

Version: 1.0
Author: Felipe Campos
Requirements: AppleScript
Cost: free
Pros: eliminates much of the need to know about FORMs
Cons: can't edit exiting FORMs, long FORMs would be tedious
Remote location: <URL:http://edb518ea.edb.utexas.edu/felipe.html>
Tricks location: <URL:http://152.1.24.177/teaching/archives/formsgenerator.hqx>


FORMSGENERATOR's purpose in life it create HTML+ FORMS. It is distributed as an uncompiled AppleScript script. By running it you will be prompted to answer many questions about the FORM you want to create. For example, it will ask you for the number of items you will be creating in your FORM and what types of items they are. Since you can't go back, it is a good idea to have your FORM well thought out ahead of time. After answering the questions exactly, FORMSGENERATOR will prompt you for a file name and location. Finally, the script will save your FORM. It will now be up to you to incorporate the FORM into a large HTML document and/or tweak the FORM's content to suit your needs.

FORMSGENERATOR is nice little utility, but any mistakes during the question and answer process are difficult to correct without starting all over.

HOTLIST2HTML

Version: 0.71
Author: Lutz Weimann
Requirements: MacWeb or MacMosaic hotlist files
Cost: free
Pros: by default, sorts hotlists
Cons: none
Remote location: <URL:ftp://ftp.ncsa.uiuc.edu/Mosaic/Mac/Related/hotlist-to-html-071.hqx>
Tricks location: <URL:http://152.1.24.177/teaching/archives/hotlist-to-html-071.hqx>

 Where Hotlist Grabber creates text files from MacWeb or Mosaic hotlists, Hotlist2HTML goes one step further and provides, by default, the option to sort the resulting text file as well.

Like Hotlist Grabber, Hotlist2HTML simply allows you to select a hotlist file and then supply an output name. then extracts the URLs from the hotlist file's resource fork and creates a text file of your URL's. By default, this utility sorts your URL's in the text file alphabetically by name. You can override this feature by holding down the shift key while selecting your hotlist file. Hotlist2HTML does not support the conversion of text files (lists of URLs) into native Mosaic nor MacWeb hotlist files.

HTML GRINDER

Version: 2.0
Author: Michael Herrick
Requirements: 1600K of RAM, pre-written HTML files, AppleScript extensions
Cost: $15 per module (shareware)
Pros: very easy to use, does small list of very specific tasks very well
Cons: limited functionality
Remote location: <URL:http://www.matterform.com/mf/grinder/htmlgrinder.html>
Tricks location:
<URL:http://152.1.24.177/teaching/archives/grinder_bundle_2.0.sea.hqx>

 HTML Grinder provides you with a set of tools to use against you existing HTML

documents. These tools presently include: find and replace, glossary, index builder, replace tagged text, sequential linker, stretch list.

By dragging and dropping a set of HTML documents on to an AppleScript droplet, HTML Grinder first creates a list of files to modify. You then select a tool to use from the menu. HTML Grinder comes a basic very handy tool called Find and Replace. This tool is free, all the others cost $15 each. To use any of the tools you first select them form the menu and complete the simple dialog box. For example, the Find and Replace tool ask you for the text you want to locate and the text you want to use instead. It works very well.

The other tools work in a similar manner. The Glossary tool is an enhanced find and replace utility allowing you to find and replace multiple words or phrase. Replace Tagged Text is yet another variation of Find and Replace allowing you to enter two HTML tags and replace the text between those tags with other text. Index Builder creates a table of contents. Sequential Linker inserts HREFs in your documents linking them to the a table of contents, next, and previous pages. Stretch List modifies you existing HTML so every thing is looks like BLOCKQUOTES and creates new HTML files containing links to the edited version. The net (all puns intended) result are HTML files resembling the Macintosh Finder while viewing windows by name or size.

HTML Grinder works very well at the things it was designed to do, but be forewarned! HTML Grinder makes extensive modifications to your existing documents. If you don't like what it did, then the best recourse is to defer to back up copies of your data. Once HTML Grinder gets a hold of your documents, it may be very difficult to go back.

HTML Markdown

Version: 1.0.1
Author: Scott J. Kleper
Requirements: HTML text
Cost: free
Pros: easy to use
Cons: only works as a drag-and-drop application
Remote location: <URL:http://htc.rit.edu/scott.html>
Tricks location: <URL:http://152.1.24.177/teaching/archives/HTML-Markdown-101.hqx>

HTML Markdown works as advertised. This is always the sign of a good program. In a sentence, HTML Markdown is specialized find-and-replace application removing HTML tags from HTML files thus creating plain text files.

By dragging and dropping any HTML file on to HTML Markdown the application proceeds to remove HTML tags from the document and prompts for the name of an output file. Using any

text editor to view the resulting file demonstrates that HTML Markdown has done what you requested with a few minor exceptions and unexpected results.

First, HTML Markdown does not seem to recognize special characters like the greater than (>) or less than (<) sign. Consequently, you may end up with strings like ">" throughout your text. Second, if your HTML document contained multiple, sequential return characters, then the resulting text file will contain these sequences as well. Within HTML, these multiple return characters go unnoticed, but they become blatantly apparent after conversion with HTML Markup.

There are a number of times when you may need a utility like HTML Markdown. One such time is when you want to format an HTML document into a word processing document. While the present WWW browsers will let you save as text, they often insert many hard returns at the end of every line. HTML Markup does not do this. Furthermore, HTML does not try to interpret things like horizontal rulers and list items. This means you have fewer find-and-replace sequences to go through when formatting your converted HTML document to a word processing document.

HTML MARKUP

Version: 1.0b1
Author: Scott J. Kleper
Requirements:
Cost:
Pros: very easy to use
Cons: immature, but acknowledged as such
Remote location: <URL:http://htc.rit.edu/scott.html>
Tricks location: <URL:http://152.1.24.177/teaching/archives/HTML-Markup-10b1.hqx>

HTML Markup is the antithesis of HTML Markdown. Written by the same author, this application tries to create HTML documents from text files.

Acknowledged as beta-ware, this program works by dragging text files on to the application. Your are then presented with a dialog box allowing you to specify whether or not the first line is the title of the document, whether or not you want any list items created, whether or not the header should be centered, et cetera. Like WebDoor (described below) HTML Markup gets is smarts by knowing where the element to mark up are located. Thus, if you put various items in specific locations of your text, the HTML Markup will work as documented. Otherwise, HTML Markup will not create the the sort of HTML documents desired.

HTML Sketcher

Version: 1.1
Author: Michael Korican
Requirements: HyperCard
Cost: $5 (shareware)
Pros: creates and maintains sets of HTML template documents
Cons: none
Remote location: <URL:>
Tricks location: <URL:http://152.1.24.177/teaching/archives/htmlSketcher_1.1.hqx>

Html Sketcher, as its name implies, creates "sketches" of HTML documents. Put another way, this little, no-nonsense HyperCard stack creates HTML templates. By opening html Sketcher and entering text into its various fields, you can create sets of HTML documents. These documents are then intended to be opened up with other, more robust HTML editors for completion. If you need to maintain many sets of documents and/or templates of documents, then this stack may be of use to you.

HTML Special Effects

Version: 2.0
Author: Tim Howland
Requirements: Netscape 1.1N
Cost: $5 (shareware)
Pros: creates complete color template files for extended Netscape documents
Cons: none
Remote location: <URL:>
Tricks location: <URL:http://152.1.24.177/teaching/archives/html-special-effects.hqx>

HTML Special Effects is another useful utility for setting the colors of your "Netscape-aware" HTML documents. It allows you to do four things: set the default font size of your document, create a marquee from the document's title, automatically create a fade in from black or white to your document's background color, and select colors of each of your document's text elements.

To use HTML Special Effects open the application and begin by choosing the colors of your document. By selecting one of the radio buttons (Background, Body text, Link text, etc.) you can then click on the color wheel and choose a color. HTML Special Effects then displays your selection in a preview window. If you are not satisfied with the color, then you can fine tune it using the plus (+) and minus (-) sign buttons. The other three options (default font, marquee, and fade) are provided through simple checkboxes and radio buttons; they're no-brainers.

Once you have made your selections you can save the options wherein you can open the resulting document and use it as template for further HTML markup.

HTML Special Effects is a nice utility. The ease in which you can create colorized documents is directly proportional to the easy in which you can create true ugly and difficult-to-read documents. You have been warned.

HTML TableTool

Version: 1.0 (?)
Author: Bertil Holmberg
Requirements: HyperCard 2.2 or HyperCard Player 2.2
Cost: free
Pros: creates simple tables nicely
Cons: could use some command key sequences or scripting
Remote location: <URL:>
Tricks location: <URL:http://152.1.24.177/teaching/archives/table-tool.hqx>


HTML TableTool is a nice little HyperCard utility for creating HTML tables from tab-delimited text files. Any text file will do as long as each item in the file is separated from every other item by a tab character. As its documentation states, these text files are intended to be the output of spreadsheet or database programs.

To use HTML TableTool, first create your tab-delimited text file. Next, open HTML TableTool and select your preferences. HTML TableTool includes options for making the first item of every row a "row header" as well as the first item of every column a "column header". It also provides options for border, cell padding, captions, center, and width. Once you've set your preferences you use the Open button to import and markup your text file. Your results are then displayed in a scrolling text field for copying or saving.

The utility includes two other useful features. One, it can translate ASCII characters greater than ASCII 126. This means it will convert things like å into &aring. Two, it indents each of your rows from the left-hand column by a number of spaces. This makes reading your tables remarkably easy.

HTML TableTool works well, but do not think it provides all the features of all tables. Notably, it does not support things like ROWSPAN, COLSPAN, nor allow you to specify the width of your border. These attributes as well as the addition of a few command-key sequences to updated and save the tables as you tweak with them are the only improvements it could use, but these improvements would just be icing on the cake.

HTML+ XTND Translator

Version: 1.0b1
Author: Leonard Rosenthol
Requirements: XTND compatible word processor
Cost: free, but payments are accepted
Pros: easy to use, straight-forward
Cons: only supports output, does not provide all markup options
Remote location: <URL:>
Tricks location: <URL:http://152.1.24.177/teaching/archives/html-plus-xtnd.hqx>


HTML+ XTND Translator takes advantages of Claris's XTND technology just like another XTND translator here, XTND HTML Translator. Consequently, you must have the XTND Translator software previously installed on your Macintosh before this utility will work. Your most like have this software if you have purchased some sort of Claris product like MacWrite Pro, Claris Works, FileMaker Pro, or even AppleSearch.

Written by Leonard Rosenthol, the author of numerous Macintosh applications, this translator works as advertised. To use the translator you first put it in the Claris Translator folder of your Claris folder of your System folder. Then, using your XTND-compatible word processor like MacWrite Pro, Claris Works, BBEdit, or WordPerfect you begin creating writing your document. Like most of the tools described here, it really behooves you complete your document first, and then mark it up.

To add markup tags you must format your document with particular characteristics. Specifically, you to create headers you must change the size of your text from 24-14 points. To create the various types of lists (ordered, unordered, menu, definition, and directory) you simply change the color of your text (red, blue, green, magenta, and cyan). You can also use bold, italics, and underlined text become STRONG, CITATION, and EMPHASIS markup tags. A particularly nice feature of this translator is that is automatically converts extended characters into their marked-up equivalents. For example, when you type an option-a (&aring), HTML+ XTND Translator automatically maps this this to &aring.

You can also add pictures to your document, then, when you save your document through the XTND translator, it extracts your pictures and automatically saves them as PICT files. To facilitate the completion of you HTML, HTML+ XTND Translator comes with an accompanying utility that converts double line break symbols (<br>) into paragraph marks (<p>). It also uses clip2gif to convert the extracted PICT files into gif files. Thus, after writing your document you can simply put all the necessary documents where you want them on your server.

The translator works, but it is not complete. I affords no way of creating links to other documents, nor things like tables. It does not support translation from HTML into your XTND-compatible word processor; in other words, it only does export and not import. On the other hand, it does what it says it can do and it does it well.

Incidentally, in the Tricks archive there is a MacWrite Pro stationary file that has been configured to used the markup option supported by HTML+ XTND Translator. To use it, download the file (INSERT URL HERE) and choose the markup options from the Style's Character menu. This way you don't have to remember what HTML+ XTND Translator supports.

RTFTOHTML

Version: 2.7.5
Author: Chris Hector
Requirements: 256K of RAM, RTF (Rich Text Format) files
Cost: free
Pros: quickly creates HTML documents
Cons: effectiveness depends on your validity of your RFT-generated files
Remote location:
<URL:ftp://ftp.cray.com/src/WWWstuff/RTF/rtftohtml_overview.html>
Tricks location: <URL:http://152.1.24.177/teaching/archives/rtftohtml-mac.hqx>

The purpose of rtftohtml is to convert rich text format (RTF) files into HTML files. In a (long) sentence, your ability to use rtftohtml effectively is directly proportional to your word processor's ability to save valid RTF files, your word processor's ability to create "paragraph styles", and your ability to interpret the html-trans file that comes along with the rtftohtml distribution.

Like the XTND translators described here, the concept is a good one. Fire up your word processor and being entering text. To denote HTML markup, define paragraph styles corresponding to attributes listed in the html-trans file that comes with rtftohtml. For example, a paragraph style called "heading 1" and applied to a line of text will ultimately surround the selected text with the <H1> and </H1> tags. Similarly, a paragraph style called "address" will generate the ADDRESS tags.

After "marking up" your document with the paragraph styles from the html-trans file, save your document as an RTF file. (RTF is intended to be cross-platform file format defined my Microsoft.) Now, use rtftohtml to open and convert your RTF file. If all goes well, a new document will be created with the same name as your original, except it will have an ".html" extension. If rtftohtml encounters any pictures in your file, it will export them as a PICT file and give them a unique name. If your file contains any tables, then it will create a rudimentary table using the PRE constructs. If your file contains any simple bold, italic, or underlined text, then it will surround the characters with those formats with the bold, italics, or underline attributes, respectively. Footnotes and headers generate separate files and table of contents pages linked to the original.

The best way to use rtftohtml is to delete all the text from the included sample_styles.rtf file and save it as a stationary file in the format of your favorite RTF-compatible word processor.

Unfortunately, not all word processors are going to read the file correctly (at least WordPerfect nor MacWrite Pro didn't), and consequently, all the enclosed styles don't get correctly translated or interpreted. Furthermore, rtftohtml expects to find certain styles while reading its RTF files. Specifically, if you want to create any hypertext links, then your word processor will have to support the "hidden" style, a style WordPerfect nor MacWrite Pro support.

In summary, rtftohtml is a very useful utility if you have Microsoft Word and/or if you are in hurry and you want to create an HTML document very quickly from a previously existing document that you can save as an RTF file.

WEBDOOR PUBLISHER

Version: 1.0
Author: Open Door Networks
Requirements: HyperCard and optionally a modem
Cost:
Pros: straight forward and almost WYSIWYG
Cons: limited feature set and layout possibilities
Remote location: <URL:http://www.opendoor.com>
Tricks location: <URL:http://152.1.24.177/teaching/archives/webdoor-publisher-10.hqx>


WebDoor Publisher could be considered an HTML editor, yet its feature set is so limiting that it is more like a utility. Open Door Network's goal is to make publishing on the World Wide Web easy. That it most certainly has done, but to really express yourself in the ways you want, you might as well learn a full-blown HTML editor.

WebDoor Publisher is a template in the form of a HyperCard stack. The template looks like a simplified WWW browser. You begin by filling in fields like the title bar, which ultimately becomes the contents of the <title></title> tags. Next, by clicking on a button you can select a graphic image to be inserted into your file. Third, you give your page another title. Fourth, you enter (or paste) what you want to communicate into the body of the template. To incorporate hypertext links and more graphic files into the body of your text you mark the locations of these items with asterisk/number symbols (**1 for example) and underlined text. Fifth, you sign your document with your email address.

Once you have finished writing your text you "create" your HTML document by clicking the Create button. The HyperCard stack then generates HTML codes for you based on the contents of the template.

By definition, because a template is being used to generate your HTML code, all of your HTML will have the exact same look and feel. In many ways, this is a sign of good graphic design. On the other hand, your pages will not be distinctive when compared to everybody else's using WebDoor. Consequently, you may want to edit the resulting HTML file anyway.

Another feature that makes WebDoor different from everything else described here is its ability to actually have your document made available on the World Wide Web without the use of your own World Wide Web server. Instead, this service can be provided to you through Open Door Networks. If you have a modem, then you can establish an account with Open Door Networks. After creating an account, you can connect to their AppleShare server, upload your documents, and instantly publish your pages on the Web. Of course they provide all these services for a fee.

Open Door Networks has come up with an easy to use method of putting your pages on the World Wide Web at the expense of direct control over your own documents. This approach will be satisfactory for many people, but for you, the readers of this book, you may find their solution a bit limiting.

XTND HTML TRANSLATOR

Version: 1.0
Author: Brian A. Sullivan and Jonathan Ryan Day
Requirements: XTND-compatible word processor, System 7
Cost: free
Pros: represents a good concept
Cons: not fully implemented
Remote location: <URL:http://ai.eecs.umich.edu/highc/software/translator.html>
Tricks location: <URL:http://152.1.24.177/teaching/archives/HTML-Translator-1.0.hqx>


Using an XTND-capable word processor like Claris Works, WordPerfect, or MacWrite, you can write documents and then supposably save them as HTML documents. HTML Translator was developed using Claris Works. Not having access to this word processor, other XTND-capable word processors (MacWrite II release 1.1 and WordPerfect 3.0) were used for evaluation. Using MacWrite, when saving any documents consistently resulted in a bus error forcing reboot of the computer.

More success was gotten using WordPerfect, but the success was limited. According to the documentation, text sizes within particular ranges were to be translated into headers. This never happened. Left-justified tabs are to be interpreted as list items, but the translator created <DD> tags, tags of which the author is not familiar. Bold, italics, and underlined text worked as advertised. Hypertext links are implemented by creating footnotes. The Translator did create hypertext links, but they were invalid. Driven by frustration, attempts were not made at importing graphics.

HTML Translator exemplifies what people want in terms of creating HTML documents, the ability to use their existing word processor in a WYSIWYG manner and "save as" HTML. Unfortunately, HTML Translator is not quite mature enough to allow people such a luxury.

GRAPHIC UTILITIES

This chapter reviews a few HTML graphic utilities.

You will inevitably want to incorporate graphics into your HTML documents. While just about any graphics program will suffice for the creation of your graphic files, you will have to modify them for the WWW medium. First, you will probably want to save them as (or convert them to) GIF images since all graphical browsers support this format inline. Another thing you may want to do is make sure your images are transparent. This removes the "blockyness" of some of your pages. Clip2gif is a great utility for both of these functions. Transparency works well for the second. Finally, if you want to incorporate QuickTime movies into the files you deliver, then you will want to use FlattenMooV so your files get delivered correctly.

One final note, it is generally not a good idea to incorporate too many graphics into your pages. Many people will be accessing your server through relatively slow modem connections. Transferring images takes a long time. Make sure the wait your reader's have to bear is worth it. Furthermore, some of your readership may have their graphics turned off or they may be using a text-only browser. Consequently, make sure your graphics "add value" to your documents, otherwise you will simple annoy your readership.

SEE ALSO

"Computers:World Wide Web: HTML Editors:Macintosh"
<URL:http://www.yahoo.com/Computers/World_Wide_Web/HTML_Editors/Macintosh/>

"Macintosh WWW Development Resources"
<URL:http://www.uwtc.washington.edu/Computing/WWW/Macintosh.html>

Chuck Shotton, "How to serve QuickTime movies" - Serving QuickTime movies from a Mac

running MacHTTP is really no different than serving any other binary document (such as GIF or JPEG files) with one exception. The movie data must be in a proper, flattened format.
<URL:http://www.biap.com/machttp/howto_qt.html>

Daniel W. Connolly, Bruno Girschweiler, and Tim Berners-Lee, "Graphics formats for WWW" - "The World-Wide web is a multimedia information space. This means that one of the design issues is dealing with various data formats. In addition to HTML and other textual formats, since the widespread availability of NCSA Mosaic and other visual interfaces to the web, more and more of the web's information is represented or augmented with data in any number of popular graphics formats."
<URL:http://www.w3.org/hypertext/WWW/Graphics/Overview.html>

Robert Lentz, "Macintosh Web Programs and Utilties"
<URL:http://www.astro.nwu.edu/lentz/mac/net/mac-web.html>

Yahoo, "Computers: World Wide Web: HTML Editors:Macintosh"
<URL:http://www.yahoo.com/Computers/World_Wide_Web/HTML_Editors/Macintosh/>

CLIP2GIF

Version: v.0.3
Author: Yves Piguet
Requirements:
Cost: free
Pros: extremely simple, easy to use, multi-purpose application graphic application that is AppleScriptable
Cons: doesn't display images
Remote location: <URL:http://iawww.epfl.ch/Staff/Yves.Piguet/clip2gif-home/>
Tricks location: <URL:>

 Clip2gif, in many ways, combines the abilities of graphic converters and Transparency into one easy-to-use program. It does this with a rather Spartan interface with no windows.

To use clip2gif simply launch it. Next select graphic options from the Options menu. Options include the ability to save files as in PICT, GIF, or JPEG formats. If the format is GIF, then you have the option of saving the file as an interlaced file and/or a transparent file. If you select JPEG, then you have a number of image quality options. In either the GIF or JPEG formats you have the options of selecting the number of colors in the image (pixel depth) and/or scaling the image.

After selecting your options, you can apply those options to graphic images in either of two ways. The first way is via the clipboard; by selecting "Save clipboard as..." from the File menu any PICT image in the clipboard is saved to the file name you specify. Alternatively, you can use the File menu's "Convert PICT file..." option to select a graphic from a dialog box.

The selected file will then be duplicated and the selected options will be applied to the newly created file.

The only problem with this utility is that you never actually see the graphics you are manipulating. This isn't necessarily bad. It's just irregular for most Macintosh-based programs.

Clip2gif is also completely AppleScriptable. This means it can become an indispensable utility in common interface gateway (CGI) scripts that create graphic images "on the fly." The example called "See My Room" in the chapter on CGI scripting does just this. Please refer to that section for more information.

FLATTENMOOV

Version: 1.0 (?)
Author: Robert Hennessy
Requirements: System 7
Cost: free
Pros: works flawlessly
Cons: none
Remote location: <URL:ftp://ftp.ncsa.uiuc.edu/Mosaic/Mac/Related/flattenmoov.hqx>
Tricks location: <URL:http://152.1.24.177/teaching/archives/flattenmoov.hqx>


If you plan to have QuickTime movies available on your server, then you should consider flattenMooV as a necessary tool for your software toolbox.

Everybody knows that QuickTime movies are relatively large files. Unfortunately, compressing QuickTime movies does not accomplish very much since they have already been compressed using the QuickTime technology itself. It makes even less sense to encode QuickTime movies into a BinHex documents since that will make the files even larger than they already are. Consequently, to preserve the movie's contents you will have to configure your server to deliver QuickTime movies as BINARY files.

To complicate matters, MacHTTP (and WebSTAR) only serve the data fork portion of any given file. QuickTime movies contain a resource called "moov" in their resource fork necessary for any QuickTime movie to play. This is where flattenMooV comes in. flattenMooV copies and duplicates the moov resource of QuickTime files into the movie's data fork. If applied to the movies you deliver through your server, your QuickTime movies will contain all the information necessary for playback in the data fork, and your movies will be playable. Furthermore, if you intend Windows-based platforms to be able to play your movies, then flattenMooV is even more necessary since the Windows-based platforms can not interpret the resource/data fork construct at all.

Understanding all this technical jargon is not necessary for actually using flattenMoov. Knowing that you need to "flatten" your movies is.

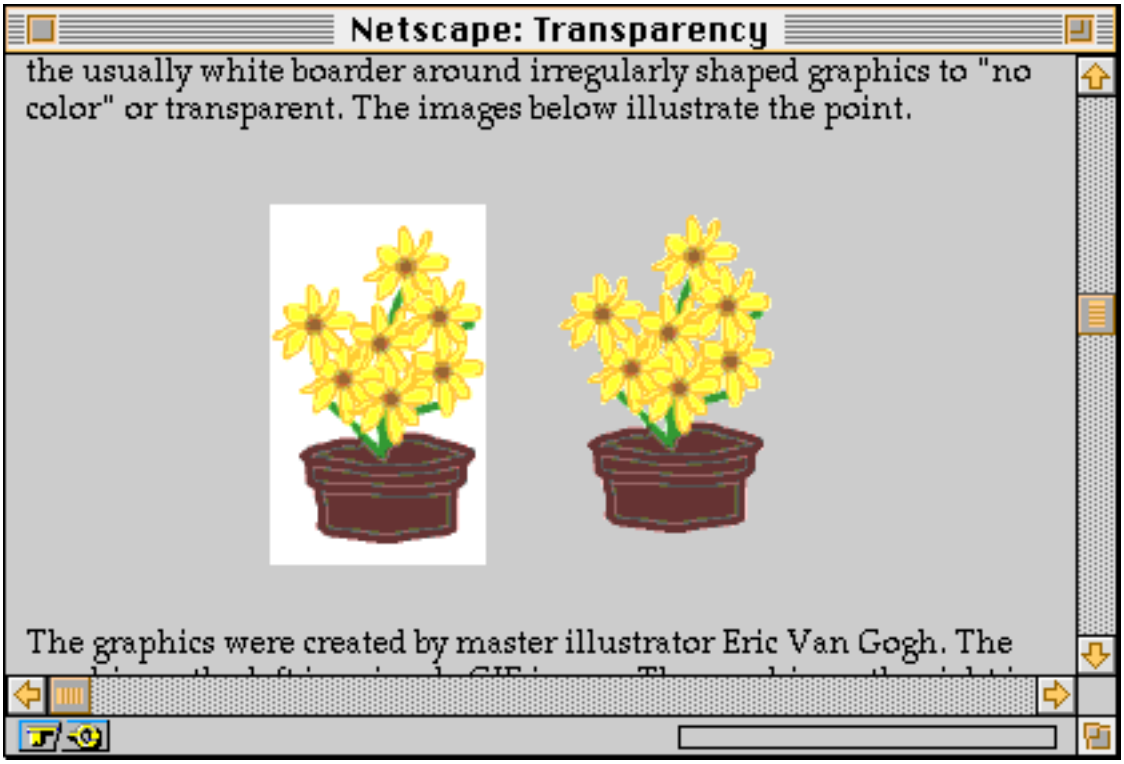
To use flattenMoov, launch it, select a QuickTime movie file, confirm that the selected file is the movie you want to "flatten", give the movie a new name, and click the Save button. Now your movies will be deliverable through your server.

flattenMooV works flawlessly.

TRANSPARENCY

Version: 1.0b3
Author: Aaron Giles
Requirements: 384K of RAM
Cost: free
Pros: extremely simple and easy to use
Cons: none
Remote location: <URL:>
Tricks location: <URL:http://152.1.24.177/teaching/archives/transparency.hqx>

Transparency is an indispensable utility for anybody incorporating GIF images into their HTML pages. This free little utility, distributed as a "fat" binary, allows you to add an element of professionalism to your pages. It does this by changing the usually white boarder around irregularly shaped graphics to "no color" or transparent. The images below illustrate the point.



The graphics were created by master illustrator Eric Van Gogh. The graphic on the left is a simple GIF image. The graphic on the right is the same graphic after the use of Transparency. While the artistic qualities of the graphics can be debated, few people would argue that the image on the left is more appealing than the image on the right. This is the value of Transparency.

Using Transparency couldn't be easier:

1. Create your graphic with what ever program you desire.
2. Save the graphic as a GIF file or convert your graphic with some sort of graphic converter.
3. Open Transparency or drag your graphic(s) on to the Transparency icon. Transparency will display your graphics.
4. For each of your graphic images, click on the image and a palette of colors appearing in the image will appear.
5. Choose the color you want to change to "no color" and let go of the mouse.
6. Alternatively, choose "None" from the very top of the palette. This is equivalent to no change or returning the image to its original state.
7. Transparency will change the selected color to transparent and display the graphic against a gray background so you can get a sense what your graphic will look like on other people's browsers.
8. Save the graphic and include it in your documents.

I feel I should be writing more about Transparency, but it is so easy to use that any more words would be wasted. This is a great utility.

3 ESSENTIAL QUALITIES OF INFORMATION SYSTEMS

This chapter enumerates 3 essential qualities of useful information systems: readability, browsability, and searchability.

In order to be useful, any information system must be readable, browsable, and searchable. This chapter enumerates guidelines and outlines qualities of readability, browsability, and searchability so you, as an information system manager, can incorporate these ideas into your products and services.

An information system, in the present context, is any organized collection of information. In our culture, information systems abound. The dash board of our cars are information systems. Maps are information systems.World Wide Web servers are no exception. While World Wide Web servers are primarily intended to be an electronic publishing medium they are also information systems. In order to be most effective, all but the smallest of information systems must be readable, browsable, as well as searchable.

All of these qualities (readability, browsability, and searchability) do not have to be equally represented in every information system. As your collection of information increases, different aspects of these qualities take on greater significance. Thus, the amount of readability, browsability, and searchability your information system exhibits depends on the type and quality of your collected data, as well as the information needs of your clientele.

Readability

Readability means good page layout.

Readability connotes an appealing graphic design and page layout. All information systems, no matter how small must incorporate principles of good graphic design. You and your information system is competing with a myriad of other information systems. If your data is not presented in a visually appealing, easy-to-read manner, then your chances of retaining the attention of your intended audience are significantly reduced.

Guidelines

- Use a consistent layout
- White space is good
- Visually organize the page; employ horizontal rules
- Keep pages short
- Include elements of contrast
- Use all stylistic elements in moderation

Use a consistent layout

Your documents reflect you, your organization, and your information. By consistently using the same layout you are creating a unified whole, an identity. With the use of a consistent layout it is easier for the reader to know when they are reading your text and not someone else's. The creation of a template file can be helpful here. The template file would consist of your standard headers, footers, logo, signature, last data updated, as well as any other stylistic features you may incorporate.

White space is good

White space is the empty areas of a page. It adds contrast and provides a place for you eyes to rest. White space is not wasted space. For this same reason, stay away from all capital letters. Capital letters are usually the same height and width. This creates a block effect reducing the white (negative) space around letters. Instead, use a combination of lower and upper case letters because it increases the amount of white space around the letters.

Visually organize your pages

In other words, group similar concepts on your page together. Employ proximity. Don't put our email address at the top of the page and the URL to your personal home page at the

bottom. Both items are electronic pointers relating to you. Group them together. When there is more than one type of information on a page, delimit the page with white space or horizontal rules.

KEEP YOUR PAGES SHORT

In general, people do not like reading text from a computer screen. Using the popular vertical scroll bar of graphical WWW browsers it is very easy to get lost in a document. In general keep you pages shorter than two or three screens in length. By keeping your pages short and to the point, the attention span of your readership will increase. Put another way, break up long pages into shorter ones.

INCLUDE ELEMENTS OF CONTRAST

A boring looking page is completely filled with text. It contains no white space, no change in fonts or font sizes, no lists, no pictures. It is boring. Elements of contrast breakup the monotony and make our page more dynamic. Elements of contrast include emphasizing some text with styles like or . Other examples include the use of very heavy horizontal rulers or very large headers. With the current version of MacWeb and MacMosaic you as a information provider can distribute preference files defining the size of style of fonts you want your readers to use. This gives you the ability to select different font families for different parts of your documents. Unfortunately, this is an extremely uncommon practice and difficult to actually implement.

USE ALL STYLISTIC ELEMENTS IN MODERATION

Be a stoic Greek, "All things in moderation." The use of too many headers gets old and the reader feels like you are shouting. Too many emphasized elements loose their distinction. To many graphics take too long to download no matter how fast your computer or network connection is.

SEE ALSO

HTML Writers Guild, "Principles of Good HTML Design" - "This is a list of general principles of quality HTML design. It is intended to educate HTML authors to the elements of good and bad HTML style. It does not seek to "control" Guild members, but rather to encourage them to adopt these practices in their everyday HTML construction. <URL:http://ugweb.cs.ualberta.ca/~gerald/guild/style.html>

Jan V. White, *Graphic Design for the Electronic Age*, (Watson-Guptill : New York 1988)
Robin Williams, The Non-Designer's Design Book (Peach Pit Press: Berkeley CA 1994)

Roy Paul Nelson, *Publication Design*, 5th ed. (Wm. C Brown: Debuque IA 1991)
Yale Center for Advanced Instructional Media, "Yale C/AIM WWW Style Manual" - This is one of the more scholarly treatments of the subject.
<URL:http://info.med.yale.edu/caim/StyleManual_Top.HTML>

BROWSABILITY

Browsability means logically classifying your data and information.

As the size of your information system grows, so does the need to logically organize your data. This implies grouping conceptual sets of data with similar conceptual sets of data. Browsability becomes apparent when it is coupled with hypertext and logical groupings of information. A browsable information system has a number of advantages.

ADVANTAGES

- Readers see entire system at a glance
- Knowledge of a vocabulary is not necessary
- Like items are grouped together
- Easy to navigate
- Fosters serendipity
- Stimulates thinking

But a solely browsable system is not without its disadvantages as well.

DISADVANTAGES

- Easy to get "lost"
- Classification system may be foreign to reader
- Classification breaks down as quality of information increases

CLASSIFICATION CHANGES OVER TIME

An easily browsable system is logically organized by topics. Effective organization your information is critical to the success of your server. This point cannot be overstated and bears repeating. Effective organization of your information is critical to the success of your server. If you want your server to be an effective information tool and you want people to use your server more than once, then it must be organized.

A PHILOSOPHY OF CLASSIFICATION

Classifying knowledge and bringing like things together have been fundamental aspects of at least Western culture since the before the Golden Age of Greece when philosophers systematized their ways of thinking. This classification process was their way of creating an intellectual cosmos from the apparent chaos of their experience. The process brought a sense of order to their disordered society. It provided a common ground for others to work from and to use as a basis for further discovery. Without organization intellectual anarchy rushes in to fill the void. Given enough time, eventually, no one is speaking the same intellectual language and communication breaks down. Stagnation sets in.

"SOUR MILK"

To make matters worse, "One person's cheese is another person's sour milk." In other words, one person's view of the world may not adequately represent the next person's view. Interpretations of the night sky represent an excellent example. Everybody in the northern or southern hemispheres have exactly the same data in which to make interpretations, the stars. Yet every culture creates their own distinct constellations and explanations about what they mean. The interpretations these people make reflect their culture. War-like cultures see warriors. Fishing cultures see boats. Wandering peoples see migrating animals.

A CULTURAL EXAMPLE

For better or for worse, classification systems of knowledge ultimately break down. This is because cultural revolutions and technological change occur. For example, the Medices of the late Middle Ages used their newly found wealth to hire artisans and craftsman. These creative individuals explored new ways of looking at the human form and rekindled an interest in humanity. These cultural perception shifts were then picked up by others throughout Europe. At the same time, technologies like the telescope offered people like Galileo a new perspective of the skies. For example, he noticed Venus moves through phases just like our moon. These observations lead Galileo to believe Venus did not revolve around the Earth as previously thought but around the Sun instead. While these two phenomenon (the rekindled interest in humanity and observations of Venus) did not by themselves alter a system of knowledge, these phenomenon represented the beginning of major intellectual shift in Western thought. Specifically, these two phenomenon contributed to the Renaissance and Reformation where Western civilization's entire intellectual basis where shaken. Therefore, old ways of thinking always seem to give way to new ways of thinking and the process begins anew. Put another way, there is no perfect intellectual organization of knowledge or information just as there is no perfect circle. Similarly, as the habits and technologies of societies change so does their interpretation of the "cosmos" they live in.

Back to the Real World

The point is, despite the dynamic nature of intellectual constructs, the organization of information and knowledge seem to be a necessary part of human existence. Since the primary purpose of information servers is to disseminate knowledge, facts, and ideas, it then follows the information they disseminate must be organized in some reasonable fashion.

Guidelines

- Know your audience
- Provide an "about" text
- Use the vocabulary of your intended audience
- Create a hierarchal system of ideas
- Create a system that is both flexible and exhaustive
- Classify by format last

Know your audience

If your intended readers cannot make sense of your server's organizational scheme, then they will only use it as a last-resort information resource. Thus follows the first and foremost guideline for a useful organizational scheme. The organizational scheme must be comprehensible to your intended audience. Think about the people who will be using your server. What are their backgrounds? What do they want? What specialized terminology do they use? In general, how do they think? Incorporate the answers to these questions into the structure of your server. To paraphrase a respected librarian, "Servers are for use" and, in order for this to happen, your organizational scheme must be understandable by the majority of your intended clientele. A thesaurus listing the vocabulary of a discipline may be indispensable in this regard, especially for those of you who are creating collections of Internet resources.

Provide "about" texts

Embed "about" texts, texts describing your organizational system, its intended audience, and how it can be used, within as much of your system as possible. You can not include too much explanatory information as long as it stays within the guidelines for good readability.

Use the vocabulary of your readership

Once you have identified your intended audience, use their terminology. Thus your readership will identify with your information system and be more likely to use again and again.

Create a Hierarchal System of Ideas

By definition, a hierarchal system of ideas begins with broad terms and is subdivided into narrower terms. There is no perfect hierarchal system of ideas, but your intended audience will bring to your system some preconceived ideas on how information on their topic should be organized. These people will have similar, but not exact preconceptions. Biologist think similarly, just as computer scientists think similarly. Create a hierarchal system fitting the preconceptions of as much of your intended audience as possible. This is called "literary warrant." At this point it may be helpful to employ the use of a specialized thesaurus or handbook written for your audience. This thesaurus or handbook will contain definitions of the discipline's terminology. These tools will help clarify and provide a structure for your hierarchal system.

Create a System That is Both Flexible and Exhaustive

Make sure the system is exhaustive as well as flexible. In other words, create a structure striving to be both enumerative and synthetic.

Enumerative classification attempts to assign designations for (to enumerate) all the single and composite subject concepts required in the system. . . . Synthetic classifications are more likely to confine their explicit lists of designations to single, unsubdivided concepts, giving the local classifier generalized rules with which to construct headings of composite subject.

Classify by Format Last

Organize materials based on format as a last resort. People usually don't care what format the data is in just as long as the answer to their query can be found. This means when you have a collection of various Internet resources group them by subject first and Internet protocol last; do not put all the telnet sessions in one section, FTP sites in another section, gopher sites in a third, WAIS sites in a fifth, and so on. Organizing your information by topic rather than form brings like things together and end-users will not have to navigate throughout your server for the information they need.

See Also

Aslib, *Proceedings of the International Study Conference on Classification for Information Retrieval* (London: Aslib, 1957)

Bohdan S. Wynar, *Introduction to Cataloging and Classification* (Libraries Unlimited: Littleton CO, 1980) pg. 394

Derek Langridge, *Approach to Classification for Students of Librarianship* (Hamden, Connecticut: Linnet Books, 1973)

Searchability

Searchability means direct information access.

The largest of information systems must include search features. These features help overcome the disadvantages of the purely browsable system. They have a number of distinct advantages.

Advantages

- Creates alternative logical classifications
- Simplifies location of known items
- Works independently of collection size

As described in the previous section, your conception of the information universe is not necessarily the same as your reader's. While you try to group things in the most logical manner, your reader's "logic" will be different than yours. Searchability can help over come this discrepancy by allowing the reader to create their own set of logically similar items.

Searchability readily lends itself to locating known items rather than making the reader browser down a number of menus to get what they want. Similarly, a reader may have used an item from you before an not put it into their holist. If you have a searching mechanisms in place, then it may be easier for the reader to find the item again.

Searchability works independently of your collection's size. Browsable systems begin to break down after the collection becomes too large. The effectiveness of searching an information system is not directly impeded by the size of the system.

Yet, purely searchable systems are not perfect either.

Disadvantages

- Must know searching syntax
- Must have a preconceived idea, phrase, or term
- Must know the structure of the data

In order to effectively search an information system, the reader must know the query language of the search engine. This may include Boolean logic or Unix regular expressions. They may have to know the meaning of right-hand truncation and the symbol for its use. While this sort of knowledge is necessary to use the system, it is irrelevant to the information itself and is seen by the reader as an impediment.

The ability to search an information system assumes your readership has a preconceived idea describing what they need. This is a notorious example of the chick and egg problem. How are you suppose to find information about a particular topic if you don't know about that topic in

the first place. In other words, the reader must come to the information with some terms or phrases describing want they want to find. Many times those terms or phrases will not be found in the collection, but synonyms will be found. It is difficult to think of many synonyms and it is difficult to "guess" at the controlled vocabulary used by the collection.

Finally, totally searchable systems require the searcher to know the data structure of the indexed collection. Is the data divided into field? If so, what are those fields and how do they specify them in their query?

What to do? Here are some guidelines for creating searchable systems.

Guidelines

- Include help texts
- Map located items to similar items
- Provide simple as well as "power user" search mechanisms

Include help texts

Just as "about" texts are necessary in a browsable system, help texts are necessary for searchable systems. Help texts describe the features and limitations of the system. They list system's data structure including fields available for searching and the contents of those fields. Help texts also list plenty of example searches and provide explanations on what the end-user should do if they encounter too many or too few results.

Map located items to similar items

After items are located with the search mechanism, there should be links to similar items. This answers the perennial question, "Can you find me more items like this one?" These links should go directly back to your browsable collection where the end-user can freely "wander". From there the end-user will have the ability to see terms that can be applied to more searches. This is where you provide the end-user with the vocabulary terms of your system, in case they are unfamiliar with your system of information organization.

Provide simple as well as "power user" search mechanisms

Simple search mechanisms will be most useful for the first-time or casual end-user. Unfortunately, these same mechanisms often return too many or too few "hits". Providing power user search mechanisms like field searching, truncation, Boolean qualifiers, and number-of-term limitations can compensate for the simple searches. Unfortunately, the cost of

these services is effectively describing the more powerful searching mechanisms to the end-user. Again, readability comes into play.

SEARCHING YOUR SERVER

This chapter reviews two methods of providing searchability to your server.

The ability to search your server is essential for all but the smallest of information systems. This chapter describes two tools enabling you to provide searching services: AppleWebSearch and TR-WWW. Both tools work as described in their documentation, but neither one is ideal since they return results in the form of file names and/or raw HTML. These tools need to return the titles of documents.

APPLESEARCH

Version: 1.5
Author: Robin Martherus
Requirements: MacHTTP 1.3.1b1 or later, AppleSearch client software, AppleSearch
Cost: free
Pros: Fast, few configuration files
Cons: AppleSearch is expensive and RAM hungry, default configurations may be difficult to edit
Remote location: <URL:http://kamaaina.apple.com/>
Tricks location: <URL:http://152.1.24.177/teaching/archives/applewebsearch.sit.hqx>

AppleSearch.acgi is a gateway program between your WWW browser, MacHTTP, and AppleSearch. AppleSearch is an indexing/document delivery technology developed by Apple. (It is a lot like the WAIS technology originally co-developed by Thinking Machines, Apple Computer, and Dow-Jones, but in reality is it based on the Personal Librarian technology.)

AppleSearch can provide two types of interfaces: an ISINDEX interface and FORMS interface. The ISINDEX interface is rudimentary but allows the same searching syntax as the FORMS interface.

To get AppleSearch to work for you, you must first purchase AppleSearch. It is available from most Macintosh mail order companies for about $1,400. It also a part of the Apple Internet Server Solution. After setting up the AppleSearch server, you must index your data. The AppleSearch.acgi application can only deliver ASCII text files, so make sure the data you intend to make searchable is simple text. Once your data is indexed, all you have to do launch the AppleSearch.acgi program and your are ready to go.

If you want to customize AppleSearch.acgi's user-interface, then you will have to edit a few TEXT resources from the AppleSearch.acgi application itself. The documentation explains how to do this, but be forewarned. Editing documents with tool like ResEdit is not like using regular text editors. Be sure to keep a backup of your work. Editing either the ISINDEX or FORMS interface to AppleSearch.acgi is not difficult, but just a bit scary if you do not have much practice with ResEdit.

AppleSearch.acgi and AppleSearch provide a fast and effective way to search a set of indexed data. Its searching features are robust and yet flexible. It supports nested Boolean queries, wild card searches, and can be used to search remote WAIS databases.

On the other hand, AppleSearch is expensive both in terms of dollars as well as computing resources. AppleSearch requires at least a 68040 processor and 5MB of RAM.

If your indexed data contains relative URLs, URLs not specifying the complete paths to other documents on our server, then the links in the documents returned by AppleSearch.acgi will not work. This is because AppleSearch includes the full path name of the documents it indexes. Consequently, since relative linked documents are being retrieved from a location other than the one intended by the document's author, the links fail.

Last, like TR-WWW, described in the next section, AppleSearch.acgi ultimately returns file names. AppleSearch.acgi does not return the HTML titles of documents. The means you, as the information provider, must save your HTML documents with the most meaningful names possible in order to help your end-users effectively evaluate the output of AppleSearch.acgi.

TR-WWW

Version: 1.3
Author: Adrian Vanzyl
Requirements: none
Cost: $50-$3000 (shareware)
Pros: No indexing required, supports relevance ranking as well as context searching
Cons: Not zippy, does not fully support Boolean logic
Remote location: <URL:http://www.monash.edu.au/informatics/tr-www.html>
Tricks location: <URL:http://152.1.24.177/teaching/archives/tr-www.sit.hqx>


TR-WWW is a shareware search engine from Monash University Medical Informatics in Australia. It is based on "Total Research, the complete research system for dealing with unstructured textual information, searching it, extracting information, generating reports and integrating this with a database." As the home page of TR-WWW states, "TR-WWW is a modified version of TR designed to work with MacHTTP. It has all the search strengths of the normal version of TR [Total Research], but requires a web client to access its functionality and to act as its user interface."

In short, TR-WWW is searches and retrieves documents from an unstructured collection of information (read text files) based on end-user input.

With a bit of tweaking, TR-WWW can provide simple or more robust interfaces to collections of data. In its simplest form, an interface can be presented only providing an input field and a submit button. The other extreme is to give the reader a choice context versus relevance ranking of hits, Boolean operations, or even the document set to be searched.

The real strength of TR-WWW is that it requires no indexing. Consequently, you can set up a mechanism for saving files in a directory and whenever requests are made to search the contents of that directory your data will be ready. Thus, TR-WWW is a good choice for dynamic data.

On the down side, TR-WWW does not seem to work well with "large" files (over 32K) nor with too many files in one directory (about 100). If the number of files is too great or the size of the files is too large, then TR-WWW times out or lacks the RAM for successful processing.

WWW SCRIPTING

This chapter outlines and demonstrates the fundamental concepts behind common gateway interface (CGI) scripting with MacHTTP and AppleScript.

The real potential of Web servers lies in their ability to run programs behind the scenes and return the results of these programs to the reader. This is known as common gateway interface (CGI) scripting. Basic CGI scripts include the ability to display the current time or the number of users who have accessed a server. More advanced and useful CGI scripts allow readers to search databases, complete survey forms and have them sent to an email address, implement imagemaps, or take pictures of remote places and have the pictures returned.

CGI scripts are made available to a Web browser through the use of simple links, specialized URLs containing a question marks (?), ISINDEX HTML tags, or HTML+ FORMs. After the user completes an HTML document containing one of these elements, a query is passed to the Web server, which is passed on to the CGI script itself. The script processes the input, formats the output into HTTP codes and/or an HTML document, and returns these items back to the WWW server. The server then passes it along to the client application.

CGI scripts can be written in almost any language. Common languages include C, Perl (MacPerl), AppleScript, Frontier (Aretha), or even HyperCard's HyperTalk. The limitation of the scripting language you choose is that it must be able to receive AppleEvents because AppleEvents are the technology by which MacHTTP communicates with the CGI program.

This chapter outlines how to write CGI scripts using AppleScript. This is because AppleScript is distributed with MacHTTP and it is the most widely supported Macintosh-based scripting language next to HyperTalk. This chapter does not discuss how to write AppleScripts in general since such a topic is beyond the scope of this book; CGI scripting is a specialized form of AppleScripting. Consequently, you are expected to have at least some AppleScripting experience.

If you do not have any experience writing AppleScripts, then please consult some of the texts and Internet resources found in the See Also section of this chapter.

The use of HyperTalk, Frontier, nor MacPerl are discussed in this chapter. HyperCard's HyperTalk is seen as relatively slow when it comes to other programming languages, and its overhead in terms of RAM, when compared to other languages, is high. This is true of MacPerl as well, although MacPerl has the slight advantage of being a cross-platform application since many operating systems support the perl programming language. Frontier (or Aretha as it is now being called), is a more robust scripting language when compared to AppleScript but its syntax is more like traditional programming. Furthermore, because of its internal data structure, it is more difficult to demonstrate CGI fundamentals without knowing a lot about Aretha itself.

SEE ALSO

"AppleScript Archives" <URL:ftp://gaea.kgs.ukans.edu/applescript/>

"Overview of CGI" - "This page contains pointers to information and resources on the Common Gateway Interface, a standard for the interface between external gateway programs and information servers." <URL:http://www.w3.org/hypertext/WWW/CGI/Overview.html>

"Common Gateway Interface" - [This is the official specification for CGI scripting.] <URL:http://hoohoo.ncsa.uiuc.edu/cgi/intro.html>

Chuck Shotton, "Using FileMaker Pro with MacHTTP" - An archive with sample forms and CGI that shows how to hook MacHTTP to FMPro. <URL:http://www.biap.com/machttp/examples/fmpro.sit.hqx>

Chuck Shotton, "HyperCard CGI Demo" - Sample stack and HTML form demonstrating how HyperCard can be used to process searches, forms, etc. using the new CGI file type and "sdoc" AppleEvent sent by MacHTTP. <URL:http://www.biap.com/machttp/ftp/hc_cgi_demo.sit.hqx>

Chuck Shotton, "Writing Search Engines for MacHTTP " - This points to an archive containing C source code for a sample application that performs searches in conjunction with MacHTTP using the "srch" AppleEvent. <URL:http://www.biap.com/machttp/ftp/search_ex.sit.hqx>

Daaron Dwyer, "CGIs In Mac C" <URL:http://www.mccom.com/CGIinC/>

Danny Goodman, *Complete AppleScript Handbook* (Random House: New York, 1994)

Dave Winer, "Aretha Website" - "Frontier is a scripting system for the Macintosh. Lots of features, lots of verbs. It used to be a commercial product, but now it's free. Why? Because I want Frontier to have a shot at becoming a standard. I think it'll be fun!" <URL:http://www.hotwired.com/staff/userland/aretha/>

Derrick Schneider, *Tao of AppleScript* (Hayden Books: Carmel, IN, 1993)

Grant Neufeld, "Grant's CGI Framework" - Grant's CGI Framework is a framework for writing 68K Macintosh CGI applications in C. It currently comes with a CodeWarrior project. Features: - AppleScriptable (including the CGI sdoc event!) - Threaded - Single function entry-point for customizing to your own needs - If you use it to write free applications, it's free to use <URL:http://arpp1.carleton.ca/grant/mac/grantscgi.html>

Ian Andrew Bell, "Macintosh WWW Tools Compendium" <URL:http://www.arpp.sfu.ca/tools/>

John O'Fallon, "Maxum Home Page" - [This page documents many of the products of Maxum.] <URL:HTTP://www.digimark.net/maxum/>

Jon Wiederspan, "Extending WebSTAR" - Below are links to pages that hopefully will help you learn how to create CGI applications to extend the capabilities of your WebSTAR server. The primary requirements for following these lessons are a willingness to learn, patience with my jokes, and a WebSTAR server to practice on. Since the language I use in these lessons is AppleScript, it would also be a good idea to have some familiarity with it, although many complete beginners find the information here sufficient to get going. <URL:http://www.uwtc.washington.edu/Computing/WWW/Lessons/START_HERE.html>

Jon Wiederspan, "Extending MacHTTP" - The following is a directory of the pages I have or hope to have soon that will help you learn how to use and/or create CGI applications to extend the capabilities of your MacHTTP server. The only requirement for following these lessons is a willingness to learn, patience with my jokes, and a MacHTTP server to practice on. <URL:http://www.biap.com/tutorials/Extending_MacHTTP/ExtendingMacHTTP.html>

Jon Wiederspan, "MacWWW - CGI Applications" - [This page lists various CGI applications for use with your Macintosh-based WWW server.] <URL:http://www.uwtc.washington.edu/Computing/WWW/Mac/CGI.html>

Jon Wiederspan, "Extending MacHTTP" <URL:http://www.uwtc.washington.edu/Computing/WWW/ExtendingMacHTTP.html>

Martin Fenner, Fred Terry, and PreFab Software, Inc., "ScriptWeb" - "This virtual site is a collaborative effort to provide a single source of information for Macintosh scripting, primarily for AppleScript and Frontier." <URL:http://www.gz.com/scriptweb/>

Mason Hale, "Frontier CGI Scripting" - This is the home of documentation and examples related to the writing and running of CGI scripts in the Frontier scripting environment. <URL:http://www.webedge.com/frontier/>

Matthias Neeracher, "MacPerl Q & A" <URL:http://err.ethz.ch/members/neeri/macintosh/perl-qa.html>

Netscape Communications, Inc., "Netscape API for the Macintosh" - "Netscape uses AppleEvents to interact with other Macintosh applications. It is scriptable, and partially recordable. Most of the events (and all Netscape-specific ones) are documented in the

Netscape's AppleEvent dictionary. You see the dictionary with the Script Editor, a scripting utility available from Apple." <URL:http://home.netscape.com/newsref/std/mac-remote-control.html>

Robert Godwin-Jones, "Guide to Web Forms and CGI Scripts for Language Learning" <URL:http://www.fln.vcu.edu/cgi/interact.html>

Sandra Silcot, "MacPerl Primer" - "This Primer is intended to assist new users get started with Macintosh Perl, and to point out salient differences for experienced Unix Perlers. This Primer is not a language reference manual, nor does it replace Matthias's documentation or Hal Wine's Frequently Asked Questions (FAQ) about MacPerl. The primer assumes you have already obtained and installed MacPerl, and that you have read the MacPerl FAQ." <URL:http://www.unimelb.edu.au/~ssilcot/macperl-primer/home.html>

Scott S. Lawton, "Scripting" - Our goal is to make this site one of the three most useful places on the Web for Macintosh script writers, focusing on AppleScript and Frontier. (We also don't mind some good -- hopefully friendly -- competition!) <URL:http://www.tiac.net/prefab/scripting.html>

Spyglass, Inc. , "Software Development Interface" - "This document describes a cross-platform API which can be used to extend the capabilities of Web browsers by integrating them with other applications. The API is specified as a set of platform-independent generic verbs which can be issued either to or from Web browsers. Platform-specific implementations of this spec are given for the AppleEvents and DDE transports. Future transports to be provided include OLE2 and TCP/IP (for UNIX)." <URL:http://www.spyglass.com:4040/newtechnology/integration/iapi.htm>

Tom Trinko, *Applied Mac Scripting* (M and T Books: New York, 1995) - The volume represents just about the only book available describing how to program Frontier.

"Hello, World!"

This section illustrates the simplest of .script files, the most rudimentary form of scripting for MacHTTP.

Here is the classic demonstration program used whenever new programming languages or techniques are demonstrated. This program, "hello-world-01.script" simply returns the text "Hello, World!"

Hello-world-01.script is one line long:

```
return "Hello, World!"
```

To create this script:

1. Launch your copy of Script Editor (or you could use any text editor)

2. Create a new file
3. Enter the single line above
4. Save it as a text file within your server's directory structure making sure you give it a file name with a .script extension

To run the script you must open up a URL through your server to the saved .script file. For example, http://your.domain.edu/hello-world.script. This is true for all the scripts described here. If you don't open a URL to your scripts through your server (as opposed to using your browser's Open option from the File menu), then the server will not pass the necessary information along to your scripts and/or they will not get executed.

The most important thing about this script is not the code itself. Rather, the most important thing is the code was saved as a text file and given a .script extension. Remember your machttp.conf file? With the MacHTTP distribution, the machttp.conf file contained the following MIME type definition:

```
SCRIPT  .SCRIPT TEXT * text/html
```

This definition tells MacHTTP that if a requested file ends in .script, then MacHTTP is suppose to:

1. Load the script as an ASCII text file into RAM
2. Prepend a number of predefined global values to the script
3. Send the modified script to AppleScript for compilation and execution
4. Return the output of the executed script back to the client application

Consequently, this is exactly happens with hello-world-01.script. To modify this script, try changing "Hello, World" to something like "Gosh, this is fun!" or even "the (current date) as string" and see what happens.

MAKING YOUR SCRIPTS HTTP-FRIENDLY

This section describes how to make your scripts conform to the HTTP and HTML standards.

There are two major problems with hello-world-01.script. First, it is not HTTP-friendly because it does not return the standard HTTP header telling the client application the origin of its input and type of document being returned. This is a responsibility of any script you write for your WWW server.

Second, the output of hello-world-01.script is an invalid HTML document; when writing scripts for your WWW server, you must format the results into valid HTML. (At the present time, many client applications will accept the raw output from hello-world-01.script, but future clients may not be so forgiving).

Consequently, the following script (hello-world-02.script), while it functions exactly the same as the previous script, is a nicer "network citizen" because it returns an HTTP header and valid

HTML:

```
-- define the standard HTTP header
set LF to ASCII character (10)
set CR to return
set CRLF to CR & LF
set http_10_header to "HTTP/1.0 200 OK" & CRLF & ¬
  "Server: MacHTTP" & CRLF & ¬
  "MIME-Version: 1.0" & CRLF & ¬
  "Content-type: text/html" & CRLF & CRLF

-- return the results as an HTML file
return http_10_header & ¬
  "<html>" & ¬
  "<head>" & ¬
  "<title>Hello, World</title>" & ¬
  "</head>" & ¬
  "<body>Hello, World!</body" & ¬
  "</html>"
```

Use the same method to create this script as the first script:

1. Launch your copy of Script Editor (or you could use any text editor)
2. Create a new file
3. Enter the single line above
4. Save it as a text file within your server's directory structure making sure you give it a file name with a .script extension

The first part of this script defines a standard HTTP header as defined by the HTTP protocol. To do so, it first creates variables for the carriage return/line feed sequence. It then defines a variable (http_10_header). This variable will describe to the client application the version of HTTP being implemented, the version of MIME being implemented, and the MIME type of information to follow. When writing CGI scripts for MacHTTP, your http_10_header will be defined like this. If you are writing scripts for WebSTAR, then you will want to replace the string "MacHTTP" with "WebSTAR".

The second part of the script returns text to the client. This text first includes the header described above and then the shortest of HTML documents marking up the "Hello, World" text.

Obviously, this script does not do very much. By adding a few lines of code and a bit of tweaking, you can make the script return the current date and time:

```
-- define the standard HTTP header
set LF to ASCII character (10)
set CR to return
set CRLF to CR & LF
set http_10_header to "HTTP/1.0 200 OK" & CRLF & ¬
  "Server: MacHTTP" & CRLF & ¬
  "MIME-Version: 1.0" & CRLF & ¬
```

```
  "Content-type: text/html" & CRLF & CRLF
-- get the date and time
set theDateAndTime to the (current date) as string

-- return the results as an HTML file
return http_10_header & ¬
  "<html>" & ¬
  "<head>" & ¬
  "<title>Hello, Reader</title>" & ¬
  "</head>" & ¬
  "<body>The date and time is " & theDateAndTime & "." & ¬
  "</body" & ¬
  "</html>"
```

To summarize, this scripts demonstrates how to create the standard HTTP header and how to send the header and marked up HTML document back to the client application. In performing these functions, your CGI scripts conform to standards.

Global Variables in .script Scripts

This section demonstrates how .script files are modified before execution to include a number of standard variables.

After MacHTTP loads a .script file into RAM and before it sends the script to AppleScript for compilation and execution, MacHTTP prepends to your script a number of predefined global variables. These variables are pieces of information sent from many (but not all) client applications and include:

path_args
: The data after the $ and before a ? in a URL

http_search_args
: All data after the ? of a URL

method
: Either GET or POST (.script files always return GET)

post_args
: (Always empty because the method is never POST)

client_address
: The IP address or name of the client's host computer

username
: The validated person accessing the script

password
: The validated password of the person accessing the script

from_user
: The name of the person (from the client's preferences) accessing the script

server_name
: The IP address or name of the computer running the server

server_port
: The port the server is running on (most likely 80)

script_name
: The name of the script to being executed

content_type
: The MIME type

referer
: The URL of the page containing the script

user_agent
: The name of the client application

Because MacHTTP prepends these variables and their values to every .script file sent to AppleScript, you can read and manipulate these values in that same .script file as shown below in hello-world-03.script:

```
-- define the standard HTTP header
set LF to ASCII character (10)
set CR to return
set CRLF to CR & LF
set http_10_header to "HTTP/1.0 200 OK" & CRLF & ¬
  "Server: MacHTTP" & CRLF & ¬
  "MIME-Version: 1.0" & CRLF & ¬
  "Content-type: text/html" & CRLF & CRLF

-- return the results, including the prepended variables
return http_10_header & ¬
  "<html>" & ¬
  "<head>" & ¬
  "<title>Hello, World</title>" & ¬
  "</head>" & ¬
  "<body>Hello, World!" & ¬
  "<p>http_search_args: " & http_search_args & ¬
  "<br>path_args: " & path_args & ¬
  "<br>post_args: " & post_args & ¬
  "<br>method: " & method & ¬
```

```
"<br>client_address: " & client_address & ¬
"<br>username:" & username & ¬
"<br>password: " & password & ¬
"<br>from_user:" & from_user & ¬
"<br>server_name:" & server_name & ¬
"<br>server_port:" & server_port & ¬
"<br>script_name: " & script_name & ¬
"<br>content_type: " & content_type & ¬
"<br>referer: " & referer & ¬
"<br>user_agent: " & user_agent & ¬
"</body>" & ¬
"</html>"
```

Consequently, based on these global variables, you can process your script differently. For example, you may want to test whether or not the client application understands "Netscapisms" by examining the contents of user_agent. If the user_agent can render "Netscapisms", then you might return one type of HTML mark up. If it doesn't, then you might return a different markup.

Adding Rudimentary Input to Your .script Files

This section describes how to add rudimentary input to your .script files by extending your URLs.

Knowing that path_args and http_search_args are variables, you can supply rudimentary input to your .script files by defining the values of these variables in a URL. To do so you need to keep in mind the following guidelines:

- path_args are defined by the text after the dollar sign ($) and before the question mark (?).
- http_search_args are defined by the text following the question mark (?).
- All path_arg and http_search_arg strings must be encoded according the specification layed out for URLs.

The following URLs all run the hello-world-03.script, but in these examples, the scripts are being called using input:

- hello-world-03.script?TryThisAtHome
- hello-world-03.script$ScriptingIsFun
- hello-world-03.script$communication?WWWServersAreAbout...

Notice how the value of script_name is consistent in the output for all three scripts.

Unfortunately, getting input in this manner is impractical; it would be difficult, if not impossible, to get your readers to enter such hard-to-read and understand URLs.

GETTING INPUT VIA ISINDEX

The use of ISINDEX tags in HTML documents provide an easier way to supply into to our scripts.

With the addition of an ISINDEX tag returned in the HTML code sent by your .script files, you can create a simple text field facilitating input. After the end-user enters text into the field created by the ISINDEX tag, the script is called again and the the input is assigned to the http_search_args variable. The script hello-world-04.script illustrates this point:

```
-- define the standard HTTP header
set LF to ASCII character (10)
set CR to return
set CRLF to CR & LF
set http_10_header to "HTTP/1.0 200 OK" & CRLF & ¬
  "Server: MacHTTP" & CRLF & ¬
  "MIME-Version: 1.0" & CRLF & ¬
  "Content-type: text/html" & CRLF & CRLF

if http_search_args = "" then

  -- http_search_args is empty; request input via ISINDEX
  return http_10_header & ¬
    "<html>" & ¬
    "<head>" & ¬
    "<title>Hello!</title>" & ¬
    "<isindex>" & ¬
    "</head>" & ¬
    "<body>Hello! What is your name?" & ¬
    "</body>" & ¬
    "</html>"

else

  -- http_search_args has value; return all variables
  return http_10_header & ¬
    "<html>" & ¬
    "<head>" & ¬
    "<title>Hello!</title>" & ¬
    "</head>" & ¬
    "<body>Hello!" & ¬
    "<p>http_search_args: " & http_search_args & ¬
    "<br>path_args: " & path_args & ¬
    "<br>post_args: " & post_args & ¬
    "<br>method: " & method & ¬
    "<br>client_address: " & client_address & ¬
```

```
    "<br>username:" & username & ¬
    "<br>password: " & password & ¬
    "<br>from_user:" & from_user & ¬
    "<br>server_name:" & server_name & ¬
    "<br>server_port:" & server_port & ¬
    "<br>script_name: " & script_name & ¬
    "<br>content_type: " & content_type & ¬
    "<br>referer: " & referer & ¬
    "<br>user_agent: " & user_agent & ¬
    "</body>" & ¬
    "</html>"

end if
```

This script works by first checking the value of http_search_args. If is it empty, then the script returns an HTML document containing the ISINDEX tag in the head of the HTML file. The existence of this tag automatically makes an input field active on the end-user's client application.

After the end-user enters some data into the resulting input field, the client application appends the input to the script's URL and sends the URL back to the server requesting processing. Notice how the URL changed to include your input.

Finally, the script executes again, but the second time http_search_args has a value and the second half of the if-then-else statement is processed.

This technique is echoed in the following script emulating that popular song, "The Name Game":

```
-- define the standard HTTP header
set LF to ASCII character (10)
set CR to return
set CRLF to CR & LF
set http_10_header to "HTTP/1.0 200 OK" & CRLF & ¬
  "Server: MacHTTP" & CRLF & ¬
  "MIME-Version: 1.0" & CRLF & ¬
  "Content-type: text/html" & CRLF & CRLF

if http_search_args = "" then

  -- http_search_args is empty; request input via ISINDEX
  return http_10_header & ¬
    "<html><head>" & ¬
    "<title>Play the Name Game</title><isindex></head>" & ¬
    "<body>" & ¬
    "<H1>Play the Name Game</H1>" & ¬
    "</body></html>"
```

```
else

  -- get our name
  set theName to word 1 of http_search_args as string

  -- initalize some variables for the next routine
  set foundVowel to 0
  set theCounter to 0
  set theSuffix to theName

  -- work through the name until we find the first vowel
  repeat until foundVowel is greater than 0

    -- increment our counter
    set theCounter to theCounter + 1

    -- get the next character
    set theCharacter to the first item of theSuffix

    -- check whether or not it is a vowel
    if "aeiouy" contains theCharacter or "AEIOUY" contains & ¬
      theCharacter then

      -- found it, exit
      set foundVowel to 1

    else

      -- left-hand truncate the name
      set theSuffix to items 2 through length of theSuffix as string
    end if

  end repeat

  -- compose the ryhmes
  set ryhmeOne to "B" & theSuffix
  set ryhmeTwo to "F" & theSuffix

  -- return the lyrics
  return http_10_header & ¬
    "<html><head>" & ¬
    "<title>The Name Game</title>" & ¬
    "<body>" & ¬
    "<h1>The Name Game</h1>" & ¬
    "<blockquote>" & theName & ", " & theName & ", Bo " & ryhmeOne & ¬
    ".<br>" & return & ¬
    "Bananna fana, Fo " & ryhmeTwo & ".<br>" & return & ¬
```

```
        "Fe Fi Fo, " & ryhmeTwo & ".<br>" & return & ¬
        "<strong>" & theName & "</strong>!" & ¬
        "</body>" & ¬
        "</html>"

    end if
```

Again, the example is rather trivial. This script first defines an HTTP header. It then examines the value of http_search_args. If it has a null value, then an ISINDEX query is returned to the client. If http_search_args is not empty, then the script searches for the first vowel in the returned string and thus creates the ryhmes. Finally, the script returns the lyrics.

The point is that you can get input from the use of an ISINDEX HTML tag placed in the head of your HTML documents. You can then examine the value of http_search_args to perform some function and return the results.

BUT DON'T USE .SCRIPT FILES

Only use the .script file technique for your simplest scripting needs.

The important points of the previous sections were were:

1. Scripts should return valid HTTP headers and HTML
2. There are a number of values passed to scripts by the WWW server
3. There a number of ways to get input for your scripts

These points are true for all scripting purposes where MacHTTP and WebSTAR are concerned.

After all that, it is not really recommended you use the .script technique for writing scripts, except for the simplest of scripts. The biggest reason for this is the server must load your entire script into RAM and prepend the client variables before anything happens. This is unnecessary overhead that can be avoided if you write your scripts as common gateway interface (CGI) scripts, described in the next sections. Also, this technique limits you to writing AppleScript scripts which may not be your forte or the best language for your particular application.

On the other hand, this technique demonstrates fundamental principles of scripting: the need for valid HTTP headers and HTML, as well as the existence of variables other than your input.

RUNNING SCRIPTS VIA APPLEEVENTS

This section introduces how to write true CGI scripts through the use of AppleEvents sent by

the server. This is the recommended method for executing scripts.

The use of AppleEvents to launch your scripts is the preferred method of running scripts. It eliminates the need for MacHTTP to load your scripts into RAM, prepend the global client variables, and most importantly, wait until the script is finished executing. Through the use of AppleEvents, much of this can be eliminated, a wider variety of scripting (programming) languages are available to you, and your scripts can run concurrently with the server application.

To make this happen using AppleScript as your scripting language you must:

1. Include a handler in your script that waits for and processes AppleEvents
2. Save your script as an application (not a text file) making sure the "Stay Open" and "Never Show Startup Screen" options are checked
3. Give your script a .cgi extension rather than a .script extension

The "Hello, World" script (hello-world-05.cgi) is revisited here to illustrate this concept:

```
-- process the AppleEvent sent to this script by the server
on «event WWWΩsdoc»

  -- define the standard HTTP header
  set LF to ASCII character (10)
  set CR to return
  set CRLF to CR & LF
  set http_10_header to "HTTP/1.0 200 OK" & CRLF & ¬
    "Server: MacHTTP" & CRLF & ¬
    "MIME-Version: 1.0" & CRLF & ¬
    "Content-type: text/html" & CRLF & CRLF

  -- return the results as an HTML file
  return http_10_header & ¬
    "<html>" & ¬
    "<head>" & ¬
    "<title>Hello, World</title>" & ¬
    "</head>" & ¬
    "<body>Hello, World!</body" & ¬
    "</html>"

end «event WWWΩsdoc»
```

This example, hello-world-05.cgi, illustrates one difference you can see and two differences you can not when compared to hello-world-02.script. The difference you can see is the addition of an AppleEvent handler. The AppleEvent handler is denoted by the addition of the on «event WWWΩsdoc» and end «event WWWΩsdoc» lines of code. This AppleEvent (WWW*sdoc) is sent by MacHTTP and WebSTAR and is used to initiate the script. It provides the means of passing the global client variables to the script as well.

Important! The AppleEvent is composed of two parts: "WWW" and "sdoc". These parts must

be delimited by the omega character (option-z). The omega character does not display in

HTML. Consequently, the asterisk (*) is used here instead. Every time you see the asterisk character in these scripts think the omega character.

The two differences you can not see entail how the script was saved. First, the script was saved as an application and not as a text file. Furthermore, the script was saved with the "Stay Open" and "Never Show Startup Screen" options checked.

Second, the script was given a name with a .cgi extension. This extension is defined in your server's configuration file and consequently it tells your server to treat any such file as a program to be executed.

GLOBAL VARIABLES AND ISINDEX REVISITED

The global client variables and the use of ISINDEX are still supported in scripts called via AppleEvents.

Just because scripts saved as CGI or ACGI files are more efficient than .script files, does not mean you have to unlearn everything about .script files. In fact, much of what you learned there can be applied here.

The first thing to understand is the continued existence of the client variables. The only difference is they are passed along to your script in a different manner; the client variables are passed to the script via AppleEvents rather than being prepended to your scripts as declared global variables. Furthermore, if you want to use any of the variables in your scripts, then you have to declare them yourself.

The following script (hello-world-06.cgi) demonstrates how to do this:

```
-- process the AppleEvent sent to this script by the server
on «event WWWΩsdoc» path_args given «class kfor»:http_search_args,
«class user»:username, «class pass»:password, «class frmu»:from_user,
«class addr»:client_address, «class post»:post_args, «class
meth»:method, «class svnm»:server_name, «class svpt»:server_port,
«class scnm»:script_name, «class ctyp»:content_type, «class
refr»:referer, «class Agnt»:user_agent

  -- define the standard HTTP header
  set LF to ASCII character (10)
  set CR to return
  set CRLF to CR & LF
  set http_10_header to "HTTP/1.0 200 OK" & CRLF & ¬
    "Server: MacHTTP" & CRLF & ¬
    "MIME-Version: 1.0" & CRLF & ¬
    "Content-type: text/html" & CRLF & CRLF
```

```
    -- return the results, including the prepended variables
    return http_10_header & ¬
      "<html>" & ¬
      "<head>" & ¬
      "<title>Hello, World</title>" & ¬
      "</head>" & ¬
      "<body>Hello, World!" & ¬
      "<p>http_search_args: " & http_search_args & ¬
      "<br>path_args: " & path_args & ¬
      "<br>post_args: " & post_args & ¬
      "<br>method: " & method & ¬
      "<br>client_address: " & client_address & ¬
      "<br>username:" & username & ¬
      "<br>password: " & password & ¬
      "<br>from_user:" & from_user & ¬
      "<br>server_name:" & server_name & ¬
      "<br>server_port:" & server_port & ¬
      "<br>script_name: " & script_name & ¬
      "<br>content_type: " & content_type & ¬
      "<br>referer: " & referer & ¬
      "<br>user_agent: " & user_agent & ¬
      "</body>" & ¬
      "</html>"

end «event WWWΩsdoc»
```

The most important line in this script is the one waiting for the WWW*sdoc AppleEvent. As described in the previous section, this line is activated when your server launches the script and sends the AppleEvent. This AppleEvent brings with it a number of parameters. These parameters represent the same variables prepended to .script files.

In order to use these parameters in your scripts, you must extract them from the AppleEvent. The path arguments of a URL (everything following the dollar sign and before the question mark) are included in the direct parameter of the AppleEvent. To use the path arguments in our script, simply put a variable (like path_args) after the on «event WWWΩsdoc» statement.

The other variables (the indirect parameters) must by extracted using a combination of the given command and AppleEvent keyword codes. The keyword codes for the remaining client variables are:

- kfor - search arguments
- user - user name
- pass - password
- frmu - from user
- addr - client address
- post - post arguments
- meth - HTTP method

- svnm - server name
- svpt - server port
- scnm - script name
- ctyp - content type
- refr - referer
- Agnt - user agent

Thus, if you wanted to use the client variable frmu (from user) in your script, then you would have to assign it to a variable in the on «event WWWΩsdoc» line something like this:

```
on «event WWWΩsdoc» given «class frmu»:from_user
```

Consequently, the variable from_user would containing the name of the person at the other end of the client program. Similarly, if you wanted to extract the referer information (the URL of the page calling your script), then you would put something like this on the on «event WWWΩsdoc» line:

```
on «event WWWΩsdoc» given «class refr»:referer
```

ISINDEX tags still work in CGI/ACGI scripts too. As the new, CGI version of the Name Game script (name-game.cgi) demonstrates:

```
-- process the AppleEvent sent to this script by the server
on «event WWWΩsdoc» given «class kfor»:http_search_args

  -- define the standard HTTP header
  set LF to ASCII character (10)
  set CR to return
  set CRLF to CR & LF
  set http_10_header to "HTTP/1.0 200 OK" & CRLF & ¬
    "Server: MacHTTP" & CRLF & ¬
    "MIME-Version: 1.0" & CRLF & ¬
    "Content-type: text/html" & CRLF & CRLF

  if http_search_args = "" then

    -- http_search_args is empty; request input via ISINDEX
    return http_10_header & ¬
      "<html><head>" & ¬
      "<title>Play the Name Game</title><isindex></head>" & ¬
      "<body>" & ¬
      "<H1>Play the Name Game</H1>" & ¬
      "</body></html>"

  else

    -- get our name
    set theName to word 1 of http_search_args as string
```

```
  -- initalize some variables for the next routine
  set foundVowel to 0
  set theCounter to 0
  set theSuffix to theName

  -- work through the name until we find the first vowel
  repeat until foundVowel is greater than 0

    -- increment our counter
    set theCounter to theCounter + 1

    -- get the next character
    set theCharacter to the first item of theSuffix

    -- check whether or not it is a vowel
    if "aeiouy" contains theCharacter or "AEIOUY" contains & ¬
      theCharacter then

      -- found it, exit
      set foundVowel to 1

    else

      -- left-hand truncate the name
      set theSuffix to items 2 through length of theSuffix as string
    end if

  end repeat

  -- compose the ryhmes
  set ryhmeOne to "B" & theSuffix
  set ryhmeTwo to "F" & theSuffix

  -- return the lyrics
  return http_10_header & ¬
    "<html><head>" & ¬
    "<title>The Name Game</title>" & ¬
    "<body>" & ¬
    "<h1>The Name Game</h1>" & ¬
    "<blockquote>" & theName & ", " & theName & ", Bo " & ryhmeOne & ¬
   ".<br>" & return & ¬
    "Bananna fana, Fo " & ryhmeTwo & ".<br>" & return & ¬
    "Fe Fi Fo, " & ryhmeTwo & ".<br>" & return & ¬
    "<strong>" & theName & "</strong>!" & ¬
    "</body>" & ¬
    "</html>"

end if
```

```
end «event WWWΩsdoc»
```

This new version of the Name Game begins by waiting for the AppleEvent from the server. When the AppleEvent is received, the search arguments (everything after the question mark of a URL) are assigned to the variable http_search_args. This is because all WWW clients append a question mark and the value of the ISINDEX field after the URL. The script then continues as it did in its .script version by evaluating the contents of http_search_args and branching accordingly.

The point of this section is two-fold. First, client variables discussed in the .scripts sections of this book are still available to you in CGI/ACGI applications. Using AppleScript, you must extract these variables from the AppleEvents manually instead of relying on the server to prepend them to our scripts.

Second, the use of ISINDEX tags to create simple text fields continues to be a valid method for getting input.

TRAPPING AND HANDLING ERRORS

This section demonstrates how to trap errors in your scripts.

Error trapping and handling is an imperative part of your CGI/ACGI scripting skills. Nothing is more frustrating to your readers than the meaningless message "Error executing script." Not only will they go away frustrated, but it will detract from your entire service. Furthermore, until AppleScripting becomes second nature to you, the lack of error trapping and handling will stunt your development skills. For these reasons, including error handling routines in your scripts is essential.

The way to trap and handle error messages in AppleScript CGI/ACGI programs is through the use of the try/on error/end try commands. What you want to do is wrap every critical routine in your script with try/on error/end try handler. Then, when an error occurs, an error message can be passed to the on error part of your handler and this handler returns valid HTML describing the error.

To demonstrate this process, a new version of the Name Game is used. As you may or may not have found out, in the previous versions of the Name Game script, if you entered a word that did not have any vowels (or a "y"), then the script returned the nasty error "Error executing script." In this new version of the Name Game (name-game-02.cgi), an error handler is included in the code. Try it. Run the Name Game again and enter a word containing no vowels and see what happens.

```
-- process the AppleEvent sent to this script by the server
on «event WWWΩsdoc» given «class kfor»:http_search_args

  -- define the standard HTTP header
  set LF to ASCII character (10)
```

```
set CR to return
set CRLF to CR & LF
set http_10_header to "HTTP/1.0 200 OK" & CRLF & ¬
  "Server: MacHTTP" & CRLF & ¬
  "MIME-Version: 1.0" & CRLF & ¬
  "Content-type: text/html" & CRLF & CRLF

-- practice good error trapping
try

  if http_search_args = "" then

    -- http_search_args is empty; request input via ISINDEX
    return http_10_header & ¬
      "<html><head>" & ¬
      "<title>Play the Name Game</title><isindex></head>" & ¬
      "<body>" & ¬
      "<H1>Play the Name Game</H1>" & ¬
      "</body></html>"

  else

    -- get our name
    set theName to word 1 of http_search_args as string

    -- initalize some variables for the next routine
    set foundVowel to 0
    set theCounter to 0
    set theSuffix to theName

    -- work through the name until we find the first vowel
    repeat until foundVowel is greater than 0

      -- increment our counter
      set theCounter to theCounter + 1

      -- get the next character
      set theCharacter to the first item of theSuffix

      -- check whether or not it is a vowel
      if "aeiouy" contains theCharacter or "AEIOUY" contains
        theCharacter then

        -- found it, exit
        set foundVowel to 1

      else
```

```
        -- left-hand truncate the name
        set theSuffix to items 2 through length of theSuffix as string
      end if

    end repeat

    -- compose the ryhmes
    set ryhmeOne to "B" & theSuffix
    set ryhmeTwo to "F" & theSuffix

    -- return the lyrics
    return http_10_header & ¬
      "<html><head>" & ¬
      "<title>The Name Game</title>" & ¬
      "<body>" & ¬
      "<h1>The Name Game</h1>" & ¬
      "<blockquote>" & theName & ", " & theName & ", Bo " & ryhmeOne
      & ".<br>" & return & ¬
      "Bananna fana, Fo " & ryhmeTwo & ".<br>" & return & ¬
      "Fe Fi Fo, " & ryhmeTwo & ".<br>" & return & ¬
      "<strong>" & theName & "</strong>!" & ¬
      "</body>" & ¬
      "</html>"

  end if

 on error msg

  -- return the error as an HTML document
  return http_10_header & ¬
    "</html><head><title>Error</title></head>" & ¬
    "<body><h1>Error</h1>" & ¬
    "An error occured: " & msg & ¬
    "</body></html>"

 end try

end «event WWWΩsdoc»
```

The only additions to this script are the try/on error/end try sections. The real meat of the demonstration is in the on error routine. This routine is called when any sort of error occurs. If you don't supply a word containing a vowel, then the script is never able to left-hand truncate the name. This generates an error and is trapped by on error. On error receives one parameter, msg. This parameter is then incorporated into a simple HTML document and returned to the server.

There are two drawbacks to the script as it stands. First, if there is some sort of error in the on

error routine, then your script will still return the nasty "Error executing script" message. All you can do here is just make sure there are no errors.
Second, you may want to wrap the try/on error/end try commands around many parts of your script. At the same time you don't want to write the same on error handler over and over again. The solution to this problem is to create a separate handler that only returns the marked up error message, and this handler can be called by the each on error handler. This is a three-step process.

Declare your HTTP header as a global variable outside the WWW*sdoc handler. You will also want to define the header outside the WWW*sdoc handler. For example, at the beginning of your script, before the WWW*sdoc handler, insert the following code:

```
-- declare the HTTP header as a global variable
global http_10_header

-- define the HTTP header as usual
set LF to ASCII character (10)
set CR to return
set CRLF to CR & LF
set http_10_header to "HTTP/1.0 200 OK" & CRLF & ¬
  "Server: MacHTTP" & CRLF & ¬
  "MIME-Version: 1.0" & CRLF & ¬
  "Content-type: text/html" & CRLF & CRLF
```

Next, within your WWW*sdoc handler, wrap try/on error/end try statements around your code. Only this time, call a new handler within your on error statement. The purpose of this handler is to create the HTML error statement:

```
-- practice good error handling
try
  -- your scripting routines go here
on error msg
  return ErrorMessage(msg)
end try
```

Finally, write the ErrorMessage handler. It could look something like this:

```
on ErrorMessage(ErrorMessage)

   -- return the error as an HTML document
   return http_10_header & ¬
     "</html><head><title>Error</title></head>" & ¬
     "<body><h1>Error</h1>" & ¬
     "An error occured: " & ErrorMessage & ¬
     "</body></html>"

end ErrorMessage
```

Overriding the "Stay Open" option

In this section a method of overriding the "Stay Open" option is presented.

All of the CGI/ACGI scripts described so far, once launched, will never quit. This is not bad since each one of these scripts takes time to load, and once loaded they will execute faster. If the scripts you write are intended to be used often, then keeping them open is a good idea, provided you have the RAM to spare. In those cases where you don't have enough RAM to spare and/or the CGI/ACGI scripts you write are intended to be used infrequently, then you may want to consider including in your scripts a method for quitting.

The following script, hello-world-07.cgi works just most of the other "Hello, World!" scripts here. It simply displays the text "Hello, World!", but unlike the other scripts, this one will quit after 30 ticks (one tick = 1/60th of a second) of idle time. Not incidentally, this technique was first described by Jon Wiederspan in his essay "Extending MacHTTP"

```
-- set up automatic quitting
global theIdletime
global theDateStamp
set theIdletime to 30
set theDateStamp to current date

-- process the AppleEvent sent to this script by the server
on «event WWWΩsdoc»

  -- practice good error trapping
  try

    -- re-initialize the date stamp for .acgi purposes
    set theDateStamp to current date

    -- define the standard HTTP header
    set LF to ASCII character (10)
    set CR to return
    set CRLF to CR & LF
    set http_10_header to "HTTP/1.0 200 OK" & CRLF & ¬
      "Server: MacHTTP" & CRLF & ¬
      "MIME-Version: 1.0" & CRLF & ¬
      "Content-type: text/html" & CRLF & CRLF

    -- return the results as an HTML file
    return http_10_header & ¬
      "<html>" & ¬
      "<head>" & ¬
      "<title>Hello, World</title>" & ¬
      "</head>" & ¬
```

```
      "<body>Hello, World!</body" & ¬
      "</html>"
  on error msg

    -- return the error as an HTML document
    return http_10_header & ¬
      "</html><head><title>Error</title></head>" & ¬
      "<body><h1>Error</h1>" & ¬
      "An error occured: " & msg & ¬
      "</body></html>"

  end try

end «event WWWΩsdoc»

on idle

  -- check to see if it is time to quit
  if (current date) > (theDateStamp + theIdletime) then quit

  -- wait a few more ticks
  return 5

end idle

on quit

  -- exit the program
  continue quit
end quit
```

The first part of this script declares two global variables (theDateStamp and theIdletime). These variables have to be global in nature because they will be used in other handlers of the script.

Next, the script assigns values to these variables. The variable, theDateStamp is self-evident. The second variable, theIdleTime defines how long the script should stay open and wait for more requests for processing. The value of theIdleTime must be long enough for your script to execute, but not so long that it hangs around in RAM too long.

Third, within the WWW*sdoc handler, theDateStamp is reinitialized since the program may have been called after it was initially launched. By reinitializing theDateStamp you can accommodate ACGI scripts.

Fourth, the program does its processing and returns the result back to the server.

Now, the script receives idle calls automatically. This is a feature of AppleScript and works just like the on idle calls in HyperCard's HyperTalk. It is in the on idle handler that theIdletime and the date stamp are compared to the current time. If the current time is greater than the sum of the date stamp and theIdletime, then the quit routine is called. Otherwise, a few more ticks

are spent waiting.

USING FORMS TO GET INPUT

This section describes HTML+ FORMs as specialized HTML documents.

A number of script input methods have been enumerated in the previous sections. Namely these methods have made use of an ISINDEX tag, customized URLs using question marks (?) and dollar signs ($), and/or client variables passed from the server to the scripts. Programmers found these methods of input limiting and devised FORMs. FORMs provide programmers with a wider range of input methods as well as the ability to include multiple data entry vehicles in one or more HTML documents.

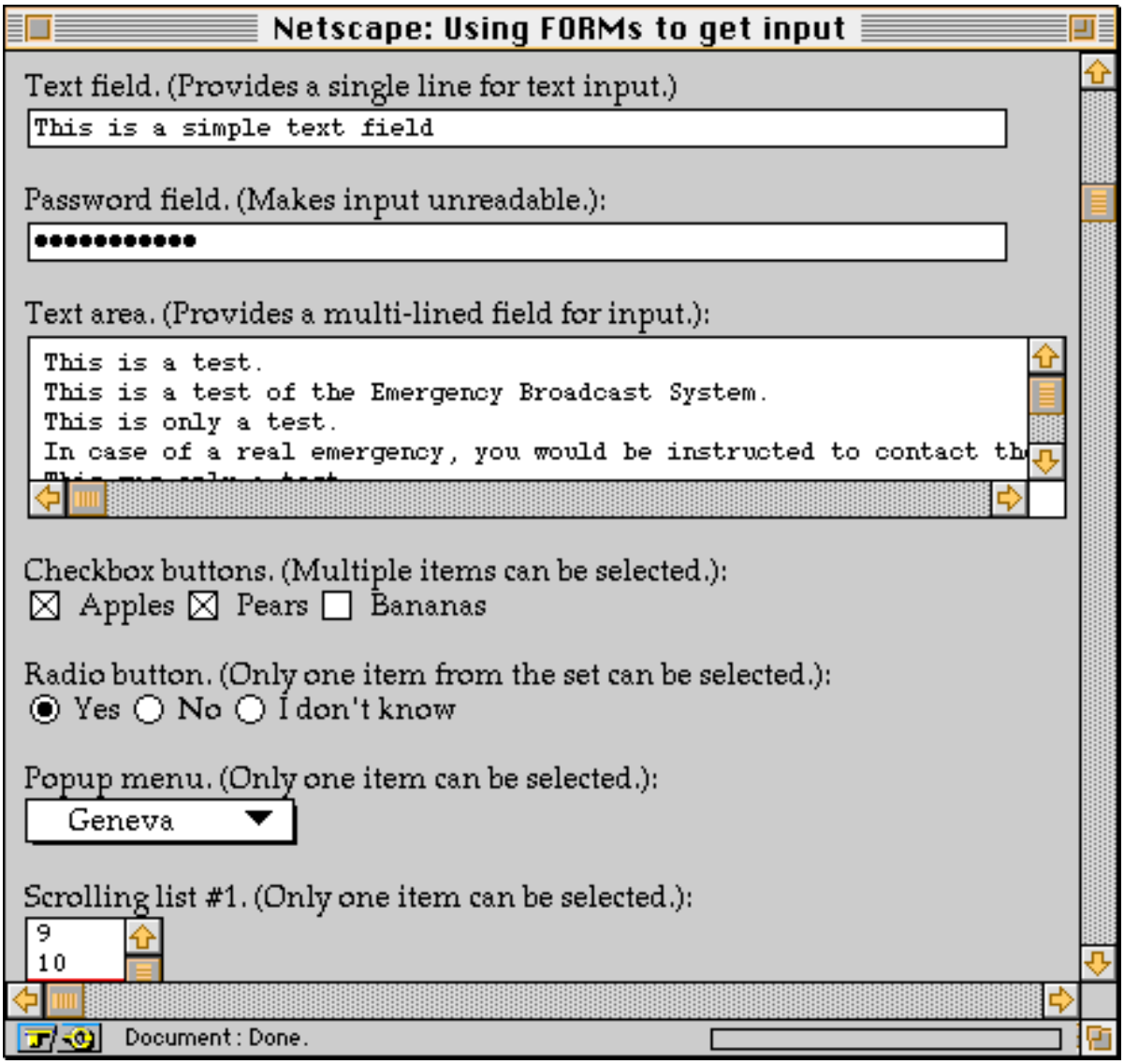
These data input (or data entry) elements include:

- Simple text fields similar to the functionality of ISINDEX tags
- Password text fields whose input echoes ambiguous characters for security reasons
- Hidden elements eliminating the possibility of data entry
- Scrolling text fields
- Radio buttons
- Check boxes
- Pop-up menus
- Scrolling lists where one item can be selected
- Scrolling lists where more than one item can be selected
- Submit buttons whose function is similar to "OK" buttons of Macintosh dialog boxes
- Reset buttons which return the data entry elements back to their default values

The data entry elements manifest themselves as extensions to HTML. (Incidentally, with the addition of FORMs to the HTML specification, the HTML specification became loosely known as HTML+.) Using any one or a combination of these elements in your HTML documents, you can get a much wider variety of input for your scripts.

By convention, HTML+ FORMs are preceded and followed by horizontal rulers (<hr>) to set the FORMs off from the rest of the HTML document.

Below is a sample HTML+ FORM:



In order to generate this sort of data entry document, you will have to enhance your HTML with various FORMs elements. Each one of these elements is described below with examples.

<FORM></FORM>

The <FORM> and </FORM> tags surround the data entry elements described below. They are used to initiate the data entry section of your HTML document. The <FORM> tag requires two attributes: ACTION and METHOD.

ACTION

This attribute declares the URL (either relative or complete) of the script to be executed after the submit button is selected.

METHOD

At the present time, this attribute can be either GET or PUT, and it describes the way the data from your FORM will be encoded before being sent to the server for processing. The GET method was the original method of sending data to WWW servers. Its main limitation is the number of bytes of information it can handle. The limit is 128K. One the other hand, when the GET method is used, the client application will echo the entire URL, complete with search arguments. This is handy if you want to replicate a complicated query sometime in the future. All you have to do is copy the URL from your browser's window and save it for future reference. The PUT method is much more commonly used. It can handle up to 1024K bytes of information in one query, and results in much cleaner but incomplete, URLs. Last, if you use the GET method, the when your FORM is submitted, then the FORMs contents will be encoded and passed to your script as search arguments. If you use the POST method, then the contents of your FORM will be encoded and passed on to your script as path arguments.

Examples:

<FORM ACTION="../scripts/hello-world-06.cgi" METHOD=GET> (More FORMs elements go here.)
</FORM>

<FORM ACTION="../scripts/hello-world-06.cgi" METHOD=POST> (More FORMs elements go here.)
</FORM>

<FORM ACTION="email.cgi" METHOD=POST> (More FORMs elements go here.)
</FORM>

<INPUT>

The INPUT tag is used to declare the existence of any of the following attributes in your FORM.

TYPE
This attribute can be any one of the following values.

TEXT
Creates a simple, single-line text input field.

PASSWORD
Just like TEXT but all letters are unreadable. This is good for passwords and sensitive information.

CHECKBOX
Produces a square button.

RADIO
Produces a small, circular button. All radio buttons defined with the same NAME

attribute will work as a group and allow you to only have one selected at a time. A common mistake when using RADIO is the lack of the CHECKED default button.

HIDDEN
This attribute allows you to define variables without letting the end-user change them. This will only work if the end-user does not download your FORM, modify it, and then resubmit it to your script.

SUBMIT
This works like the standard "OK" button in Macintosh-based dialogs. By selecting the SUBMIT button, the contents of the FORM are sent to the script defines by the ACTION attribute of the FORM tag.

RESET
This attribute is used to return the FORM to its original state after the end-user has made some changes.

NAME
This declares, in the programming sense of the word, the existence of a variable. The value created by the FORM will be put into the variable given by the NAME attribute.

VALUE
When used in conjunction with types TEXT and PASSWORD, VALUE is the default value the NAMEed variable will have. VALUE is not required for TEXT and PASSWORD types. VALUE is a required attribute for types CHECKBOX, RADIO, SUBMIT, RESET, and HIDDEN.

SIZE
This is a required attribute when used in conjunction with the TEXT and PASSWORD attributes. It defines the number of characters wide the resulting field will be. The SIZE attribute is meaningless for anything but TEXT and PASSWORD.

MAX
Again, this is a required attribute for the TEXT and PASSWORD attributes. When used in conjunction with the TEXT and PASSWORD attributes, it defines the maximum number of characters that can be entered into the defined field. It makes little sense for the MAX value to be smaller than the SIZE value. The value of MAX can be greater than the value of SIZE.

CHECKED
This is an optional attribute for the RADIO and CHECKBOX input types. If defined, then the resulting button (radio or checkbox) will be selected. By their nature, at least one radio button, in any given set, should be CHECKED. Otherwise, the set of buttons should be checkboxes instead.

Examples:

<INPUT TYPE="text" NAME="textField" VALUE="This is a simple text field" SIZE=65 MAX=65>

<INPUT TYPE="password" NAME="passwordField" VALUE="watermellon" SIZE=65 MAX=65>

<INPUT TYPE="hidden" NAME="hiddenElement" VALUE="Here is the hidden value">

<INPUT TYPE="checkbox" NAME="checkboxButtons" VALUE="bananas">

<INPUT TYPE="checkbox" NAME="checkboxButtons" VALUE="apples" CHECKED>

<INPUT TYPE="radio" NAME="radionButton" VALUE="no">

<INPUT TYPE="radio" NAME="radionButton" VALUE="yes" CHECKED>

<INPUT TYPE="submit" VALUE="OK">

<INPUT TYPE="reset" VALUE="Reset">

<TEXTAREA></TEXTAREA>

This tag is used to create a multi-lined scrolling field. A common misconception is that this field will automatically wrap text enter into it. This is not true and there is nothing you, as a scripter can do about it. To add default text to the scrolling fields defined by TEXTAREA, you must insert your test between the beginning and ending tags. Each one of the following attributes are required in the TEXTAREA tag set.

NAME
This attribute is used to declare the variable that will contain the contents of the TEXTAREA data once it is sent off for execution.

ROWS
This is the number of vertical rows the TEXTAREA field will occupy.

COLS
This is the number of characters wide the field will occupy.

Examples:

<TEXTAREA NAME="textAreaField" ROWS=4 COLS=65>This is a test.</TEXTAREA>

<TEXTAREA NAME="comment" ROWS=2 COLS=40></TEXTAREA>

<SELECT></SELECT>

SELECT is used to create scrolling lists and pop-up menus. Within scrolling lists one or more items can be "selected". Pop-up menus limit the selection to only one item.

NAME

This is the name of the variable that will be used to contain the value of the selected item(s).

SIZE
This is the number of visible selections. If SIZE equals one, then the resulting input element will be a pop-up menu. A value of 1 is the default SIZE when SIZE is not specifically defined. If SIZE is greater than 1, then the resulting input element will be that many rows deep.

MULTIPLE
If included in the SELECT tag, then more than one option can be available for selecting from the list of options. This is usually accomplished on the client end by shift- or command-clicking the items in the resulting field.

OPTION
This is a required parameter. The OPTION tag is used to denote the items appearing in the pop-up menu or scrolling list. To use the OPTION tag you simply proceed the names of each of your items with an <OPTION> and make sure the complete set of options is surrounded by the <SELECT> and </SELECT> tags.

SELECTED
Used within the OPTION parameter, this attribute defines the associated OPTION as a default selection.

Examples:

```
<SELECT NAME="popUpMenu">
<OPTION SELECTED>Geneva
<OPTION>Monaco
<OPTION>New York
<OPTION>Palatino
<OPTION>Times
</SELECT>

<SELECT NAME="scrollingListOne" SIZE=4>
<OPTION> 9
<OPTION> 10
<OPTION SELECTED> 12
<OPTION> 14
<OPTION> 18
<OPTION> 24
<OPTION> Other
</SELECT>

<SELECT NAME="scrollingListTwo" SIZE=4 MULTIPLE>
<OPTION SELECTED> Plain
<OPTION> Bold
<OPTION> Italic
<OPTION> Underline
```

<OPTION> Superscript
<OPTION> Subscript
</SELECT>

Now with the FORM elements defined for you, it is time to see a complete example. Here, in raw HTML, is the FORM used to create the example at the beginning of this section

```
<HR>
<FORM ACTION="../scripts/hello-world-06.cgi" METHOD=GET>

Text field. (Provides a single line for text input.)<br>
<INPUT TYPE="text" NAME="textField" VALUE="This is a simple text field"
SIZE=65 MAX=65><p>

Password field. (Makes input unreadable.):<br>
<INPUT TYPE="password" NAME="passwordField" VALUE="watermellon" SIZE=65
MAX=65><p>

Text area. (Provides a multi-lined field for input.):<br>
<TEXTAREA NAME="textAreaField" ROWS=4 COLS=65>This is a test.
This is a test of the Emergency Broadcast System.
This is only a test.
In case of a real emergency, you would be instructed to contact the
nearest WebMaster.
This was only a test.
</TEXTAREA><p>

Checkbox buttons. (Multiple items can be selected.):<br>
<INPUT TYPE="checkbox" NAME="checkboxButtons" VALUE="apples" CHECKED>
Apples
<INPUT TYPE="checkbox" NAME="checkboxButtons" VALUE="pears" CHECKED>
Pears
<INPUT TYPE="checkbox" NAME="checkboxButtons" VALUE="bananas">
Bananas<P>

Radio button. (Only one item from the set can be selected.):<br>
<INPUT TYPE="radio" NAME="radionButton" VALUE="yes" CHECKED> Yes
<INPUT TYPE="radio" NAME="radionButton" VALUE="no"> No
<INPUT TYPE="radio" NAME="radionButton" VALUE="I don't know"> I don't
know<p>

Popup menu. (Only one item can be selected.):<br>
<SELECT NAME="popUpMenu">
<OPTION SELECTED>Geneva
<OPTION>Monaco
<OPTION>New York
<OPTION>Palatino
<OPTION>Times
</SELECT><p>
```

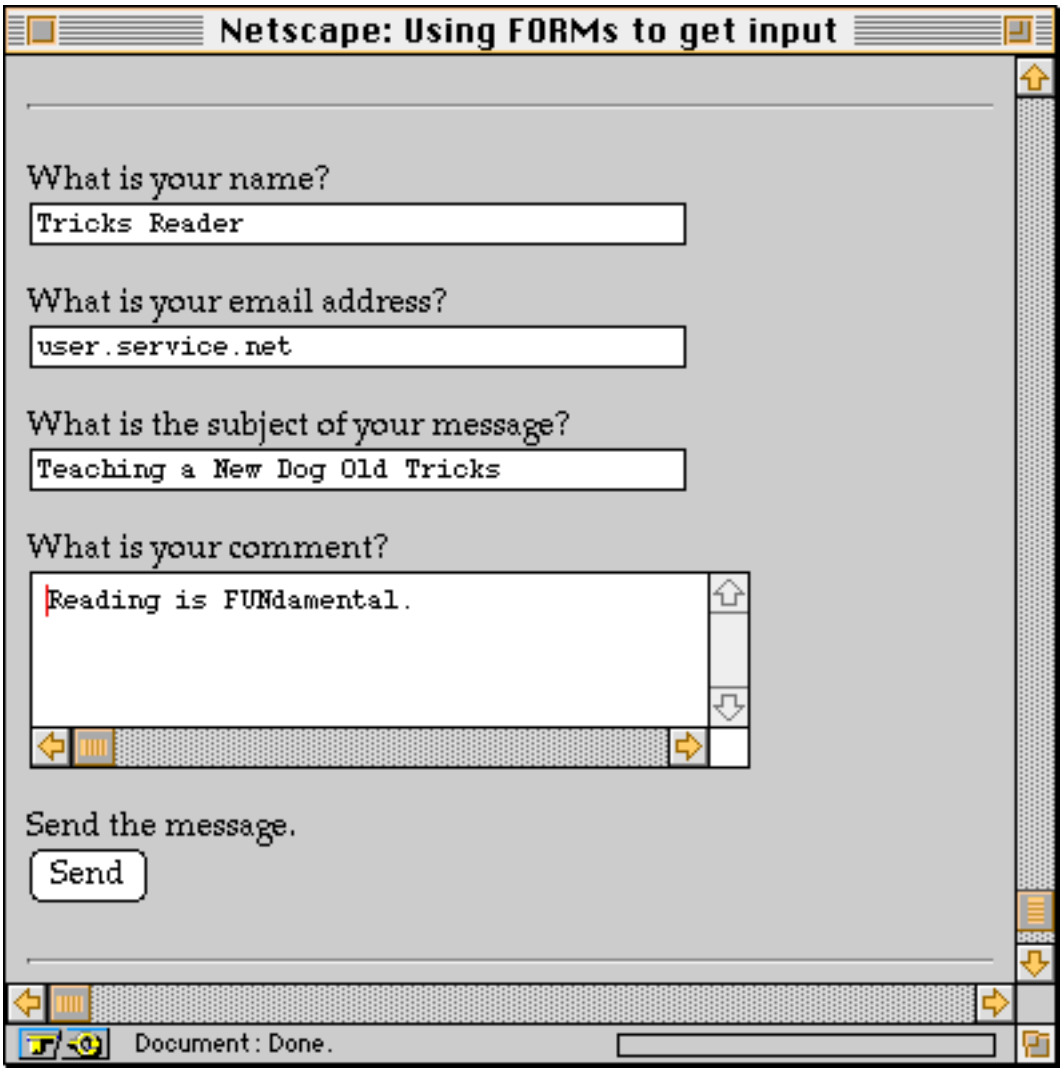
```
Scrolling list #1. (Only one item can be selected.):<br>
<SELECT NAME="scrollingListOne" SIZE=4>
<OPTION> 9
<OPTION> 10
<OPTION SELECTED> 12
<OPTION> 14
<OPTION> 18
<OPTION> 24
<OPTION> Other
</SELECT><p>

Scrolling list #2. (Multiple items can be selected usually using shift-
or command-click combinations.):<br>
<SELECT NAME="scrollingListTwo" SIZE=4 MULTIPLE>
<OPTION SELECTED> Plain
<OPTION> Bold
<OPTION> Italic
<OPTION> Underline
<OPTION> Superscript
<OPTION> Subscript
</SELECT><p>

Hidden attribute. (Not displayed, but contains a value.):<br>
<INPUT TYPE="hidden" NAME="hiddenElement" VALUE="Here is the hidden
value"><p>

Submit button. (This is like the OK button of dialog boxes.):<br>
<INPUT TYPE="submit" VALUE="OK"><p>

Reset button. (Used to return the FORM to its default state.):<br>
<INPUT TYPE="reset" VALUE="Reset"><p>

</FORM>

<HR>
```

There are a number of things needing to be stated at this point. First, while the FORMs tags are complicated, they should be available to any HTML editor worth its weight in salt; creating these sorts of tags is not any more complicated than other HTML as long as your editor adequately supports them.

Second, the distinction between the GET and POST methods is subtle. The important thing to remember is that the GET method is limited to 128K bytes of information and ultimately manifests itself as the search arguments in the AppleEvent sent to your script. On the other hand, the POST method can accommodate 1024K of information and manifests itself as the path arguments in the AppleEvent.

To demonstrate the second point, return to the FORM at the beginning of this section. It uses the GET method to send the data to the script. The FORM below uses the same ACTION

attribute as the one above (ACTION="../scripts/hello-world-06.cgi"), but since the METHOD is POST, the resulting output of hello-world-06.cgi is different. Notice how the the values of post_args, method, and content_type change.



DECODING AND READING THE INPUT IN YOUR SCRIPTS

This section explains how to decode and read the contents of variables once they have been sent to your scripts.

As you may have noticed from the previous sections, the actual data sent by the client application through the server and received by your scripts look pretty cryptic. If you used the .script technique or the GET method in your FORMS, then every space in your original input

got translated into a plus sign (+). You also may have noticed the names of your variables, a few equals signs (=), and a few ampersands (&) as well. This translation process is a part of the Uniform Resource Locator specification and is all a consequence of the GET method. For example, a piece of the search arguments may look like this:

```
textField=This+is+a+simple+text+field&passwordField=watermellon
```

If you used the POST method in your FORMs, then more translations occur. Specifically, all the reserved characters for URLs get translated into their hexadecimal equivalents. An example of the POST arguments could include:

```
addition=2+%2B+3+%3D+5
```

This translates to:

```
addition=2 + 3 = 5
```

These translation processes make the content of your search arguments and/or POST arguments difficult to humanly read. Furthermore, in order to do any processing on the contents of these variables you need to parse out the necessary information. This section describes how to accomplish this goal.

The hard way to translate (decode) these search and POST arguments is through the judicious use of repeat loops and various incarnations of the AppleScript's text item delimiters command. Frankly, this is just too difficult.

The next easiest way to decode and parse the search and/or POST arguments is through the combined use of two OSAX: Decode URL and Tokenize. Both of these OSAX are free and described in the next chapter, "OSAX for WWW scripting", but they have been superseded by the following OSAX.

Parse CGI is the OSAX of choice when it comes to decoding and parsing your search and/or POST arguments. This shareware tool is indispensable. Its cost is well worth the time and effort is saves you in, to turn a corporate phrase, "opportunity costs."

To demonstrate how to use Parse CGI, the following FORM is used to send input to a new script Dear Santa.

What would you like to say to Santa Claus?
Dear Santa,

I have been very good this year.
Please send me an Internet connection for Christmas this year.

Sign your name.

Send or reset the message.

Here is the FORM in raw HTML. Note the METHOD is POST and therefore the contents of the FORM are encoded and will be passed to the script as POST arguments. Note also the values of the NAME fields ("comment" and "signature"):

```
<HR>

<FORM ACTION="../scripts/dear-santa.cgi" METHOD=POST>

What would you like to say to Santa Claus?<br>
<TEXTAREA NAME="comment" ROWS=4 COLS=65>Dear Santa,

I have been very good this year.
Please send me an Internet connection for Christmas this year.
</TEXTAREA><p>

Sign your name.<br>
<INPUT TYPE="text" NAME="signature" VALUE="Cybernaut" SIZE=40
MAX=40><p>

Send or reset the message.<br>
<INPUT TYPE="submit" VALUE="Send"> <INPUT TYPE="reset" VALUE="Reset">

</FORM>

<HR>
```

Here is the script used to analyze the example:

```
-- process the AppleEvent sent to this script by the server
on «event WWWΩsdoc» given «class post»:post_args

  -- practice good error trapping
  try

    -- define the HTTP header as usual
    set LF to ASCII character (10)
    set CR to return
    set CRLF to CR & LF
    set http_10_header to "HTTP/1.0 200 OK" & CRLF & ¬
      "Server: MacHTTP" & CRLF & ¬
      "MIME-Version: 1.0" & CRLF & ¬
      "Content-type: text/html" & CRLF & CRLF

    -- decode the data from the POST arguments
    set theFORMData to parse CGI arguments post_args

    -- extract each variable from the FORM data
    set theComment to CGI field "comment" from theFORMData
    set theSignature to CGI field "signature" from theFORMData
```

```
    -- return the results as an HTML file
    return http_10_header & ¬
      "<html>" & ¬
      "<head>" & ¬
      "<title>Reply</title>" & ¬
      "</head>" & ¬
      "<body>" & ¬
      "<b>post_args</b>: " & post_args & "<p>" & ¬
      "<b>comment</b>: " & theComment & "<p>" & ¬
      "<b>signataure</b>: " & theSignature & ¬
      "</body" & ¬
      "</html>"

  on error msg

    -- return the error as an HTML document
    return http_10_header & ¬
      "</html><head><title>Error</title></head>" & ¬
      "<body><h1>Error</h1>" & ¬
      "An error occured: " & msg & ¬
      "</body></html>"

  end try

end «event WWWΩsdoc»
```

(Whew!) There are three important points to be made about this script.

First, since the FORM used the POST method, the server-defined AppleEvent keyword post had to be used to initialize the post_args variable. This was done on the second line of the script.

Next, the POSTed data needed to be decoded and parsed into a set of variable/data records. This was done with the line:

```
set theFORMData to parse CGI arguments post_args
```

The result is an AppleScript list put into the variable theFORMData.

Third, using the NAME attributes of the FORM, each one of the INPUT elements were extracted from the AppleScript list, in this case, theFORMData. This procedure was done with the following AppleScript:

```
set theComment to CGI field "comment" from theFORMData
set theSignature to CGI field "signature" from theFORMData
```

To bring the script home, the raw POST arguments, as well as the parsed variables were formatted into an HTML document and returned.

While this script does not perform any useful function, it demonstrates the best method to-date on how to extract the POST arguments sent to your server's scripts. At the same time, there are a few more options possible from the Parse CGI OSAX. These other options account for SELECT MULTIPLE INPUT elements. They also account for possibility that a particular element may have been given a null value. These options are left up you to explore and are described in the OSAX's dictionary.

Finally, these procedures would be exactly the same if you were to use the GET method in your scripts or the ISINDEX technique of getting input, except you would perform the parse

CGI arguments command on the search arguments instead of the POST arguments.

OSAX FOR WWW SCRIPTING

This chapter reviews a number of OSAX particularly useful for Macintosh-based WWW server scripts.

As pointed out in the previous chapter, adding scripts to your server's suite of functions represents the real power of the World Wide Web.

Listed here are a number of AppleScript OSAX and OSAX collections of particular use to WWW server scripting.

Even though the documentation in these various scripts says you do not have to reboot your computer after you install them, it is a good idea to do so anyway since that seems to solve more than a few installation problems.

ACME SCRIPT WIDGETS

Version: 1.1
Author: Wayne Walrath
Requirements: AppleScript
Cost: free
Pros: provides useful string manipulation tools
Cons: none
Remote location: <URL:>
Tricks location: <URL:http://152.1.24.177/teaching/archives/acme-script-widgets.hqx>


This set of OSAX provides a number of string manipulation tools useful in server

scripting. Given a string and a delimiter, Tokenize parses strings of characters into lists of strings. This is (was) particularly useful in the extraction of variables from search and POST arguments of server scripts. Much of this script's functionality, as far as server scripts are concerned, has been superseded by the Parse CGI OSAX. Replace OSAX, on the other hand has not been replaced by another package. This tool too is very useful for string manipulations as well.

DECODE URL

Version: 1.0
Author: Chuck Shotton
Requirements: AppleScript
Cost: free
Pros: decodes URLs from their hexadecimal equivalents
Cons: none
Remote location: <URL:>
Tricks location: <URL:http://152.1.24.177/teaching/archives/decode-url.hqx>


As you may or may not know, WWW browsers encode reserved characters in URLs when requesting files from remote services. The happens most frequently when accessing gopher services and delivering the data from GET or POST queries. While your scripts never get the URL in its entirety, there are parts of it you may have to decode. This is what the Decode URL OSAX is for. A good example of its use is within the redirect.acgi. Unfortunately, redirect.acgi has been saved as a read-only script. If you want to undo what the browsers have done to our URL, then Decode URL is an OSAX for you.

ENCODE URL

Version: 1.0
Author: Chuck Shotton
Requirements: AppleScript
Cost: free
Pros: encodes restricted characters of URLs into hexadecimal equivalents
Cons: none
Remote location: <URL:>
Tricks location: <URL:http://152.1.24.177/teaching/archives/encode-url.hqx>


This OSAX performs the same operations on URLs that WWW browsers perform on URLs when they send any type of FORMs data to a server. In other words, Encode URL translates the reserved characters of URLs (like plus signs, colons, or equals signs) into their

hexadecimal equivalents. This is useful when used in conjunction with WebRunner (described in the "Enhancing Browsers" chapter) or when writing scripts that perform some sort of redirection.

Parse CGI OSAX

Version: 1.2
Author: Alex Powers and Mark Kriegsman
Requirements: AppleScript
Cost: $10 (shareware)
Pros: quickly and easily parses search and POST arguments into variables
Cons: none
Remote location: <URL:http://marquis.tiac.net/software/parse-cgi.html>
Tricks location: <URL:http://152.1.24.177/teaching/archives/parse-cgi-osax-12.hqx>


This is the quinessencial OSAX for WWW scripting on a Macintosh with AppleScript. With two simple commands (parse CGI arguments and CGI field) you can very quickly decode and parse the any search or POST arguments sent to a script. This script beats Decode URL and Tokenize since it essentially combines the functions of these OSAX into one and does it more efficiently.

Script Tools

Version: 1.3-0
Author: Mark Alldritt
Requirements: AppleScript
Cost: free
Pros: contains a "grep" tool useful for text processing
Cons: none
Remote location: <URL:>
Tricks location: <URL:http://152.1.24.177/teaching/archives/script-tools.hqx>


Script Tools is a collection of OSAX. The most useful OSAX for WWW server scripting is the Regular Expressions OSAX. If you are familiar with the Unix environment, or if you use BBEdit as your text editor, then you may have heard of "regular expressions" before. In short, regular expressions represent a method of describing, searching, and replacing text strings using a sets of symbols. The whole process is very cryptic but very powerful. Using regular expressions brings to mind magical incantations whose force is equal to Merlin the Magicians. In any event, Script Tools and the Regular Expressions OSAX are a recommended addition to your OSAX collection if your scripts do a lot of text manipulation.

TCP/IP Scripting Addition

Version: 1.2
Author: Atul Butte
Requirements: AppleScript
Cost: $0-$5000 (shareware)
Pros: facilitates TCP/IP communications through AppleScripts
Cons: none
Remote location: <URL:>
Tricks location: <URL:http://152.1.24.177/teaching/archives/tcp-scripting-addition.hqx>

The TCP/IP Scripting Additions provide you with the means of opening TCP/IP connections to remote services through AppleScript. Theoretically, this means you could write your own WWW server using this scripting addition and AppleScript. In fact, with the distribution of the TCP/IP Scripting Addition comes a gopher server in the form of an AppleScript application. Realistically, this tool can be used in our server scripts to create very simple Internet clients. For example, you can use the TCP/IP Scripting Additions to send email from within a server script. You could write a program implementing the finger protocol and provide an interface to your script through a WWW server. You could use the TCP/IP Scripting Addition to write a URL checking program. Besides the more practical purposes, learning how to use the TCP/IP Scripting Additions will give you a better insight into the inner workings of TCP/IP communications.

IMAGE MAPPING

This chapter describes how implement imagemapping on your server.

Imagemapping is one of the unique features of WWW servers and one of the most fun. In a sentence, imagemapping allows you to create a graphic image, display it in your HTML files using the IMG tag and the ISMAP attribute, and assign URLs to different regions of the graphic. Thus, when somebody clicks on parts of the graphic, the point defining where they clicked is sent from the client application to the server. The server passes this information along to an imagemapping CGI program. The imagemapping CGI program "maps" the point to a list of predefined URLs, and finally, the mapped URL is returned to the client in turn "redirecting" the client application to another document.

The imagemap of Europe below illustrates this concept. Each dot is associated with an HTML file containing a graphic, a picture of water from that location. The cartoon balloon and mini speaker are associated with an "about" text and a sound of running water, respectively. Any region that is not a dot nor a graphic is associated with a friendly error message. By clicking on any of the map elements (dots or graphics), your client program (Netscape, for example) sends an (x, y) coordinate to the server who passes it along to a CGI program. The CGI reads its configuration file and maps the point to a URL. This URL is then passed back up the chain to your client application who is automatically sent to a different URL, a URL displaying pictures of bottled water. Give it a try, and hopefully it will be fun!

Imagemapping example - "Waters of Europe"

Implementation

Actually implementing an imagemap is a 5 step process:

1. Create your graphic
2. Create an imagemap configuration file
3. Setup and configure your imagemapping CGI program
4. Incorporate the graphic into an HTML file
5. Test

Step #1: Create your graphic

Use your favorite painting or drawing program to create your graphic. Like any other graphics on your server, keep in mind your potential audience. In other words, ask yourself what types of computing platforms might your readers be using and what types of Internet connections may they have. In general, try not to make your graphics too large in terms of file size as well as area. It is a good ideas not to use more than 256 colors in your graphics and make sure the resolution is no more than 72 dots per inch (dpi). Save your graphics as a PICT file, a standard image format available on all Macintosh graphics editing programs.

Step #2: Create an imagemap configuration file

Using either Hyper Mapedit or WebMap (described below), create an imagemap configuration file. In general, you will use these programs to associate regions of the graphic (circles, rectangles, or irregular polygons) with URLs. By selecting their Save option, you will have

created the configuration file. It is a good idea to save the configuration files with a .map extension for recognizability.

Step #3: Setup and configure your imagemapping CGI program

The next step is to set up your imagemapping CGI program. Each of the reviewed programs (Mac-ImageMap, or MapServe) have different configuration methods and consequently, these methods are described in each of their respective sections below.

Step #4: Incorporate the graphic into an HTML file

Edit your HTML file to include your graphic. If you haven't done so already, you will have to convert your graphic into a GIF image with a converter program like clip2gif or GIFConverter. When adding the IMG tags to your HTML document, be sure to include the ISMAP attribute so the client applications know to send point coordinates where the user clicked. For example, the tags denoting the inclusion of a graphic in your HTML may look like this:

```
<IMG SRC="../graphics/water-map.gif" ISMAP>
```

The IMG denotes a image. The SRC specifies its location. The ISMAP tells the client that the graphic is a clickable map.

Next, you will have to actually make the image hot by applying a CGI anchor to it. This is done almost like any other anchor. Instead of inserting the URL of an HTML document, you insert the URL of your imagemapping CGI program and the imagemap configuration file created in Step #2. For example, suppose you were using MapServe as your CGI program and you had created a configuration file named "water.map", then your HTML code would look something like this:

```
<A HREF=../scripts/mapserve.cgi$water-map.map><IMG
SRC=../graphics/water-map.gif ISMAP></A>
```

The example can be interpreted this way:

1. Display the image water-map.gif from the graphics folder
2. Make the image a link and send any clicks on the image to the program called mapserve.cgi in the scripts folder telling mapserve.cgi to use the configuration file water-map.map.

Note how this example takes advantage of relative URLs.

Step #5: Test

Finally, give your new imagemap a test drive, by accessing your newly created HTML page through your server. It is important to do this because if you don't go through your server, then the point information will never get passed to the CGI application.

Imagemapping Software

The programs below will help you automate much of the processes described above.

See Also

Jon Wiederspan, "Maps for MacHTTP"
<URL:http://www.uwtc.washington.edu/Computing/WWW/Map.html>

Hyper Mapedit

Version: 1.0
Author: Nathan Gasser
Requirements: HyperCard 2.1 or greater
Cost: free
Pros: "works as advertised"
Cons: sometimes PICT files do not import
Remote location: <URL:>
Tricks location: <URL:http://152.1.24.177/teaching/archives/hyper-mapedit.hqx>


Hyper MapEdit is a useful program for creating imagemap configuration files. It has got to be one of the oldest Macintosh-based WWW utilities. (It's dated November 23, 1993.) What's more, it still works great! The program requires HyperCard 2.1 or better. This is very important because Hyper MapEdit does not work with HyperCard Player.

If you are going to use Hyper MapEdit to write your imagemap configuration file, then the first step is to create your graphic and save it in PICT file format. After launching Hyper MapEdit, a number of palettes appear. By clicking the "Load Picture..." button on one of those pallets you are prompted for a graphic to import. If successful, Hyper MapEdit will import your picture and convert it to a black and white image. (This procedure has met with mixed success and you may have to resort to copying your graphic to the clipboard and pasting it into Hyper MapEdit.)

Now that the image is displayed you begin defining regions of your graphic using the circle, rectangle, and irregular polygon buttons. This is done by clicking on the buttons and then clicking on the graphic. Circles are defined by a radius. Rectangles are defined by opposite corners. Irregular polygons are defined by a number of points. After each region is defined you are prompted for a URL to be associated with the region. When you are finished defining your map, click the "Save Map..." button. You will be prompted for a default URL (a URL to return when the end-user does not click in a defined region) and then a file name for your map. It is a good idea to give your map a .map extension, and it is a good idea to save it in the same folder as your imagemapping CGI program.

Hyper MapEdit is not as pretty as WebMap, but it is just as effective. Furthermore, its free.

WEBMAP

Version: 1.0.1
Author: Rowland Smith
Requirements: 2MB RAM
Cost: $20 (shareware)
Pros: uses Macintosh interface well, saves configurations in resource fork of the graphics
Cons: none
Remote location: <URL:http://www.city.net/cnx/software/webmap.html>
Tricks location: <URL:http://152.1.24.177/teaching/archives/webmap.sea.hqx>

 WebMap has many more features than Hyper MapEdit and it is much more graphical in nature. Furthermore, it supports a wider range of input and output options when compared to Hyper MapEdit.

To begin with, GIF files as well as PICT files are supported by WebMap. Simply choose Open from the File menu and select your image. The resulting image is displayed in full color. Next, set up your preferences using the Preference menu choice of the Edit menu. For the most part you will want to WebMap to create NCSA imagemap files and not CERN imagemap files. NCSA imagemap files are the only file format supported by the imagemap CGI programs described later. On the other hand, if you have other WWW servers in your domain, then you may have a need for the CERN imagemap format.

The next step is to select either the circle, rectangle, or irregular polygon tool from the floating palette. Once selected, simply draw the desired shape. After the region is defined by the shape you can select, move, and resize the shape on the graphic at will. When a region is defined, an "untitled" URL will appear to the right. By double-clicking on the URL you are prompted for a real URL, the URL you want referred back to the end-user's client application. These URLs are very easy to edit if you make a mistake. One of the nicest features of WebMap is that these URLs are incorporated as resources into the actual graphic itself. This makes editing the imagemap later on much more convenient. When you are done, just select Save from the File menu.

WebMap uses the Macintosh interface to its best advantage. You would be hard pressed to find a better imagemapping configuration file editor.

IMAGEMAP-PL

Version: 1.6
Author: Scott Atwood
Requirements: MacPerl runtime software
Cost: free
Pros: can be used a learning tool
Cons: too much overhead to be effective
Remote location: <URL:>
Tricks location: <URL:http://152.1.24.177/teaching/archives/imagemap16.sit.hqx>

Imagemap-pl is a PERL script used for the purposes of handling imagemaps. When compared to Mac-Imagemap and MapServe, Imagemap-pl requires too much overhead to run effectively. On the other hand, since Imagemap-pl is not distributed in a compiled format, you can examine its code to learn a bit of the PERL scripting language and how imagemapping can be done. Maybe you can write a better program.

MAC-IMAGEMAP

Version: 1.3
Author: Lutz Weimann
Requirements: MacHTTP 2.0 or later, System 7
Cost: free
Pros: supports automatic quitting, good documentation
Cons: configuration file at first difficult to interpret
Remote location: <URL:http://weyl.zib-berlin.de/imagemap/Mac-ImageMap.html>
Tricks location: <URL:http://152.1.24.177/teaching/archives/image-map-13.hqx>

Mac-ImageMap is free imagemapping application that can run both as a CGI as well as an ACGI application. To run it in either mode you just need to change its name from imagemap.cgi to imagemap.acgi.

Mac-Imagemap supports the standard region definitions (oval, rectangle, polygon), but it also supports the point region. You will have to edit your .map files by hand to include this feature since WebMap nor Hyper Mapedit support this region type.

Like MapServe, Mac-Imagemap has been redesigned to work hand-in-hand with WebSTAR.

Specifically, it will work through WebSTAR's ACTION handler. Using this feature can insure that more people will have access to your imagemaps since they will not be hindered by the "CERN proxy bug" that incorrectly encodes dollar signs ($) of URLs into %24. Using the ACTION handler also makes writing your HTML simpler and may eliminate the need for an imagemap.config file.

The most difficult part about using Mac-Imagemap is writing its configuration file. When running Mac-Imagemap as a CGI or ACGI application, a configuration file is necessary. Even though the documentation is verbose and plentiful, the non-standard format of the configuration file is difficult to decipher. Furthermore, this configuration is married to the host computer; you can not create a generic configuration file and move it from server to server. This is not a huge problem, just something you have to keep in mind.

If you are low on RAM, then Mac-Imagemap supports a StayOpen attribute in its configuration file. Consequently, you can configure Mac-Imagemap to quit after it is done processing. Mac-Imagemap also supports a test mode for new maps and the ability to include the email address of the server's maintainer in error messages.

Mac-Imagemap functions as advertised. Once you learn how to interpret the imagemap.config file, you will be pleased with its results. Learning this file's parameters is well worth the $20 you might spend on MapServe.

MAPSERVE

Version: 1.5
Author: Kelly Campbell
Requirements: System 7, MacHTTP 2.0 or later
Cost: $20 (shareware)
Pros: easy-to-set up for MacHTTP
Cons: doesn't allow automatic quitting
Remote location:
<URL:http://www.spub.ksu.edu/other/machttp_tools/mapserve/mapserve.html>
Tricks location: <URL:http://152.1.24.177/teaching/archives/mapserve1.5.sit.hqx>

MapServe is an easy-to-setup imagemapping program. It is run as an ACGI (asynchronous CGI) program and consequently allows your WWW server to process other requests while MapServe is processing its own.

To use MapServe you must first create your graphic and create an imagemap configuration file. These processes were described earlier. The next step is to save MapServe somewhere in your WWW server's directory structure. Preferably, you would save MapServe is the same location where you save other application-type files (.CGI, .ACGI, .script files).

In order to run MapServe, you must create a preferences file. This is done by choosing

Preferences from the File menu. Doing so you will be prompted for two things. First, you will be prompted for the location of your WWW server. This is easy enough. Just click the "Set Path" button and locate your server. This preference is important if you have or have not used relative URLs in your imagemap configuration file(s). Second, you will be prompted as to whether or not you want MapServe to return the IP address (or Internet name) of MapServe's host computer, or whether you want your WWW server to do this. This option is helpful if you have turned off the DNS setting in MacHTTP or WebSTAR and you want the remote client application to know your host computer's Internet name instead of just its IP number.

Making MapServe do its thing is now a matter of adding the ISMAP tag to your graphic and making the image hot by surrounding the entire IMG tag with an anchor. For example:

```
<a href="../scripts/mapserve.acgi$mapserve.map"><img
src="../graphics/mapserve.gif" ISMAP></a>
```

This HTML will first display a graphic (mapserve.gif). Since its specification concludes the ISMAP attribute, the client application knows the extract point locations when the graphic is clicked on and append them to any URLs sent to the server. Furthermore, when the end-user clicks on the graphic, it will send a URL to the server calling the program mapserve.acgi and specifying the mapserve.map imagemap configuration file.

Mapserve works very well. Its configuration file is necessary, but does not have to be created on the same computer hosting the MapServe. (Although, it is in your best interest to do so.) Furthermore, Mapserve has been enhanced to work well with WebSTAR and some of WebSTAR's special features. The only drawback to MapServe is that it does not support the option to quit after it has done its processing.

THE END... OF THE BEGINNING

This chapter summarized the contents of the book.

This is the end of the manuscript, but hopefully the beginning of your adventures on the Web.

This book has tried to explain the concepts behind the World Wide Web starting with the client/server model of computing and ending with principles of good information systems design. In between were descriptions and reviews of software applications, HTML techniques, CGI scripting, and hopefully more than a few good examples.

This book was not intended to be the end-all of Macintosh-based World Wide Web books. In fact, this book was intended to be just the opposite; this book is a starter kit, a tool helping you get familiar with the technology.

Now that you have become familiar with the workings of an HTTP server, you are encouraged to follow up on the things that peak your interest. Browse the list of readings in the bibliography or collection of Macintosh-related Internet resources, read some of the Usenet newsgroups specifically geared to the World Wide Web including:

- comp.infosystems.www.advocacy
- comp.infosystems.www.announce
- comp.infosystems.www.authoring.cgi
- comp.infosystems.www.authoring.html
- comp.infosystems.www.authoring.images
- comp.infosystems.www.browsers.mac
- comp.infosystems.www.servers.mac
- comp.infosystems.www.users

Consider joining one or two electronic mailing lists and/or newsgroups dealing with your subject matter. There are bound to be at least a few and they are bound to be talking about the World Wide Web to some degree.

Remember that the World Wide Web is essentially about communication, and you are using a new technology to accomplish ongoing, eternal goals; you are "teaching a new dog old tricks." You as well as thousands (and soon to be millions) of people are using this technology to share their ideas, messages, and products with whom ever will "listen." While you are not encouraged to bring up a Web server just because everybody else is, you are encouraged to explore the use of this technology and see whether or not it can be applied to your particular milieu.

Lastly, remember the concepts of readability, browsability, and searchability. They are the keys to providing an information service people will come back to again and again.

Appendix A: Special Characters and Entities

The table below, heavily based on Martin Ramsch's "iso8859-1 table", is a list of special characters, their HTML code equivalents, as well as their entity equivalents. See also "Hypertext Markup Language - 2.0 - Appendices."

By inserting these codes or the entities into your HTML documents, you can make your documents more expressive and appeal to a wider audience. For example, if you want to put the cents symbol (¢) into your documents, then enter "¢" and it should be rendered correctly in your browser application.

Even though these codes represent a standard encoding scheme, your browser may not display all of them correctly because of the font you are using.

Character	Description	Code	Entity
"	quotation mark	"	"
&	ampersand	&	&
<	less-than sign	<	<
>	greater-than sign	>	>
	non-breaking space		
¡	inverted exclamation mark	¡	¡
¢	cent sign	¢	¢
£	pound sign	£	£
¤	currency sign	¤	¤
¥	yen sign	¥	¥
¦	broken vertical bar	¦	¦
§	section sign	§	§
¨	spacing diaresis	¨	¨
©	copyright sign	©	©

Character	Description	Code	Entity
ª	feminine ordinal indicator	ª	ª
`	angle quotation mark, left	«	«
¬	negation sign	¬	¬
–	soft hyphen		
®	circled R registered sign	®	®
¯	spacing macron	¯	&hibar;
°	degree sign		
±	plus-or-minus sign	±	±
"	superscript 2	²	²
"	superscript 3	³	³
´	spacing acute	´	´
µ	micro sign	µ	µ
¶	paragraph sign	¶	¶
·	middle dot	·	·
¸	spacing cedilla	¸	¸
¸	superscript 1	¹	¹
º	masculine ordinal indicator	º	º
»	angle quotation mark, right	»	»
1/4	fraction 1/4	¼	¼
1/2	fraction 1/2	½	½
3/4	fraction 3/4	¾	¾
¿	inverted question mark	¿	¿
À	capital A, grave accent	À	À
Á	capital A, acute accent	Á	Á
Â	capital A, circumflex accent	Â	Â
Ã	capital A, tilde	Ã	Ã
Ä	capital A, dieresis or umlaut mark	Ä	Ä
Å	capital A, ring	Å	Å
Ç	capital C, cedilla	Ç	Ç
È	capital E, grave accent	È	È
É	capital E, acute accent	É	É
Ê	capital E, circumflex accent	Ê	Ê
Ë	capital E, dieresis or umlaut mark	Ë	Ë
Ì	capital I, grave accent	Ì	Ì
Í	capital I, acute accent	Í	Í
Î	capital I, circumflex accent	Î	Î
Ï	capital I, dieresis or umlaut mark	Ï	Ï
	capital Eth, Icelandic	Ð	Ð
Ñ	capital N, tilde	Ñ	Ñ
Ò	capital O, grave accent	Ò	Ò
Ó	capital O, acute accent	Ó	Ó
Ô	capital O, circumflex accent	Ô	Ô
Õ	capital O, tilde	Õ	Õ
Ö	capital O, dieresis or umlaut mark	Ö	Ö
◊	multiplication sign	×	×
Ø	capital O, slash	Ø	Ø
Ù	capital U, grave accent	Ù	Ù

Character	Description	Code	Entity
Ú	capital U, acute accent	Ú	
Û	capital U, circumflex accent	Û	Û
Ü	capital U, dieresis or umlaut mark	Ü	Ü
	capital Y, acute accent	Ý	Ý
	capital THORN, Icelandic	Þ	Þ
ß	small sharp s, German (sz ligature)	ß	ß
à	small a, grave accent	à	à
á	small a, acute accent	á	á
â	small a, circumflex accent	â	â
ã	small a, tilde	ã	ã
ä	small a, dieresis or umlaut mark	ä	ä
å	small a, ring	å	å
æ	small ae diphthong (ligature)	æ	æ
ç	small c, cedilla	ç	ç
è	small e, grave accent	è	è
é	small e, acute accent	é	é
ê	small e, circumflex accent	ê	ê
ë	small e, dieresis or umlaut mark	ë	ë
ì	small i, grave accent	ì	ì
í	small i, acute accent	í	í
î	small i, circumflex accent	î	î
ï	small i, dieresis or umlaut mark	ï	ï
	small eth, Icelandic	ð	ð
ñ	small n, tilde	ñ	ñ
ò	small o, grave accent	ò	ò
ó	small o, acute accent	ó	ó
ô	small o, circumflex accent	ô	ô
õ	small o, tilde	õ	õ
ö	small o, dieresis or umlaut mark	ö	ö
÷	division sign	÷	÷
ø	small o, slash	ø	ø
ù	small u, grave accent	ù	ù
ú	small u, acute accent	ú	ú
û	small u, circumflex accent	û	û
ü	small u, dieresis or umlaut mark	ü	ü
	small y, acute accent	ý	ý
	small thorn, Icelandic	þ	þ
ÿ	small y, dieresis or umlaut mark	ÿ	ÿ

APPENDIX B: RUNNING SERVERS WITHOUT THE INTERNET

This text describes how to run your WWW server without a TCP (Internet) connection.

There are many times when you may want to run your WWW server even though you are not directly connected to the Internet. For example, you may be giving a presentation. You may be at home without a connection. Your Internet provider happens to be "down."

At first glance, since your WWW server depends on the Internet (more specifically the transmission control protocol or TCP) to communicate with client applications, you may think it is not possible to run your WWW server without an Internet connection. This is not true. While it is true that without an Internet connection you will not be able to communicate with others on the Internet, you will be able to run our WWW server locally.

The "trick" is in configuring your MacTCP control panel document in a special way. This technique was first illustrated by Grant Neufeld in his documents "Running MacHTTP Locally Without A Network" and "Using TCP on a LocalTalk Network." Full and complete credit is given to him for these ideas.

To run your WWW server on your local computer without an Internet connection, follow these steps:

Make sure AppleTalk is turned on
: AppleTalk is the Macintosh's native networking protocol. Make sure it is turned on by opening your Chooser and selecting the Active button. You may have to restart your computer to complete this part of the process.

Configure MacTCP for AppleTalk communications

Open your MacTCP control panel document. One of the icons presented there will be LocalTalk. Select the LocalTalk icon. This tells MacTCP to get its Internet configuration via the AppleTalk protocol.

Give yourself a gateway, address class, and name server designation

Click the More... button. From the resulting dialog box:

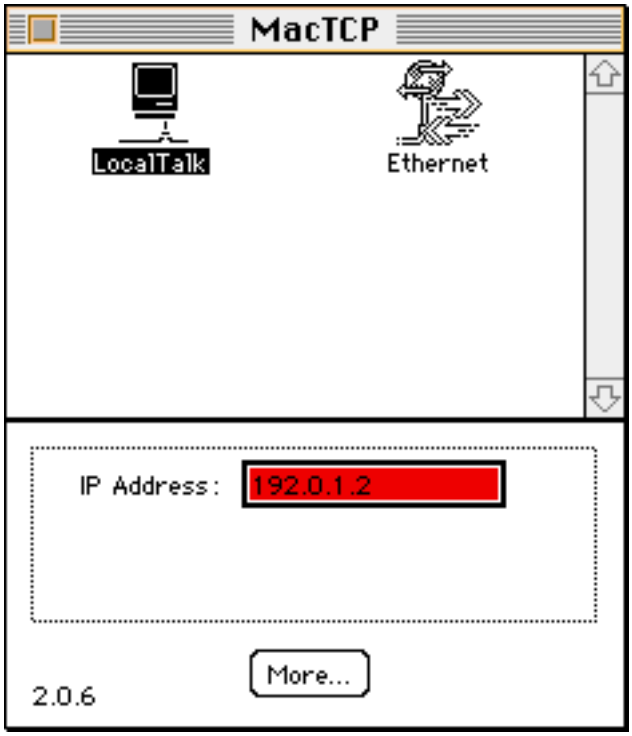
1. Select Manual from the "Obtain Address" section
2. Enter 0.0.0.0 for your gateway address
3. Select C as the class
4. Make sure the there is no information in the Name Server section

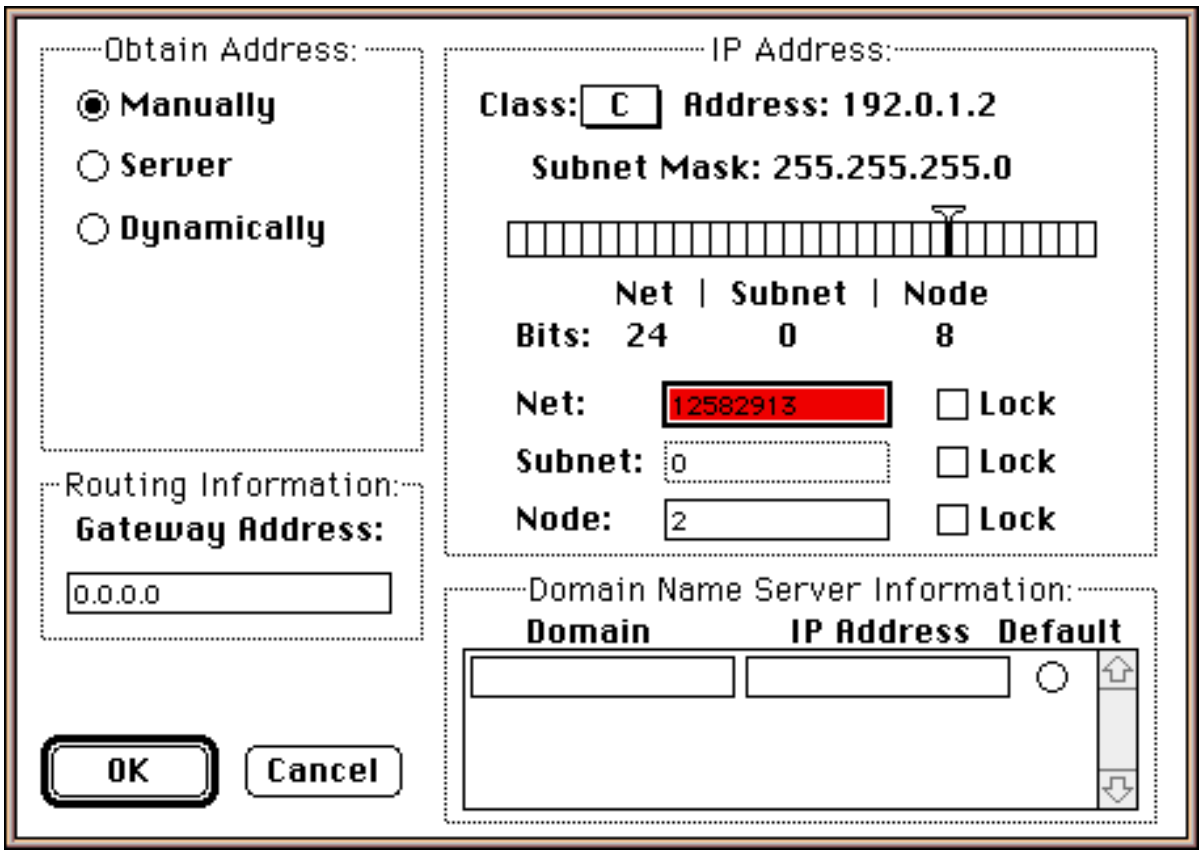
These settings tell our Macintosh that you have a static IP address (Manual), but you don't have a gateway. Your address class is C, and you don't use a name server (a database of Internet names).

Assign yourself an IP number

Close the dialog box by clicking the OK button. Finally, give yourself an IP number. Specifically, give yourself an IP number that is a class C address. Grant suggests 192.0.1.2. Again, you may have to restart your computer in order for the changes to take effect.

When the configuration is complete, your MacTCP control panel should look very similar to the dialog boxes below.





After the changes have taken effect (this may be after you restart), you can launch your WWW server and it will think it is connected to the Internet. More importantly, you will now be able to access our WWW server normally except you will have to connect to it via it new IP number. Again, to use this new configuration, you will have to access your server through the proper IP number, namely 192.0.1.2 or what ever IP number you assigned yourself in Step #4 above.

There are at least two caveats with this procedure. First you must tell your WWW server not to do DNS (domain name server) look ups. By not doing so, your WWW server will try to resolve IP numbers can cause problems. Second, CGI scripts that return referer information will not work. This includes imagemapping programs like MapServe and Mac-Imagemap. This happens because the IP address passed by your client application to the server is not quite valid. This incorrect information is then passed on to the CGI script in turn being passed back to you. Its a vicious circle.

The process of moving back and forth between various MacTCP configurations can be tedious. Luckily there is a utility call MacTCP Switcher that can save MacTCP configurations and update them. You may find this utility indispensable if you tend to "trick" MacTCP often.

Running your server on a AppleTalk-only network

You can extend this technique to many Macintoshes on an AppleTalk-only network. By configuring every Macintosh on a AppleTalk-only network exactly as outlined above, except incrementing every Macintosh's IP number by 1 (192.0.1.3, 192.0.1.4, 192.0.1.5, etc.), then you can run a server on one of these Macintoshes and each of the computers on your network will be able to access the server. Just as above, the server will have to be access through it IP number, what ever that is. Consequently, you can use your WWW server to disseminate information within your local area network without going through the Internet.

See Also

Grant Neufeld, "Running MacHTTP Locally Without A Network" - This text describes how to configure MacTCP so your WWW server will run even though you may not be physically connected to a network. <URL:http://arpp1.carleton.ca/machttp/doc/setup/nonetwork.html>

Grant Neufeld, "Using TCP on a LocalTalk Network" - Grant describes how to run a WWW server on an AppleTalk-only network without the Internet.
<URL:http://arpp1.carleton.ca/machttp/doc/setup/localtalk.html>

John Norstad, "MacTCP Switcher" - This handy utility saves and restores MacTCP configurations. <URL:ftp://ftp.acns.nwu.edu/pub/jlnstuff/mactcp-switcher/mactcp-switcher-11.sea.hqx>

Appendix C: Selected Readings

Thes pages lists selected readings to supplement the texts.

Browsability

Aslib, *Proceedings of the International Study Conference on Classification for Information Retrieval* (London: Aslib, 1957)

Bohdan S. Wynar, *Introduction to Cataloging and Classification* (Libraries Unlimited: Littleton CO, 1980) pg. 394

Derek Langridge, *Approach to Classification for Students of Librarianship* (Hamden, Connecticut: Linnet Books, 1973)

CGI Scripting

"AppleScript Archives" <URL:ftp://gaea.kgs.ukans.edu/applescript/>

"Overview of CGI" - "This page contains pointers to information and resources on the Common Gateway Interface, a standard for the interface between external gateway programs and information servers." <URL:http://www.w3.org/hypertext/WWW/CGI/Overview.html>

"Common Gateway Interface" - [This is the official specification for CGI scripting.] <URL:http://hoohoo.ncsa.uiuc.edu/cgi/intro.html>

Chuck Shotton, "Using FileMaker Pro with MacHTTP" - An archive with sample forms and

CGI that shows how to hook MacHTTP to FMPro.
<URL:http://www.biap.com/machttp/examples/fmpro.sit.hqx>

Chuck Shotton, "HyperCard CGI Demo" - Sample stack and HTML form demonstrating how HyperCard can be used to process searches, forms, etc. using the new CGI file type and "sdoc" AppleEvent sent by MacHTTP.
<URL:http://www.biap.com/machttp/ftp/hc_cgi_demo.sit.hqx>

Chuck Shotton, "Writing Search Engines for MacHTTP " - This points to an archive containing C source code for a sample application that performs searches in conjunction with MacHTTP using the "srch" AppleEvent.
<URL:http://www.biap.com/machttp/ftp/search_ex.sit.hqx>

Daaron Dwyer, "CGIs In Mac C" <URL:http://www.mccom.com/CGIinC/>

Danny Goodman, *Complete AppleScript Handbook* (Random House: New York, 1994)
Dave Winer, "Aretha Website" - "Frontier is a scripting system for the Macintosh. Lots of features, lots of verbs. It used to be a commercial product, but now it's free. Why? Because I want Frontier to have a shot at becoming a standard. I think it'll be fun!"
<URL:http://www.hotwired.com/staff/userland/aretha/>

Derrick Schneider, *Tao of AppleScript* (Hayden Books: Carmel, IN, 1993)
Grant Neufeld, "Grant's CGI Framework" - Grant's CGI Framework is a framework for writing 68K Macintosh CGI applications in C. It currently comes with a CodeWarrior project. Features: - AppleScriptable (including the CGI sdoc event!) - Threaded - Single function entry-point for customizing to your own needs - If you use it to write free applications, it's free to use <URL:http://arpp1.carleton.ca/grant/mac/grantscgi.html>

Ian Andrew Bell, "Macintosh WWW Tools Compendium"
<URL:http://www.arpp.sfu.ca/tools/>

John O'Fallon, "Maxum Home Page" - [This page documents many of the products of Maxum.] <URL:HTTP://www.digimark.net/maxum/>

Jon Wiederspan, "Extending WebSTAR" - Below are links to pages that hopefully will help you learn how to create CGI applications to extend the capabilities of your WebSTAR server. The primary requirements for following these lessons are a willingness to learn, patience with my jokes, and a WebSTAR server to practice on. Since the language I use in these lessons is AppleScript, it would also be a good idea to have some familiarity with it, although many complete beginners find the information here sufficient to get going.
<URL:http://www.uwtc.washington.edu/Computing/WWW/Lessons/START_HERE.html>

Jon Wiederspan, "Extending MacHTTP" - The following is a directory of the pages I have or hope to have soon that will help you learn how to use and/or create CGI applications to extend the capabilities of your MacHTTP server. The only requirement for following these lessons is a willingness to learn, patience with my jokes, and a MacHTTP server to practice on.
<URL:http://www.biap.com/tutorials/Extending_MacHTTP/ExtendingMacHTTP.html>

Jon Wiederspan, "MacWWW - CGI Applications" - [This page lists various CGI applications for use with your Macintosh-based WWW server.]
<URL:http://www.uwtc.washington.edu/Computing/WWW/Mac/CGI.html>

Jon Wiederspan, "Extending MacHTTP"
<URL:http://www.uwtc.washington.edu/Computing/WWW/ExtendingMacHTTP.html>

Martin Fenner, Fred Terry, and PreFab Software, Inc., "ScriptWeb" - "This virtual site is a collaborative effort to provide a single source of information for Macintosh scripting, primarily for AppleScript and Frontier." <URL:http://www.gz.com/scriptweb/>

Mason Hale, "Frontier CGI Scripting" - This is the home of documentation and examples related to the writing and running of CGI scripts in the Frontier scripting environment.
<URL:http://www.webedge.com/frontier/>

Matthias Neeracher, "MacPerl Q & A" <URL:http://err.ethz.ch/members/neeri/macintosh/perl-qa.html>

Netscape Communications, Inc., "Netscape API for the Macintosh" - "Netscape uses AppleEvents to interact with other Macintosh applications. It is scriptable, and partially recordable. Most of the events (and all Netscape-specific ones) are documented in the Netscape's AppleEvent dictionary. You see the dictionary with the Script Editor, a scripting utility available from Apple." <URL:http://home.netscape.com/newsref/std/mac-remote-control.html>

Robert Godwin-Jones, "Guide to Web Forms and CGI Scripts for Language Learning"
<URL:http://www.fln.vcu.edu/cgi/interact.html>

Sandra Silcot, "MacPerl Primer" - "This Primer is intended to assist new users get started with Macintosh Perl, and to point out salient differences for experienced Unix Perlers. This Primer is not a language reference manual, nor does it replace Matthias's documentation or Hal Wine's Frequently Asked Questions (FAQ) about MacPerl. The primer assumes you have already obtained and installed MacPerl, and that you have read the MacPerl FAQ."
<URL:http://www.unimelb.edu.au/~ssilcot/macperl-primer/home.html>

Scott S. Lawton, "Scripting" - Our goal is to make this site one of the three most useful places on the Web for Macintosh script writers, focusing on AppleScript and Frontier. (We also don't mind some good -- hopefully friendly -- competition!)
<URL:http://www.tiac.net/prefab/scripting.html>

Spyglass, Inc. , "Software Development Interface" - "This document describes a cross-platform API which can be used to extend the capabilities of Web browsers by integrating them with other applications. The API is specified as a set of platform-independent generic verbs which can be issued either to or from Web browsers. Platform-specific implementations of this spec are given for the AppleEvents and DDE transports. Future transports to be provided include OLE2 and TCP/IP (for UNIX)."
<URL:http://www.spyglass.com:4040/newtechnology/integration/iapi.htm>

Tom Trinko, *Applied Mac Scripting* (M and T Books: New York, 1995) - The volume represents just about the only book available describing how to program Frontier.

EDITING HTML

"Guides to Writing HTML Documents" - "These guides are about writing style, which HTML constructs to use when, when to divide up documents into multiple parts, etc. Also see information on the HTML language itself, learning HTML, editors, and converters." <URL:http://union.ncsa.uiuc.edu/HyperNews/get/www/html/guides.html>

"ANSI/ISO 8859-1 Coded Character Set" - "This list, sorted numerically, is derived from ANSI/ISO 8859-1 8-bit single-byte coded graphic character set." <URL:http://www.utirc.utoronto.ca/HTML_2.0/html-spec_11.html#SEC90>

Bill Spurlock, "HTML Writers Guild Website" - The Guild provides information resources for members in the form of FAQs (Frequently Asked Questions) and answers to those questions. FAQs are especially helpful to new Guild members and beginners in the art of Web authoring. <URL:http://www.mindspring.com/guild/>

David Siegel, "David Siegel's High Five" - This site demonstrates what can be done with HTML and Netscapisms. <URL:http://www.best.com/~dsiegel/high_five/high_five.html>

Eamonn Sullivan, "Crash course on writing documents for the Web" - "wrote this to help people in PC Week Labs to write documents for our web server. It's designed for someone who wants to put a page on the web but could care less about most of the technical details and doesn't want to read a book. I hope it's helpful." <URL:http://www.ziff.com:8002/~eamonn/crash_course.html>

Eric Tilton, "Composing Good HTML" - "As the Web continues to explode in its own inimitable fashion, it is becoming more and more important to write HTML that conforms to certain guidelines. Specifically, with the current diversity of clients for the Web (and we can only expect to see more!), it's become important to write HTML that will look good on any client, and not just on the specific client which the author may have access to." <URL:http://www.cs.cmu.edu/~tilt/cgh/>

Hakon W Lie, "HTML Style sheets" - "This page contains pointers to information about style sheets in the context of HTML. A mailing list, www-style@w3.org, has been started to host discussions on this topic. Feel free to add/delete yourself or browse the archive." <URL:http://www.w3.org/hypertext/WWW/Style/>

HTML Writers Guild, "Principles of Good HTML Design" - This is a list of ``general principles" of quality HTML design. It is intended to educate HTML authors to the elements of good and bad HTML style. It does not seek to ``control" Guild members, but rather to encourage them to adopt these practices in their everyday HTML construction. <URL:http://ugweb.cs.ualberta.ca/~gerald/guild/style.html>

Iam Graham, "HTML Documentation"
<URL:http://www.utirc.utoronto.ca/HTMLdocs/NewHTML/htmlindex.html>

Jeff Barry, "The Hypertext Markup Language (HTML) and the World-Wide-Web: Raising ASCII Text to a New Level of Usability." *The Public Access Computer Systems Review*, 5 no. 5 (1994):5-62 <URL:ftp://ftp.lib.ncsu.edu/pub/stacks/pacsr/pr-v5n05-barry-hypertext>

John Price-Wilkin, "Using the World Wide-Wide Web to Deliver Complex Electronic Documents: Implications for Libraries." *The Public Access Computer Systems Review*, 5 no. 3 (1994): 5-21 <URL:ftp://ftp.lib.ncsu.edu/pub/stacks/pacsr/pr-v5n03-price-wilkin-using>

Larry Aronson, *HTML Manual of Style* (Ziff-Davis Press: Emerville, CA 1994) - This is one of the first HTML books to appear on the market.
Lars Aronsson, "TEI Filelist -- files of the Text Encoding Initiative" - This list of files is one of the best collections describing the Standardized Generalized Markup Language (SGML).
<URL:http://www.lysator.liu.se/runeberg/teip3files.html>

Laura Lemay, *Teach Yourself Web Publishing with HTML in a Week* (SAMS Publishing: Indianapolis, IN 1995) - This easy-to-read book is thorough and complete.
Martin Ramsch, "iso8859-1 table" - [This is a list of the codes used to generate special characters in HTML.] <URL:http://www.uni-passau.de/~ramsch/iso8859-1.html>

National Center for Supercomputing Applications, "A Beginner's Guide to URLs"
<URL:http://www.ncsa.uiuc.edu/demoweb/url-primer.html>

NCSA, "HTML Primer " <URL:http://www.ncsa.uiuc.edu/demoweb/html-primer.html>

Robert Lentz, "Macintosh Web Programs and Utilties"
<URL:http://www.astro.nwu.edu/lentz/mac/net/mac-web.html>

Tim Berners-Lee and Dan Connolly, ed., "HyperText Markup Language (HTML)" - "HTML is a markup language for hypertext which is understood by all WWW clients. Here we discuss the HTML language, i.e. its syntax and semantics, including information on the history of the language, status of the standard, and development issues."
<URL:http://www.w3.org/hypertext/WWW/MarkUp/MarkUp.html>

Tom Magliery, "Mag's Big List of HTML Editors"
<URL:http://sdg.ncsa.uiuc.edu/~mag/work/HTMLEditors/>

Tom Savola with Alan Westenbroek and Joseph Heck, *Special Edition Using HTML* (Que: Indianapolis, IN 1995) - This book, while rather expensive described in detail HTML as well as providing many pieces of software and documentation on its CD-ROM.

Vannevar Bush, "As We May Think" *Atlantic Monthly* 176 (July 1945): 101-108
<URL:http://www.csi.uottawa.ca/~dduchier/misc/vbush/as-we-may-think.html>

Wm. Dennis Horn, "HTML Documents: A Mosaic Tutorial"
<URL:http://fire.clarkson.edu/doc/html/htut.html>

Yale Center for Advanced Instructional Media, "Yale C/AIM WWW Style Manual" - This is one of the more scholarly treatments of the subject.
<URL:http://info.med.yale.edu/caim/StyleManual_Top.HTML>

EXTENDING BROWSERS

Dave Winer, "Aretha Website" - "Frontier is a scripting system for the Macintosh. Lots of features, lots of verbs. It used to be a commercial product, but now it's free. Why? Because I want Frontier to have a shot at becoming a standard. I think it'll be fun!"
<URL:http://www.hotwired.com/staff/userland/aretha/>

Martin Fenner, Fred Terry, and PreFab Software, Inc., "ScriptWeb" - "This virtual site is a collaborative effort to provide a single source of information for Macintosh scripting, primarily for AppleScript and Frontier." <URL:http://www.gz.com/scriptweb/>

Netscape Communications, Inc., "Netscape API for the Macintosh" - "Netscape uses AppleEvents to interact with other Macintosh applications. It is scriptable, and partially recordable. Most of the events (and all Netscape-specific ones) are documented in the Netscape's AppleEvent dictionary. You see the dictionary with the Script Editor, a scripting utility available from Apple." <URL:http://home.netscape.com/newsref/std/mac-remote-control.html>

Spyglass, Inc., "Software Development Interface" - "This document describes a cross-platform API which can be used to extend the capabilities of Web browsers by integrating them with other applications. The API is specified as a set of platform-independent generic verbs which can be issued either to or from Web browsers. Platform-specific implementations of this spec are given for the AppleEvents and DDE transports. Future transports to be provided include OLE2 and TCP/IP (for UNIX)."
<URL:http://www.spyglass.com:4040/newtechnology/integration/iapi.htm>

GRAPHIC UTILITIES

"Computers:World Wide Web: HTML Editors:Macintosh"
<URL:http://www.yahoo.com/Computers/World_Wide_Web/HTML_Editors/Macintosh/>

"Macintosh WWW Development Resources"
<URL:http://www.uwtc.washington.edu/Computing/WWW/Macintosh.html>

Chuck Shotton, "How to serve QuickTime movies" - Serving QuickTime movies from a Mac running MacHTTP is really no different than serving any other binary document (such as GIF or JPEG files) with one exception. The movie data must be in a proper, flattened format.
<URL:http://www.biap.com/machttp/howto_qt.html>

Daniel W. Connolly, Bruno Girschweiler, and Tim Berners-Lee, "Graphics formats for WWW" - "The World-Wide web is a multimedia information space. This means that one of the design issues is dealing with various data formats. In addition to HTML and other textual formats, since the widespread availability of NCSA Mosaic and other visual interfaces to the web, more and more of the web's information is represented or augmented with data in any number of popular graphics formats."
<URL:http://www.w3.org/hypertext/WWW/Graphics/Overview.html>

Robert Lentz, "Macintosh Web Programs and Utilties"
<URL:http://www.astro.nwu.edu/lentz/mac/net/mac-web.html>

Yahoo, "Computers: World Wide Web: HTML Editors:Macintosh"
<URL:http://www.yahoo.com/Computers/World_Wide_Web/HTML_Editors/Macintosh/>

Helper Applications

Scott D. Nelson, "WWW Viewer Test Page" - This pages helps you test your WWW browser for helper applications. By clicking on the "test" buttons a file will be downloaded to your machine and then your browser will try to open it. If it doesn't open the file, then this page lists helper applications that you can use to "fix" the problem. <URL:http://www-dsed.llnl.gov/documents/WWWtest.html>

HTML Editors

"Computers:World Wide Web: HTML Editors:Macintosh"
<URL:http://www.yahoo.com/Computers/World_Wide_Web/HTML_Editors/Macintosh/>

"Macintosh WWW Development Resources"
<URL:http://www.uwtc.washington.edu/Computing/WWW/Macintosh.html>

IETF (Internet Engineering Task Force), "IETF HyperText Markup Language (HTML) Working Group" <URL:http://www.ics.uci.edu/pub/ietf/html/>

Yahoo, "Computers: World Wide Web: HTML Editors:Macintosh"
<URL:http://www.yahoo.com/Computers/World_Wide_Web/HTML_Editors/Macintosh/>

IMAGE MAPPING

Jon Wiederspan, "Maps for MacHTTP"
<URL:http://www.uwtc.washington.edu/Computing/WWW/Map.html>

MacHTTP and WebSTAR

Bill Doerrfeld, "MacHTTP-Talk Digests"
<URL:http://www.blueworld.com/bluespace/bluedata/MacHTTP-Talk_Search.html>

Brad Schrick, "Brad's WebSTAR/MacHTTP Pages" - [This is the most comprehensive list of MacHTTP/WebSTAR servers available.] <URL:http://brad.net/webstar/>

Chuck Shotton, "How to serve Mac documents" - Information on serving documents from Mac-specific applications like MS Word, Excel, or MacDraw Pro. (This info is somewhat outdated by the addition of MIME types in MacHTTP 1.2.4 and later.)
<URL:http://www.biap.com/machttp/howto_docs.html>

Chuck Shotton, "MacHTTP Technical Reference" - This document is intended to provide a reference to MacHTTP features and capabilities. [This document is also available as a part of the complete MacHTTP 2.2 distribution.]
<URL:http://www.biap.com/documentation/technical_ref.html>

Chuck Shotton, "MacHTTP Questions and Answers"
<URL:http://www.biap.com/documentation/QandA.html>

Chuck Shotton, "MacHTTP Security" - MacHTTP has two techniques for providing access controls. MacHTTP can restrict incoming connections based on the IP address or domain name of the remote client. Also, MacHTTP implements the "Basic" authentication scheme supported by most WWW clients and servers. This method allows you to assign access controls to files or folders that require a remote user to enter a valid username and password before they are allowed to access the information.
<URL:http://www.biap.com/tutorials/Security.html>

Chuck Shotton, "Performance Tuning for MacHTTP" - There are lots of different ways to get MacHTTP to "go real fast." This document discusses a few of the factors that affect MacHTTP's performance and how you can adjust them.
<URL:http://www.biap.com/tutorials/Performance.html>

Eric Bickford and Brad Schrick, "Mac Webmasters Consultant Directory" - "This Web site lists more then 150 consultants available throughout the world who are experts in electronic publishing on the Internet's World Wide Web using Apple Macintosh and compatible computers." <URL:http://www.macweb.com/consultants/>

Grant Neufeld, "Macintosh World Wide Web Frequently Asked Questions"
<URL:http://arpp1.carleton.ca/machttp/doc/>

Ian Andrew Bell, "Macintosh WWW Tools Compendium"
<URL:http://www.arpp.sfu.ca/tools/>

Robert Lentz, "Macintosh Web Programs and Utilties"
<URL:http://www.astro.nwu.edu/lentz/mac/net/mac-web.html>

StarNine, "StarNine Mailing List Maintenance"
<URL:http://emod.starnine.com/starnine/ListSTAR-Administration/Address-List-Editor.html>

StarNine, "WebSTAR by StarNine Technologies" - Based on Chuck Shotton's MacHTTP, WebSTAR(TM) 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(TM) movies. And yet, using WebSTAR is as easy as AppleShare(r). Plus, it's faster than many Web servers running on UNIX. <URL:http://www.starnine.com/webstar/webstar.html>

Stephen Collin, "Web 66" - <URL:http://web66.coled.umn.edu/>

Stephen E. Collins, "Classroom Internet Server Cookbook" - This is cookbook gives the recipes for setting up an Internet server in a classroom.
<URL:http://web66.coled.umn.edu/Cookbook/contents.html>

Verity Inc., "Recent MacHTTP/WebSTAR messages sorted by date"
<URL:http://asearch.mccmedia.com/MacHTTP_Talk/>

Verity Inc., "Virtual Library Search [Macintosh]" - This page allow you to search the archives of the MacHTTP Mailing List.
<URL:http://asearch.mccmedia.com/MacHTTP_Talk/search.html>

NO NETWORK COMPUTING

Grant Neufeld, "Running MacHTTP Locally Without A Network" - This text describes how to configure MacTCP so your WWW server will run even though you may not be physically connected to a network. <URL:http://arpp1.carleton.ca/machttp/doc/setup/nonetwork.html>

Grant Neufeld, "Using TCP on a LocalTalk Network" - Grant describes how to run a WWW server on an AppleTalk-only network without the Internet.
<URL:http://arpp1.carleton.ca/machttp/doc/setup/localtalk.html>

John Norstad, "MacTCP Switcher" - This handy utility saves and restores MacTCP configurations. <URL:ftp://ftp.acns.nwu.edu/pub/jlnstuff/mactcp-switcher/mactcp-switcher-11.sea.hqx>

Readability

HTML Writers Guild, "Principles of Good HTML Design" - This is a list of ``general principles" of quality HTML design. It is intended to educate HTML authors to the elements of good and bad HTML style. It does not seek to ``control" Guild members, but rather to encourage them to adopt these practices in their everyday HTML construction. <URL:http://ugweb.cs.ualberta.ca/~gerald/guild/style.html>

Jan V. White, *Graphic Design for the Electronic Age*, (Watson-Guptill : New York 1988)
Robin Williams, The Non-Designer's Design Book (Peach Pit Press: Berkeley CA 1994)

Roy Paul Nelson, *Publication Design*, 5th ed. (Wm. C Brown: Debuque IA 1991)

Yale Center for Advanced Instructional Media, "Yale C/AIM WWW Style Manual" - This is one of the more scholarly treatments of the subject.
<URL:http://info.med.yale.edu/caim/StyleManual_Top.HTML>

Uniform Resource Locators

Daniel W. Connolly, "WWW Names and Addresses, URIs, URLs, URNs, URCs" - "Addressing is one of the fundamental technologies in the web. URLs, or Uniform Resource Locators, are the technology for addressing documents on the web. It is an extensible technology: there are a number of existing addressing schemes, and more may be incorporated over time." <URL:http://www.w3.org/hypertext/WWW/Addressing/Addressing.html>

National Center for Supercomputing Applications, "A Beginner's Guide to URLs"
<URL:http://www.ncsa.uiuc.edu/demoweb/url-primer.html>

URI working group of the Internet Engineering Task Force, "Uniform Resource Locators" - Try also, <URL:http://www11.w3.org/hypertext/WWW/Addressing/URL/Overview.html>.
<URL:http://www11.w3.org/hypertext/WWW/Addressing/URL/URL_TOC.html>

World Wide Web

"World Wide Web" - [This URL will take you to a terminal-based WWW browser.]
<URL:telnet://telnet.w3.org:23/>

Aaron Anderson, "Mac Net Journal" <URL:http://www.dgr.com/web_mnj/>

Alan Richmond, "WWW Development" <URL:http://www.charm.net/~web/Vlib/>

Bob Alberti, et al., "Internet Gopher protocol"
<URL:gopher://boombox.micro.umn.edu/11/gopher/gopher_protocol>

CERN, "[Summary of HTTP Error Codes]"
<URL:http://info.cern.ch/hypertext/WWW/Protocols/HTTP/HTRESP.html>

CERN European Laboratory for Particle Physics , "CERN Welcome" - CERN is one of the world's largest scientific laboratories and an outstanding example of international collaboration of its many member states. (The acronym CERN comes from the earlier French title: "Conseil Europeen pour la Recherche Nucleaire") <URL:http://www.cern.ch/>

CNIDR, "freewais Page" <URL:http://cnidr.org/cnidr_projects/freewais.html>

Distributed Computing Group within Academic Computing Services of The University of Kansas, "About Lynx" <URL:http://kufacts.cc.ukans.edu/about_lynx/about_lynx.html>
Internet Engineering Task Force (IETF), "HTTP: A protocol for networked information" - HTTP is a protocol with the lightness and speed necessary for a distributed collaborative hypermedia information system. It is a generic stateless object-oriented protocol, which may be used for many similar tasks such as name servers, and distributed object-oriented systems, by extending the commands, or "methods", used. A feature if HTTP is the negotiation of data representation, allowing systems to be built independently of the development of new advanced representations.
<URL:http://www.w3.org/hypertext/WWW/Protocols/HTTP/HTTP2.html>

Jon Wiederspan, "Macintosh WWW Information" - "This is everything I have been able to gather about software, information sources, and online documentation that will help you put up a WWW site on your Macintosh computer. Someday soon I'll have some spiffy graphics, but that's not why you came, right? So here's the straight stuff for now."
<URL:http://www.uwtc.washington.edu/Computing/WWW/Mac/Directory.html>

Karen MacArthur, "World Wide Web Initiative: The Project" - [This site hosts many standard concerning the World Wide Web in general.] <URL:http://www.w3.org/>

Mary Ann Pike, et al., *Special Edition Using the Internet with Your Mac* (Que: Indianapolis, IN 1995)

NCSA, "NCSA Home Page" <URL:http://www.ncsa.uiuc.edu/>

NCSA, "NCSA Mosaic Home Page"
<URL:http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/Docs/help-about.html>

NCSA, "NCSA Mosaic for the Macintosh Home Page"
<URL:http://www.ncsa.uiuc.edu/SDG/Software/MacMosaic/MacMosaicHome.html>

NCSA, "NCSA Mosaic for Microsoft Windows Home Page"
<URL:http://www.ncsa.uiuc.edu/SDG/Software/WinMosaic/HomePage.html>

NCSA HTTPd Development Team, "NCSA HTTPd Overview"
<URL:http://hoohoo.ncsa.uiuc.edu/docs/Overview.html>

Software Development Group (SDG) at the National Center for Supercomputing Applications, "SDG Introduction" <URL:http://www.ncsa.uiuc.edu/SDG/SDGIntro.html>

Thomas Boutell, "World Wide Web FAQ" - "The World Wide Web Frequently Asked Questions (FAQ) is intended to answer the most common questions about the web." <URL:http://sunsite.unc.edu/boutell/faq/www_faq.html>

Tim Berners-Lee, Roy T. Fielding, and Henrik Frystyk Nielsen, "Hypertext Transfer Protocol" - "The Hypertext Transfer Protocol (HTTP) has been in use by the World-Wide Web global information initiative since 1990. HTTP is an application-level protocol with the lightness and speed necessary for distributed, collaborative, hyper media information systems. It is a generic, stateless, object-oriented protocol which can be used for many tasks, such as name servers and distributed object management systems, through extension of its request methods (commands). A feature of HTTP is the typing and negotiation of data representation, allowing systems to be built independently of the data being transferred." <URL:http://www.w3.org/hypertext/WWW/Protocols/Overview.html>

Ulrich Pfeifer, "FreeWAIS-sf" <URL:http://ls6-www.informatik.uni-dortmund.de/freeWAIS-sf/>

University of Kansas, "KUfact Online Information System" <URL:http://kufacts.cc.ukans.edu/cwis/kufacts_start.html>

University of Minnesota Computer & Information Services Gopher Consultant service, "Information about gopher" <URL:gopher://gopher.tc.umn.edu/11/Information%20About%20Gopher>

Vannevar Bush, "As We May Think" *Atlantic Monthly* 176 (July 1945): 101-108 <URL:http://www.csi.uottawa.ca/~dduchier/misc/vbush/as-we-may-think.html>

WAIS, Inc., "WAIS, Inc." <URL:http://www.wais.com/>

APPENDIX D: SOFTWARE INDEX

These pages lists all the software reviewed in this book. The items are arranged by topic and then alphabetically.

CGI SCRIPTING OSAX AND TOOLS

ACME SCRIPT WIDGETS

- **Remote location:** <URL:>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/acme-script-widgets.hqx>

DECODE URL

- **Remote location:** <URL:>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/decode-url.hqx>

ENCODE URL

- **Remote location:** <URL:>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/encode-url.hqx>

Parse CGI OSAX

- **Remote location:** <URL:http://marquis.tiac.net/software/parse-cgi.html>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/parse-cgi-osax-12.hqx>

Script Tools

- **Remote location:** <URL:>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/script-tools.hqx>

TCP/IP Scripting Addition

- **Remote location:** <URL:>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/tcp-scripting-addition.hqx>

Configuration File Editors for Imagemaps

Hyper Mapedit

- **Remote location:** <URL:>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/hyper-mapedit.hqx>

WebMap

- **Remote location:** <URL:http://www.city.net/cnx/software/webmap.html>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/webmap.sea.hqx>

CONVERTERS AND UTILITIES FOR HTML DOCUMENTS

COLORHEX

- **Remote location:** <URL:http://firehorse.com/colorhex/>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/ColorHEXb0.93.sit.hqx>

COLOUR HEXER

- **Remote location:** <URL:http://www.arts.unimelb.edu.au/Horwood/hcf.html>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/colour-hexer.hqx>

FORMSGENERATOR

- **Remote location:** <URL:http://edb518ea.edb.utexas.edu/felipe.html>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/formsgenerator.hqx>

HOTLIST2HTML

- **Remote location:** <URL:ftp://ftp.ncsa.uiuc.edu/Mosaic/Mac/Related/hotlist-to-html-071.hqx>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/hotlist-to-html-071.hqx>

HTML GRINDER

- **Remote location:** <URL:http://www.matterform.com/mf/grinder/htmlgrinder.html>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/grinder_bundle_2.0.sea.hqx>

HTML MARKDOWN

- **Remote location:** <URL:http://htc.rit.edu/scott.html>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/HTML-Markdown-101.hqx>

HTML Markup

- **Remote location:** <URL:http://htc.rit.edu/scott.html>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/HTML-Markup-10b1.hqx>

HTML Sketcher

- **Remote location:** <URL:>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/htmlSketcher_1.1.hqx>

HTML Special Effects

- **Remote location:** <URL:>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/html-special-effects.hqx>

HTML TableTool

- **Remote location:** <URL:>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/table-tool.hqx>

HTML+ XTND Translator

- **Remote location:** <URL:>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/html-plus-xtnd.hqx>

RTFtoHTML

- **Remote location:** <URL:ftp://ftp.cray.com/src/WWWstuff/RTF/rtftohtml_overview.html>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/rtftohtml-mac.hqx>

WebDoor Publisher

- **Remote location:** <URL:http://www.opendoor.com>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/webdoor-publisher-10.hqx>

xtnd HTML Translator

- **Remote location:** <URL:http://ai.eecs.umich.edu/highc/software/translator.html>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/HTML-Translator-1.0.hqx>

Extending WWW Browsers

Flypaper

- **Remote location:** <URL:http://www.pass.wayne.edu/~eric/flypaper>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/latest_flypaper.sit.hqx>

Scripting WWW Browsers

- **Remote location:** <URL:http://152.1.24.177/teaching/archives/scripting-WWW-browsers.hqx>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/scripting-WWW-browsers.hqx>

WebRunner

- **Remote location:** <URL:http://www.pass.wayne.edu/~eric/webrunner>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/latest_webrunner.sit.hqx>

Graphic Utilities

Clip2Gif

- **Remote location:** <URL:http://iawww.epfl.ch/Staff/Yves.Piguet/clip2gif-home/>
- **Tricks location:** <URL:>

FlattenMoov

- **Remote location:** <URL:ftp://ftp.ncsa.uiuc.edu/Mosaic/Mac/Related/flattenmoov.hqx>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/flattenmoov.hqx>

Transparency

- **Remote location:** <URL:>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/transparency.hqx>

Helper Applications

Acrobat Reader

- **Remote location:** <URL:ftp://ftp.ncsa.uiuc.edu/Mosaic/Mac/Helpers/acrobat-reader-20.hqx>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/acrobat-reader-20.hqx>

Drop Stuff w/Expander Enhancer

- **Remote location:** <URL:ftp://ftp.ncsa.uiuc.edu/Mosaic/Mac/Helpers/drop-stuff-with-ee-352.hqx>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/drop-stuff-with-ee-352.hqx>

Fast Player

- **Remote location:** <URL:ftp://ftp.ncsa.uiuc.edu/Mosaic/Mac/Helpers/fast-player-110.hqx>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/fast-player-110.hqx>

Ghostscript (MacGS Viewer)

- **Remote location:** <URL:http://www.glyphic.com/glyphic/projects/macgs.html>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/ghostscript.hqx>

GIF Converter

- **Remote location:** <URL:ftp://ftp.ncsa.uiuc.edu/Mosaic/Mac/Helpers/gif-converter-237.hqx>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/gif-converter-237.hqx>

GraphicConverter

- **Remote location:** <URL:ftp://ftp.ncsa.uiuc.edu/Mosaic/Mac/Helpers/graphic-converter-207.hqx>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/graphic-converter-207.hqx>

JPEGView

- **Remote location:** <URL:ftp://ftp.ncsa.uiuc.edu/Mosaic/Mac/Helpers/jpeg-view-331.hqx>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/jpeg-view-331.hqx>

MacBinary II+

- **Remote location:** <URL:http://hyperarchive.lcs.mit.edu/HyperArchive/Archive/cmp/mac-binary-ii-plus-101.hqx>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/mac-binary-ii-plus-101.hqx>

MacGZip

- **Remote location:** <URL:http://persephone.cps.unizar.es/general/gente/spd/gzip/gzip.html>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/mac-gzip-03b3.hqx>

Sound Machine

- **Remote location:** <URL:ftp://ftp.ncsa.uiuc.edu/Mosaic/Mac/Helpers/sound-machine-21.hqx>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/sound-machine-21.hqx>

SoundApp

- **Remote location:** <URL:http://hyperarchive.lcs.mit.edu/HyperArchive/Archive/snd/util/sound-app-151.hqx>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/sound-app-151.hqx>

Sparkle

- **Remote location:** <URL:ftp://ftp.ncsa.uiuc.edu/Mosaic/Mac/Helpers/sparkle-24.hqx>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/sparkle-24.hqx>

Stuffit Expander

- **Remote location:** <URL:ftp://ftp.ncsa.uiuc.edu/Mosaic/Mac/Helpers/stuffit-expander-352.hqx>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/stuffit-expander-352.hqx>

Tar for the Macintosh

- **Remote location:** <URL:>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/tar.hqx>

HTML EDITORS

ALPHA

- **Remote location:** <URL:http://www.cs.umd.edu/~keleher/alpha.html>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/Alpha-html-0.20.sit.hqx>

ARACHNID

- **Remote location:** <URL:http://sec-look.uiowa.edu/about/projects/arachnid-page.html>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/arachnid-1.5.4b-full.sea.hqx>

BBEDIT HTML EXTENSIONS

- **Remote location:** <URL:http://www.uji.es/bbedit-html-extensions.html>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/bbedit-html-ext.hqx>

BBEDIT HTML TOOLS

- **Remote location:** <URL:http://www.york.ac.uk/~ld11/BBEditTools.html>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/BBEdit-HTML-Tools.hqx>

HIGH TEA

- **Remote location:** <URL:http://www.mistral.co.uk/cbuzz>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/high-tea.hqx>

HTML EDITOR FOR THE MACINTOSH

- **Remote location:** <URL:http://dragon.acadiau.ca/~giles/HTML_Editor/Documentation.html>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/html-editor-10.hqx>

HTML Pro

- **Remote location:** <URL:http://www.ts.umu.se:80/~r2d2/shareware/shareware.html>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/html-pro-105.hqx>

HTML SuperText

- **Remote location:** <URL:http://www.potsdam.edu/HTML_SuperText/About_HTML_S.html>
- **Tricks location:** <URL:>

HTML Tag Macros

- **Remote location:** <URL:>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/html-tag-macros.hqx>

HTML Web Weaver

- **Remote location:** <URL:http://www.student.potsdam.edu/web.weaver/about.html>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/html-web-weaver.hqx>

HTML-HyperEditor

- **Remote location:** <URL:http://balder.syo.lu.se/Editor/HTML-HyperEditor.html>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/HTML-HyperEditor1.0.S.sea.hqx>

HTML.edit

- **Remote location:** <URL:http://nctn.hq.nasa.gov/tools/htmledit/HTMLEdit.html>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/HTMLEdit.hqx>

HTMLWriter

- **Remote location:** <URL:http://www.uwtc.washington.edu/JonWiederspan/HTMLEditor.html>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/HTMLWriter0.9d4.sit.bin>

Simple HTML Editor (SHE)

- **Remote location:** <URL:http://www.lib.ncsu.edu/staff/morgan/simple.html>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/SHE.hqx>

Site Writer Pro

- **Remote location:** <URL:http://www.rlc.dcccd.edu/Human/SWPro.htm>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/Site-Writer-Pro.hqx>

Webtor

- **Remote location:** <URL:http://www.igd.fhg.de/~neuss/webtor/webtor.html>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/webtor0.9a.sea.hqx>

HTTP Servers

FTPd

- **Remote location:** <URL:>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/FTPd-300.hqx>

HTTPD4Mac

- **Remote location:** <URL:http://130.246.18.52/>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/httpd4mac-123a.hqx>

MACHTTP

- **Remote location:** <URL:http://www.biap.com/>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/machttp-2.2.sit.hqx>

IMAGEMAP CGI PROGRAMS

IMAGEMAP-PL

- **Remote location:** <URL:>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/imagemap16.sit.hqx>

MAC-IMAGEMAP

- **Remote location:** <URL:http://weyl.zib-berlin.de/imagemap/Mac-ImageMap.html>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/image-map-13.hqx>

MAPSERVE

- **Remote location:**
  <URL:http://www.spub.ksu.edu/other/machttp_tools/mapserve/mapserve.html>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/mapserve1.5.sit.hqx>

MANAGING BOOKMARKS AND URLS

CLAY BASKET

- **Remote location:**
  <URL:http://www.hotwired.com/Staff/userland/clay/news_187.html>
- **Tricks location:** <URL:http://152.1.24.177/teachings/archives/clay-basket.hqx>

CYBER LINK

- **Remote location:** <URL:http://www.rtz.com/Cyber_Link_Info.html>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/cyber-link-10.hqx>

CYBERFINDER

- **Remote location:** <URL:http://hyperarchive.lcs.mit.edu/HyperArchive/Archive/comm/tcp/cyber-finder.hqx>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/cyber-finder.hqx>

DUKE OF URL

- **Remote location:** <URL:http://www.kei.com/duke-of-url/>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/duke-of-url-10-as.hqx>

HOTLIST GRABBER

- **Remote location:** <URL:ftp://ftp.ncsa.uiuc.edu/Mosaic/Mac/Related/hotlist-grabber-101.hqx>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/hotlist-grabber-101.hqx>

HOTLIST SORTER

- **Remote location:** <URL:ftp://ftp.ncsa.uiuc.edu/Mosaic/Mac/Related/hotlist-sorter-10b3.hqx>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/hotlist-sorter-10b3.hqx>

INTERNET SITE

- **Remote location:** <URL:http://www.er.uqam.ca/nobel/simoneau/aoce/InternetSite1-2.sea.hqx>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/InternetSite1-2.sea.hqx>

URLKey

- **Remote location:** <URL:http://www.city.net/cnx/software/urlkey.html>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/URLKey_1.0.7.sea.hqx>

Searching

AppleWebSearch

- **Remote location:** <URL:http://kamaaina.apple.com/>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/applewebsearch.sit.hqx>

TR-WWW

- **Remote location:** <URL:http://www.monash.edu.au/informatics/tr-www.html>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/tr-www.sit.hqx>

Server Enhancements

Annotate

- **Remote location:** <URL:>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/Annotate.cgi.sit.Hqx>

Count WWWebula

- **Remote location:** <URL:http://www.edb.utexas.edu/greg/ftp/programming/counter/index.cgi>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/CountWWWebula1.0a9.hqx>

FOLDER-LIST2

- **Remote location:** <URL:http://www.phy.utulsa.edu/Physics/files/folderList2.hqx>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/folderList2.hqx>

INDEX CGI

- **Remote location:** <URL:ftp://edb518ea.edb.utexas.edu/ftp/acgi/index.cgi.hqx>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/index.cgi.hqx>

KELLY'S ERROR.ACGI

- **Remote location:** <URL:http://www.spub.ksu.edu/other/machttp_tools/error/>
- **Tricks location:**
  <URL:http://152.1.24.177/teaching/archives/Error.acgi.0.6b4.sit.hqx>

MACHTTP APPLEGUIDE

- **Remote location:**
  <URL:ftp://192.253.114.3/pub/MacHTTP_stuff/MacHTTPGuide_v1.1.sit.bin>
- **Tricks location:**
  <URL:http://152.1.24.177/teaching/archives/MacHTTPGuide_v1.1.sit.bin>

RANDOMGIF.ACGI

- **Remote location:** <URL:http://edb518ea.edb.utexas.edu/felipe.html>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/randomgif.acgi.hqx>

REDIRECT.ACGI

- **Remote location:** <URL:http://www.netdreams.com/net.dreams/software.html>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/redirect.hqx>

ServerStat Lite

- **Remote location:** <URL:http://www.ericse.ohio-state.edu/kitchen-sink/ss.html>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/server-stat-lite.hqx>

WebStar/MacHTTP Indexer

- **Remote location:** <URL:http://www.dow.com/~haberman/wsi/wsi.html>
- **Tricks location:**
  <URL:http://152.1.24.177/teaching/archives/webstarindexer2.2.sit.hqx>

WebStat

- **Remote location:** <URL:>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/webstat.sit.hqx>

World Wide Web Browsers

America Online

- **Remote location:** <URL:http://www.blue.aol.com/>
- **Tricks location:** <URL:>

eWorld World Wide Web Browser

- **Remote location:** <URL:http://www.eworld.com/>
- **Tricks location:** <URL:>

MacWeb

- **Remote location:** <URL:ftp://ftp.einet.net/einet/mac/macweb/macweb.latest.sea.hqx>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/macweb.latest.sea.hqx>

NCSA Mosaic

- **Remote location:**
  <URL:ftp://ftp.ncsa.uiuc.edu/Mosaic/Mac/NCSAMosaic200B12.68K.hqx>
- **Tricks location:**
  <URL:http://152.1.24.177/teaching/archives/NCSAMosaic200B9.68K.hqx>

Netscape

- **Remote location:** <URL:ftp://ftp.mcom.com/netscape/mac/>
- **Tricks location:** <URL:http://152.1.24.177/teaching/archives/netscape-1.1N.hqx>

TCP/Connect II

- **Remote location:** <URL:http://www.intercon.com/>
- **Tricks location:** <URL:>

APPENDIX E: MACINTOSH-RELATED INTERNET RESOURCES

This set of pages lists numerous Macintosh-related Internet resources. This list does not include items of a World Wide Web nature since those items are represented as a part of the book.

APPLE CORPORATION SERVERS

- ALUG (Apple Library Users Group) Newsletter <URL:gopher://info.hed.apple.com/11/Apple%20Library%20Users%20Group/ALUG%20Newsletter>

- Apple Business System Division - "This anonymous FTP archive (abs.apple.com) is maintained by Apple's Apple Business Systems Division Product Technical Support department. It is used - to distribute beta and experimental software to field test sites - to receive debugging data from test sites and others - to distribute a variety of software and information which may be peripherally related or completely unrelated to Apple's business, and which is offered for purely informational purposes." <URL:ftp://abs.apple.com/>

- Apple Computer Higher Education Gopher Server - "We also want to maintain useful information for the higher education community. This folder is continuously being developed." <URL:gopher://info.hed.apple.com/>

- Apple Computer, Incorporated <URL:http://www.apple.com/>

- Apple Press Releases
  <URL:gopher://info.hed.apple.com/11/Apple%20Press%20Releases>

- Apple Product Descriptions
  <URL:gopher://info.hed.apple.com/11/Product%20Information>

- Apple product listings and product support
  <URL:http://www.apple.com/documents/productsupport.html>

- Apple-Internet Mailing Lists Home Page - "This is the home page for the Apple-Internet family of mailing lists. It contains pointers to information sources of use to people using Macintosh computers on the Internet." <URL:http://abs.apple.com/apple-internet/>

- Macintosh Product Guide, by Apple Computer, Inc. - "This database contains information about Macintosh products from publishers around the world. There are currently around 14,000 products listed, with more added every month."
  <URL:http://www.apple.com.au/MPG/>

COMPANIES

- Adobe <URL:http://www.adobe.com/>

- Apple Computer, Incorporated <URL:http://www.apple.com/>

- ClarisWeb, by Claris <URL:http://www.claris.com/>

- Fairgate Technologies CodeWarrior Support - "Welcome to the officially supported WWW site for CodeWarrior support! Metroworks has generously agreed to provide us source code, examples, marketing material, and other CW-related data for your browsing pleasure. If you like this site, make sure to let MW know you appreciate their helping to make this Internet resource possible."
  <URL:http://www.iquest.com/~fairgate/cw/cw.html>

- Farallon - "Farallon makes innovative plug-and-play networking products to attach personal computers to AppleTalk, TCP/IP, and 10BASE-T networks to help users navigate quickly and easily to enhance their communications. Farallon's award winning EtherWave, Timbuktu, and Replica products adhere to industry standards yet are flexible and simple to install and use, which makes them compelling to both network administrators and network users. " <URL:http://www.farallon.com/>

- Global Village Communication - "Global Village develops and markets communication products for personal computer users. Visit the different areas in The Village to learn more about communicating from your computer, including faxing, accessing on-line services and the Internet, and connecting to remote networks."
  <URL:http://www.globalvillag.com/>

- Hewlett-Packard - "Hewlett-Packard designs, manufactures and services electronic products and systems for measurement, computation and communications. Our basic business purpose is to create information products that accelerate the advancement of knowledge and improve the effectiveness of people and organizations. The company's products and services are used in industry, business, engineering, science medicine and education in approximately 110 countries." <URL:http://www.hp.com>

- NEC <URL:http://www.nec.com>

- Power Computing Corporation - "[A maker of Macintosh clones.]" <URL:http://www.powercc.com/>

- Symantec - "Symantec Corporation is a leading software company with award-winning application and system software for Windows, DOS, Macintosh and OS/2 computer systems. Founded in 1982, Symantec has grown rapidly through the success of its products and a series of fourteen acquisitions resulting in a broad line of business and productivity solutions. The company has several enterprise-wide products that have been introduced recently and others that are under development." <URL:http://www.symantec.com/>

FAQs

- Buying and Selling Macintosh Computers, Software and Peripherals - "comp.sys.mac.faq, part 4: ... comp.sys.mac.apps ... Copyright 1993,1994 by Elliotte Harold ... Please see section 5.8 of the general FAQ if you wish to redistribute, revise or republish this document in any way. ... Archive-name: macintosh/apps-faq ... Version: 2.3.2 ... Last-modified: August 13, 1994 ... Address comments to elharo@shock.njit.edu ..." <URL:http://rever.nmsu.edu/~elharo/faq/wantedfaq.html>

- comp.sys.mac.comm Frequently Asked Questions <URL:http://www.cis.ohio-state.edu:80/hypertext/faq/usenet/macintosh/comm-faq/top.html>

- Frequently Asked Questions About Macintosh Application Software - "comp.sys.mac.faq, part 4: ... comp.sys.mac.apps ... Copyright 1993,1994 by Elliotte Harold ... Please see section 5.8 of the general FAQ if you wish to redistribute, revise or republish this document in any way. ... Archive-name: macintosh/apps-faq ... Version: 2.3.2 ... Last-modified: August 13, 1994 ... Address comments to elharo@shock.njit.edu ..." <URL:http://rever.nmsu.edu/~elharo/faq/appsfaq.html>

- Frequently Asked Questions about Macintosh System Software - "comp.sys.mac.faq, part 4: ... comp.sys.mac.apps ... Copyright 1993,1994 by Elliotte Harold ... Please see section 5.8 of the general FAQ if you wish to redistribute, revise or republish this document in any way. ... Archive-name: macintosh/apps-faq ... Version: 2.3.2 ... Last-modified: August 13, 1994 ... Address comments to elharo@shock.njit.edu ..." <URL:http://rever.nmsu.edu/~elharo/faq/systemfaq.html>

- Macintosh FAQs <URL:http://www.astro.nwu.edu/lentz/mac/faqs/home-faqs.html>

- Macintosh Frequently Asked Questions - "comp.sys.mac.faq, part 1: ... Introduction to the Macintosh Newsgroups ... Copyright 1993, 1994 by Elliotte Harold ... Please see section 5.8 below if you wish to ... distribute or revise this document in any way. ... Version: 2.3.3 ... Last-modified: November 4, 1994 ... Address comments to elharo@shock.njit.edu" <URL:http://rever.nmsu.edu/~elharo/faq/generalfaq.html>

- Macintosh PowerPC Frequently Asked Questions - "This FAQ was created in response to a request for a PowerPC FAQ in comp.sys.mac.hardware. It exists to answer basic questions about the future of Macintosh and its relation to the PowerPC series of microprocessors." <URL:http://www.cis.ohio-state.edu:80/hypertext/faq/usenet/macintosh/PowerPC-FAQ/faq.html>

- Miscellaneous Frequently Asked Questions - "comp.sys.mac.faq, part 4: ... comp.sys.mac.apps ... Copyright 1993,1994 by Elliotte Harold ... Please see section 5.8 of the general FAQ if you wish to redistribute, revise or republish this document in any way. ... Archive-name: macintosh/apps-faq ... Version: 2.3.2 ... Last-modified: August 13, 1994 ... Address comments to elharo@shock.njit.edu ..." <URL:http://rever.nmsu.edu/~elharo/faq/miscfaq.html>

- This is the Frequently Asked Question (FAQ) list for A/UX 3.x.x - "This FAQ list is intended to cut down on the number of "often asked questions" that make the rounds here on comp.unix.aux. Also included you'll find a few words of wisdom as well as some general information for the A/UX community. This list assumes that you are familiar with Unix (to some extent) but are curious about A/UX's eccentricities. The list will concentrate on A/UX 3.x.x but may also have info about previous versions. If you don't understand something in the FAQ List, and a "Point of Contact" isn't specified, then contact me and I'll attempt to help or else point you to someone who can. In any case, let me know how I can make the list more clear." <URL:http://jagubox.gsfc.nasa.gov/aux/FAQ.aux>

GUIDES

- Info-Mac Mirrors <URL:http://www.astro.nwu.edu/lentz/mac/net/info-mac-mirrors.html>

- Internet resources for Macintosh support, by Jon Wiederspan <URL:http://www.uwtc.washington.edu/Computing/Internet/MacintoshResources.html>

- Internet Starter Kit for Macintosh - "The purpose of this page is to provide links to some of what I feel are the most important Internet sites and to keep readers of Internet Starter Kit for Macintosh (and anyone else who cares) informed about developments of interest to Macintosh users." <URL:http://www.tidbits.com/tidbits/index.html>

- jagubox's A/UX Home Page - "This is the A/UX directory, which contains _loads_ of

neat, useful, and needed A/UX stuff." <URL:http://jagubox.gsfc.nasa.gov/aux/>

- Mac ftp Sites - "This page is based on version 3.9.1 of the information maintained by Bruce Grubb (with permission). Corrections and/or additions should be sent to him (although it would be nice to notify the owner of this page also). The original list is copyrighted 1991 by Mike Gleason." <URL:http://www.realtime.net/~jones/mac-ftp-sites.html>

- Mac Software Catalog - "This service is only possible thanks to the enormous effort by the The University of Michigan's Macintosh Public Domain and Shareware Archive ... The index is mirrored daily from London, and the catalog is updated automatically. Note that we do not mirror the contents of the archive, but simply present an interface for existing mirrors." <URL:http://web.nexor.co.uk/public/mac/archive/welcome.html>

- Macintosh Index - "Welcome to the Macintosh Index. Here, you will find extensive information on the Internet that relates to the Macintosh." <URL:http://ici.proper.com/1/mac>

- Macintosh Internet Software, by Robert Lentz <URL:http://www.astro.nwu.edu/lentz/mac/software/mac-internet.html>

- Macintosh Internet software, by Gavin Bell - "Most of the below are ftp url references to the Sunsite UK archive held at Imperial College. The choice of software on this page is mainly my own, this is what I use daily and have found to be stable and useful. All comments after the software listings are my own opinions of the software. There is also a list of viewers and convertors for video and sound files, as well as utilities for compressed data. I will try to keep all the ftp url references correct, but some may slip a version for a couple of weeks or so." <URL:http://www.qub.ac.uk/sigweb/mac-comms-utils.html>

- Macintosh Resources <URL:http://www.astro.nwu.edu/lentz/mac/>

- Macintosh WWW Pointers, by Jay J. Myers - "This page is devoted primarily to items of interest to Macintosh users." <URL:http://www.nmia.com/~jjm/home.html>

- UBUdex - a HyperIndex to //Mac.Archive.UMich.edu/mac - "This HTML index has been running since 8 February 1994 and is "rebuilt" from the 00index.txt files of the /mac/* directories in mac.archive.umich.edu. Ideally, it will be FTP'ed nightly, HTMLized with a TPU routine (hey, I'm on a VAX/VMS system and Perl isn't part of my vocabulary) and FTP'ed into the local HTTP server to be made available. One of the future plans for this index is to take advantage of HTML forms and permit access to the same files via the mirror sites. The MacHTTP server can do forms, but MacMosaic doesn't have form support yet. " <URL:http://ubu.hahnemann.edu/UBUdex/UBUdex.html>

- utexas mac archive, by Chris W. Johnson <URL:http://wwwhost.ots.utexas.edu/mac/main.html>

HARDWARE

- Apple Product Descriptions <URL:gopher://info.hed.apple.com/11/Product%20Information>

- Apple Products <URL:http://microstore.ucs.sfu.ca/PriceLists/Apple/AppleProducts.html>

- Clock Chipping Home Page - "To ease access to some of the most up to date information on crystal oscillator swapping/acceleration on your Apple Macintosh computer I've started this Clock Chipping Home Page. Please send me any information you think should be added." <URL:http://bambam.cchem.berkeley.edu/~schrier/mhz.html>

- Index of reviews of Macintosh products - "The reviews listed here are from MacWorld and MacUser magazines (United States editions), starting with the January 1993 issues. You can search for any word in the name of the product or the name of the manufacturer.... This service is an exclusive feature of Internet Computer Index (ICI). The indexes of reviews are prepared by Proper Publishing for exclusive use on ICI. We hope you find them useful." <URL:http://ici.proper.com/1/mac/reviews>

- Macintosh Product Guide, by Apple Computer, Inc. - "This database contains information about Macintosh products from publishers around the world. There are currently around 14,000 products listed, with more added every month." <URL:http://www.apple.com.au/MPG/>

- Prices of used Macintoshes and PCs <URL:http://ici.proper.com/1/mac/used>

PROGRAMMING

- Developer Services and Products - "These World-Wide Web pages are created and maintained by Apple Computer for Apple Developers. They contain information provided by Apple to help you to develop your applications and solutions. All of the content found on these pages as well as on our ftp site can be found on our Developer CD Series, eWorld, and AppleLink. These products can be ordered through APDA." <URL:http://www.info.apple.com/dev/developerservices.html>

- Dylan - "Dylan is a new language developed at Apple. It is a bold new effort to create a powerful, practical tool for writing mainstream commercial applications. We believe it combines the best qualities of static languages (small, fast programs) with the best qualities of dynamic languages (rapid development, code that's easy to read, write and maintain)." <URL:http://www.cambridge.apple.com/dylan/dylan.html>

- Fairgate Technologies CodeWarrior Support - "Welcome to the officially supported WWW site for CodeWarrior support! Metroworks has generously agreed to provide us source code, examples, marketing material, and other CW-related data for your browsing

pleasure. If you like this site, make sure to let MW know you appreciate their helping to make this Internet resource possible."
<URL:http://www.iquest.com/~fairgate/cw/cw.html>

- MacPerl Q and A <URL:http://err.ethz.ch/members/neeri/macintosh/perl-qa.html>

PUBLICATIONS AND PUBLISHERS

- Info Mac digests - "This list contains the digests from the list server at the Stanford University of California. The digests have been converted to an interactive HTML format using the program digester (last change 5 December 1994) developed by Andre van der Ham who works at the The Control Laboratory of the Delft University of technology." <URL:http://dutera.et.tudelft.nl/people/vdham/info-mac/index.html>

- Information Alley - "The goal of the Information Alley is to help Apple computer users get full use of their Apple computers, peripherals, and software. Feel free to distribute The Information Alley to anyone. ... The "Information Alley" is published in two formats: Adobe Acrobat 2.0 and No Hands Common Ground. Please note you must download the Acrobat reader separately and that the Acrobat reader requires 2.5 MB of RAM and is currently incompatible with QuickDraw GX. The Common Ground reader requires 1 MB of RAM." <URL:http://www.info.apple.com/info.alley/info.alley.html>

- Mac Net Journal - "Mac Net Journal Online is a hypertext version of the downloadable Mac Net Journal. Both versions are generally the same, but evolution is bound to create a difference between the two in time. . . Check out our subscriptions & comments section (subscriptions, by the way, are free!)." <URL:http://www.netaxs.com/~aaron/>

- Mac Net Journal Home Page - "[Mac Net Journal is] produced on a single basic principle: To provide Macintosh users with a resource for helping them get the most of their Internet connection time and their downloads From the beginning, Mac Net Journal has targeted the average user as its audience. Subsequently, as a matter of policy we do not: Charge for our services Review products that cost more than $75 Those principles guide most of our actions here at Mac Net Journal. We strive to provide an informative, entertaining multimedia presentation for all to enjoy." <URL:http://www.netaxs.com/~aaron/>

- MacSense - "MacSense is a new electronic magazine (based in Canada) which seeks to provide interesting and insightful Macintosh information to an international market." <URL:http://www.achilles.net/~helix/issues/>

- MacUser - "Here, you can find selected top stories from the latest MacUser, an index of past stories, contact information and more. Articles from the latest issue of MacUser will appear here each month as the issue hits the newsstands."
<URL:http://zcias3.ziff.com/%7Emacuser/>

- MacWEEK - "Here, you can find selected top stories from the latest MacWEEK, reviews from past issues, contact information and more. Articles from the latest issue of

MacWEEK will appear here each Monday, as the issue reaches subscribers' mailboxes. MacWEEK publishes weekly with the exception of August 28, November 20 and December 18 and 25." <URL:http://zcias3.ziff.com/%7Emacweek/>

- MODEMnotes - "MODEMnotes is an electronic magazine (e-zine) for online Macintosh enthusiasts. If you've stumbled on to this page without having seen an issue of MODEMnotes, you can ftp the latest issue by clicking below. MODEMnotes is in its second year of production, and this Web page is part of an effort to extend its availability and accessibility to its readers." <URL:http://www.aloha.com/~swami/modemnotes.html>

- PowerPC News - "PowerPC News is a free electronic magazine for the Internet Community. It is published every two weeks via email, and is read by over 44,000 readers worldwide. We don't clutter up your mailbox with the whole issue, instead readers are just sent the contents page, and can retrieve the stories that they want." <URL:http://power.globalnews.com/>

- Reviews of Macintosh software, hardware, and peripherals <URL:http://rever.nmsu.edu/~elharo/faq/reviews.html>

- TidBITS - "TidBITS is a free weekly electronic publication that reports on interesting products and events in the computer industry, currently with an emphasis on the world of the Macintosh. In addition to weekly issues, we occasionally publish formal review issues and special issues focusing on a single topic. We feel that publications like TidBITS will become an important medium of exchange as the world becomes more electronically connected. TidBITS has a number of advantages over paper publications that ensure its popularity." <URL:http://www.dartmouth.edu/pages/TidBITS/TidBITS.html>

- Well Connected Mac <URL:http://rever.nmsu.edu/~elharo/faq/Macintosh.html>

- ZiffNet/Mac - "ZiffNet/Mac, also known as ZMac, is an online publishing division of Ziff Communications. It provides an online connection to MacUser and MacWEEK, and brings news, information, carefully selected and tested shareware and freeware, and exclusive original software to thousands of CompuServe, eWorld and AppleLink users. " <URL:http://www.ziff.com/~zmac/>

REVIEWS

- Apple Press Releases <URL:gopher://info.hed.apple.com/11/Apple%20Press%20Releases>

- Index of reviews of Macintosh products - "The reviews listed here are from MacWorld and MacUser magazines (United States editions), starting with the January 1993 issues. You can search for any word in the name of the product or the name of the manufacturer.... This service is an exclusive feature of Internet Computer Index (ICI).

The indexes of reviews are prepared by Proper Publishing for exclusive use on ICI. We hope you find them useful." <URL:http://ici.proper.com/1/mac/reviews>

- Reviews of Macintosh software, hardware, and peripherals <URL:http://rever.nmsu.edu/~elharo/faq/reviews.html>

SOFTWARE

- [an Apple FTP archive, Austin, TX] <URL:ftp://ftp.austin.apple.com/>

- APDA - "APDA is Apple's worldwide source for hundreds of development tools, technical resources, training products, and information for anyone interested in developing applications for Apple computer platforms. Customers receive the APDA Tools Catalog featuring all current versions of Apple and the most popular third-party development tools. APDA offers convenient payment and shipping options including site licensing." <URL:http://www.info.apple.com/dev/apda.html>

- AppleScript Archives <URL:ftp://gaea.kgs.ukans.edu/applescript/>

- Crystal Info-Mac Mirror at UALR <URL:ftp://crystal.ualr.edu/mac/info-mac/>

- Index of reviews of Macintosh products - "The reviews listed here are from MacWorld and MacUser magazines (United States editions), starting with the January 1993 issues. You can search for any word in the name of the product or the name of the manufacturer.... This service is an exclusive feature of Internet Computer Index (ICI). The indexes of reviews are prepared by Proper Publishing for exclusive use on ICI. We hope you find them useful." <URL:http://ici.proper.com/1/mac/reviews>

- Info-Mac Archive (Summex), by Stanford <URL:ftp://sumex-aim.stanford.edu/info-mac/>

- Info-Mac HyperArchive Root - "The HyperArchive page is the root of a hypertext subtree containing a mirror of the Sumex Info-Mac archives for the Macintosh. It is designed for browsing, allowing fast access to the brief descriptions at the start of .hqx files and reverse-chronological summaries. The mirror is updated, and this hypertext regenerated, nightly between 5 and 6 AM EST." <URL:http://hyperarchive.lcs.mit.edu/HyperArchive/HyperArchive.html>

- Mac ftp Sites - "This page is based on version 3.9.1 of the information maintained by Bruce Grubb (with permission). Corrections and/or additions should be sent to him (although it would be nice to notify the owner of this page also). The original list is copyrighted 1991 by Mike Gleason." <URL:http://www.realtime.net/~jones/mac-ftp-sites.html>

- Mac Software Catalog - "This service is only possible thanks to the enormous effort by the The University of Michigan's Macintosh Public Domain and Shareware Archive ...

The index is mirrored daily from London, and the catalog is updated automatically. Note that we do not mirror the contents of the archive, but simply present an interface for existing mirrors." <URL:http://web.nexor.co.uk/public/mac/archive/welcome.html>

- Macintosh Product Guide, by Apple Computer, Inc. - "This database contains information about Macintosh products from publishers around the world. There are currently around 14,000 products listed, with more added every month." <URL:http://www.apple.com.au/MPG/>

- Macintosh Word Processing <URL:http://www.astro.nwu.edu/lentz/mac/software/wp.html>

- Microlib/Mac Software Archive - "Welcome to the University of Texas at Austin Computation Center's archive of Macintosh freeware and shareware. ... NOTE: The pages that follow make heavy use of graphics. If your network connection is slow, or if you simply want the best possible performance, turn off the "Auto-Load Images" option in your WWW client. Individual graphics can then be forced to load at any time by clicking on them." <URL:http://wwwhost.cc.utexas.edu/cc/microlib-mac/main.html>

- MIDnet's view of INFO-MAC <URL:http://www.mid.net:80/INFO-MAC/>

- UBUdex - a HyperIndex to //Mac.Archive.UMich.edu/mac - "This HTML index has been running since 8 February 1994 and is "rebuilt" from the 00index.txt files of the /mac/* directories in mac.archive.umich.edu. Ideally, it will be FTP'ed nightly, HTMLized with a TPU routine (hey, I'm on a VAX/VMS system and Perl isn't part of my vocabulary) and FTP'ed into the local HTTP server to be made available. One of the future plans for this index is to take advantage of HTML forms and permit access to the same files via the mirror sites. The MacHTTP server can do forms, but MacMosaic doesn't have form support yet." <URL:http://ubu.hahnemann.edu/UBUdex/UBUdex.html>

- University of Michigan <URL:ftp://mac.archive.umich.edu/mac/>

- utexas mac archive, by Chris W. Johnson <URL:http://wwwhost.ots.utexas.edu/mac/main.html>

USENET NEWSGROUPS

- comp.sys.mac.advocacy <URL:news:comp.sys.mac.advocacy>

- comp.sys.mac.announce <URL:news:comp.sys.mac.announce >

- comp.sys.mac.apps <URL:news:comp.sys.mac.apps >

- comp.sys.mac.comm <URL:news:comp.sys.mac.comm >

- comp.sys.mac.digest <URL:news:comp.sys.mac.digest>
- comp.sys.mac.games <URL:news:comp.sys.mac.games>
- comp.sys.mac.graphics <URL:news:comp.sys.mac.graphics>
- comp.sys.mac.hardware <URL:news:comp.sys.mac.hardware>
- comp.sys.mac.misc <URL:news:comp.sys.mac.misc>
- comp.sys.mac.portables <URL:news:comp.sys.mac.portables>
- comp.sys.mac.programmer <URL:news:comp.sys.mac.programmer>
- comp.sys.mac.scitech <URL:news:comp.sys.mac.scitech>
- comp.sys.mac.system <URL:news:comp.sys.mac.system>
- comp.sys.mac.wanted <URL:news:comp.sys.mac.wanted>
- USENET News Archives - "We are currently archiving the comp.sys.mac, comp.sys.newton, comp.unix.aux, and comp.protocols.appletalk news groups. From this page you can either browse through the archive listings manually or search for specific information using our WAIS based search engine." <URL:http://www.support.apple.com/wwwdocs/usenet_archive.html>

USER GROUPS

- ACUMUG WWW Home Page - "This web is an (extremely) unofficial resource for the Arkansas College & University Macintosh Users Group. A great deal of fleshing out is still necessary. As those annoying signs in department stores say, "Please pardon our progress..." This web contains mostly Apple & Arkansas-related resources. Please mail suggestions and comments to the address at the bottom of the page." <URL:http://acumug.ualr.edu/>
- Alaska Macintosh Users Group (AAUG) - "The Alaska Macintosh Users Group (AAUG) is a non-profit 501-C3 organization incorporated in the State of Alaska. It's purpose is to provide education and assistance to it's members in the use of Macintosh Computers and related products. If you are not a member, there are several ways you can join. One is to just come to a meeting. Meetings are generally open to the public. Although you may be urged to join if you attend a meeting, the sell will be soft, a hard sell is not necessary once you become aware of all the services offered. To find out about meeting schedules you can call the Information Hotline, (907) 338-7363. However, to get yourself on the mailing list and eligible to log on to the Bulletin Board System you may wish to mail-in your membership application. An application can be found in this system." <URL:http://www.lasertone.com/AAUG.html>

- ALUG (Apple Library Users Group) Newsletter <URL:gopher://info.hed.apple.com/11/Apple%20Library%20Users%20Group/ALUG%20Newsletter>
- APDA - "APDA is Apple's worldwide source for hundreds of development tools, technical resources, training products, and information for anyone interested in developing applications for Apple computer platforms. Customers receive the APDA Tools Catalog featuring all current versions of Apple and the most popular third-party development tools. APDA offers convenient payment and shipping options including site licensing. " <URL:http://www.info.apple.com/dev/apda.html>

- Apple Business System Division - "This anonymous FTP archive (abs.apple.com) is maintained by Apple's Apple Business Systems Division Product Technical Support department. It is used - to distribute beta and experimental software to field test sites - to receive debugging data from test sites and others - to distribute a variety of software and information which may be peripherally related or completely unrelated to Apple's business, and which is offered for purely informational purposes." <URL:ftp://abs.apple.com/>

- Apple Network Manager's Association (ANMA) Home Page - "Tired of user group meetings where you answer all the questions? Feel that MacIS is too large for your 40 node Mac network? Then the Apple Network Manager's Association, or ANMA, is a user group for you. ANMA is a group of Macintosh network managers who manage 5-50,000 Macs, and who are looking for peers to share their wisdom, triumphs and failures with. Our meetings generally consist of trading network problem stories looking for solutions and a vendor product demonstration pertaining to networking. Above all, we have a good time doing it! So if this sounds like a group for you, come to one of our meetings in your area. If there isn't one in your area, start a chapter! The more ANMA chapters that there are, the more Apple and the rest of the industry will come to respect, and listen to what we have to say. " <URL:http://www.hsas.washington.edu/ANMA/ANMA.html>

- Apple Research Partnership Program Carleton University - "The ARPP (Apple Research Partnership Program) began in November 1988. It is a joint venture between Apple Canada and Carleton University, and is intended to help develop new research applications for Apple equipment. Under this co-operative arrangement, Apple funds the work of three student consultants at Carleton who work 8 hours a week, undertaking research projects on donated equipment while the University provides space, administrative support, and a staff member to oversee the program." <URL:http://arpp1.carleton.ca/>

- Apple-Internet Mailing Lists Home Page - "This is the home page for the Apple-Internet family of mailing lists. It contains pointers to information sources of use to people using Macintosh computers on the Internet." <URL:http://abs.apple.com/apple-internet/>

- Carleton Macintosh Users Group (MUG) - "ARPP sponsors meetings for Macintosh Users in the Carleton community. Each meeting focuses on a particular topic, presented by the ARPP consultants. There is usually open discussion during the meeting, as well. The ARPPs usually try to put together a disk of shareware/public-domain software for

those attending the meeting (bring a new, blank, disk or $2 to trade for a disk with the software on it)." <URL:http://arpp1.carleton.ca/arpp/mug/>

- JHMI Mac Users Group, by JHMI (Johns Hopkins Medical Institution) <URL:http://mug.welch.jhu.edu/>

- Lawrence Apple Users Group - "The Lawrence Apple Users Group serves the interests of Apple/Macintosh computer users in the Lawrence, Kansas, area. The LAUG is the premiere user group in the region, offering its members unparalleled support, informative and topical monthly programs, and a top-notch newsletter/disk every month. In addition, members are allowed extended access to the MacRocosm BBS, the only BBS in Kansas with a graphical interface. Dues for the LAUG are barely more than a dollar per month, a small price to pay for all the goodies you get in return! Sound interesting? Read on..." <URL:http://ukanaix.cc.ukans.edu/~laug/index.html>

- Mac-Supporters - "Mac-Supporters is a Mailbase distribution list. Its members are mainly involved with the support of Macintosh computers in educational establishments in the UK, though we have a few guest members from companies concerned with producing Macintosh software." <URL:http://www.sys.uea.ac.uk/MacSupporters/MacSupporters.html>

- Mac-Supporters - "Mac-Supporters is a Mailbase distribution list. Its members are mainly involved with the support of Macintosh computers in educational establishments in the UK, though we have a few guest members from companies concerned with producing Macintosh software." <URL:http://www.sys.uea.ac.uk/MacSupporters/MacSupporters.html>

- Macintosh Support Center of Western Kentucky University <URL:http://www.msc.wku.edu/Dept/MSC/Macintosh/Home.html>

- MacMAD Home Page - "MacMAD is a Macintosh user group devoted to helping each other explore the possibilities opened by the Apple Macintosh. We publish a monthly newsletter called MAC BITS, operate the MacMAD BBS (407-952-5325), and hold meetings on the first Wednesday and third Tuesday of each month. Other member benefits include a new member disk, discounts at some area dealers, and the use of club property." <URL:http://spindata.iu.net/MacMad/>

- NJMUG Web, by New Jersey Macintosh Users Group - "NJMUG is a non-profit organization to promote general knowledge of the Macintosh and compatible hardware and software for its members, to help them make better personal choices when purchasing Macintosh compatible products and to help them make better use of and enjoy the Macintosh. ... NJMUG promotes the dissemination of information about the Macintosh through general meetings, a monthly newsletter, an electronic bulletin board system (BBS), Novice, Business & Multimedia Special Interest Groups (SIGS)and a public domain shareware disk library." <URL:http://www.intac.com/njmug/njmug.html>

- QCMUG Online - "This page is a service of the Quad Cities Macintosh Users Group." <URL:http://www.infonet.net/showcase/newt/qcmug/>

- San Diego Macintosh User Group - "The San Diego Macintosh User Group is the 6th largest user group in the United States (source: InfoWorld Magazine). It involves itself with creating a learning atmosphere at it's well-attended meetings, and providing it's members with a spectrum of services tailored to every level of Mac user. ... Meetings are held the first Wednesday of the month at the Price Center Theater on the campus of U.C. San Diego. Meetings start at 6:30 p.m. with a Q&A period and start in earnest at 7 p.m. There is a small slide info session, and a shareware demonstration. The meeting then progresses to a major vendor presentation and a prize drawing near the end of the meeting." <URL:http://www.qualcomm.com/users/gnash/sdmug.html>

- South Jersey Apple/Mac Users Group (SJAUG) - "The South Jersey Apple/Mac Users Group, based in Cherry Hill, N.J., is the connection for cutting edge news and information. Full of professionals and hobbyists, its members meet on the third Friday of each month at Lenape High School in Medford, N.J., where members have access to demos, Apple II and Macintosh "clinics" to help cure your computer's ills, a disk and book library, and much more." <URL:http://www.voicenet.com/voicenet/homepages/reeltime/SJAUG.html>

- Stanford/Palo Alto Macintosh User Group (SMUG) - "SMUG (Stanford/Palo Alto Macintosh User's Group) is a group of people like you who want to get the most out of their Macintosh. SMUG members meet to listen to guest speakers and share information." <URL:http://www.svpal.org/cgi-bin/gopher2html/comm/smug/>

- ZiffNet/Mac - "ZiffNet/Mac, also known as ZMac, is an online publishing division of Ziff Communications. It provides an online connection to MacUser and MacWEEK, and brings news, information, carefully selected and tested shareware and freeware, and exclusive original software to thousands of CompuServe, eWorld and AppleLink users." <URL:http://www.ziff.com/~zmac/>