

WEB PUBLISHER'S

CONSTRUCTION KIT

Second Edition
of best-selling
**HTML Web
Publisher's
Construction
Kit**

with

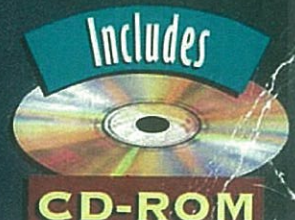
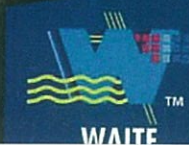
HTML 3.2

Publishing Your
Own HTML Pages on
the Internet

INCLUDES

NETMANAGE CHAMELEON,
SLIPKNOT, HOTMETAL,
HTML ASSISTANT, WINHTTPD,
LVIEW PRO, WHAM, CGI SCRIPTS
FOR UNIX AND WINDOWS,
NCSA'S HTTPD FOR UNIX,
AND MORE!

COVERS THE
LATEST VERSIONS
OF NETSCAPE
AND HTML



WEB PUBLISHER'S

CONSTRUCTION KIT

Second Edition
of best-selling
**HTML Web
Publisher's
Construction
Kit**

with

HTML 3.2

Publishing Your
Own HTML Pages on
the Internet

INCLUDES

NETMANAGE CHAMELEON,
SLIPKNOT, HOTMETAL,
HTML ASSISTANT, WINHTTPD,
LVIEW PRO, WHAM, CGI SCRIPTS
FOR UNIX AND WINDOWS,
NCSA'S HTTPD FOR UNIX,
AND MORE!

**COVERS THE
LATEST VERSIONS
OF NETSCAPE
AND HTML**



WAITE
GROUP
PRESS™



CD-ROM

DAVID FOX, TROY DOWNING

WEB PUBLISHER'S

CONSTRUCTION KIT

with HTML 3.2

Publishing Your
Own HTML Pages on
the Internet



DAVID FOX
TROY DOWNING

Publisher • Mitchell Waite
Associate Publisher • Charles Drucker

Acquisitions Manager • Jill Pisoni

Editorial Director • John Crudo
Managing Editor • John Crudo
Copy Editor • Scott Calamar, LightSpeed Publishing
Technical Editor • Miko Matsumura

Production Director • Julianne Ososke
Production Manager • Cecile Kaufman
Production Editor • Mark Nigara
Cover Design and Production • Sestina Quarequio and Karen Johnston
Cover Illustration • Rafael Lopez
Illustrations • Kristin Peterson
Production • Michele Cuneo

© 1996 by The Waite Group, Inc.®
Published by Waite Group Press™, 200 Tamal Plaza, Corte Madera, CA 94925.

Waite Group Press™ is a division of Sams Publishing.

All rights reserved. No part of this manual shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, desktop publishing, recording, or otherwise, without permission from the publisher. No patent liability is assumed with respect to the use of the information contained herein. While every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained herein.

All terms mentioned in this book that are known to be registered trademarks, trademarks, or service marks are listed below. In addition, terms suspected of being trademarks, registered trademarks, or service marks have been appropriately capitalized. Waite Group Press cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any registered trademark, trademark, or service mark.

The Waite Group is a registered trademark of The Waite Group, Inc.
Waite Group Press and The Waite Group logo are trademarks of The Waite Group, Inc.
All other product names are trademarks, registered trademarks, or service marks of their respective owners.

Printed in the United States of America
96 97 98 • 10 9 8 7 6 5 4 3 2

Fox, David, 1973-
Web publisher's construction kit with HTML 3.2 / David Fox, Troy Downing.
p. cm.
Includes index.
ISBN: 1-57169-079-4

1. HTML (Document markup language) 2. World Wide Web (Information retrieval system)

I. Downing, Troy. II. Title.
QA76.76.H94F693 1996
025.04--dc20

96-30564
CIP

TABLE OF CONTENTS

INTRODUCTION.....	XXIV
PART I CONNECTING TO THE WEB	I
1 CATCHING THE INTERNET IN A WEB.....	3
2 WEB BROWSERS.....	49
3 LYNX.....	63
4 SLIPKNOT.....	89
5 NCSA MOSAIC.....	113
6 NETSCAPE ATLAS.....	137
7 A BRIEF LOOK AT OTHER BROWSERS.....	173
PART II CREATING WEB PAGES	193
8 WHAT CAN I DO?.....	195
9 THE GAME PLAN.....	275
10 THE HYPERTEXT MARKUP LANGUAGE.....	297
11 TEXT.....	313
12 GRAPHICS.....	375
13 SOUND.....	433
14 INTERACTIVITY.....	447
15 CGI SCRIPTS.....	479
16 HTML EXTRAS.....	545
17 JAVASCRIPT.....	593
18 JAVA.....	627
19 OTHER WEB RESOURCES.....	657
20 CONVERTING, TRANSLATING, OR CHEATING.....	675
21 HTML ASSISTANT.....	695
22 HoTMetaL.....	715
PART III WEAVING A WEB OF YOUR OWN	745
23 WHERE TO PLACE YOUR HTML DOCUMENTS.....	747
24 STARTING YOUR OWN WEB SITE.....	779
A SLIP SERVICE PROVIDERS.....	833
B OTHER SOFTWARE SOURCES.....	839
C WEB HELP RESOURCES.....	849
D ABOUT THE CD.....	855
INDEX.....	859

CONTENTS

PART I CONNECTING TO THE WEB	I
I CATCHING THE INTERNET IN A WEB	3
HOW THE WEB WAS WOVEN	6
CERN: The Concept	7
NCSA: The Tool	7
NETSCAPE: The Pretty Face	7
The New Fabric	8
THE WEB'S EVOLUTIONARY EDGE	8
WORLD WIDE WEB LAYOUT	9
The Server	10
The Client Browser	10
NAVIGATING WEB PAGES	10
Hypermedia Links	11
Search Indexes	13
THE HTTP PROTOCOL	14
URLS, URLS, URLS	15
The Composition of a URL	17
ACCESSING THE WEB	21
What Computer Do I Need?	21
What About the Modem?	22
What Type of Internet Account Do I Need?	22
SLIP ACCOUNTS	23
TCP/IP	24
Connecting with Windows 95 Dial-Up Networking	25
Installing Trumpet Winsock for Windows 3.1	28
Manual Login	31
Login Scripting	32
Logging Out	34
UNIX SHELL WEB BROWSERS	34
Text: Lynx or WWW	35
Graphics: SlipKnot	36
THE INTERNET ADAPTER	36
Determining Your Unix Platform	36
Getting Your License Code	37
Getting the Software	39
Installing The Internet Adapter	40
Configuring Winsock for The Internet Adapter	40
Exiting TIA	42

TABLE OF CONTENTS

HIGH-SPEED LEASED LINES	42
Permanent SLIP	42
High-Speed Lines	43
High-Speed Hardware	44
Connection Companies	44
SHOOTING DOWN TROUBLES	45
Your Modem Won't Respond	45
The "DNS Lookup Failed" Message	46
The "Unable to Resolve Host Name" Message	46
WHAT NOW?	46
 2 WEB BROWSERS.	 49
WHAT TYPE OF SOFTWARE IS A BROWSER?	51
WHICH BROWSER SHOULD I USE?	53
HELPFUL HANDS	54
Viewers	54
External Applications	54
The Talking MIME	55
TELNETING TO A PUBLIC BROWSER	57
WWW VIA E-MAIL	59
BROWSING WITHOUT THE INTERNET ALTOGETHER	60
Stand-Alone Browsers	60
Minimizing Winsock	60
Nullsock	60
WHAT NOW?	61
 3 LYNX	 63
LESSON #1: THE LYNX'S MEOW	67
LESSON #2: NAVIGATING	68
Following Lynx Links	69
Link Info	69
LOADING A SPECIFIC URL	69
Halting a Load	70
Exiting Lynx	70
LESSON #3: THE HISTORY LIST	70
LESSON #4: SCREEN-WASHING	71
LESSON #5: PRINTING A PAGE	72
Saving	72
Mailing	72
Printing	72
LESSON #6: GETTING TO THE SOURCE	73
Saving the Source	73
Returning to the Regular View	74
LESSON #7: EDITING A PAGE	74
LESSON #8: BOOKMARKS	74
Adding a Bookmark	75
Viewing Bookmarks	75
LESSON #9: THE INDEX	76

LESSON #10: SEARCHING	76
Page Searching	76
Index Documents	77
LESSON #11: SAVING A MULTIMEDIA FILE	77
LESSON #12: FILLING OUT FORMS	78
The Form of a Form	78
User Validation	80
LESSON #13: USENET NEWS POSTING	80
Selecting a Group	80
Browsing Articles	81
Reading an Article	81
Posting an Article	82
LESSON #14: SENDING E-MAIL	82
LESSON #15: THE LYNX OPTIONS MENU	83
Editor	85
Bookmark File	85
Personal Mail Address	85
Searching Type	85
VI Keys	86
Emacs Keys	86
Keypad as Arrows or Numbered Links	86
User Mode	86
Local Execution Scripts or Links	87
WHAT NOW?	87

4 SLIPKNOT	89
TYING THE KNOT: INSTALLING	92
HOW SLIPKNOT SLIPS	93
SETTING UP THE TERMINAL	93
Setting Up the Host	93
Setting Up Communications	97
Setting Up the Terminal	97
LESSON #1: CONNECTING	98
Logging In	98
Logging Out	98
LESSON #2: NAVIGATING	99
Following a Link	100
Snagging a Particular URL	101
Stopping a Load	101
Loading a Local File	101
Retrieve Again!	102
LESSON #3: TURNING OFF IMAGES	102
LESSON #4: RETRIEVING MULTIMEDIA FILES	102
Viewing	103
Saving	103
FTP Files	103
LESSON #5: YOU'RE HISTORY	103
Cleaning Up	104

TABLE OF CONTENTS

Cleaning Up for Good	104
LESSON #6: THE PERMANENT SAVE	105
Local Keepers: Saving a Complete Web Page	105
Saving a Page	106
Remote Keepers: Bookmarks	106
Placing a Bookmark	106
LESSON #7: CHECKING OUT THE SOURCE	107
LESSON #8: CONFIGURING VIEWERS	107
Editing LView or WPLAN	107
Adding a New Viewer	108
LESSON #9: MAKING IT LOOK PRETTY	109
Setting Colors	109
Customizing Fonts	109
Background	110
LESSON #10: SWITCHING TO THE TERMINAL	112
LESSON #11: UPGRADING	112
WHAT NOW?	112
5 NCSA MOSIAC	114
INSTALLING MOSAIC	116
Installing WIN32S-OLE	116
Installing the Mosaic Browser	116
LESSON #1: NAVIGATING USING THE VIEW WINDOW	117
Following Hyperlinks	118
Moving to a Specific URL	119
The Right Mouse Button	119
Halting: The Secret Button	120
LESSON #2: IGNORING IMAGES	121
LESSON #3: MULTIMEDIA FILES	122
LESSON #4: HISTORY LIST	122
LESSON #5: THE HOTLIST	122
Getting Hot: Quickly Creating a Hot Item	124
Customizing Your Mosaic Menu	124
The Quicklist	126
Sharing Hotlists	126
LESSON #6: KEEPING THE PAGE	126
Save It	127
Load It to Disk	127
Print It	127
LESSON #7: SEARCHING	128
LESSON #8: THE SOURCE OF IT ALL	128
LESSON #9: FORMS	128
LESSON #10: ANNOTATING	131
LESSON #11: GOOD NEWS AND BAD NEWS	131
LESSON #12: FTP	132
LESSON #13: E-MAIL	133
Sending Mail	133
Mail to Developers	133
LESSON #14: PRESENTATION MODE	133

LESSON #15: VIEWERS AND OTHER HELPERS	134
LESSON #16: OTHER MOSAIC OPTIONS	135
The Fonts You Want	135
Colors	135
Home Sweet Home	136
WHAT NOW?	136
6 NETSCAPE ATLAS	137
INSTALLING NETSCAPE	142
LESSON #1: SCAPING THE NET	142
Following Hyperlinks	144
Framed!	145
Moving to a Specific URL	145
Loading Indicators	145
Stop It!	145
The Web Page Appears	146
Safe and Secure	146
LESSON #2: GRAPHICLESS: NO AUTO LOADING	146
Loading Later	147
Hither and Dither	148
LESSON #3: RETRIEVING MULTIMEDIA FILES	148
LESSON #4: ANCIENT HISTORY	149
LESSON #5: BOOKMARKS	150
Dealing with Bookmarks	150
Adding or Deleting Bookmarks	151
Organizing Bookmarks	152
The Bookmark Files	154
LESSON #6: KEEPING A WEB PAGE	154
Saving to Disk	155
Loading	155
Printing	155
Mail Document	156
LESSON #7: SEARCHING A PAGE	156
LESSON #8: GETTING TO THE SOURCE	157
LESSON #9: FILLING IN FORMS	158
Filling 'er Out	159
User Validation	160
LESSON #10: THE NEWS	160
Setting Up a News Server	160
Getting the News	161
Subscribing to Newsgroups	161
Browsing Articles	161
Reading an Article	163
Posting an Article	164
LESSON #11: E-MAILING	165
Opening The Mail Window	166
Sending Mail	166
Receiving Mail	167
LESSON #12: GRABBING FTP FILES	168

UPLOADING AN FTP FILE	168
LESSON #13: CONFIGURING VIEWERS	169
LESSON #14: CUSTOMIZING YOUR NETSCAPE EXPERIENCE	170
You've Got the Look	171
There's No Place Like Home	171
WHAT NOW?	172
7 A BRIEF LOOK AT OTHER BROWSERS	173
SPRY AIR MOSAIC	176
Features	176
Getting It	177
WINWEB	177
Features	178
Getting It	179
CELLO	180
Features	180
Getting It	181
SPYGLASS ENHANCED MOSAIC	181
Features	181
Getting It	181
NETCRUISER'S WEB BROWSER	182
Features	182
Getting It	182
QUARTERDECK MOSAIC 2.0	183
Features	183
Getting It	183
MICROSOFT'S INTERNET EXPLORER	183
Features	183
Getting It	184
NETMANAGE INTERNET CHAMELEON	184
INSTALLING THE CHAMELEON	185
SIGNING UP FOR AN INTERNET PROVIDER ACCOUNT	185
REGISTERING THE CHAMELEON SOFTWARE	187
CONNECTING TO THE INTERNET	188
WEB BROWSING with WebSurfer	190
OTHER INTERNET TOOLS	192
WHAT NOW?	192
PART II CREATING WEB PAGES	193
8 WHAT CAN I DO?	195
NAVIGATE: PLACES TO START	198
Best of the Web	198
Global Network Navigator	198
Einnet Galaxy	198
Yahoo	200
INFORM: TERRIFIC TEXT	201
Manuals: NCSA Mosaic	201
Helpful Information: OncoLink	202

A New Type of Text	202
HOME-MAKING: NO PLACE LIKE HOME PAGES	203
Corporate Pages	203
University Home Pages	205
Government Home Pages	207
Personal Home Pages	208
TEACH: EDUCATION	210
Geography Skills	210
Eric	211
Britannica Online	211
ENTERTAIN: GROOVY GRAPHICS	212
TV: Interactive TV Index	212
Movies: Buena Vista	212
Music: Underground Music Archive	212
Art: ArtSource	214
Sports: Internet Baseball Information Center	214
PUBLISH: PAPERS, BOOKS, MAGAZINES, AND MORE	215
Academic Papers	215
Multimedia Magazines	216
'Zines	217
Books	218
SHOWCASE: PUTTING YOUR TALENTS ONLINE	218
SELL: TAKING CARE OF BUSINESS	220
Stock Quotes	220
Networking over the Net	221
Job Searching	221
BUY: SHOPPING	221
The Internet Shopping Network	222
Direct Ordering	222
SPEAK FREE: POLITICS	224
PeaceNet	224
FedWorld	224
OBSERVE: THE NEWS	225
Headline News	225
Weather	226
INTERACT: FILL-IN FORMS	226
Personal Ads	226
Databases	226
Security	227
FIND: MAPS	227
Interactive Weather Map	227
Xerox Map Server	228
Map of Mars	228
TOUR: MUSEUMS	230
Le WebLouvre	230
Exploratorium	231
Museum of Paleontology	231
VISIT: VIRTUAL PLACES	232
San Diego	232

TABLE OF CONTENTS

Paris	233
EXPO Ticket Office	234
Downtown Anywhere	235
Internet Town Hall	236
BROWSE: LIBRARIES	236
SURPRISE: OTHER NEAT GIMMICKS	238
LabCam	238
The Cyrano Server	239
Lite-Brite	239
The Virtual Frog	240
Say	240
Bianca's Smut Shack	241
A Puzzler	242
THE LATEST AND GREATEST	243
WHAT NOW?	272

9 THE GAME PLAN 275

ENTER THE WEBMASTER	277
LESSON #1: WHO'S YOUR AUDIENCE?	278
Does the Audience Exist?	278
What's Already Out There?	279
Focusing In	279
LESSON #2: WHAT'S YOUR OBJECTIVE?	281
Sound and Video	282
Talking Back	282
Security	282
LESSON #3: THE HARD SELL	282
Astound!	283
Inform!	284
Entertain!	284
LESSON #4: OVERALL DESIGN	285
Point of Entry	285
The Home Page	285
Permanence	287
LESSON #5: PROBLEM PAGES	288
Art Bombardment	288
Graphic Gluttony	288
Multimedia Moderation	289
Textual Terseness	289
Java Jabbering	289
LESSON #6: GETTING THE RIGHT LOOK	290
Consistent Controls	291
The Look Should Fit the Structure	292
Cross-Browser Dressing	292
LESSON #7: LINKS TO OTHER PAGES	293
Stale Links	293
Senseless Links	293
LESSON #8: STUDYING THE STATS	294

WHAT NOW?	295
10 THE HYPERTEXT MARKUP LANGUAGE	297
WHERE TO START	300
Steal This Text	300
Convert This Text	302
HTML HISTORY	302
HTML FUNCTIONALITY	303
LIFE, LIBERTY, AND GOOD HTML	304
LESSON #1: HTML ELEMENTS	306
Specifying HTML	307
The Header	307
The Body	309
THE FUTURE OF HTML	310
WHAT NOW?	310
II TEXT	313
DESIGN FUNDAMENTALS	316
Consistency	317
Signing Your Page	317
LESSON #1: SPECIAL CHARACTERS	317
Punctuation	318
Enhanced Characters	318
Other Characters	320
LESSON #2: SECTION HEADINGS	322
Style: Drafting Good Headings	323
Aligning Headings	324
LESSON #3: PARAGRAPHS	324
LESSON #4: LINE BREAKS	326
No Break	327
Word Break	327
LESSON #5: TEXT STYLE	327
Physical Styles	328
Logical Styles	329
LESSON #6: SECTION STYLE	330
Preformatted Text	330
Blockquotes	332
Addresses	333
Styles to Watch Out For	333
LESSON #7: FONT SIZE AND COLOR	335
Relative Font Size	336
The Base Font	336
Making It Easier	336
Font Color	336
LESSON #8: CENTERING	337
LESSON #9: COMMENTS	337
LESSON #10: LISTS	338
Ordered Lists	339
Unordered Lists	341

TABLE OF CONTENTS

Customizing Lists	343
Definition Lists	343
Plain Lists	345
Menu Lists	346
Directory Lists	346
LESSON #11: HORIZONTAL LINES	346
In Thickness and in Health	347
Setting the Line's Width	347
Aligning Your Lines	347
Using a Solid Bar	348
LESSON #12: TABLES	349
Text Tables	349
Graphical Tables	349
HTML+ Tables	350
LESSON #13: HYPERTEXT	356
The Anchor Element	356
Links: HREF	358
Anchors: The Name Attribute	360
Title	363
Defining Relationships	364
Methods	366
LESSON #14: RELATIONSHIPS AND HYPERPATHS	366
Using Link	366
Nodes	367
Path	368
LESSON #15: MATHEMATICAL EQUATIONS	368
LESSON #16: DYNAMIC DOCUMENTS: PUSH OR PULL?	370
Server Push	370
Client Pull	370
LESSON #17: MARQUEES DE HAPPY	371
LESSON #18: POLISHING	373
WHAT NOW?	373

12 GRAPHICS 375

GRAPHIC FORMATS	378
GIF	378
JPEG	379
GRAPHIC VIEWERS	379
LView	379
WinGIF	381
LESSON #1: INLINED IMAGES	382
The IMG Command	383
Distant Images	383
Hyperimages	383
Crossing the Border	384
LESSON #2: ALIGNMENT	384
Absolute Alignment	385
Floating Alignment	386
Mind in the Gutter	387

Clearing Space After Line Breaks	387
LESSON #3: TEXT ALTERNATIVES	388
LESSON #4: IMAGE SIZING	389
Auto-Scaling	390
Percentage Auto-Scaling	391
LESSON #5: TRANSPARENT GIFS	392
STEP 1: Isolate the Background	392
STEP 2: Selecting a Background Color	394
STEP 3: Making the Background Transparent	395
LESSON #6: CREATING SPACERS	399
LESSON #7: PERKS, GIZMOS, AND FRILLS	400
Bullets	400
Buttons / Icons	402
Bars	403
LESSON #8: MAKING GRAPHICS HUMBLE	403
Shrink Me	404
Thumbnails	406
The Flip Trick	407
LESSON #9: INTERLACED GIFS AND PROGRESSIVE JPEGs	408
Interlacing a GIF Image	408
Progressive JPEGs	409
LESSON #10: CLICKABLE IMAGEMAPS	410
STEP 1: Create the Image	411
STEP 2: Create the Map	411
MapEdit 1.1.2	413
STEP 3: Write the HTML	416
STEP 4: Set Up the Server	417
STEP 5: The Text-Based Index	418
LESSON #11: CLIENT-SIDE IMAGE MAPS	418
LESSON #12: WHAT'S YOUR BACKGROUND?	420
Background Color	421
Foreground Color	422
Getting Backgrounds	422
LESSON #13: EMBEDDING	423
THE SILVER SCREEN	423
Lights, Camera... Action?	424
QuickTime	424
MPEGPLAY	425
GRABBING GRAPHICS	425
Playing Picasso	425
Capturing	426
Image Hunting	427
WHAT NOW?	431
13 SOUND	433
HEARING AIDS	436
Installing SPEAK.EXE	437
Configuring SPEAKER.DRV	437
JAMMIN'	438

WPLANY	438
WHAM	439
LESSON #1: HYPER SOUND	440
LESSON #2: AMBIENT SOUND	442
SOUNDING OFF	443
Audio Formats	443
Recording It Yourself	444
WHAM	444
Searching For That Right Sound	445
WHAT NOW?	445
14 INTERACTIVITY	447
HOW A FORM WORKS	450
LESSON #1: A FORM IS BORN	452
Action	452
Method	452
ENCTYPE	453
TARGET	453
LESSON #2: INPUT FIELDS	453
TYPE	454
NAME	455
DISABLED	456
MIN/MAX	456
LESSON #3: TEXT FIELDS	456
Text Field Variations	457
VALUE	458
SIZE	458
MAXLENGTH	458
LESSON #4: CHECKBOXES OR RADIO BUTTONS	458
VALUE	459
Checkboxes	459
Radio Buttons	460
Prechecked	461
LESSON #5: THE RESET AND SUBMIT BUTTONS	461
Customizing the Label	462
Using Icons	462
LESSON #6: TEXTAREAS	462
Size	463
WRAP	463
LESSON #7: SELECTION LISTS	464
SIZE	465
Multiple Selections	466
Options Attributes	466
LESSON #8: PUTTING IT ALL TOGETHER	467
A Basic Form	467
A Sample Message Form	468
LESSON #9: UPLOADING A FILE	470
LESSON #10: BUTTONS	471
LESSON #11: INCLUDING HIDDEN TEXT	471

LESSON #12: SEARCH INDEXES	472
ISINDEX	472
HREF	473
PROMPT	473
Book-Like Index Documents	473
LESSON #13: TESTING IT OUT	474
LESSON #14: WHAT THE DATA LOOKS LIKE	474
Search Indexes	475
The GET Method	475
POST	476
THE FUTURE OF FORMS	476
WHAT NOW?	477
15 CGI SCRIPTS	479
LESSON #1: WHAT IS CGI?	482
What Can I Do with a CGI Script?	484
LESSON #2: THE STANDARD CGI SCRIPTS	485
LESSON #3: HOW TO USE PRE-EXISTING SCRIPTS	488
Constructing a CGI URL	488
Specifying a Script Within a Form	489
Specifying a Script from the "Open URL" Interface	489
LESSON #4: HOW TO USE A SCRIPT TO ACCESS OTHER APPLICATIONS	490
LESSON #5: WRITING YOUR OWN SCRIPTS	493
Writing a Simple Script	493
Using Environment Variables in Scripts	497
Location and Status Headers	499
Security with CGI Scripts	501
LESSON #6: HANDLING FORM DATA	503
What Does Form Data Look Like?	503
HTTP Cookies	520
LESSON #7: SAMPLE SCRIPTS FOR UNIX, WINDOWS, AND MACINTOSH SERVERS	524
Unix	524
Easy Counter	530
Server-Push	533
Perl	535
DOS/Windows Scripts	539
Macintosh Scripts	541
WHAT NOW?	544
16 HTML EXTRAS	545
LESSON #1: INLINED VIDEO WITH INTERNET EXPLORER	548
When to Play It	549
Should There Be a Control Strip?	549
LOOP-D-LOOP	550
LESSON #2: FRAMING WEB PAGES	550
The Frame Document	551
Supporting The Frame-Impaired	558
LESSON #3: TARGETING A FRAME	559
Framing Yourself	560
The Acme Catalog	561

LESSON #4: CHARGING UP NETSCAPE PLUG-INS	564
How to Install a Plug-in	568
How to Embed a Live Object	569
LESSON #5: LIVEMEDIA: STREAMED VIDEO AND AUDIO	569
Live Audio	570
Live Video	571
LESSON #6: OTHER STREAMED VIDEO AND AUDIO PLUG-INS	572
Voxware	572
VDOLive	575
PreVU	576
RealAudio	576
LESSON #7: LIVE3D AND THE VIRTUAL-REALITY MODELING LANGUAGE	578
Walking Through 3D Worlds	579
Creating Your Own VRML Files	581
LESSON#8: USING SHOCKWAVE	589
Installing Shockwave	589
Afterburning	589
Simply Shocking	591
WHAT NOW?	592
17 JAVASCRIPT	593
WHAT JAVASCRIPT DOES	596
WHAT JAVASCRIPT DOESN'T DO	597
INSERTING JAVASCRIPT INTO YOUR WEB PAGE	597
THE LANGUAGE	600
Objects	600
Functions	601
Methods	602
Variables	603
Event Handlers	603
JAVASCRIPT EXAMPLES	606
LUSCIOUS LINKS	606
After That Mouse!	606
You're History	607
FANTASTIC FORMS	609
Making Forms Smart	609
Processing Forms	610
Error-Checking	610
Interest Calculator Example	615
Beautifying The Page	616
Scrolling Marquee	616
BUILT-IN OBJECTS AND FUNCTIONS	620
The String Object	621
The Math Object	622
The Date Object	622
Built-in Functions	625
WHAT NOW?	626
18 JAVA	627
JAVA'S JOLT	630

THE BIG APPLET	630
The ABCs of Alpha and Beta	632
An Applet a Day... ..	632
PLACES TO GET APPLETs	632
Applets from the Java Products Group	633
USING EXISTING APPLETs	636
Some Applets to Bite Into	637
Animation	638
Sound	642
Blinking/Nervous Text	642
Scrolling Marquee	643
Image Maps	646
CREATING YOUR OWN APPLT	649
The Java Developer's Kit	650
Writing The Applet	651
Compiling It	653
Using It	653
Java Development Packages	654
WHAT NOW?	654
19 OTHER WEB RESOURCES	657
LESSON #1: GOPHER	660
LESSON #2: NET NEWS	662
LESSON #3: E-MAIL	665
LESSON #4: WIDE AREA INFORMATION SERVICE	667
LESSON #5: FILE TRANSFER PROTOCOL	668
What Happens to Downloaded Files?	669
How Can I Send a File Using FTP?	671
LESSON #6: TELNET	671
What Does Telnet Allow Me to Do?	671
What Happens When I Initiate a Telnet Session?	671
LESSON #7: ACCESSING LOCAL FILES	672
LESSON #8: URL SCHEMES	673
WHAT NOW?	673
20 COVERTING, TRANSLATING, OR CHEATING	675
WORDPERFECT TO HTML	679
Installing It	679
Writing Your Document	680
Converting	681
MICROSOFT WORD TO HTML	681
ANT_HTML	681
CU_HTML	687
The Internet Assistant	690
Other Tools	691
Netscape Navigator Gold	691
BOOKMARKS TO HTML	692
Lynx	692
Mosaic	693
Netscape	693

Cello	693
WHAT NOW?	693
21 HTML ASSISTANT.....	695
INSTALLING THE ASSISTANT	698
LESSON #1: FROM TEXT TO HTML	698
LESSON #2: CREATING FROM SCRATCH	700
LESSON #3: THE TOOLBAR	700
Selection Tools	701
Insertion Tools	703
LESSON #4: INSERTING HYPERLINKS	703
External Links	704
Internal Links	705
Recycling URLs	706
LESSON #5: IMAGES	706
LESSON #6: USER TOOLS	706
Creating a User Tool	707
Entering Enter	708
LESSON #7: FINDING OR REPLACING	708
Finding	708
Replacing	708
LESSON #8: TOOLING WITH TOOLS	708
Undo	709
Auto Repeat	709
Putting an HTML Document on Hold	709
LESSON #9: TESTING 'ER OUT	709
Specifying a Browser	710
The Final Test	710
LESSON #10: PRINTING	710
LESSON #11: URL FILES	710
Grabbing URLs	711
Editing URLs	711
Combining URL Files	712
Saving It	712
LESSON #12: CONVERTING A BOOKMARK, HOTLIST, OR URL FILE TO HTML	712
WHAT NOW?	713
22 HoTMetaL.....	715
INSTALLING IT	718
LESSON #1: STARTING WITH A TEXT FILE	719
LESSON #2: STARTING FROM SCRATCH	719
Templates	720
Saving	721
LESSON #3: MARKING IT UP	721
Inserting	722
Surrounding	725
Changing	725
Deleting	726
The Smart List	726

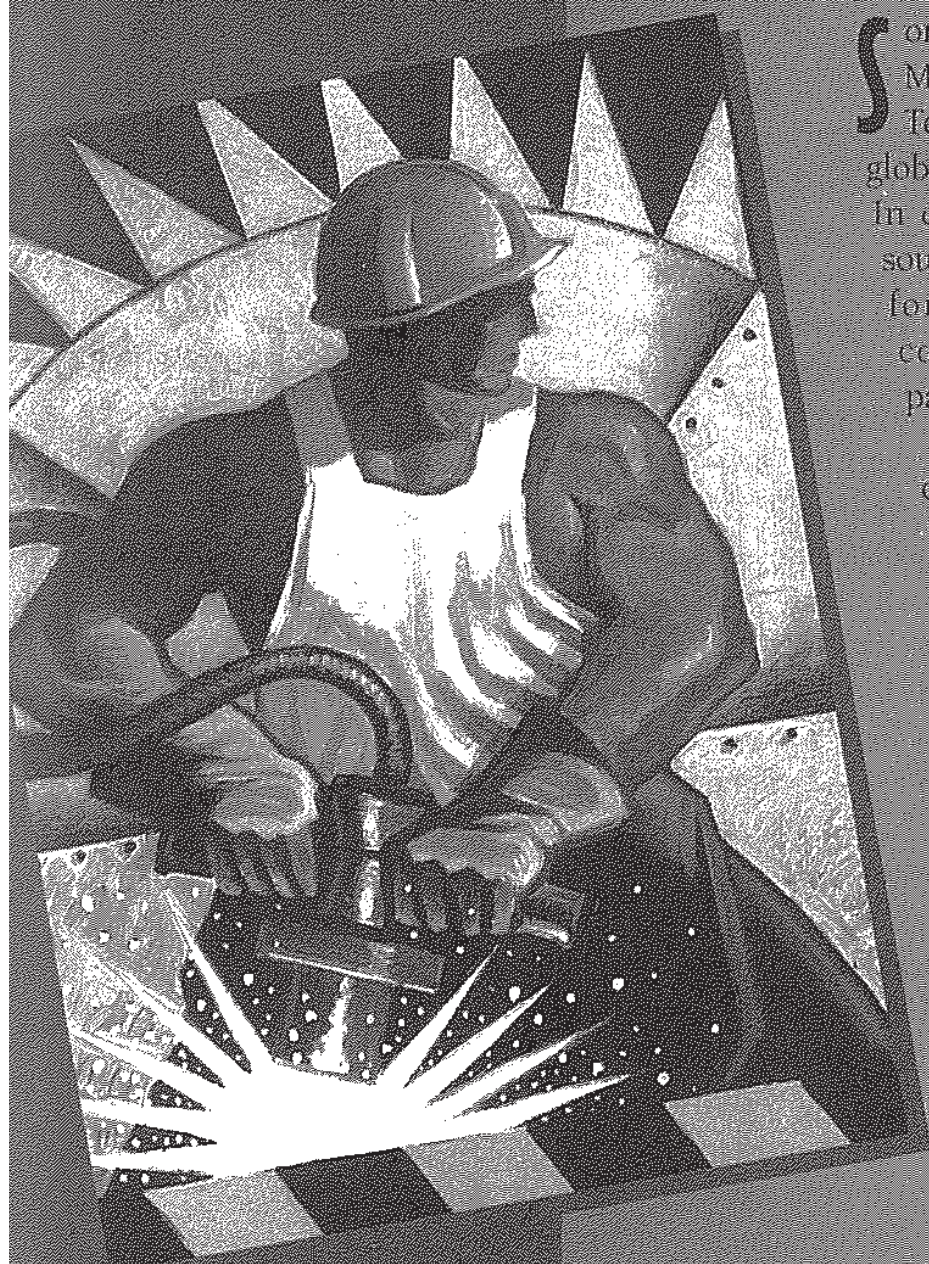
Pinning	726
LESSON #4: SPECIAL CHARACTERS	727
LESSON #5: WORKING WITH HYPERLINKS	728
Creating External Links	728
Creating Internal Links	729
LESSON #6: WORKING WITH GRAPHICS	730
Image Attributes	730
Viewing Images	731
LESSON #7: WORKING WITH FORMS	731
Input Field	731
TEXTAREA	733
Selection Lists	734
LESSON #8: FINDING SOMETHING	735
The Search Term	735
Search Options	736
Searching	737
LESSON #9: BREAKING THE RULES	737
LESSON #10: STYLES	737
Character	738
Separation	739
Load Styles	739
LESSON #11: TESTING IT OUT	740
Hiding Tags	740
Publishing	741
Previewing	741
LESSON #12: THE HTML BACKBONE	741
The Context Window	742
The Structure Window	742
WHAT NOW?	743

PART III WEAVING A WEB OF YOUR OWN. 745

23 WHERE TO PLACE YOUR HTML DOCUMENTS	747
LESSON #1: TROUBLESHOOTING YOUR HTML DOCUMENTS	750
LESSON #2: HTML DOCUMENT ANALYSIS SERVICES	752
LESSON #3: FREE WEB POSTING SPOTS	755
LESSON #4: WEB SPACE FOR SALE	757
LESSON #5: POSTING	769
LESSON #6: UPLOADING DOCUMENTS	771
Uploading via FTP	771
Transferring Files with a Modem	772
LESSON #7: INSTALLING PAGES ON AN EXISTING WEB SERVER	772
LESSON #8: ANNOUNCING YOUR WEB PAGES	775
WHAT NOW?	777
24 STARTING YOUR OWN WEB SITE	779
LESSON #1: YOUR WEB SERVER	782
LESSON #2: WEB SERVER SOFTWARE	782
LESSON #3: SERVER SOFTWARE SOURCES	785

TABLE OF CONTENTS

LESSON #4: TIPS AND CAUTIONS	786
Type of Machine	786
Memory	787
LESSON #5: SECURITY!	788
Transmission Security and Data Encryption	789
Pretty Good Privacy Encryption	791
Access Control, Passwords, and Other Filters	793
LESSON #6: HARDWARE	800
LESSON #7: INSTALLING A WEB SERVER	801
Unix Server Installation	801
Netscape's Latest Servers	812
Windows 95 And NT Servers	815
Windows 3.1 Server Installation	816
Macintosh Server Installation	826
LESSON #8: HYPERMEDIA GATEWAYS	831
WHAT NOW?	831
A SLIP SERVICE PROVIDERS	833
B OTHER SOFTWARE SOURCES	839
C WEB HELP RESOURCES	849
D ABOUT THE CD	855
INDEX	859



Some call it W3 for short, or just WWW. Most folks refer to it as "the Web." Techies may go so far as to call it a global-network hypermedia retrieval tool. In other words, it can shoot graphics, sounds, databases, software, movies, and formatted text across the world, combining everything on one Web page—that is, one screen.

In theory, the World Wide Web consists of the entire 20-million-person Internet. It includes every gopher menu site, FTP file library, telnet public-access computer, Usenet newsgroup, and WAIS database. And when any other hot Internet resources come along, it'll probably include them, too. All in all, the Web is the largest body of computerized data in the world.

The Web is called the Web, however, because it links together these pages and resources in a way that lets you

reach any point on it from any other. Imagine strands of silk, stretching from every spot to dozens or hundreds of other spots. With tens of thousands of potential sites, that's a lot of virtual silk. If you tried to draw what all this would look like, you'd probably come up with—that's right—an enormous psychedelic web. That sounds like a tangled mess, but you can navigate it through an easy point-and-click interface known as *hypertext*. We'll talk more about hypertext later in this chapter.

But the true magic of the Web isn't its size or its multimedia snazziness. It isn't the fact that individual Web pages can be seen by people worldwide, no matter what computer or operating system they use. The true magic of the Web is that anybody can do it. Anybody can weave Web pages—complete with hypertext and everything else—and spin them onto the Web for the world to see. And doing it is easier than you might think.

HOW THE WEB WAS WOVEN

The *Internet*—the interconnected network of networks—came into being during the chilliest part of the cold war. The Defense Department linked together computers around the world to form something known as Arpanet. Slowly this network melted into the civilian sector. Looking to expand its communication power, a university or major corporation would tie into Arpanet, and students and employees were soon linked to the minds and data of other students and employees. The Net spread like a virus. By 1993, the Internet had nearly 15 million users in almost 70 countries.

The Internet rapidly turned into a global city, complete with virtual public libraries, storefronts, business offices, art galleries, museums, and even bars. There were some private sites, locked up tight, but most were open to anyone who knew where to look. Libraries offered card catalogs. Universities offered all sorts of academic papers. Software companies offered support files. There were places to post messages and places to chat. This was great stuff. It added up to a lot of text.

All this text was hard to sort through, and, frankly, it had a tendency to get boring. If you wanted a particular article, first you had to find what and where it was, then you had to figure out a way to get at it. You might have to use a newsreader program to grab it. Or maybe you'd need to log in to someone else's computer by typing something like *telnet somewhere.com*. Or maybe you would have to use the file transfer protocol (FTP) to download the file.

Utilities like gopher came into being. By selecting from a jungle of menus, you could retrieve files and articles, or log in to remote sites. There were a few images and sounds online, but you needed specialized hardware

and software to take advantage of them. Gopher was a step better than the barrage of Internet commands, but still far from perfect. Different documents were hard to sift through. It didn't feel like a newspaper or magazine or even a TV show, it felt like a blizzard of text and photos and files swirling around you.

CERN: The Concept

In 1989, a bunch of High Energy Physics dudes at the Conseil Européen pour la Recherche Nucléaire (CERN)—led by an ambitious Tim Berners-Lee—developed a way to share all sorts of information over the Internet. CERN was an international organization, and members across the world wanted a way to communicate quickly and consistently. The World Wide Web was born. Using the Web, a team of scientists in Geneva using Sun computers and a Unix operating system could read about particle propulsion tests stored on a PC in Dallas, Texas. Everything could be hassle-free.

NCSA: The Tool

But it wasn't quite hassle-free until a few years later, when the National Center for Supercomputing Applications (NCSA) developed an attractive way to look at this universe of information. One lovely summer day in 1993, NCSA officially released *Mosaic* to the public. The Mosaic software could consistently and elegantly display Web graphics, text styles, hypertext, and other files on the same page like a sort of, yup, mosaic. The Mosaic Web browser was released as noncommercial software for the benefit and use of the entire Internet community. Versions were created for all major platforms—Windows, Macintosh, Unix, Sun, and Silicon Graphics workstations. Before Mosaic, the Web was like a bunch of TV signals without the TV.

NETSCAPE: The Pretty Face

Netscape took Mosaic and went ahead to the next level with it. First they made it faster so that graphics loaded in sooner and crisper. They cleaned up some of the menus, making it much easier to use bookmarks and to set up helper applications. But most important, they added special extensions to HTML that gave people the power to lay out their Web pages with a professional magazine-like look. Netscape is still going strong, adding new extensions with every new release of its Navigator browser. This book covers all the latest Netscape additions.

The New Fabric

Much like the Internet itself, the Web caught fire without really trying. New Web users put their own pages of information over the Web. This new variety of information attracted more Web readers, who in turn put more information on the Web, which in turn . . . you get the idea. Before 1993, the Web had about 100 official sites. After Mosaic came out, Web traffic steadily increased. A year later, the Web had nearly 10,000 sites.

As fast PCs and modems became less expensive, the everyday computer user could transfer large files such as graphics, movies, and sound. Soon these media found their way onto the Web. The old fascinating but frustrating Internet suddenly became streamlined, organized, and really, really neat to look at.

And this isn't ancient history. The most exciting aspect of all this is that it *just* happened. The Web is in its infancy right now and it's growing fast. It seems likely that it will serve as the backbone for tomorrow's video teleconferencing, interactive television, and shared virtual reality. You can be a cyberspace pioneer by weaving your own little Web corner today.

THE WEB'S EVOLUTIONARY EDGE

The World Wide Web isn't the first tool for fishing data out of the Internet, but it's the best so far—and it will keep on evolving. Here are the features that make the Web the global superhighway hypermedia information retrieval system of choice:

HTML The Web is simple. Once you're online with a program like Mosaic, all you need to access a world of data is a pair of eyes, a mouse, and a healthy curiosity.

HTML Documents look great. Instead of plowing through endless text files, you'll find a mix of fonts, styles, and layouts. You'll even get graphics and text together on the same screen, presented seamlessly.

HTML The Web keeps growing. It can deal with gopher, FTP, telnet, and other Internet tools now, and built-in features will let it adapt as new standards, protocols, data types, and languages develop.

HTML The Web can be accessed anytime, anywhere, any which way. You can surf the Web from most any type of computer or operating system, and from any country and any part of the Internet.

HTML The Web lets you go anywhere even when you don't know where you're going! With hypermedia links, each Web page points to lots of others; you can skip around the Internet randomly, pursuing any topic that interests you from site to site.

HTML The Web has no owner. With the right equipment and software, anybody can read the Web or distribute data on it—even you.

All this sounds great, but how is it possible? Read on . . .

WORLD WIDE WEB LAYOUT

Most any type of network consists of computers known as *clients* and *servers*. The server is a powerful computer that does all the grunt work, holding most of the files, memory, data, and resources. Individual PCs usually act as clients. Though weaker and dumber, clients exploit the mighty server.

The Web can be thought of as a vast network of data. As such, it has its own servers and clients, as illustrated in Figure 1-1. The servers hold all the

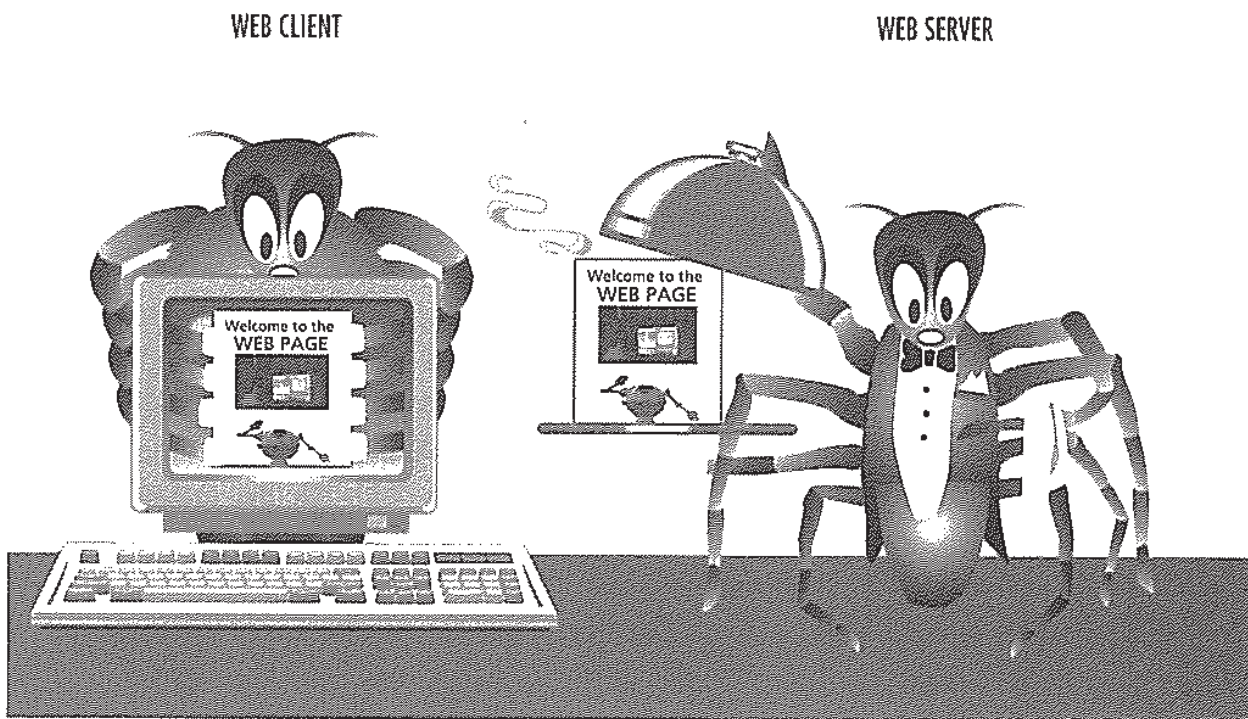


Figure 1-1 Web servers distribute Web pages; Web clients bring them to users' computer screens

Web data and dish it out, much like publishers or TV stations. The clients can access the Web data itself. Client programs like Mosaic are called *Web browsers*, since you use them to browse through the Web.

To access the Web, you need to install a browser on your computer, turning it into a Web client. Your browser then uses the Internet to connect to another computer running the Web server software. The server serves up whichever Web page your client asks for. Your client then slaps the Web document onto your screen. The Web is nothing more than a tangle of these client-server interactions, all happening thousands of times a second.

The Server

A Web server can be much smaller and simpler than you may expect. Most major Internet providers have their own Web servers. If you have a permanent Internet connection, you can even install a server on your own machine.

You can learn how to install and configure a Web server in Chapter 21, *Starting Your Own Web Site*. Chapter 20, *Where To Place Your HTML Documents*, discusses existing Web servers who can publish your Web pages for you.

The Client Browser

Since the advent of Mosaic, dozens of browsers have hit the market. Some are commercial, others are perfectly free. Each browser has its own perks, bells, and whistles. The next chapter discusses browsers in general. Chapter 5, *NCSA Mosaic*, is a hands-on tutorial of the Mosaic software. Other chapters discuss the popular Netscape browser, the Lynx text-only browser, and the revolutionary SlipKnot browser.

NAVIGATING WEB PAGES

A Web page can be anything you imagine it to be: a hypertext document, graphics, sound, animation, tabular data, software files, a discussion group, a search index, a specialized program, and more. Most every Web site has a standard, default page known as the *home page*. This page acts as a sort of front door. Anybody who accesses a site usually starts off at that site's home page.

Many people create personal home pages for themselves, including things like their photograph, current projects, addresses, hobbies, even complete resumes. These *author pages* stick a human face onto the Web.

Most browsers use uniform fonts and styles. This means every Web page will have a similar layout and look. Web pages can contain everything

from academic articles to shopping catalogs to art galleries to travel guides. Chapter 8, What Can I Do?, is a photo-essay tour of some of the world's most popular Web pages. Some pages will haunt you, others will teach you, most will entertain you. In general, though, there are only two ways you can interact with a Web page:

HTML Click on a *hypermedia link* to access another page, program, or Internet service.

HTML Search for a keyword within a *search index* or fill in input fields in a form.

Hypermedia Links

You're cozy in your favorite armchair reading a really good book about the Civil War when you come across a footnote that has something to do with the death of General Lee. You turn to the back of the book and find out that there's another complete book about General Lee's last days. You rush out to the bookstore and buy that book. It cites still other books. References like these let you keep on learning forever.

Hypertext is the same idea, computerized and made instant. You're sitting at your computer reading about the Civil War and you notice that the words "General Lee's death" are printed in blue. You click on the blue words with your mouse and an entire article about General Lee's last stand pops onto your screen. Hypertext combines documents by topics, not by sequence. That means you can jump from Tickling to Feathers to Birds to Penguins to The Arctic to Snow to Frosty-the-Snowman. Hypertext works the same way your mind does—by association. Using hypertext you can access an infinite range of data, make sense of it faster than you ever have before, and have a lot more fun doing it.

The World Wide Web uses hypertext to link all its pages together. And since a page can be a movie, a graphic, a song, a database, an Internet resource, or a computer file, hypertext melds with multimedia to form *hypermedia*. See Figure 1–2 for an illustration of hypermedia.

A Web page may have no *hyperlinks* or hundreds of them. Sometimes a Web page is nothing more than a huge list of links, all geared toward one subject. There are two types of hyperlinks:

HTML Text links

HTML Graphic links

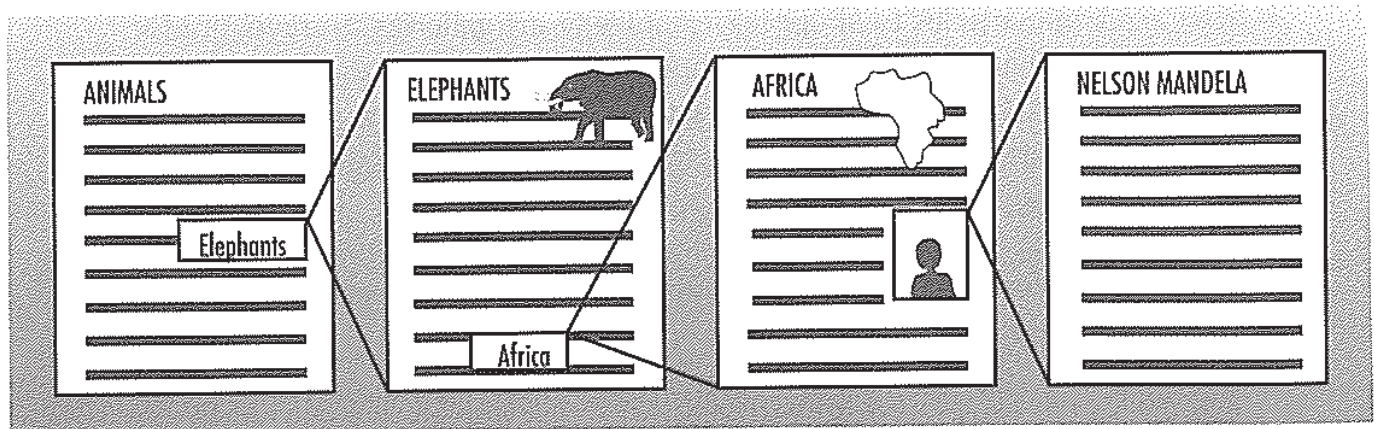


Figure 1-2 Hyper about hypermedia: a word or image on one page acts as the gateway to a different topic's page

A Web browser generally highlights any text link. For example, links in Mosaic and Netscape are generally underlined and colored blue, as shown in Figure 1-3. Cello surrounds the text with a perforated box. Lynx displays all links in bold print.

Graphic links may be small icons, buttons, logos, or any other photo, illustration, or design. A graphic link is usually surrounded by a blue border, also shown in Figure 1-3. To follow a graphic link, just click anywhere on the image. Some graphic links are *imagemaps*—clicking on different parts of the image transports you to different places. See the section on *imagemaps* in Chapter 12, Graphics, for more details.

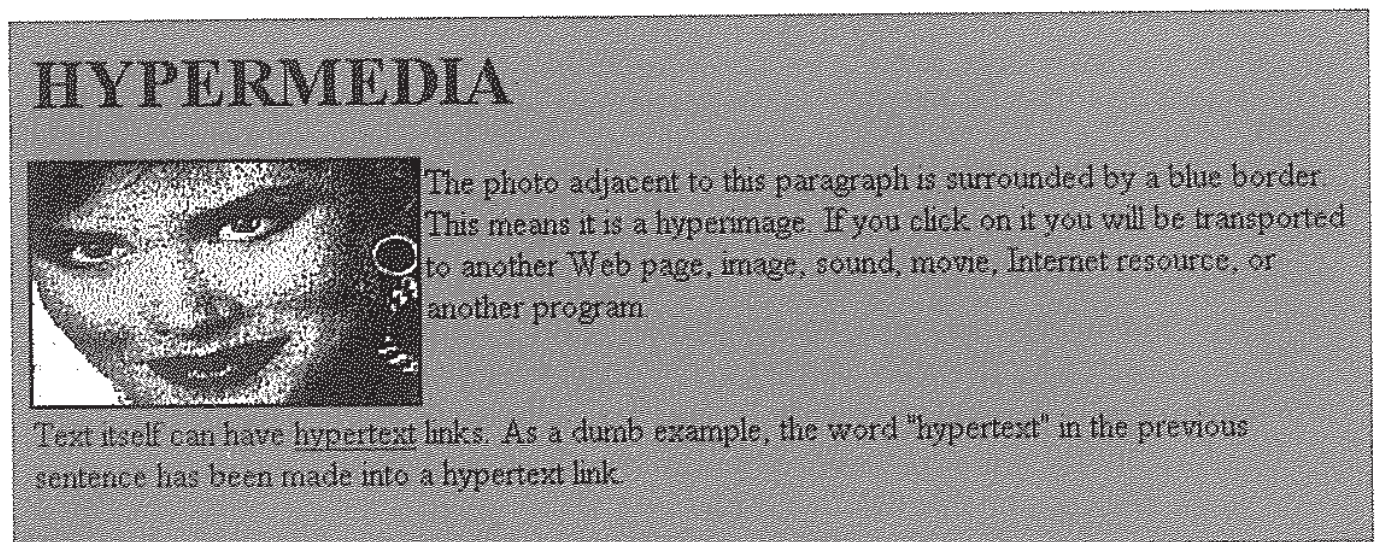


Figure 1-3 A typical Web page with various hyperlinks

You never know exactly where a hyperlink will take you or exactly what will happen. Be brave, be whimsical, follow your gut. Most every browser has a Previous Page button or Back command. If you jump somewhere you don't really want to be, you can return to the previous page with the press of a key. Most browsers also have a *history list* that shows you all the pages you've visited and lets you return to any of them with ease.

Search Indexes

Hypertext makes it easy to get lost. That's part of the appeal. You can start out reading about Sperm Whales and end up reading about the Prince of Wales. But what if you want to find pages dealing with a specific topic?

Luckily, the Web is full of indexes. An index is a special Web page that allows you to type in a search term or keyword. The index then spits out a list of Web pages matching your search. You can then click on any page that catches your interest and transfer there automatically. Some indexes are manual: A person keeps track of all the Web pages he or she comes across. Others are created using *search robots*—small programs that crawl through the Web, following every hyperlink, keeping a list of every new place or topic they come across. Figure 1-4 shows a typical index search screen.

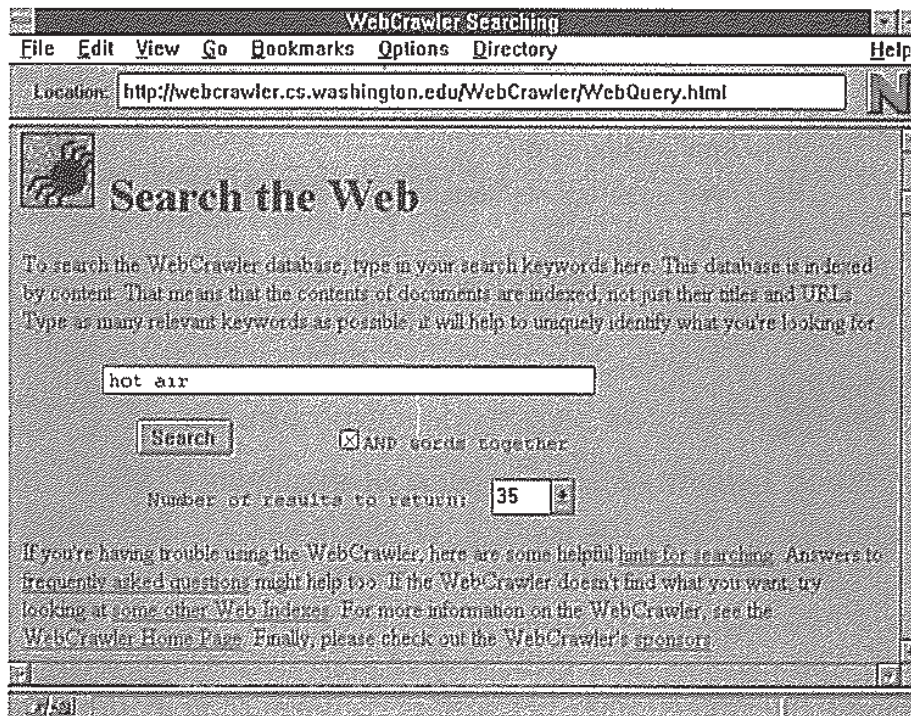


Figure 1-4 Searching for a Web starting point

Many indexes allow you to search the full contents of thousands of Web pages. Other indexes let you search for titles, keywords, or geographical locations. Some indexes have a series of forms or menu items you can customize, allowing you to find exactly what you want. In most cases, you fill in a search term or phrase in a box and then click the Submit or Search button.

Some of the most popular indexes, and their URLs, are listed in Table 1–1. If you don't know what a URL is, check out the URLs, URLs, URLs section later in this chapter.

Table 1–1 Places to go Web hunting for that topic, file, or resource of interest

Search Index	URL
AliWeb Archie Search	http://web.nexor.co.uk/aliweb/doc/aliweb.html
ArchiePlex Form File Search	http://www.lerc.nasa.gov/Doc/archieplex-httpd.html
CUI Index of Search Engines	http://cui_www.unige.ch/meta-index.html
CUI Web Catalog	http://cui_www.unige.ch/w3catalog
CUSI Index of Search Engines	http://Web.nexor.co.uk/susi/cusi.html
Excite	http://www.opentext.com/
Home Pages Broker	http://www.town.hall.org/brokers/www-home-pages/query.html
Infoseek	http://www.infoseek.com/
JumpStation Search Page	http://www.stir.ac.uk/jsbin/js
Lycos	http://www.lycos.com/
Magellan	http://www.mckinley.com/
OpenText	http://www.opentext.com/
WebCrawler	http://webcrawler.cs.washington.edu/WebCrawler/WebQuery.html
Who Where?	http://www.whowhere.com/
WWW: World Wide Web Worm	http://www.cs.colorado.edu/home/mcbryan/WWW.html
Yahoo	http://www.yahoo.com/
Zorbamatic Title Search	http://www.rns.com/cgi-bin/nomad

THE HTTP PROTOCOL

When communication fails it's usually because of intelligence. Maybe the speaker is more intelligent than the listener, or vice versa. Or, more likely, two people are intelligent in very different ways—say, one is an expert in Swahili and the other speaks only Latin. Intelligence is a similar problem over the Internet. You can design a great way to ship data—fast, accurate, with lots of options. But unless other net surfers conform to your standard, they won't see your data.

The Web works so well because, at its heart, it's so stupid and trustworthy. The Web doesn't care what type of data you pump out or what program you use to view the data. It treats everything the same. This means everybody can use their own browsers and their own data formats, and it'll all be friendly. You don't need to use different programs to access different types of data—the Web handles it all without complaining. All Web clients should be able to communicate with all Web servers.


The secret behind this is a simple, no-frills data protocol known as the *Hypertext Transfer Protocol* or *HTTP*. HTTP has the capacity to deal with most any Internet tool, file format, or resource. A Web browser client and the HTTP server communicate, coordinating things before a Web page gets sent. If you could hear them talk, they'd sound something like this:

BROWSER: Hey, here's the page I want, here's how I want it.

HTTP SERVER: Uhh... sure. Can you handle JPEG graphics?


BROWSER: Yup. I sure can.


HTTP SERVER: Ah, good. Here you go, then. Here's your page, hot off the press.


 **NOTE:** Web servers are usually called *httpd*, which stands for HTTP Daemon. A daemon is a program that hides in the computer's memory and is always active. To learn all about servers, including how to set one up, flip to Chapter 21, *Starting Your Own Web Site*.

URLS, URLS, URLS

A URL (or *Universal Resource Locator*) is the address of a document, file, or other Internet resource. If you're familiar with the Internet, you know that each person has his or her own e-mail address. Each resource also has its own address. The World Wide Web gives every article, newsgroup, database, computer, file, or other Internet fragment its very own URL. A URL can consist of these elements:

 The scheme, protocol, or type of Internet *service* the resource is using.

 The name of the *host* computer on which the resource is stored.

 The *port* number of the service.

 The directory where the resource is stored.

HTML The name of the resource itself.

HTML A term to search for in an index.

Some of these attributes are optional, depending upon the type of resource. The port attribute, for example, is almost never used.

A URL will typically look like the phrase shown in Figure 1-5. In general, you don't need to be an expert at crafting or even decoding URLs. A browser always shows you the current document's URL, usually at the top of the page. Whenever you use your Web browser to move anywhere or retrieve any file, its URL is shown. It's a good idea, then, to brief yourself about the basic types of URLs so when you come across them, you'll know what to expect.

If you're reading a magazine, newspaper, or book and come across a URL that sounds pretty groovy, you can usually just type it into your Web browser. Your browser will then retrieve, display, or save the appropriate information on your hard disk.

KEY NOTE: Since a URL usually accesses a Unix file system, upper and lower case are important. Be careful. Most URLs are expressed in lower case. Also, unlike DOS, Unix file and directory names can be any length. Don't be surprised when you come across a 20 letter filename. Another difference is directories: Unix doesn't use backslashes and drive-letter colons. The DOS file C:\DIR1\DIR2\DIR3\FILE.HTM, for example, is expressed in a URL as file:///C1/dir1/dir2/dir3/file.htm.

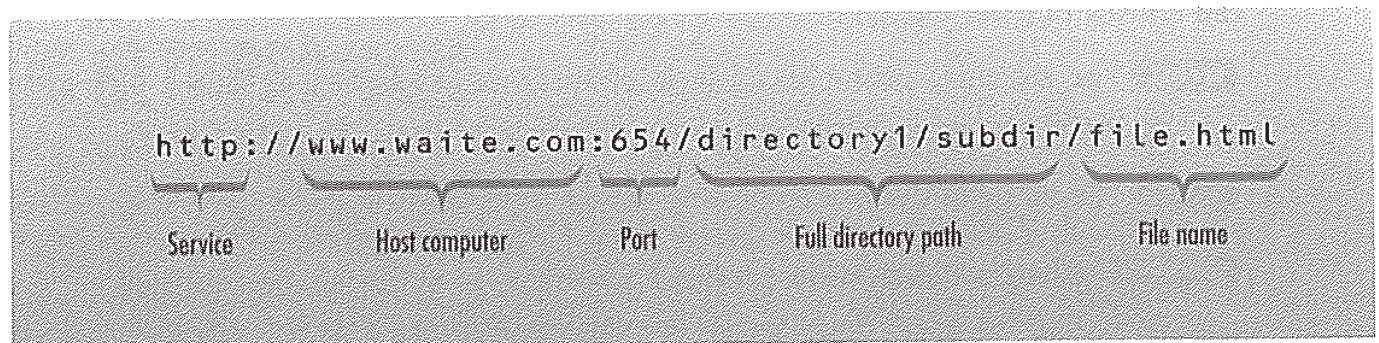


Figure 1-5 The breakdown of a typical URL

The Composition of a URL

The first part of the URL is always the name of a given service (see the next section for a list of services) followed by a colon. The second part of most URLs is the full Internet address of the machine where the resource is located. A host address always begins with a double slash: //. Most addresses are located at default ports, but in some cases you need to specify a specific port by following the address with a colon and then typing the port number. To specify the default port, simply leave out the colon.

After the host computer address, you need to specify the exact directory where the given resource is located. Type a slash between each level in the directory. Finally you can end the URL with the filename of the resource. This filename can be a hypertext document, a database, an index, a text file, a sound file, a graphic, or pretty much any other computerized resource in the world.

In some cases, the filename is followed by the hash mark (#) to specify a particular section within the file. A filename may also include a question mark (?) if it is an index or fill-in form, designating a term to be searched for. Search terms following a question mark are usually separated by the plus sign (+) and may use the equals sign (=). You may also come across a percent sign followed by two letters or numbers (%d1), which represents various extended ASCII characters.

Services

A URL always begins with the name of a service. You can use URLs to access the following services:

 http

 ftp

 file

 gopher

 mailto

 news

 telnet

 tn3270

 rlogin

 wais

New services are always being implemented as new Internet tools, protocols, and resources are created. If you come across a URL not listed here, try it out and see what happens.

Most of the time, you use URLs to call another hypertext Web page using the http service. You can also link documents to ftp file libraries, gopher menus, telnet machines, e-mail addresses, WAIS databases, or any other Internet resources. Chapter 16 discusses these resources in more detail.

Whenever you click on a hyperlink, you're actually requesting a specific URL. The Web server searches far and wide until it finds the URL, which it then delivers to your screen.

HTTP


The http service quickly and consistently calls up any Web page, text, graphic, or other supported item. Most Web pages are written using the *Hypertext Markup Language (HTML)*, which is covered in Chapter 10. HTML files generally have the .html extension, such as *index.html*.

To reference the file *index.html* in the */web/smith/* directory of a computer at address *www.smartpants.edu*, for example, you'd use the following URL:

```
http://www.smartpants.edu/web/smith/index.html
```


Most Web servers automatically search for a default file named *index.html* or *home.html*, depending on the brand of server and its current settings. If this is the case, you don't need to specify a filename to get an index. Just enter


```
http://www.smartpants.edu/web/smith/
```


 **NOTE:** Another protocol called *https://* is used to send secure Web pages back and forth. Microsoft and Netscape both have secure commerce Web servers. A secure page is *encrypted*, which means you can read classified information, submit credit card numbers, or exchange any other sensitive data over the Web without worrying that someone will intercept your transmission. See Chapter 24 for more information about server security.


Here are some sample valid—but fictional—URLs, ranging from the most basic to the most complicated:


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


 <http://www.moolah.com/money/moremoney/>


 <http://www.moolah.com/money/moremoney/money.html>


 <http://www.testlabs.edu/Experiments-In-Science/Monkeys.html>

 <http://www.testlabs.edu/Experiments%2FIn%2FScience/Crocodiles>

 <http://strange.org:1234/strangeness/hmm.html>

 <http://strange.org/strangeness/hmm.html#novelty>

 <http://bigindex.edu.ca/reference/dictionary?cumquat>


 http://abc.123.com:1234/abcdefg/hijk/lm/n/o/p/q/r/s/t/u/v/search?h*%20this%20now

FTP, File

A URL using the ftp or file service retrieves a file. To log in to the FTP archive at Netcom, for example, you would use the URL:

`ftp://ftp.netcom.com/`

Most browsers then show you an index of files. Each file type may even be preceded by a special icon, letting you know whether it's a directory, a text file, an image, or whatnot. You can then click on various directories until you come across a file you're interested in. Just click on that file to download it onto your hard disk or display it on your screen.

 **NOTE:** The Web server automatically logs you in as an anonymous user. Most public FTP sites happily accept anonymous visitors. You can, however, specify a specific username and password if you've got them; this may get you wider access to the site. See the FTP section in Chapter 16, Other Web Resources, for more information.

You can also use a URL to access a particular directory directly. For example, the URL

`ftp://ftp.netcom.com/pub/pc/win3/games/`

would automatically place you in the `/pub/pc/win3/games/` directory, as long as such a directory exists.

You can even use a URL to immediately download a specific file. For example, if you wanted the file *pacboy.zip* in the */pub/pc/win3/games/* directory, you would use the following URL:

```
ftp://ftp.netcom.com/pub/pc/win3/games/pacboy.zip
```

The file service is almost identical to the FTP service, but is a quicker way of retrieving local files. You can use file URLs to load and display sound, image, or text files, as follows:

```
file://www.smartpants.edu/soundoff.au
```

Gopher

To connect to a gopher menu, simply use the gopher URL followed by the appropriate gopher code:

```
gopher://gopher.uchicago.edu/
```

Most browsers print the gopher menu on your screen, complete with different icons for different types of services.

You can use the gopher service to access everything available through ordinary gopher menus: phone books, search indexes like Veronica, images, sounds, or files. For example, to retrieve a specific phone-book index, your URL would look something like

```
gopher://mainman.herenorthere.edu:1234/phones.txt
```

Mailto

This is a relatively new URL that lets you send e-mail to a specific user. For example:

```
mailto:dsf4185@graphics.cs.nyu
```

Browsers which support the mailto URL automatically enter a text editor, allowing you to fill in a subject line, return address, and message. Your browser can then send your letter across the Web quickly and easily.

News, NNTP

Some browsers use the news service to access your news server, retrieving an entire newsgroup, a specific message, or accessing anything using Usenet's NNTP (*Network News Transfer Protocol*). To access a particular newsgroup, you'd use a URL similar to

```
news:alt.cream.cheese
```

You can even access a specific list of groups by using search marks like the wildcard asterisk or the question mark. For example, to see every group that starts with *alt.p*, you could use the URL:

```
news:alt.p*
```

Telnet, rlogin, tn3270

You can use the telnet, rlogin, or tn3270 services to log in to a remote computer. Once you access one of these URLs your browser automatically dials up the remote system. You can use this to enter Internet-based bulletin boards, mainframes, MUD games, or other distant resources.

The telnet, rlogin, and tn3270 services are essentially the same. In general, you use telnet for most connections and tn3270 to telnet to 3270-type mainframes. A typical telnet URL would look like:

```
telnet://bbs.neato.com:1234
```

WAIS

The WAIS service links the Web to a *Wide Area Information Service* (WAIS) database or document. You can include specific search terms, databases, or other WAIS items.

ACCESSING THE WEB

You now know more about the history and theory of the Web than you really need, but you probably can't picture how everything fits together. The only way to grasp this unique medium is to experience it. Depending on where you live, the type of computer you have, and your technical expertise, connecting to the Web can range from a few minutes work to a day or two of detailed software installation and setups. Don't sweat it too much, though. This book guides you through the steps. In fact, you may already be connected to the Web without knowing it.

What Computer Do I Need?

Most any computer worth its weight in salt can access the Web. That's part of the Web's appeal. However, this book focuses on PCs running Windows. If you have a choice, get a 486 or better PC with at least 8 MB of memory, a hard disk with lots of free room, and Windows 95. To take full advantage of this book, you need a CD-ROM drive as well.

You also want a Super VGA monitor so Web graphics look their best. It's also a great plus if your PC has a sound card and speakers, allowing you to hear speech, music, or sound effects, all at CD quality.

What About the Modem?

If your computer is already connected to the Internet through a Local Area Network (LAN) or some other direct connection, then you won't need a modem. However, in most cases you need to call up a commercial Internet provider. A modem lets your computer communicate over the phone lines.


If you want graphics, sounds, and other huge files delivered to you in less than an eternity, you need a high-speed modem. Get one of these:

 V.32bis 14,400 baud

 V.34 28,800 baud

The V.32bis, and V.34 specifications are very important. These are standard protocols used by most Internet providers, so you want your modem to use them too. Modems using other protocols will usually transfer data just fine, but not at their advertised best speeds.

If you're just buying a modem, the 28,800 is probably the most cost-effective choice, providing a good mix of speed and economy. You should also stick to a brand name; they don't cost much more than no-name brands, and they're less likely to have bugs or other errors to slow them down.

 **NOTE:** If you want to use a 9600-baud or slower modem to access the Web, you should use a text-only browser like Lynx. This browser doesn't require a direct SLIP Internet connection; any Internet account will do.






What Type of Internet Account Do I Need?

You have several choices. The most convenient way of accessing the Web is to become part of a network. If you belong to a large corporation, organization, military group, or university, your institution probably already has Internet access. If your machine is connected to a Local Area Network (LAN), then all you may need is some software allowing your PC to communicate via the Internet. Ask your system administrator if your personal PC has Internet connectivity. The system administrator can also help you set everything up.

If you're using a home computer, however, you need to use your modem to connect to the Internet. In most cases, the easiest way to do this is to call up a commercial Internet provider and use its computing power.

You may already be connected to the Internet through a standard Unix *shell* account. This means you dial up a service using a terminal emulator and can type commands like *telnet*, *ftp*, *gopher*, and *mail*. In most cases, this standard shell account is not powerful enough to connect efficiently to the Web. Instead, you need to make your home computer an actual part of the vast network. This requires network-level access, which is known as SLIP or PPP. However, there are several options available to the standard Internet user.

The next few sections cover these methods of hooking up to the Web:

-  SLIP and PPP accounts
-  Standard Internet Unix shell accounts running a text-only browser
-  Standard Internet accounts running the SlipKnot browser, a full graphical browser that runs hand-in-hand with Lynx
-  Standard Internet accounts running The Internet Adapter, a complete SLIP emulator for the Unix shell
-  Direct leased-line connections to an Internet backbone

SLIP ACCOUNTS

SLIP has nothing to do with banana peels. Rather, it stands for *Serial Line Internet Protocol*. CSLIP is compressed SLIP, a variation typically used over the telephone. PPP is the *Point to Point Protocol*. All three protocols do the same thing for you: They let your computer connect directly to the Internet through phone lines instead of through network wires.

If you have a regular old Unix shell, your computer acts only as a window for another large computer, somewhere up in some office building, which is attached to the Internet. A SLIP account skips the middleman—your personal machine actually becomes another corner of the Internet (see Figure 1–6). This means you can easily and conveniently use any PC Internet client, including Mosaic, Netscape, or any other Web browser that comes along. You can also use a SLIP account with cool graphical Windows Internet programs like WinGopher, the Trumpet Winsock newsreader, or Eudora e-mail. These shareware products are discussed in Appendix B. SLIP access is also generally faster than a Unix shell.

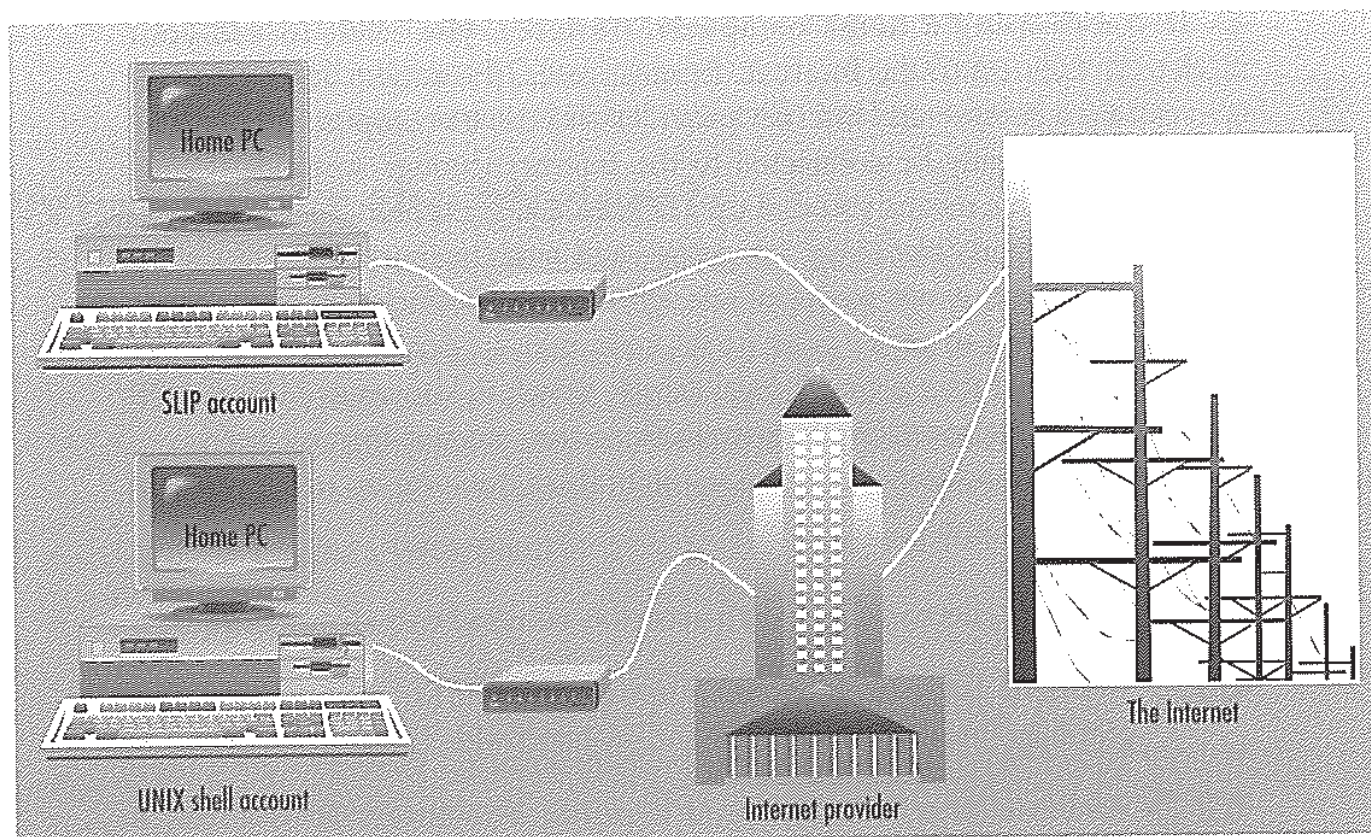


Figure 1-6 A SLIP account compared to a standard Unix shell account


Most every major Internet provider is beginning to offer SLIP or PPP accounts. Some of the most popular are listed in Appendix A, Internet Providers. SLIP often costs a few dollars more than a standard Unix shell account, though prices are steadily decreasing. You should be able to find an unlimited SLIP account for about \$10 to \$20 per month.

TCP/IP

SLIP itself just connects your machine to the rest of the Internet. But just as a telephone line is worthless without a telephone, a SLIP account is worthless if it can't carry data back and forth. To do anything useful with SLIP, you need to speak the same language as the rest of the Internet: the *TCP/IP*, *Transmission Control Protocol/Internet Protocol*.

The Internet is comprised of millions of machines worldwide. TCP/IP is the standard they all use to talk to each other. Every Internet client or Web browser speaks using TCP/IP *packets*—small bundles of computerese. Before you can do anything with your account, you'll need some pretty smart software that can speak over a SLIP line using these TCP/IP command packets.

We're dealing with a lot of different software products: The Web browser, Internet clients such as Eudora e-mail, the TCP/IP translator, and the SLIP connection. All this software is combined in what's called a TCP/IP *stack*. Luckily, there's a solution to this mess: Put a sock in it. Yes, there's a type of software that coordinates the whole TCP/IP stack for you: the *Socket*. The most versatile and popular socket for Windows 3.1 is Trumpet Winsock. Windows 95 comes with software called Dial-Up Networking that hooks you into the Internet in a jiffy.

 **NOTE:** Many Internet providers will give you their own TCP/IP stack software with its own instructions. In almost all cases, however, this will be either Trumpet Winsock or Chameleon Winsock, the only Shareware Winsocks on the market. Each Winsock program has its own *WINSOCK.DLL* Windows library. Other commercial Winsock products you might come across include:

 Spry

 Novell

 FTP

 NetManage

 PC/NFS

Connecting with Windows 95 Dial-Up Networking

Windows 95 comes with a built-in fast 32-bit Winsock. When you first install Windows, it should ask you if you want to set up an Internet connection. You can choose to use Microsoft's Internet Assistant through the Microsoft Network. Click on the desktop's Microsoft Network icon to get signed on. Fill in the information when asked. Click on the Explore the Internet icon to download or install the latest version of Internet Explorer.

However, if you want more control, you might want to set up a link to the Internet provider of your choice. You can easily do this using the Windows Dial-Up Networking program. First be sure Dial-Up Networking is installed. Click on the My Computer icon in the upper left corner. Do you see a Dial-Up Networking folder? If not, install it by opening the Control Panel folder. Select Add/Remove Programs. Choose the Windows Setup tab. Double-click on Communications. Be sure the Dial-Up

Networking checkbox is checked. Click OK and be sure your Windows 95 disc is handy.

Once Dial-Up Networking is installed, you can find it in the main My Computer folder. Open the folder and double-click on the Make New Connection icon. Type in the name of your Internet provider and choose which modem you'll be using—in case you have more than one modem. Click Configure if you've never used your modem before to be sure all the port and speed settings are correct. When done, click Next>.

Type the phone number of your Internet provider in the Telephone number slot. If necessary, adjust the area code and country code. Click Next>. Click Finish when done. Your new Internet connection will now appear in the Dial-Up Networking folder. Right click on your new connection and select Properties to set up the advanced Internet options. Click on Server Type. In most cases, you want to make sure that the Type of Dial-Up Server is: "PPP: Windows 95, Windows NT 3.5, Internet." If your Internet provider doesn't offer a PPP account, you may be able to use a SLIP account by selecting "SLIP: Windows 95". All other options on this screen should be check-marked, except for *Log on to Network* (you'll want to log on manually in most cases) and *Require encrypted password*. Now click on TCP/IP Settings. A screen as in Figure 1-7 appears. Check with your Internet provider for the appropriate settings. If you have a personal IP address, fill it

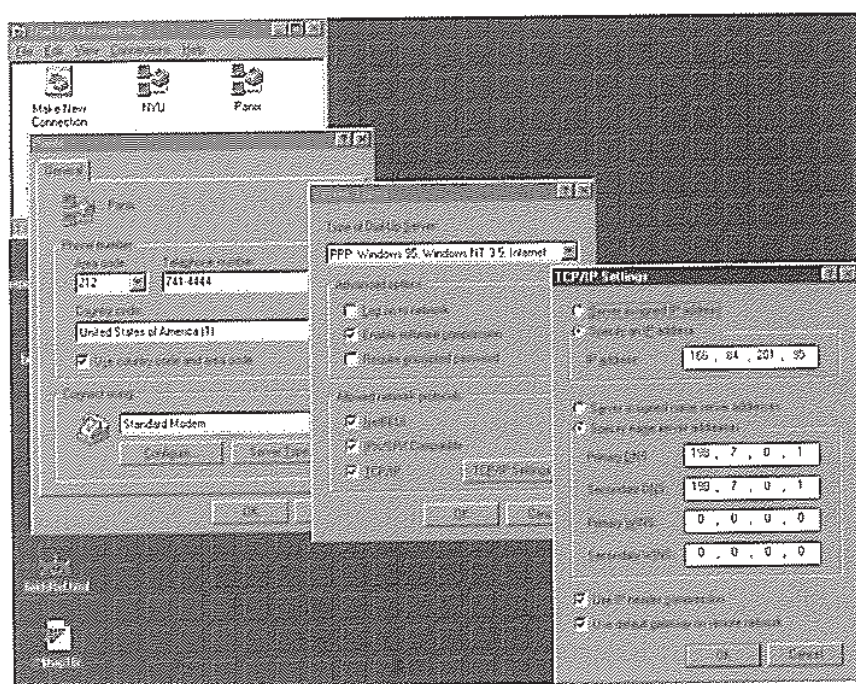


Figure 1-7 Configuring WINSOCK in Windows 95

in the top IP address slot. If you are assigned a dynamic IP address every time you call, click on "Server assigned IP address."

Be sure to fill in the correct Primary *DNS* (*Domain Name Server*)—this is the address of the machine that looks up every Web page or address you type. Your Internet provider may also have a Secondary DNS in case the first one is busy; if so, fill in the address. Click on Use UP header compression and on "Use default gateway on remote network." Click OK.

Go back to the first Dial-Up Networking screen—the one with the phone number and modem type. Click on Configure. Choose the Options tab. In order to log in to your Internet provider manually, you need to be sure the "Bring up terminal window after dialing option" is checked. Do *not* check the "Bring up terminal window before dialing" box. Now keep on clicking OK until all windows close and you return to the Dial-Up Networking folder.

NOTE: You may want to consult with your Internet provider to be sure of the options. Some companies may even have a custom Dial-Up Networking icon they can send you with all the options intact.

To log on to your Internet Assistant, all you need to do now is double-click the icon you just set up. Click on the Connect button. A Status window appears, showing you the progress as your modem dials and connects. Eventually a Post-Dial Terminal Screen should appear, as in Figure 1-8. When

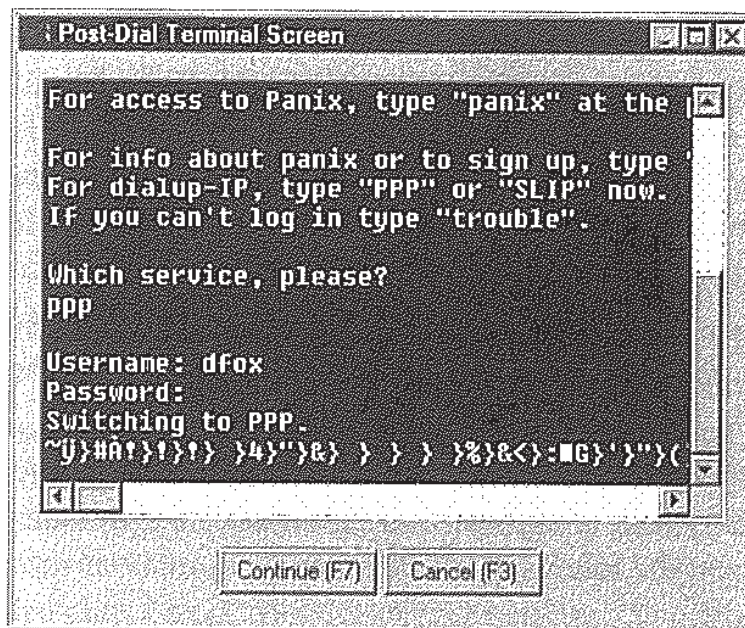


Figure 1-8 Post-Dial Terminal Screen

asked for which service, type in *PPP*—you may need to enter this before or after you type in your name and password, depending on the Internet provider. Type in your account username and password when prompted.

If all goes well you should be told that PPP is enabled and should see a string of strange characters. Click the Continue button. The status window will then let you know that Windows 95 is trying to connect to your provider. Eventually you should see the message: *Connected at 28,800 baud*—or whatever speed you're actually connected at. That's it—you're wired! Minimize the status window and use Netscape or any other Internet tool you desire.

Installing Trumpet Winsock for Windows 3.1

Trumpet Winsock can be found at the site listed in Appendix D. To install everything, just unzip and copy the entire directory to your hard disk. You may want to create a directory similar to

```
C:\WINSOCK
```

You now have a choice. If you're familiar with DOS, you can add this directory to your path by changing your AUTOEXEC.BAT file. For example, if your current path is

```
PATH C:\WINDOWS;C:\DOS;C:\UTIL
```


then you add your Winsock directory to the end of the line, after a semicolon:

```
PATH C:\WINDOWS;C:\DOS;C:\UTIL;C:\WINSOCK
```


You should now reboot your computer to make the new path take effect.

If you prefer, you can just move the *WINSOCK.DLL* file to your *C:\WINDOWS* directory—or any other directory in your path.

You should also set up an icon for Winsock's *TCPMAN.EXE* file in your Windows Program Manager, since it's a program you'll be accessing quite often.

 **NOTE:** Trumpet Winsock is not freeware. If you use Winsock regularly, you should register your version for \$20. Read the registration information in the included *INSTALL* document.

You now need to set up Winsock so it knows how to communicate. At this point, you should already have a SLIP account at an Internet provider. Your provider should give you the following information:

 **HTML** Phone. The telephone number of your Internet provider's SLIP computers. Ideally, this should be a local call.

- HTML** Username. The name you'll use to log in to the system; it may be your last name, a combination of your initials, or a meaningless ID number.
- HTML** Password. The code you'll use to assure the system that you're the one using your account. It can be any series of letters or numbers. Keep track of upper- and lowercase letters—if your password is Froggie, “froggie” may not get you into the system.
- HTML** IP Address. Your provider's Internet Protocol address, in the IP number form 1.2.3.4. The value for this may be the word *bootp* (in lowercase) or the value 0.0.0.0.
- HTML** Name Server. The IP number of the machine you'll be calling to look up Internet domain names.
- HTML** Gateway. The address of the Internet gateway or router being used.
- HTML** Time Server. The IP number of the Time server machine. Usually unused or the same as the Name server.
- HTML** Domain Suffix. The address of your Internet provider, in the form *name.com*. You may need several of these.

The first time you run Winsock, the Network Configuration window appears, as in Figure 1–9.

To return to this window in the future, select File, Setup.

Check the Internal SLIP checkbox. Type the COM port your modem is connected to in the SLIP Port box. This should be a number from 1 to 4. For example, if your modem is wired to COM2, type 2. Fill in your modem's maximum baud rate. Do not use commas as part of the number. If you're running 14,400 baud you should fill in the maximum data rate:

19200

Select the Hardware Handshake checkbox unless your Internet provider tells you not to. Typically, you need to check the Van Jacobson CSLIP Compression box too. This means you'll be using CSLIP instead of SLIP, which is a common way of sending SLIP over telephone lines. If you know your modem can handle DCD (RLSD) or DSR Online Status Detection, you can click one of these values. Otherwise, click None.

Network Configuration

IP address: 128.122.237.130

Netmask: 255.255.255.0

Default Gateway: 128.122.237.10

Name server: 128.122.128.24 12

Time server: 128.122.128.24

Domain Suffix: nyu.edu

Packet vector: 00

MTU: 1500

TCP RWIN: 4096

TCP MSS: 1460

Demand Load Timeout [secs]: 5

TCP RTO MAX: 60

☒ Internal SLIP

SLIP Port: 1

Baud Rate: 19200

☒ Hardware Handshake

☒ Van Jacobson CSLIP compression

Online Status Detection

☒ None

☐ DCD (RLSD) check

☐ DSR check

OK Cancel

Figure 1-9 Configuring Trumpet Winsock


You can now fill in the IP Address, Name Server, and other values. If you're supposed to use several Domain Suffixes, separate each by a space. You now need to fill in some specific data speed values. The values for these depend on whether you're using SLIP or CSLIP compression. If your Internet provider doesn't give you exact values, try the defaults:

HTML MTU. This is the Maximum Transmission Unit, which depends on the Maximum Segment Size (MSS). Winsock recommends you use the MSS value plus 40.

HTML TCP RWIN. The TCP Receive Window. Winsock recommends you use your MSS number times 3 or 4.

HTML TCP MSS. The TCP Maximum Segment Size. This number should be around 512 for SLIP or around 255 or less for CSLIP.

You should leave all other values at their defaults, unless your Internet provider tells you otherwise. When everything is filled in, click OK. A message appears letting you know that you must restart Trumpet Winsock before any of your settings will take effect. Exit Winsock by selecting File, Exit and then reload it.

 **NOTE:** If your machine is connected directly to the Internet through a Local Area Network, you need to install a TCP/IP packet driver such as WINPKT. You can then use SlipKnot to handle the stack. Internal SLIP should be unchecked. Ask your system administrator for the value of your institution's IP Address, NetMask, Default Gateway, and all other values.

Manual Login

You now need to call your Internet service provider to be sure everything works. At first, make the call manually. Eventually, you want to create a script file that can automatically dial the Internet provider, log in using all appropriate names and passwords, and start up your SLIP connection using the proper commands.

To use the manual login, select Dialler, Manual Login. The following message should appear:

```
Manually dialing.
AFTER LOGGING IN, PRESS THE <ESC> KEY TO RETURN TO NORMAL SLIP
PROCESSING
SLIP DISABLED
```

You now need to use your modem's built-in dialing commands to call up your Internet provider. The dial command is ATDT followed by a valid phone number. For example, if your Internet provider is located at 555-1234, you would type

```
ATDT 555-1234
```

The Winsock program then becomes a generic terminal-type communications program. What you type next depends on your Internet provider. The provider should give you full instructions. In some cases, you may need to type the word *SLIP*, or just press **ENTER**. In any case, you should eventually see a login prompt similar to

```
Login:
```

Type your username and press **ENTER**. You should now be asked for your Password:

Type in the password. In most cases, you need to start the SLIP protocol by typing the word

```
slip
```

A message similar to the following should now appear:

```
Switching to SLIP.
Annex address is 123.456.789.10. Your address is 123.456.7.8.
```


What you do next depends on whether you have a *static* SLIP account or a *dynamic* one. If your Internet provider assigned you your own personal IP number then you have a static account and everything should be set up fine. You should now run the *bootp* protocol to get specific addresses, ports, and other information from your network. In most cases, you need only type:

BOOTP

Press **[ESC]** to stop the manual login process. The following message should appear:


SLIP ENABLED

However, many Internet providers use dynamic accounts instead of static ones. These providers have several valid IP numbers and assign them on a per-call basis. When you dial in on a dynamic account, you get one of the IP numbers no one else happens to be using at the moment. To finish the login process you need to select File, Setup and type your assigned IP number in the IP Address box. Click OK. A message appears telling you that you need to close Winsock before any change can take effect. Press the **[ESC]** key, exit Winsock (by selecting File, Exit), and then restart Winsock.


You can now run Mosaic, Netscape, Eudora, FTP, or any other whizbang Internet software. If things don't seem to be working, check out the Shooting Down Troubles section later in this chapter. Otherwise, you're ready to set up an automatic script.


LOGIN SCRIPTING

If your manual login seemed to work just fine, it should be easy to create an autopilot that dials you up, logs you in, and sets everything running. Many Internet providers can give you a customized Winsock script file if you ask. The file should be called *login.cmd*. Copy the file to your Winsock directory. To log in now, just select Dialler, Login.

 **NOTE:** You can have Winsock automatically call your Internet provider every time you run it. Select Dialler, Options. Click the Automatic Login On Startup Only radio button and select OK.

There are several useful script commands you need to know:

 **input** *<timeout>* *<message>*. Waits *<timeout>* seconds for the *<message>* to appear.

 **output** *<message>*. Sends the *<message>* over the modem.

HTML *password <prompt>*. Open a dialog box asking the user for the password.

HTML *username <prompt>*. Open a dialog box asking the user for a username.

HTML *BOOTP*. Run the bootp protocol.

HTML *#*. A comment.

HTML *\13*. Send a carriage return; the same as pressing the **ENTER** key.

HTML *\i*. Send the specified IP address.

HTML *\p*. Send the password that was obtained in the Password dialog box.

HTML *\u*. Send the username that was obtained in the Username dialog box.

The idea is to write a small program that mimics the exact steps you performed during the manual login. Every script begins with the ATDT dialing command. You should then wait for a specific prompt. For example, if the first thing your Internet provider requires you to do is type *login* at the *Welcome>>* prompt, the beginning of your script would look like:

```
#Call up your Internet provider
output atdt555-1234\13
#Wait for the CONNECT message
input 30 CONNECT
#Send <ENTER> to kick the system into action
output \13
#Wait for the Welcome>> prompt; you only need to specify the last
#few letters.
input 30 come>>
#Type the login command
output login\13
```

The best way to begin writing a script is to edit Winsock's sample login script and fill in the proper phone number or connect values. Select Dialler, Edit Scripts. The File Open dialog box appears. Select the *login.cmd* program and choose OK. A sample script follows. You should replace the sections in boldface print with valid values.

```
# Dial-in script to a typical SLIP Internet provider
#
#                               Initialize modem:
output at\13
#                               Wait for 'OK' from modem
input 10 OK\n
```

Continued on next page

Continued from previous page

```
#           Dial the Internet provider's phone number
output atdt555-1234\13
#           Wait for the CONNECT message
input 30 CONNECT
#           Wait for Username: prompt
input 10 name:
#           Prompt user for username using a dialog box
username Username
#           Type out that username
output \u\13
#           Wait for Password: prompt
input 10 word:
#           Prompt user for password using a dialog box
password Password
#           Type out that password
output \p\13
#           Invoke the SLIP protocol by typing the word "slip"
#           You may have to move this line prior to the
#           logging-in step.
output slip\13
#           Wait for IP address string
input 30 Your address is
#           Parse address (for dynamic SLIP accounts)
#           If you have a static slip account you can
#           delete the next line
address 30
input 30 \n
display \n
display Connection Attempt Complete!
display \n
#           Get network information using BOOTP protocol
BOOTP
```

Logging Out

To hang up manually, select Dialler, Manual Login. Type three pluses:

+++

and then type the modem hang-up command:

ATH






You can also log out using the standard logout script. Select Dialler, Bye.

UNIX SHELL WEB BROWSERS

A Unix shell is the traditional way of accessing the Internet from a home computer. You dial up an Internet provider's modem and, using a terminal program, you control the provider's computer and talk through it to the Internet.

This gives you command of the Internet, but it has many limitations. Since you're using somebody else's computer, you are limited to a simple form of communication. In most cases, you can only view or type text. Anything like graphics gets far too complicated.

A SLIP account actually makes your home computer a part of the Internet. But sometimes even SLIP isn't the answer. For instance:

-  Your modem may be too slow; less than 9600 baud.
-  SLIP may be too expensive in your area.
-  There may be no local SLIP provider.
-  Your computer may not have the resources to run SLIP.
-  You're already connected to the Internet using a great Unix shell and don't want to give this up.

Just because you're limited to a Unix shell account doesn't mean the situation is hopeless. Far from it. There are several surefire ways to access the Web from your Unix shell. The easiest way is to use a Unix-shell-compatible Web browser, such as the text-only Lynx or WWW or a graphical browser such as SlipKnot.

Text: Lynx or WWW

The simplest solution is to get a text-only browser like Lynx or WWW. These browsers run on your Internet provider's Unix machines, not on your home PC. Most Unix servers, however, should already have a working copy ready for you to use. Lynx and WWW allow you full access to the Web with one catch: You won't be able to see any graphics, hear any sounds, or be wowed by any other multimedia perks. The Lynx browser is covered fully in Chapter 3. If you can't run a local version of Lynx or WWW, you can telnet to it. See the Telnetting To A Public Browser section in the next chapter.

To access Lynx or WWW, you can usually just go to your Unix prompt and type

```
Lynx
```

```
or
```

```
WWW
```

Graphics: SlipKnot


MicroMind's SlipKnot is a terribly clever graphical Windows Web browser that works using dial-up Unix shell accounts. You need no SLIP, PPP, or TCP/IP software of any kind. SlipKnot loads up Web pages much more sluggishly than a SLIP Web browser, but if your modem is fast enough it's a very viable option. SlipKnot also has a number of other nice features. For instance, once you load up a Web page, you can load it again instantaneously. You can also view several Web pages at once.

SlipKnot is covered fully in Chapter 4.

THE INTERNET ADAPTER

If you can't access a SLIP account, the best solution is to emulate SLIP over your Unix shell. For \$25, The Internet Adapter (TIA) software by Cyberspace Development, Inc., does this task most excellently. If Winsock is the socket, TIA is the three-pronged adapter that plugs the Internet into your home computer. Once TIA is installed, you can use Mosaic, Netscape, or any other multimedia Internet tools the exact same way you would over a true SLIP line. A Unix Internet account equipped with TIA usually runs just as fast as any other SLIP.

Since TIA rides atop your standard Unix shell, it's a good idea to be fairly familiar with basic Unix and Internet commands before you try installing it. You might want to check out a command-line Internet book such as the Waite Group's *Internet How-To*.

 **NOTE:** If a SLIP account costs much more per month than a standard Unix shell account, The Internet Adapter often pays for itself quickly.

Determining Your Unix Platform

The Internet Adapter does not run on your home PC—it is Unix software and is stored and accessed on your Internet provider's hard disk. Before you install TIA, be sure you have enough space. If not, you may have to request or buy additional space from your Internet service provider. Also beware: Some Internet providers do not allow you to use TIA because it interferes with their regular SLIP service.

All Unixes are not built the same. There are many platforms, or flavors, of the Unix operating system. Likewise, there are many different versions of TIA depending on which Unix shell you are using. TIA supports the following hardware/operating system configurations:

HTML Sun Sparc: Solaris 2.x or SunOS 4.1.x

HTML 386/486: BSDI 1.x, SCO, or LINUX

HTML DEC Alpha: OSF/1 2.0

HTML DEC: ULTRIX 4.3

HTML IBM RS/6000: AIX 3.2

HTML SGI: IRIX 4.0

HTML HP 9000: HPUX 9.0

If you're not sure what system you're using, you can ask your Internet provider. Alternatively, you can run a special script program to help you guess your brand of Unix. This script is not PC software. You'll need to use your Unix shell account to download the script directly via FTP. To log in to the FTP archive, type

```
ftp marketplace.com
```

When asked for a login name, type

```
anonymous
```

When asked for your password, type your e-mail address. Now type the following sequence of commands at the *FTP>* prompt to grab the file:

```
cd tia
ascii
get config.guess
bye
```

To make the script executable (able to be run), type

```
chmod u+x config.guess
```

And then type the following to run the script:

```
./config.guess
```

If worst comes to worst, you can download each version of TIA and try them out until you find one that seems to work.

Getting Your License Code

Since there are several different versions of TIA, you need to download TIA directly from its creators. Cyberspace Development issues free evaluation

copies of TIA that you can try for 14 days. After that point, the software is automatically disabled.

To get a copy of the evaluation software, you need to order it, along with your own personal license code. To snag the order form, send a blank e-mail letter to: tia-single@marketplace.com. With most Unix accounts, you would type:

```
mail tia-single@marketplace.com
```

You could then press **ENTER** when asked for a subject and press **CTRL-D** to send the blank letter.

Cyberspace Development mails a user form to you immediately. When you get the form, load it up using any text editor and fill it in. You should fill in an answer to each question between the dotted lines and after the colon. For example, when it says

```
-----|
# FIRST NAME:
-----|
```

you would put the cursor after the colon and type


```
-----|
# FIRST NAME: David
-----|
```

assuming, of course, your first name actually was David. Be sure to put an **X** next to the Evaluation order type when asked, unless you'd like to use your credit card and order a registered copy of the software right away.

```
-----|
# ORDER TYPE: Evaluation (X) Purchase ( )
-----|
```

When you're done, mail back the completed form to tia-single@marketplace.com. If you have any questions or problems with this process, you can send mail to:

```
tia-order-desk@marketplace.com
```

 **NOTE:** After evaluating TIA for 14 days you may decide to buy a registered license code. You can resubmit the same form, this time including your credit card or other payment details. You can also order TIA by phone using a credit card by calling SoftAware at 310-314-1466.

Shortly, you get two messages back: your License Code and Installation Instructions.

Getting the Software

Once you receive your TIA Evaluation License Code, save the e-mail message in a file in your *home directory* called *.tia*. You can do this using the *extract* command of your Unix mail program. Read the e-mail message that contains your TIACODE and then type:

```
s .tia
```

Alternatively, you can create the *.tia* file in your current directory using your favorite Unix text editor. Print or write down your TIACODE, Login Name, and Domain Name. You can then create a *.tia* file that looks similar to the following:

```
TIACODE:abcd22ggdg2h3g4yutfgfj2h3j4h3jh3jh2j3h3
Login Name: <jsmith>
Domain Name: <smartpants.edu>
```



NOTE: If you purchase a registered TIACODE, the process is the same. Be sure to put the latest code in your *.tia* file. You should also keep a copy of the code in a separate file or on your home PC so that if your Unix provider's hard disk crashes, you can easily reinstall TIA.

You're now ready to download the TIA program itself. You'll need to know what Unix platform you're working on. To access the FTP archives, type the following command at your Unix prompt:

```
ftp marketplace.com
```

When asked for your login name, type

```
anonymous
```

When prompted for a password, type your complete e-mail address. You can now access the TIA file by changing to the proper directory. Change to the *tia* directory by typing

```
cd tia
```



NOTE: Some versions of the software may be available in the *tiabeta* directory.

To see the available versions, type

```
dir
```

You can now retrieve the proper version of TIA by using the *get* command. For example, if you need the *sparc.solaris.tia* version, you would type the following series of commands:

```
binary
get sparc.solaris.tia
bye
```

Installing The Internet Adapter

You should now have a working copy of TIA in your UNIX machine's home directory. You can rename the file you just retrieved to *tia* using the *mv* command. For example:

```
mv sparc.solaris.tia tia
```

You now need to make the TIA software executable by using the *chmod* command:

```
chmod u+x tia
```

You can verify that The Internet Adapter is working by typing:

```
./tia
```

The following message should appear:

```
The Internet Adapter (tm) 1.0 for BSDI Unix
Copyright (c) 1994 Cyberspace Development, Inc.
Single License
Ready to start your SLIP software.
```



WARNING: An evaluation copy of TIA only works for 14 days from date of issue, so it needs to get the current date from your Unix machine. The evaluation version will not run at all if it can't read the date. Unfortunately, some Internet providers do not allow public access to the time utilities. If you hit on this problem, you can still try TIA if you can persuade your Internet provider to allow time-server access, or if you go ahead and buy a registered copy of TIA.

Configuring Winsock for The Internet Adapter

Once The Internet Adapter is successfully installed on your Unix system, you can run the Trumpet Winsock program as usual. Instead of dialing a SLIP account, you dial up your Unix shell account and run The Internet Adapter. Before you make the call, however, you should fine-tune some of Winsock's settings so that it works in tandem with TIA.

The Setup

To set up Winsock, run it and select File, Setup. Be sure the Internal SLIP box is checked. You now need to fill in a number of values. Since TIA only emulates a SLIP account, there is no actual IP address. Instead, you should use the dummy IP address of 192.0.2.1. Your Netmask should be 255.255.255.0.

The TIA program itself can figure out what value you need to put in the Name Server box. Dial up your Unix account and type:

```
./tia -address
```


TIA tells you the IP address for the Gateway and the Name server. If your machine has several Name servers, separate each with a space. If this command doesn't work, you can try Unix's own Name server command:

```
whois -h internic.net domain.xxx
```

If neither of the above two commands seems to work, you'll need to ask your Internet provider for the IP number of the Name server.

You can leave the Time Server and Domain Suffix boxes blank. You now need to set the data flow values. Use the following values:

 MTU: 1500

 TCP RWIN: 4096

 TCP MSS: 1460

 TCP RTO MAX: 60

 Demand Load Timeout: 5

You should now fill in your Winsock communications settings, as described earlier in this chapter. Fill in the correct SLIP Port, Baud Rate, and Hardware Handshake values. Be sure the Van Jacobson CSLIP Compression attribute is *not* checked.

Trumpet Winsock should now be a perfect match with The Internet Adapter. Click OK, exit Trumpet, and restart it.

Dialing Manually


You can now use Trumpet Winsock to dial your Unix system manually, as described earlier in this chapter. Select Dialler, Manual Login. Dial up your system and log in the way you normally would. Instead of typing *slip*, however, you should type:

```
./tia
```

You should now see the message:

```
Ready to start your SLIP software
```

Press the **[ESC]** key to end the manual login sequence. Hearty congratulations to you! You now have full SLIP access to the Internet.

 **NOTE:** If your Unix Internet account uses a series of menus, you need to find some way of getting to the Unix command prompt. Ask your Internet provider if a standard shell account is available. If not, you'll need to create a Winsock script to navigate the proper menu choices.

Dialing with a Script

Alternatively, you can create a Winsock script that logs into your Unix shell account and runs the TIA program. This script should look like the one earlier in this chapter, with the additional line:

```
output ./tia\13
```

Exiting TIA

To quit TIA manually, select the Manual Login item from Winsock's Dialler menu. Slowly type **CTRL-C** five times to exit the TIA program. You'll return to the Unix prompt, where you can use Unix Internet commands as usual. You can rerun TIA at any time.

HIGH-SPEED LEASED LINES

SLIP/PPP accounts are useful for most home Internet surfers, but they lack two very important attributes: high speed and permanence. You should investigate high-speed leased lines if you fall into one of the following categories:

HTML You're a person or company that wants to be connected to the Internet all the time for fast and easy access.

HTML You want to start your own Internet site, complete with Web server.

HTML You want to become a full-fledged Internet provider selling access to the Internet to other people.

KEY NOTE: If you want to become your own Internet Provider, a good place to start finding information is the FAQ at:

<http://www.amazing.com/internet/faq.html>

Permanent SLIP

If you wish, you can use a standard 28.8 modem to establish a "permanent" SLIP/PPP connection to the Internet. Many companies will rent you leased lines that allow uninterrupted access. You can find all sorts of companies, with all sorts of rates. A 28.8k connection line may cost from \$20-\$100 a month. Contact your local phone company for information.

Note that a 28.8 connection is never good for more than one user at a time. If you want to deal with several connections at once, you'll need a high-speed line.

High-Speed Lines

If you want to connect at higher speeds and greater permanence, you'll need some even more complicated equipment and will need to connect to an official *Internet backbone*. There are dozens of backbone companies around the world and many smaller companies that can sell you time-shared access to a backbone.

You can generally connect at the following speeds:

- HTML** T3—This 44.5 Mbps connection gives you the fastest possible connection to the Internet. If you're planning on starting a major site with millions of users you should check out T3. A T3 line is very, very expensive, but may be worth it if you expect many clients, users, visitors, or customers.
- HTML** 10 Mbps—This level of connection falls directly below a T3 and usually involves sharing T3 lines with other people.
- HTML** Microwave—Faster than a T1, slower than 10 Mbps.
- HTML** T1—A full T1 connection is 1.54 Mbps. You can have around 200 simultaneous users utilizing a T1 line at any time. Although this requires some expensive start-up equipment, the line will usually run you anywhere from \$400 to \$1200 a month.
- HTML** Fractional T1—You can share a T1 line with other people at 768, 512, 384, 256, or 128 Kbps. Setup costs will be the same as a full T1, but monthly charges will range from around \$300 to \$850.
- HTML** ISDN—You can buy ISDN line ranging from 64 to 128 Kbps. An ISDN line generally costs anywhere from \$200 to \$400 a month. The modem itself will run you anywhere from \$1000 to \$1500.
- HTML** 56 Kbps—This high-speed modem is a low-cost solution for permanent Internet connectivity. The routing equipment only costs about \$400 and the price is only \$150-\$250 per month.

Note that prices vary wildly. Check out all the providers in your area and make sure you understand all the options they offer before settling on one high-speed line.

Some companies offer *frame relay*, which is a lower-priced way of connecting. A leased line is set up between you and the *frame relay access*

point. Many people may be connected to the relay point. The frame relay will then take all these split signals and stream through one high-speed pipe into the Internet provider. This lets you share a line with other people and drastically reduces costs.

High-Speed Hardware

To connect directly to a Internet backbone provider, you'll need to do more than rent high-speed lines from your phone company. You'll need to factor in the following equipment:

HTML LAN (*Local Area Network*)—If you don't already have a Local Area Network running, you'll need to set one up. You can sometimes use a simple Ethernet card. The number of computers you'll need depends on the speed and type of Internet connection you want. The more data, the more machines you'll need to split the work.

HTML A Router—This is a box or card that connects your local network to the Internet. It's a smart piece of hardware, able to figure out what data should go to the Internet. It then routes this data to the appropriate location so that everything moves as quickly as possible. There are also software routers available in certain cases which let you turn one of your computers into a router. A router costs about \$2000.

HTML CSU/DSU—This is a special type of high-speed communications hardware, similar to a modem, which can be used for 56K, T1, or T3 lines. This connects into your router and can send the router's signal over the telephone company's lines. Prices range from \$250 to \$3500 depending on the speed.

HTML ISDN Modem—A specialized modem for ISDN.

HTML Cables—You'll need specialized cables to hook everything up.

Some Internet provider companies will make you buy the routing and CSU/DSU equipment for both ends of the connection; other companies already have all the receiving equipment they need.

Connection Companies

Appendix A lists several premiere companies which offer high-speed connection options. You should also check out DLIST, which lists providers

who offer dedicated line connections. Send mail to dlist@ora.com to get an updated copy of DLIST.

Or check out:

HTML Yahoo

http://www.yahoo.com/Business/Corporations/Internet_Access_Providers/

HTML The List—<http://www.thelist.com/>

SHOOTING DOWN TROUBLES

Stacks can often tumble. The complex combinations of TCP/IP software, socket software, SLIP software, Internet client software, Web browsers, your Internet provider's Unix system, your machine, and Windows itself can cause lots of headaches.

Whenever you have a problem, be sure to check each component individually. If you're using The Internet Adapter, run it and be sure it doesn't give you any error messages. If it does, you may need a different version of the software. If you're running a Winsock script that doesn't seem to work, first try logging in manually to see if the problem is with your script or with Winsock itself. Make sure your modem is connected when you think it should be: Just pick up the phone and listen; if you hear a low-pitched grumbling noise, you're online.

That said, here are a few common errors and easy ways to squash them:

Your Modem Won't Respond

If you try dialing but nothing seems to be happening, this could indicate one of several problems:

HTML Your modem isn't attached to your computer. Check to be sure the cables are inserted tightly.

HTML Your modem isn't attached to your phone line. Plug a phone into the line to be sure you're getting a dial tone.

HTML Your modem isn't turned on.

HTML Your modem hasn't been properly reset or initialized. Check your modem owner's manual for guidelines. If possible, contact your modem's manufacturer.

HTML Your port isn't configured properly in Windows. Run the Windows Control Panel and select the Port icon. Select the port your modem is connected to and click the Settings button. Be sure the settings correspond to your modem's baud rate. Note that a 14,400 baud modem's port should be set to 19,200. The port should also be set for 8 data bits, no parity, 1 stop bit, and the Flow Control type should be Hardware.

If your modem works fine one moment but then suddenly stops responding, the modem might just be a little tipsy. Turn off the modem and turn it back on or reboot your computer.

The “DNS Lookup Failed” Message

This is probably the most common error message you're likely to run across. It means your Web browser is having trouble accessing remote URLs.

Most likely, there's a wrong value in your Winsock or Dial-Up Networking setup window. Double-check the IP numbers of your Domain server and your Name server. Contact your Internet provider to be sure they gave you the right information in the first place. All these numbers can get confusing; you may have copied something down wrong. Using the wrong IP number is as useless as dialing the wrong phone number.

The “Unable to Resolve Host Name” Message

If everything seems to run, but your Internet tools have problems connecting to specific hosts, your Domain Name Server (DNS) information is probably wrong. Check Winsock's or Dial-Up Networking's setup and double-check the Name server entry.

WHAT NOW?

You're now connected to the Internet, either using SLIP or The Internet Adapter. Your next step is to install, configure, and learn to use a Web browser so you can start flipping through those tens of thousands of multimedia Web pages. Chapter 2 introduces you to the wild world of the Web browser. Later chapters deal with particular brands of browsers:

HTML Chapter 3 is a hands-on guide to using the text-only Lynx browser.

HTML Chapter 4 covers the SlipKnot graphical Unix shell browser.

HTML Chapter 5 shows you everything you need to know about the acclaimed NCSA Mosaic browser.

HTML Chapter 6 guides you through the sleek Netscape browser.

2



The Internet is a jumble of computers, wires, satellites, modems, and other fancy hardware. The World Wide Web is a vast structure of multimedia data distributed over the Internet. It takes a Web browser to grab, handle, and display all this data, making it useful to you. This chapter is an introduction to browsers of all types.

WHAT TYPE OF SOFTWARE IS A BROWSER?

Perhaps a better question is what type of software isn't a browser? A Web browser isn't like any other program you've ever come across. In some ways, it's a text editor, able to load up files and display them on your screen. It's also a graphics package, able to handle certain types of images—often alongside the text. On the other

side of the coin, a Web browser is a full-featured online navigation tool, often combining common Internet artifacts such as e-mail, news, FTP file transfers, and telnet.

You'll find literally hundreds of browsers available for different computers and operating systems. The PC alone has nearly 40 different browsers for you to choose from. Most browsers share the following features:

- HTML** Hypertext document viewing.
- HTML** Text formatting. That is, you see text broken into paragraphs, lists, headings, or other elements with each element having its own unique appearance.
- HTML** Internet service access. Usually including gopher, telnet, news NNTP, FTP, or WAIS.
- HTML** History list maintenance. So you can get back to previously-visited Web pages or other URLs.
- HTML** Bookmark list (also called hotlist) maintenance. To keep track of your favorite Web pages or other URLs.
- HTML** Interactive forms. Fill-in forms complete with blank fields, check-boxes, radio buttons, and menus.
- HTML** Usenet news reading and posting.
- HTML** E-mail sending.
- HTML** Authentication. This lets users keep certain pages secure by requiring passwords or special signatures for access.

Browsers that run on a windowing operating system (such as Unix X-Windows, Macintosh, or Microsoft Windows) have these additional qualities:

- HTML** Mouse-driven commands.
- HTML** Ability to process full hypermedia documents.
- HTML** Ability to work with embedded graphics, tables, and complex mathematical equations.
- HTML** Wide range of fonts, font sizes, special characters, and font styles.

WHICH BROWSER SHOULD I USE?

In theory, all browsers should be similar. Browsers are supposed to be uniform, handling a common Web language. However, the latest browsers each come out with their own fancy features and commands hoping to become the standard. This means some Web pages look great using one browser and pretty shoddy using another. This, in a way, undermines the whole global idea of the Web. On the other hand, many of these new features eventually make their way into the mainstream creating an ever more useful and beautiful Web.


So what was supposed to be uniform has become competitive. There are currently tons of Web browsers out on the market with new products announced each month. Luckily, there are only four or five browsers that seem to have withstood the rigor of the real-world Web.

If there is a standard, NCSA Mosaic is probably it. This doesn't mean it works the fastest (it doesn't), is the easiest to use (nope), or has the most features (uh-uh). This does mean that tons of people use Mosaic, and thus most Web pages will look pretty good on it. NCSA also provides Mosaic absolutely free to the Internet community. Chapter 5 discusses the Mosaic browser.


The best Web browser is clearly Netscape. The Netscape software works the fastest, has the most bells and whistles, and is the easiest to use and the least prone to errors. All told, it has a pretty darn slick look to it. Netscape's latest release, unfortunately, is not free to all users. Chapter 6 covers Netscape for you.

As far as the 30 or so other available Web browsers are concerned, most are basically Mosaic clones. Although you'll find some excellent products, these third-party browsers are used by a tiny percentage of Webbers. One of these browsers may, however, have that one particular feature you've been dying for. You'll find a brief survey of Cello, WinWeb, Spry AIR Mosaic, and other Windows browsers in Chapter 7.

If you're not using a SLIP account or The Internet Adapter (both discussed in the previous chapter) and if you don't have a networked connection to the Internet, then you only have two choices when it comes to browsing:

 You can use a text-based browser such as Lynx or WWW. These browsers are usually lightning-fast and come complete with all popular features. The big drawback is that there are, of course, no graphics, font sizes, styles, lines, or other special characters. This

drastically lowers the aesthetic value of a Web page. Chapter 3 covers the Lynx browser.

 The SlipKnot browser is your second option. It can display all basic multimedia Web pages, complete with text styles, fonts, and graphics. SlipKnot is not free, however, and it takes quite a long time for each Web page to load. Chapter 4 gives you a hands-on guide to the SlipKnot browser.

HELPFUL HANDS

In many ways, a Web browser acts like its own operating system. It knows how to handle most every type of data or resource stored over the Internet. If it can't display or otherwise use the data, it is smart enough to pass the data on to a program that can. These secondary programs are known as viewers because they're typically used to view fancy graphics or photos.

A browser can also perform many actions. It can, of course, hop around the Web using hypertext links. It can also usually send or retrieve a file from an FTP archive, send e-mail, or even act as a full Usenet newsreader. However, there are some things even a Web browser can't do itself yet, like telnet to remote computers. If you ask a browser to perform an action it can't handle, it'll usually run a more suitable program for you. These programs are known as external applications.









Viewers

Most Web browsers can't actually play sounds or run movies. However, a browser can retrieve the sound or movie file and automatically hand this file over to a viewer. In this way, viewers piggyback on the browsers, greatly extending their powers.

For example, most Windows browsers display any movies or animations using the Windows MPEG viewer. If the browser can't recognize the file, it'll generally read it in as plain text. Most browsers allow you to configure your favorite brand of viewer for each type of file.

External Applications

If you're using a Unix browser, most external applications should be a basic part of your Internet provider's service. If you're using your home PC with a SLIP line, however, there are some resources you might eventually want:

-  A telnet client to log you in to remote computers.
-  A newsreader to read or post Usenet news.
-  An FTP client to send or receive files over the Internet.
-  A talk client to let you hold a split-screen live chat with anybody else on the Internet.
-  An IRC client to get you into Internet Relay Chat (IRC), where thousands of users chat about thousands of topics.
-  A finger client to tell you about people who are currently logged onto the Internet.
-  An archie client to search for particular files over the Internet.
-  A gopher client to run the gopher menuing system.

Many Web browsers already have these programs built in. In some cases, however, you may want to use a separate client because it is faster, easier to use, or offers more commands. Most browsers allow you to designate any external applications you want for any given function.

More information about obtaining and using this Internet client software can be found in Appendix B, Other Software Sources.

The Talking MIME

The Web combines graphics, video, sound, text, hypertext, and more. Keeping track of what's what and how to use it can get confusing. After all, every file is stored in its own unique way—and there are a lot of possibilities.

Web servers use a coding scheme called the Multipurpose Internet Mail Extension (MIME) to differentiate different types of data. Whenever your browser asks for a particular file, the server lets the browser know what MIME type it'll be dealing with. Web browsers have the ability to recognize most popular types of files. Some files can be displayed on the fly. Others need to be sent to external viewers.

In most cases, the browser recognizes a file by its extension. Most every file on your computer has a specialized extension. For example, Microsoft Word files usually end in .DOC while plain text files end in .TXT. When you open a document using your favorite word processor, for example, your

software often guesses what format the file is in based on its extension. Web browsers use filename extensions similarly.

KEY NOTE: Most Web documents are stored on Unix operating systems, not on DOS. Unix extensions are not limited in length. For example, `the.history.of.the.world` is the name of a valid Unix file. Also unlike DOS, Unix is case-sensitive. This means that the file called `HELLO` is a different file from the one called `hello`. On Unix systems, Web hypermedia files generally end with the lowercase extension `.html`.

Since a Web document can have links not only to other Web documents but to graphics, sounds, text, Internet resources, and so on, it's important for both you and your Web browser to know what type of file is currently being dealt with. Table 2-1 lists some standard extensions and file formats.


Table 2-1 Standard extensions for file formats found on the Web

File Extension	File Format
Text	
<code>.txt</code>	Plain text file (extension is sometimes <code>.text</code> on non-DOS systems)
<code>.rtf</code>	Rich text format
Graphics	
<code>.gif</code>	CompuServe Graphics Interchange Format for any type of image
<code>.tif</code>	Targa Image File Format (extension is <code>.tiff</code> on non-DOS systems)
<code>.jpg</code>	Joint Photographic Experts Group (extension is <code>.jpeg</code> on non-DOS systems)
<code>.bmp</code>	Windows or OS/2 bitmap
<code>.tga</code>	Truevision Targa
<code>.pcx</code>	Windows Paint pixelmap
<code>.rle</code>	Run Length Encoded image
<code>.xbm</code>	A black and white bitmap from the X operating system
<code>.xpm</code>	A color pixelmap from the X operating system
<code>.ps</code>	PostScript printer-output file
Video / Animation	
<code>.mpg</code>	A Motion Picture Experts Group video or animated movie file (extension is <code>.mpeg</code> on non-DOS systems)
<code>.mov</code>	QuickTime movie (extension is <code>.moov</code> on non-DOS systems)
<code>.avi</code>	Audio/Video Interleave Windows multimedia movie
<code>.fli</code>	Autodesk Animator flick
<code>.gl</code>	Grasprt animation
Audio	
<code>.au</code> or <code>.snd</code>	Sun or NeXT audio

File Extension	File Format
.aif	Audio Image File Format (extension is .aiff on non-DOS systems)
.iff	IFF
.voc	Soundblaster Creative Voice
.wav	Windows Waveform
.sf	IRCAM
.mod or .nst	Amiga Mod
.snd or .fssd	Miscellaneous Amiga, PC, or Mac
.ul	US telephony
Compression	
.Z	A file compressed using compress (Unix)
.zip	A file compressed using pkzip
.gz	A file compressed using gzip (Unix)
.uue	An uuencoded binary file (Unix)

TELNETING TO A PUBLIC BROWSER

Maybe you're not sold yet. I mean, maybe the Web sounds kinda interesting, kinda fun, but you just aren't sure what it is that makes the Web so talked-about, so hyped. Before you buy a SLIP account or The Internet Adapter, you want a test drive.

 **NOTE:** Telneting to a public browser is also useful if you want to access a page's text very quickly.

If you already have an Internet account, you should run the Lynx or WWW browser, described in the next chapter. At your Unix prompt, try typing:

```
Lynx
```

```
or
```

```
www
```

If nothing happens with either of these commands, you can grab a very basic tour of the Web by using telnet to jump to a public Web browser. Almost every Unix account in the world offers the telnet command. Public WWW sites are listed in Table 2-2. Type telnet followed by the name of the site. For example, to log in to the Funet site, which contains a public Lynx browser, you would type:

```
telnet info.funet.fi
```


When asked to

Login:

type

www

When you telnet to a remote site, you may be asked to

Select interface?

in which case you can try typing

lynx

to access the Lynx text-based browser.

A special beginner's page usually appears, as shown in Figure 2-1. There are no mouse or cursor controls. To access a hypertext link, type in the number or word that appears in brackets. You can also type the word back to go to a previous page, or quit to exit.

Be warned: The generic WWW browser is the most ill-mannered, simple-minded browser you could find, but it should give you some idea of how the Web works. Lynx is a little more functional, but is still limited to simple text only.

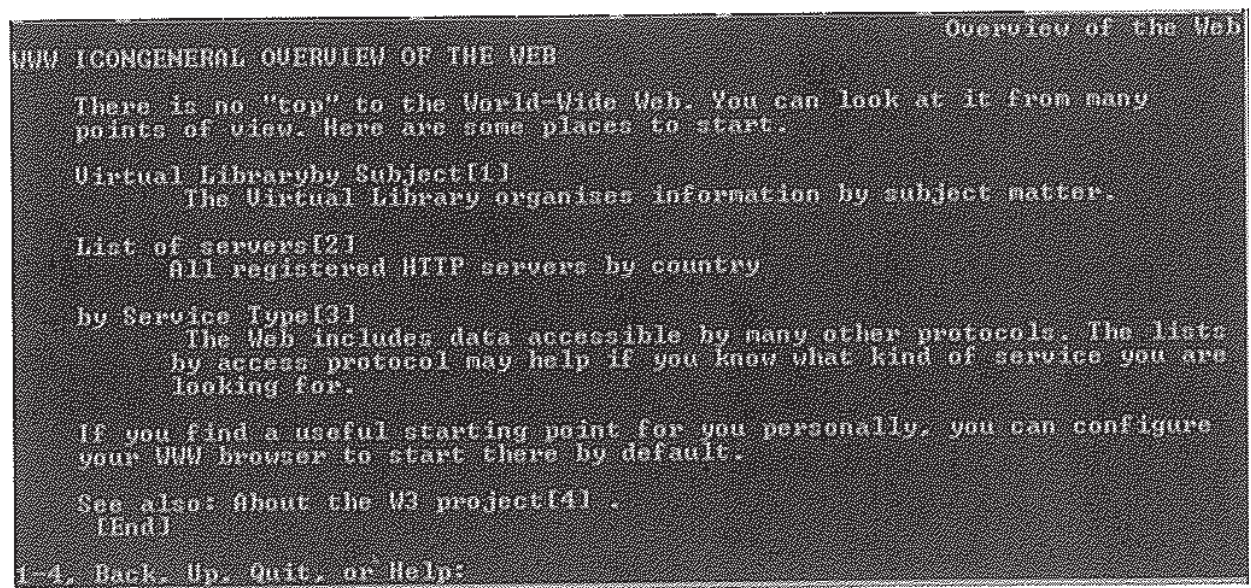


Figure 2-1 The simplest Web browser in the world

Table 2-2 Publicly accessible Web browsers

Browser	Public Site
WWW	fatty.law.cornell.edu
WWW	www.edu.tw
WWW	fserv.kfki.hu
WWW	telnet.w3.org
WWW	www.njit.edu
Lynx	info.funet.fi
Lynx	www.cc.ukans.edu
Generic	www.huji.ac.il
Generic	sun.uakom.cs

You can now flip through this book, especially Chapter 8, What Can I Do? If you see a page that looks interesting, note its URL. Telnet to a public browser, press G, and then type in that exact URL. The page is loaded up for you. These simple browsers may require you to type go URL. For instance, to jump to <http://www.smartpants.edu/> you could type

```
go http://www.smartpants.edu/
```

WWW VIA E-MAIL

If you're really in a hurry, or if the telnet command doesn't work for you, you can have the contents of a Web page mailed to you. You'll need to know the exact URL of the document you want. There are two addresses you can try. To send mail, type the following at your Unix prompt:

```
mail server@mail.w3.org
```

or

```
mail listserv@info.cern.ch
```

You should leave the subject blank. As the body of your message, type
send [URL]

and replace [URL] with the actual URL of the page you want to see. The CERN site automatically processes the request, loads up the page, and sends the text directly to your mailbox.

Each hypertext link in the document will be followed by a bracketed number, such as [3]. At the bottom of the mail message, there'll be a listing of each number and its corresponding URL. If you want to follow one of the URLs, just send off another message.

BROWSING WITHOUT THE INTERNET ALTOGETHER

Maybe you haven't subscribed to your SLIP Internet account yet and want to browse through some .html files on your hard disk. Or maybe you're designing HTML Web pages on your local PC and want to check them out. Since most browsers (Cello, Netscape, and so forth) communicate via TCP/IP stacks, you'll need to make them think that your machine is actually connected to a network.

Some companies are also investigating ways of creating small, private Webs. A Local Area Web is not distributed via the Internet, but over the company's own LAN. Such companies may want to use their own networking software. However, they could also set it up so employees could use Mosaic, Netscape, or any other browser to distribute confidential company information in a sleek hypermedia format.

Stand-Alone Browsers

The latest versions of Mosaic and SlipKnot are stand-alone browsers. When you run them, they load up just fine even if you're not using socket or TCP/IP software. You can then use the File, Open command to bring up any document or file stored on your hard disk. Giving browsers a stand-alone mode seems to be the trend, since many people use their browsers to look at local Web pages.

Minimizing Winsock

Many other browsers, however, give you a strange error message if you try to run them with no other Internet software installed:

```
Application Error
Call To Undefined Dynalink
```

However, you don't actually have to be connected to your Internet provider to use a browser. Just run the Winsock program in the background by selecting the control box in the upper left corner of the window and choosing the Minimize item. Most browsers should run just fine.

Nullsock

If you don't have a modem or network connection at all, you can still play with Web browser programs if you install a fake socket library. This fake library, known as nullsock, works in lieu of Winsock's true WINSOCK.DLL.

The nullsock file is available at the site listed in Appendix D. Download and copy the WINSOCK.DLL file to your Windows system directory:

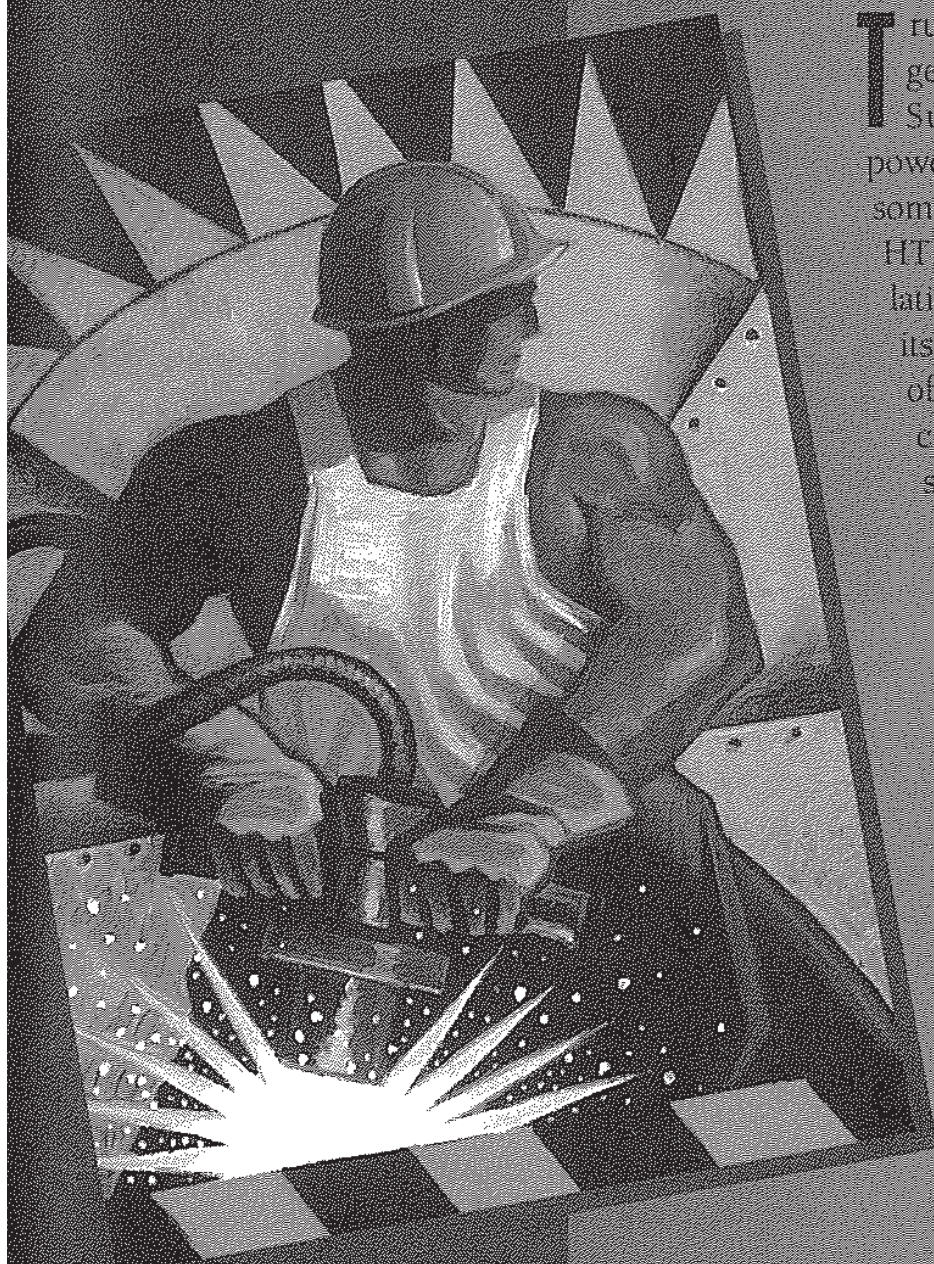
`C:\WINDOWS\SYSTEM`

Be sure to erase Winsock's real WINSOCK.DLL. You can now run Winsock along with your favorite browser or other Internet software. Winsock, of course, is crippled; it can't do much more than initialize.

WHAT NOW?

You're now ready to install and use a browser on your own machine. The next five chapters cover the most popular types of browser programs. Chapter 3 begins with an overview of Lynx, a Unix browser.

18



True to its name, Java will wake you up, get you wired, and make you shake. Sure, simple HTML Web pages are powerful in their own right. But there are some dire limitations. You can't coerce an HTML page to actually figure out a calculation or respond to a question or change its look on-the-fly. Even with the power of Javascript, there are still limits—especially when it comes to graphics or sound. You can't easily cycle through a fast animation, or click on a three-state button, or play a graphical game. Java turns the Web into a full-functioned software platform.






The Java libraries are like a special version of the powerful C++ programming language, specially tuned to zoom over the Internet. Java is all sorts of things: object-oriented, network secure, expandable, etc. If you're the programming type you can use Java to write full spreadsheets,

databases, 3D graphics packages, games, etc. There are several great books about Java out on the market, such as the Waite Group's *Java Primer Plus*.

But even better, you can use existing Java programs, which are scattered like bread-crumbs all throughout the net (and on the enclosed CD), to add spice, power, and heavy interactivity to your Web pages. You can use many Java programs as-is—just drop them onto your Web server and unleash their power. Others require only simple program changes, which any amateur programmer can easily perform.

JAVA'S JOIT

Java was created as a powerful backbone to help with the following:

-  **Interactivity:** You can use the full range of mouse or keyboard movements to perform actions. This means you can have pull-down menus, buttons, dialog boxes, sliders, smart text fields, custom checkboxes or radio buttons, and most any other type of input. This adds a great deal to the simple interactivity allowed by HTML forms.
-  **Animation:** Java allows a variety of simple graphics effects. These range from bringing up a series of images in fluid animation, playing a small video, to fancy graphical graphics, and live-rendered 3D graphics.
-  **Sound:** Java has sound libraries.
-  **Math:** Stock portfolios, other types of statistics, spreadsheets, and calculators.
-  **Games:** Folks have written Java versions of Missile Command, Tetris, Solitaire (see Figure 18-1), baseball, poker, roulette, pong, hangman, trivia puzzlers, and other simple games. These little applets are more than fun time-wasters; they can allow you to use similar interfaces to control real-life robots, numbers, facts, or other machinery.

THE BIG APPLLET

Programs written in Java are called *applets*. The way Java works is simple: You write a program. It's compiled into a small package called *byte code* which works the same on any computer platform. A well-written Java

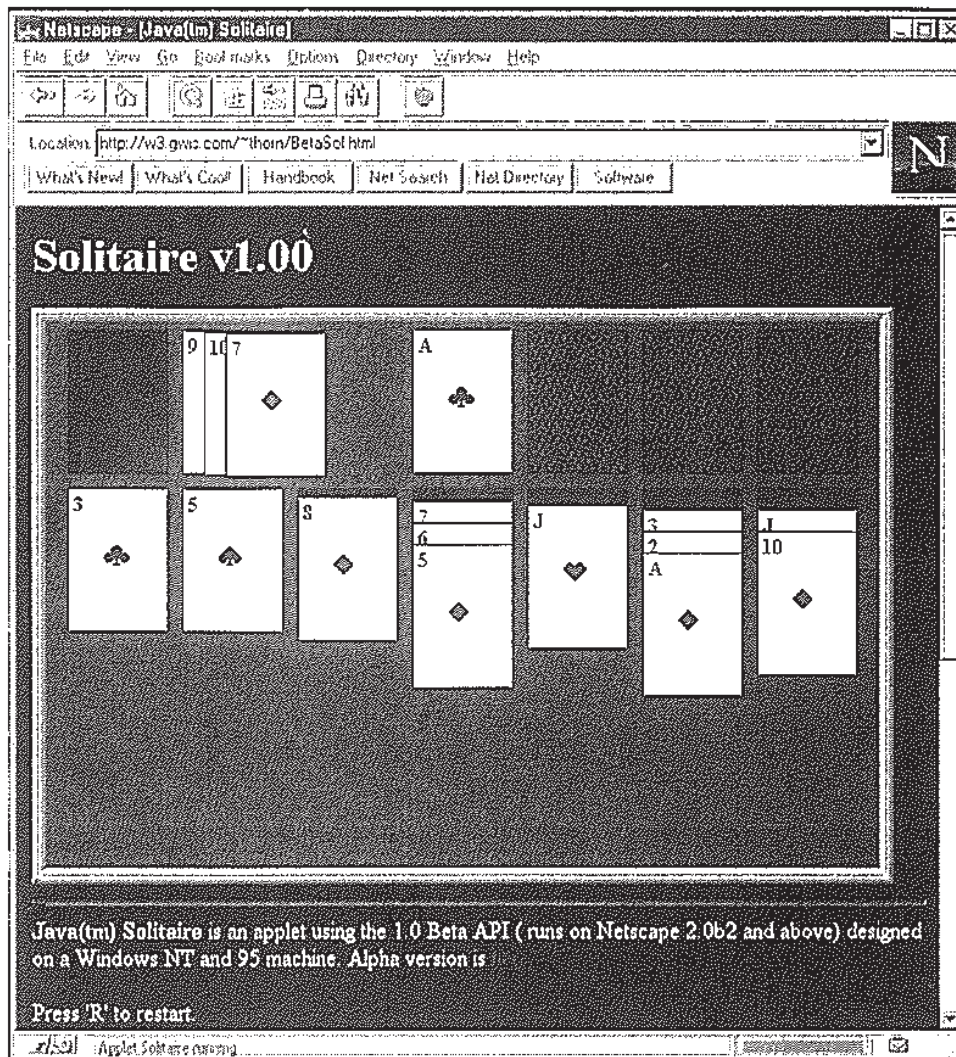


Figure 18-1 Lonesome tonight? Fire up Netscape and play a game of Java solitaire

applet, then, will work on any Java-supported browser, whether it's on a Mac or a PC or a Unix machine.

You use Netscape's APPLET element to drop Java applets smack onto your Web pages. Applets will then be downloaded any time that Web page is accessed.

The applet is then loaded and run on the user's home computer. This means you don't need to worry about wasting the server's computing power, time, or energy. Since everything happens locally, multimedia effects and interactivity are much, much faster. The applet simply requests pictures, info, or other data from the server as it needs it. It's a perfect little package: It plugs right in and gives Netscape extra abilities and powers.

So which browsers support Java? Glad you asked.

The ABCs of Alpha and Beta

There are two versions of Java: Alpha and Beta. Beta is the latest version, and allows special extensions and parameters that alpha cannot handle.

The alpha version lacks many of the functions, scalability, and stability of the beta version. Unfortunately, applets written in the beta format will not work with alpha browsers and vice versa. Eventually the official first version of Java will be released. You should expect major releases of Java every few months.

An Applet a Day...

The only browsers that currently support Java applets are Netscape 2.0 and Sun's HotJava. In addition, Microsoft is planning to release a Java-enabled version of Internet Assistant any day now.

HotJava was written especially for Java applets, but it only supports the old alpha standard. Since HotJava is rarely used as a day-to-day browser, the alpha standard is quickly becoming outdated. This chapter, therefore, focuses entirely on the beta standard, which Netscape handles like a charm.

HOTJAVA

The HotJava browser is the world's first Java-compatible Web browser. Developed by Sun Microsystems, this browser is available for Windows 95 or NT and Solaris; a Mac version is in the works. If you're interested, you can check out the HotJava browser at:

<http://java.sun.com/>


NETSCAPE 2.0

The latest version of Netscape supports Java fully. See Chapter 6, Netscape Atlas for information on downloading and installing Netscape. Or check out the following page for more details:


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


PLACES TO GET APPLETS


Applets don't grow on trees, but downloading them is pretty darn easy nonetheless. Thousands of public-domain applets are available throughout the Web, ready for you to use. Many of them include sample source code, which lets you fine-tune the applet to fit your Web page goals.

 **NOTE:** Be sure to give credit where it's due! If an applet is copyrighted, be sure to get permission before you use it. A lot of time and energy goes into writing applets, and besides, there's never any harm in common courtesy. An applet's author might even give you some hints on how to modify the applet to better suit your needs.

You can download applets from the following sources:


 JARS: The Java Applet Locator is a place where only the most useful applets are stored. <http://www.surinam.net/java/jars/category.htm>


 Gamelan: The largest archive of Java applets to date. You can find applets in the following categories: Arts and Entertainment, Business and Finance, Educational, Miscellaneous, Multimedia, Network and Communications, News, Programming in Java, Special Effects, Utilities, and a fine list of Java-rich Web Sites. If it isn't here, it probably ain't anywhere. <http://www.gamelan.com/>


 Sun: These basic applets all come with source code, and are a must if you plan to write your own applets. Programmers are invited to borrow the source code from these applets to suit their own needs. <http://www.javasoft.com/applets/applets.html>


Applets from the Java Products Group

The enclosed CD-ROM contains the following applets, courtesy of Sun Microsystems. Visit the URLs to get updates, instructions, or to see examples of the applets in action.

 Hangman: A full version of every third grader's favorite game. <http://java.sun.com/applets/applets/Hangman/index.html>

 Catch the Jumping Box: Watch a box leap around the screen. Try to nab it with your mouse. <http://java.sun.com/applets/applets/JumpingBox/example1.html>

 Tic Tac Toe: A version of the world's simplest game. Complete with fun sound effects. <http://java.sun.com/applets/applets/TicTacToe/example1.html>

 Multilingual Word Match Game: A neat game that lets you line up pictures with words. You can play in English, German, French,

Italian, Hebrew, Japanese, and an number of other languages.
<http://java.sun.com/applets/applets/WordMatch/index.html>

HTML Bubbles: A neat applet where bubbles rise and rise and rise.
<http://java.sun.com/applets/applets/Bubbles/index.html>

HTML ImageLoop: This is the world's original animation applet.
<http://java.sun.com/applets/applets/ImageLoop/index.html>

HTML Sound Player: A simple audio applet.
<http://java.sun.com/applets/applets/AudioItem/index.html>

HTML A General-Purpose Animator: Yet another animation applet, with advanced features.
<http://java.sun.com/applets/applets/Animator/example1.html>

HTML Blinking Text: Watch text blink.
<http://java.sun.com/applets/applets/Blink/example1.html>

HTML Bouncing Heads: A neat little box where the heads of Sun programmers bounce around like pinballs.
<http://java.sun.com/applets/applets/BouncingHeads/example1.html>

HTML Under Construction: A little animated icon with Duke (Java's mascot) jackhammering, complete with sound.
<http://java.sun.com/applets/applets/UnderConstruction/example1.html>

HTML Scrolling Images: A neat applet that lets you scroll images.
<http://java.sun.com/applets/applets/ScrollingImages/example1.html>

HTML Tumbling Duke: An animation of Duke doing cartwheels across the screen.
<http://java.sun.com/applets/applets/TumblingDuke/example1.html>

HTML Live Feedback ImageMap: An image-map with three-state buttons, sounds, and more.
<http://java.sun.com/applets/applets/ImageMap/example1.html>

HTML 3D Model Viewer: Rotate an actual 3D model.
<http://java.sun.com/applets/applets/WireFrame/example1.html>

HTML 3D Chemical Models: View the structures of chemicals.
<http://java.sun.com/applets/applets/MoleculeViewer/example1.html>

- HTML** Fractal Figures: A small fractal, zoom in further and further to see stranger and stranger shapes.
<http://java.sun.com/applets/applets/Fractal/example1.html>
- HTML** Animated Sorting Algorithms: Watch graphically as an algorithm takes a list of numbers of sorts them.
<http://java.sun.com/applets/applets/SortDemo/example1.html>
- HTML** Clock: What time is it?
<http://java.sun.com/applets/applets/Clock/index.html>
- HTML** Spreadsheet: A live spreadsheet. Fill in some numbers, and this applet will calculate the sums automatically.
<http://java.sun.com/applets/applets/SpreadSheet/example1.html>
- HTML** Bar Chart: Create a bar chart on-the-fly.
<http://java.sun.com/applets/applets/BarChart/example1.html>
- HTML** GraphLayout: Create a graph.
<http://java.sun.com/applets/applets/GraphLayout/example1.html>
- HTML** ArcTest: Learn how to create an arc using Java.
<http://java.sun.com/applets/applets/ArcTest/example1.html>
- HTML** CardTest: Learn how to swap cards.
<http://java.sun.com/applets/applets/CardTest/example1.html>
- HTML** DitherTest: Learn how to dither graphics.
<http://java.sun.com/applets/applets/DitherTest/example1.html>
- HTML** DrawTest: Learn techniques of drawing to the screen.
<http://java.sun.com/applets/applets/DrawTest/example1.html>
- HTML** GraphicsTest: Learn how to maximize Java graphics.
<http://java.sun.com/applets/applets/GraphicsTest/example1.html>
- HTML** ImageTest: Learn how to load in images.
<http://java.sun.com/applets/applets/ImageTest/example1.html>
- HTML** SimpleGraph: Learn how to create a simple graph.
<http://java.sun.com/applets/applets/SimpleGraph/example1.html>

USING EXISTING APPLETS

Applets always end with the extension `.class`. To use an applet, simply copy the `.class` file onto your Web server, in the same directory as your Web pages.

You can now use the `<APPLET>` and `</APPLET>` tags within your HTML document to call up the applet. Use the `CODE` attribute to tell Netscape the name of the applet. For instance, to embed an applet called `AnimatedLogo.class`, you could use

```
<APPLET CODE="AnimatedLogo.class">
</APPLET>
```

If the applet isn't in the current directory, you can still use it. Simply use the `CODEBASE` attribute to specify where the applet is located. You can use an applet anywhere on the Web in this way. For example, if the `AnimatedLogo` applet is stored at Smarty Pants University, you could write a tag similar to the following:

```
<APPLET CODEBASE="http://www.smartypants.edu/java/applets/" CODE="AnimatedLogo.class">
```

As an applet first loads, it appears as a gray rectangle. You can specify how big you want this rectangle to be by using the `HEIGHT` and `WIDTH` attributes. You can also include the `VSPACE` and `HSPACE` attributes to create a vertical and horizontal margin around the applet, respectively.

You can also use the `ALIGN` attribute to place the applet exactly where you want it. `Align` takes any valid Netscape alignment value, just like the `` element (see Chapter 12, Graphics, for more details): *left*, *right*, *top*, *texttop*, *middle*, *absmiddle*, *baseline*, *bottom*, or *absbottom*.

For example, if you want your applet to be 50 pixels wide, 75 high, aligned to the right of the page, with a right and left border of 5 pixels and a top and bottom border of 10 pixels, you would write:

```
<APPLET CODE="AnimatedLogo" WIDTH=50 HEIGHT=75 HSPACE=5 VSPACE=10 ALIGN=right>
</APPLET>
```

Depending on the applet, you may need to specify special parameters or inputs containing your own custom sound files, graphics, or URLs. These special instructions appear between the `<APPLET>` and `</APPLET>` container, in the guise of the `<PARAM>` element.

Each input variable in a Java applet should have its own name and value type. An applet may want a number, a list of numbers, a text string, a valid URL, a floating point number, or any other type of information. The best way of determining how an applet's parameters are supposed to be passed is to find a sample page that uses that applet. Sneak a peek at the source code and notice how the `<PARAM>` tag is laid out. For instance, the `AnimatedLogo`

applet might take two values: the name of the first animation graphic (*animSource*) and the number of frames in the animation (*numFrames*):

```
<APPLET CODE="AnimatedLogo" WIDTH=50 HEIGHT=75>
<PARAM NAME=animSource value="logo.gif">
<PARAM NAME=numFrames value=20>
</APPLET>
```

Some Java applets go so far as to communicate with each other. If you plan on using more than one applet on a page, you may want to name each applet using the NAME attribute to make it unique:

```
<APPLET CODE="AnimatedLogo.class" WIDTH=50 HEIGHT=75 NAME="Logo"></APPLET>
```



STYLE TIP: If a browser doesn't support Java, it will simply skip over this tag. Some non-Java browsers will, however, print whatever text follows the ALT (alternate text) attribute, as with the <SRC> element. For instance, you could have a java tag similar to the following:


```
<APPLET CODE="AnimatedLogo" WIDTH=50 HEIGHT=75 ALT="An Animated Logo"></APPLET>
```

An even better solution is to put some explanatory text between the <APPLET> and </APPLET> tags. This text will be shown by all non-Java browsers, but ignored by Netscape:


```
<APPLET CODE="AnimatedLogo.class" WIDTH=50 HEIGHT=75>
[If you had a Java-Compatible Browser You Would Be Seeing An Animated Logo Here. Too bad!]
</APPLET>
```


Some Applets to Bite Into


This section covers the following applets, all of which can be found in the Java directory of the enclosed CD-ROM.

 Animation: Perform simple animations, creating a small animated logo, movie, or neat effects on your Web page.

 Sound: Add audio to your Web page.

 Nervous Text: Jiggle some of the text on your Web page to draw attention to it.

 Scrolling Marquee: Add a scrollable line to your Web page.

 Image Map: Make image maps with three-state buttons, sounds, and other neat interactive effects.

These useful applets are ready to use. All you need to do is fill in the details and the applets will take care of the rest.

Animation

Sun's Advanced Animation applet lets you perform all sorts of neat animations. You can create an animated logo, a little film, an animated diagram, or any other sort of moving graphic you desire. This applet will display each frame cleanly and without any bad side-effects.

The applet not only displays a series of frames, it lets you specify the order of the frames, how the animation should loop, the location of the frames, and how fast the animation should occur. This applet also lets you specify a full soundtrack, or play certain sounds alongside certain frames. There's a bunch of other neat features, too. For instance, you can drop a background image behind the animation. You can also create a loading image that is displayed while the frames are loading into memory.

Setting Up the Animation Frames

Your animation can be as many frames as you wish. Create the animation using any program you desire. Three-dimensional graphics packages such as Animator Pro or 3D Studio can do a great job at helping you design some fantastic moving images. Or you can simply use a tool like Windows Paintbrush or Photoshop to sketch the animation yourself.

Save each frame of the animation as a JPEG or GIF graphic. Each frame does not need to be the same size, though the animation may not run smoothly if you don't use uniform frame sizes. Name the bitmaps consecutively by tacking on a number to the end of the name. For example, if you have a five frame animation of a mouth chattering, as in Figure 18-2, name the graphics *Mouth1.gif*, *Mouth2.gif*, *Mouth3.gif*, *Mouth4.gif*, and *Mouth5.gif*.

The applet is called *Animator.class* and is located in the `\Java\Animator` directory of the enclosed CD-ROM. You will also need the *ImageNotFoundException.class* and *ParseException.class*.

Put the Animator applets and all the images in the same directory as your Web pages.

How to Use the Animator

Use the `<APPLET>` tag to drop the Animator anywhere you like within your Web page. Set the `WIDTH` attribute to the pixel width of the widest frame of your animation. Set `HEIGHT` to the tallest height. For example, if your *Mouth1.gif* is the biggest frame, with a height of 100 and a width of 150, you would begin the Animator applet as follows:

```
<APPLET CODE="Animator.class" WIDTH=150 HEIGHT=100>
```

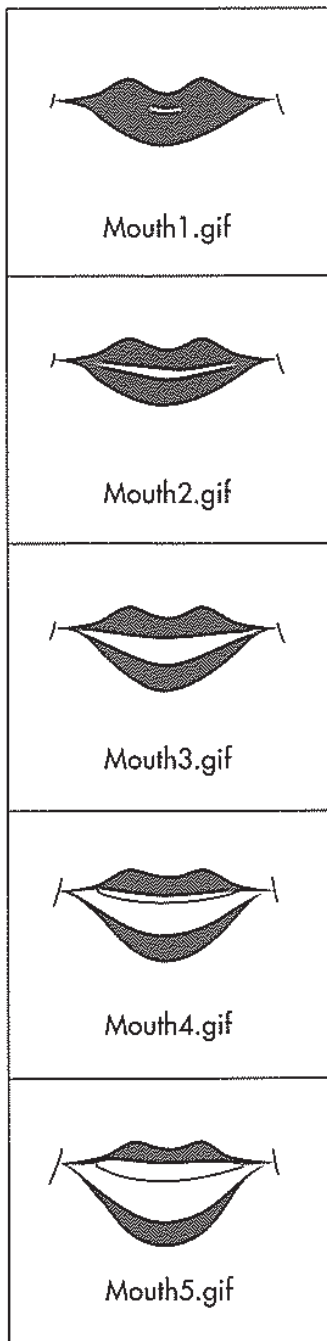



Figure 18-2 Talkin' up a storm

You now need to set up the parameters, to define how the animation works. There are many parameters:

HTML DIRECTORY: Which directory are the animation frames stored in?

HTML **STARTUP**: An image that should be displayed when the applet first loads, while the frames are loading into memory.

HTML **BACKGROUND**: An image that the frames should be painted against.

HTML **STARTIMAGE**: The number of the first frame in the animation.

HTML **ENDIMAGE**: The number of the final frame.

HTML **NAMEPATTERN**: A pattern that will help the applet determine the name of each frame, using the values in **STARTIMAGE** and **ENDIMAGE**. If you have less than ten frames of animation, simply specify the full name or URL of the images you wish to load then substitute the symbol *%N* where the numeral should be. For example, if your frames are named *Mouth1.gif* to *Mouth5.gif*, your pattern would be: "*Mouth%N.gif*". If you have more than ten frames of animation, you can specify that there are two or more digits of numerals by specifying the number of digits in place of *N*. For instance, if you have a set of frames called *Frame001.jpg* to *Frame100.jpg*, you could set the pattern as follows: "*Frame%3.jpg*", since there are three digits tagged onto the end of each animation file.

HTML **PAUSE**: The number of milliseconds to pause between each image. This is 100 by default.

HTML **PAUSES**: If you'd like to pause after a certain frame, you can specify the millisecond delay on a frame-to-frame basis. Separate each value by the pipe symbol (*|*). For example, if you want to pause 100 milliseconds after the first frame, the default value (specified by the **PAUSE** parameter) after the second frame, 200 after the third frame, 300 after the fourth, and 400 after the fifth, you would set the value to: "*100|200|300|400*".

HTML **REPEAT**: Should the animation loop? By default this is set to "true". If you don't want the animation to repeat, set this to "false".

HTML **POSITIONS**: Set the (X,Y) position for each frame, using the syntax: *X@Y*. Separate each frame value with the pipe symbol (*|*). Keep this value blank if you simply want to use the previous frame's position.

For instance, to position the first frame at (100,100) and the fifth frame at (150,150), your value would be: "100@100|||150@150".

HTML IMAGES: In what order should the frames play? You can create any order you like, simply separating each frame number by the pipe symbol (|). For example, to play the animation backwards and then forwards you could use: "5|4|3|2|1|2|3|4|5".

HTML SOUNDSOURCE: The directory that holds the audio files.

HTML SOUNDTRACK: An audio file that should be played all throughout the animation.

HTML SOUNDS: If you wish to play a specific sound file as a certain frame is shown, you can set this here. For instance, to play the sound *Scream.au* during the fourth frame of the animation, you would use the value: "|||scream.au|".

You need only use the parameters you want. The rest of them will take default values. For instance, if you want to play a simple mouth animation straight through, simply use the STARTIMAGE and ENDIMAGE parameters, along with the NAMEPATTERN parameter, as in the following code:

```
<APPLET CODE="Animator.class" WIDTH=150 HEIGHT=100>
<PARAM NAME="STARTIMAGE" VALUE="1">
<PARAM NAME="ENDIMAGE" VALUE="5">
<PARAM NAME="NAMEPATTERN" VALUE="Mouth%N.gif">
</APPLET>
```

NOTE: If you set ENDIMAGE to a number lower than STARTIMAGE, the animation will be played backwards.

If you want to play a special mouth animation—the mouth opens and then closes—you can specify the frames explicitly using the IMAGES parameter. You can add, say, a very short pause of only 10 milliseconds between each frame. You can then add in some special extras: Put a lovely Mountain lake graphic in back of the mouth. Bring up a bitmap at the beginning that has the words: "Please Wait!" on it. Have a voice say, "Hello!" whenever frame 5 loads. You can accomplish all this with the following:

```
<APPLET CODE="Animator.class" WIDTH=150 HEIGHT=100>
<PARAM NAME="STARTUP" VALUE="wait.gif">
<PARAM NAME="BACKGROUND" VALUE="lake.gif">
<PARAM NAME="PAUSE" VALUE="10">
<PARAM NAME="SOUNDS" SOUNDS="|||Hello.au">
```

Continued on next page

Continued from previous page

```
<PARAM NAME="IMAGES"
VALUE="Mouth1.gif|Mouth2.gif|Mouth3.gif|Mouth4.gif|Mouth5.gif|Mouth4.gif|Mouth
3.gif|Mouth2.gif|Mouth1.gif">
</APPLET>
```

Sound

The `AudioItem` applet lets you drop a sound into the Web page. This sound is loaded as soon as the Web page is loaded and will play as soon as it is in memory.

Simply drop the `AudioItem.class` file into the same directory as your Web page. Specify the name of the sound using the “snd” parameter. You can add more than one sound to a page by separating each sound with the pipe symbol (`|`). For instance, to have a page say, “Hello!” and then “Welcome to my page!” you could write:

```
<APPLET CODE="AudioItem" WIDTH=15 HEIGHT=15>
<PARAM NAME="snd" VALUE="Hello.au|Welcome.au">
</APPLET>
```

Blinking/Nervous Text

The `Blink` and `NervousText` classes let you spice up your Web page with fancy text effects. The `Blink` applet will take a series of words and make the words go crazy: text'll change colors as random words disappear and then reappear, as in Figure 18-3. If you need to draw attention to something, this is a heck of a way to do it.

To use the `Blink.class`, copy it to your Web server. And then add the following lines to your Web page:

```
<APPLET CODE="Blink.class" WIDTH=300 HEIGHT=100>
<PARAM NAME=lbl VALUE="The text here will blink like nuts, man!">
<PARAM NAME=speed VALUE="4">
</APPLET>
```

Put the text you'd like to blink as the value of the `lbl` parameter line. You can also adjust the speed of the blinking by setting the value of the `speed` parameter.

The `NervousText.class` applet is similar. Instead of blinking text, however, it'll appear as if it's imbibed too much coffee—it'll vibrate and shake, fast enough to seem weird, but slow enough to read, as in Figure 18-4. This is accomplished by dropping the `NervousText.class` onto your Web server and then specifying the text you'd like with the `text` parameter.

```
<APPLET CODE="NervousText.class" WIDTH=300 HEIGHT=50>
<PARAM NAME=text VALUE="Man oh man am I noivous!">
</APPLET>
```

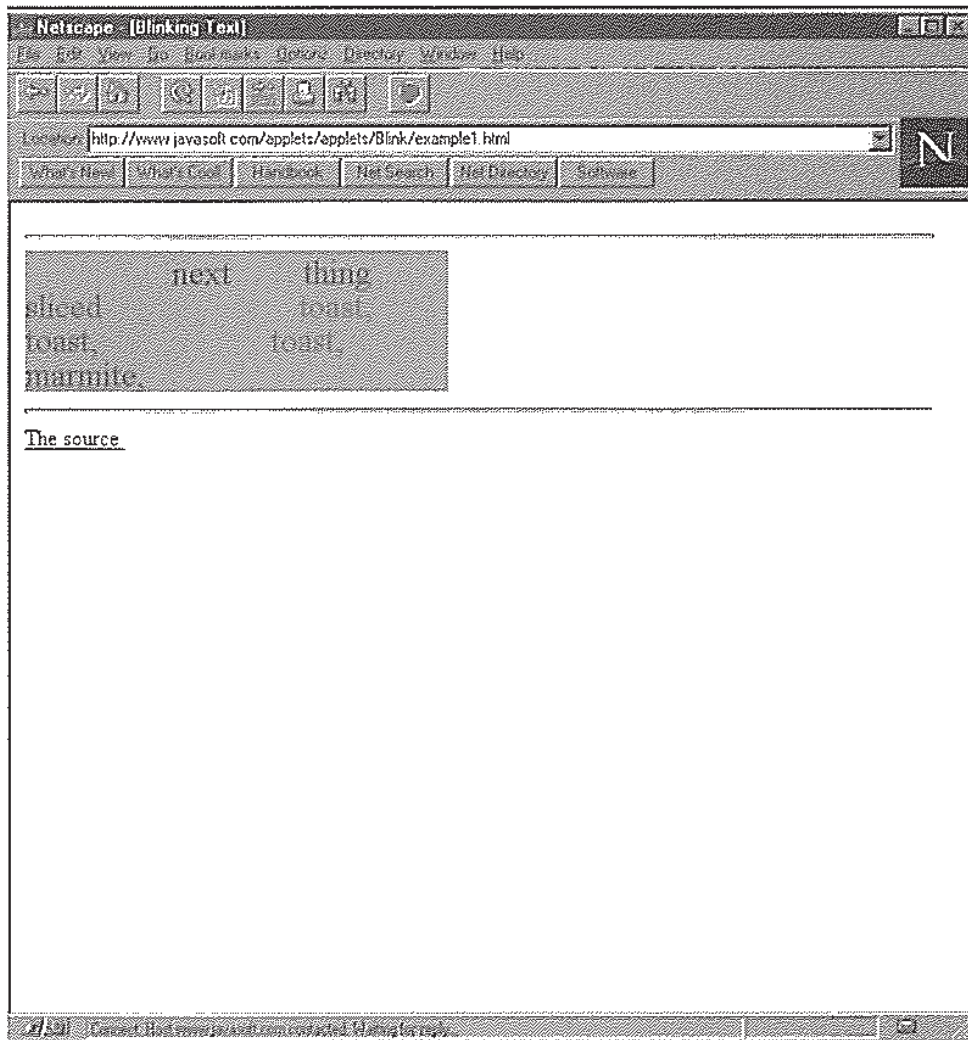


Figure 18-3 An example of blinking text

Scrolling Marquee

The marquee class scrolls a message across your Web page screen. This is a fantastic way to grab someone's attention, or to put a lot of information in a little space. Many pages scroll advertisements across the top of their page using this technique. Other people use it just to make their page stand out and seem snazzy, as in Figure 18-5.

You can grab the Marquee class at:

<http://www.quebec.net/~oliver/marquee/Marquee.class>

The class takes the following parameters:

HTML label: The text you'd actually like to scroll. This string can be as long as you wish.

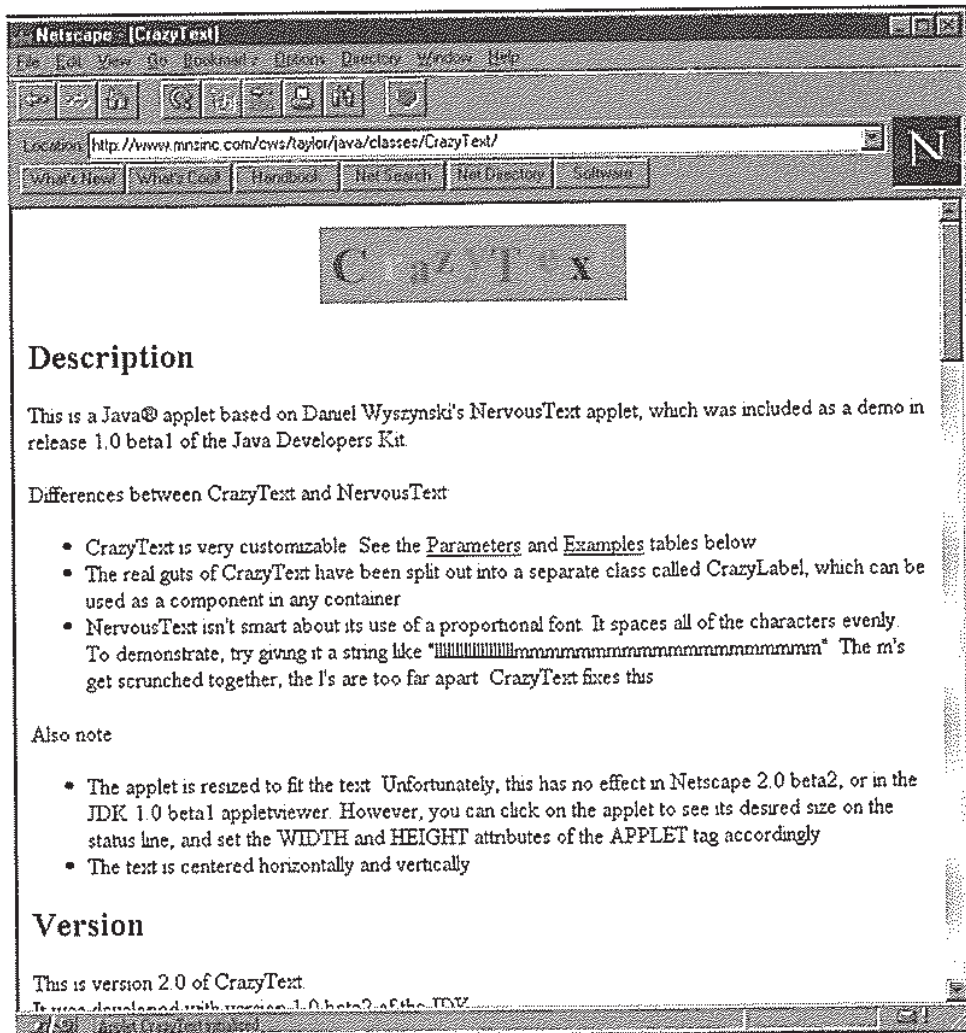


Figure 18-4 Text anxiety

HTML font: Which font family should be used? Some standard options are: *TimesRoman*, *Courier*, or *Helvetica*.

HTML ptsize: What should the text's point size be? Standard paper type is 12 points.

HTML step: How fast should the text scroll? Start with an average value of 10 and then make it faster or slower depending on how fast you think your audience can read.

There are six optional color parameters that you can use to customize the look of the marquee. Set colors by specifying each value of an RGB triplet. You can refer to the chart in Chapter 12, Graphics.

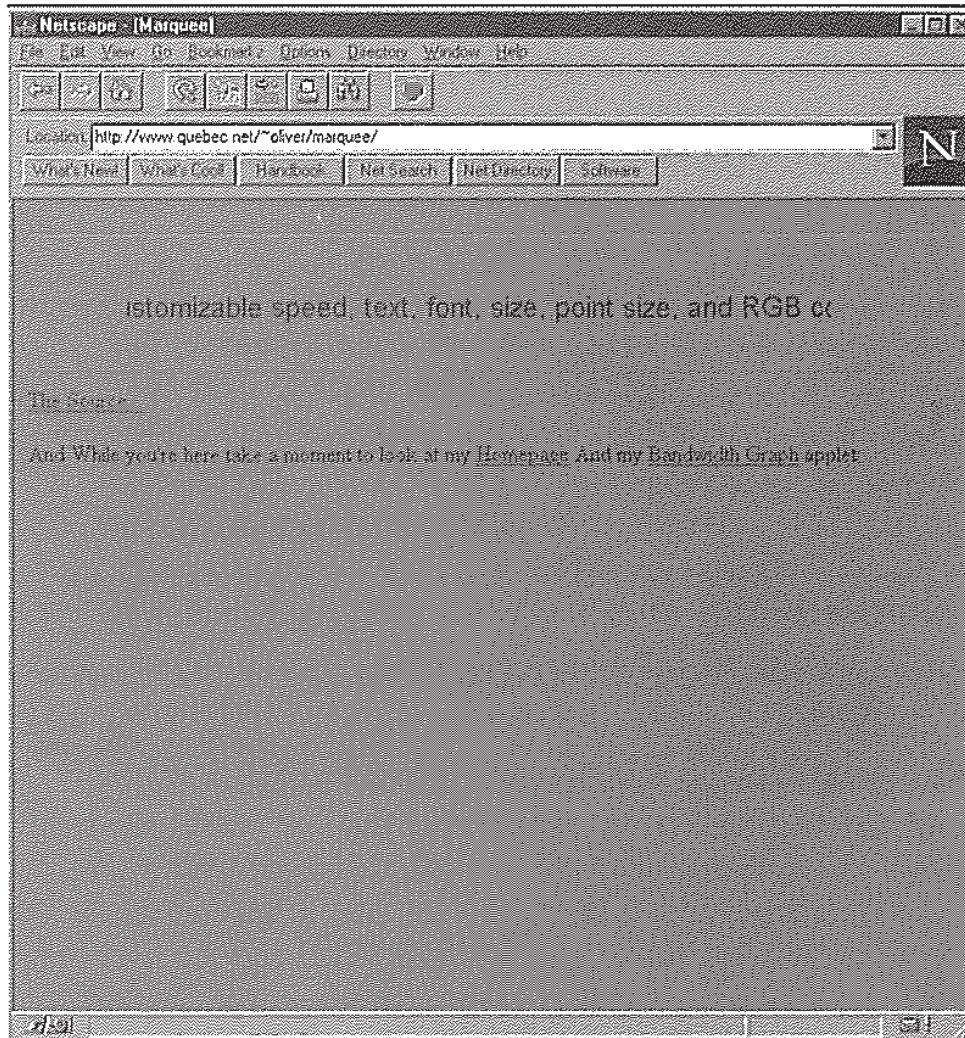


Figure 18-5 Text scrolling across the screen

HTML fred: The foreground's red value. This takes a number from 0 to 255. Use 255 if you want to pump up the red to the maximum. Use 0 if you want no red at all.

HTML fgreen: The foreground's green value.

HTML fblue: Foreground blue value.

HTML bred: The background red value.

HTML bblue: The background's blue value.

HTML bgreen: Background's green value.

So, if you wanted to scroll the message: "Howdy! Welcome to my home page... If you explore here long enough you'll find all sorts of fantastic and wonderful things... Peace... Love..." in green type against a black background you could use the code:

```
<CENTER>
<APPLET CODE="Marquee.class" WIDTH=500 HEIGHT=50>
<PARAM NAME=label VALUE="Howdy! Welcome to my home page... If you explore here long
enough you'll find all sorts of fantastic and wonderful things... Peace... Love...">
<PARAM NAME=font VALUE="TimesRoman">
<PARAM NAME=ptsize VALUE="24">
<PARAM NAME=step VALUE="10">
<PARAM NAME=fred VALUE="10">
<PARAM NAME=fgreen VALUE="255">
<PARAM NAME=fblue VALUE="10">
<PARAM NAME=bred VALUE="0">
<PARAM NAME=bblue VALUE="0">
<PARAM NAME=bgreen VALUE="0">
</APPLET>
</CENTER>
```

Image Maps

Sun's Image Map applet is a feature-packed class that lets you create highly interactive image maps. When you roll your mouse cursor over a standard HTML image map nothing much happens. You just point and click. But Java Image Maps come alive—some areas animate when you roll over them. Other areas highlight. Some parts may print a message. Other parts may play a sound effect.

Place the ImageMap.class file into the same directory on your server as your Web page. Tell the class which GIF image you'd like to use as the image map itself using the *img* parameter. For instance:

```
<PARAM NAME=img VALUE="imagemap.gif">
```

You can also set the *highlight* parameter, which lets you tell the applet how to highlight any hot regions. Simply set the value to "brighter" followed by the percentage of highlight to apply. For example, if you want a hot region in the imagemap to be lit up 50 percent brighter, use

```
<PARAM NAME=highlight VALUE="brighter50">
```

You can now define each hot area. Number the first area with a parameter called "area1", the second area as "area2", etc. The value of each hot area consists of several parameters, each separated by a comma. The first parameter is the type of area you'd like to create. The second and third parameters are the top corner's x and y positions of the hot area. The fourth and fifth parameters are the bottom corner's x and y positions.

You can create the following area types:

HTML *SoundArea*: Use this parameter if you want to play a sound. Specify the name of the au-format audio file. For instance, to play the *boom.au* sound any time the area from (260,180) to (120,60) is entered, use

```
<PARAM NAME=area1 VALUE="SoundArea,260,180,120,60,boom.au">
```

HTML *DelayedSoundArea*: Like a *SoundArea*, this will cause a sound to chime whenever somebody rolls their mouse within the hot area. However, you can specify a delay for the sound, in tenths of a second. If the mouse rolls out of the hot area before the delay time, the sound will *not* play. This lets you do cool tricks, like playing several sounds within the same hot area, depending on how long the mouse cursor is in there. So to play the *boom.au* sound after one second whenever the area from (260,180) to (120,60) is entered, use

```
<PARAM NAME=area1 VALUE="DelayedSoundArea,260,180,120,60,10,boom.au">
```

HTML *NameArea*: This allows you to print some text in the Netscape status bar. Just specify the message as the sixth parameter. For example, to print the message "Click here!" in the status bar when the area from (260,180) to (120,60) is entered, use

```
<PARAM NAME=area1 VALUE="NameArea,260,180,120,60,Click here!">
```

HTML *HighlightArea*: This will highlight an area any time you roll the mouse over it. So, to brighten the area from (260,180) to (120,60) when it is rolled over, use

```
<PARAM NAME=area1 VALUE="HighlightArea,260,180,120,60">
```

HTML *LinkArea*: To simply and quickly jump to another Web page or other URL whenever a hot area is clicked, use this parameter. If you want the area from (260,180) to (120,60) to jump to the Netscape home page when clicked, type

```
<PARAM NAME=area1 VALUE="LinkArea,260,180,120,60,http://www.netscape.com">
```

HTML *HrefButtonArea*: This area jumps to another URL when clicked. This is similar to a *LinkArea*, however it gives the hot area the feel of an actual three-dimensional button. When you click the button, it will highlight to make it seem as if it has been pressed. So, to jump to Netscape's home page when the area from (260,180) to (120,60) is clicked, you would type

```
<PARAM NAME=area1 VALUE="HrefButtonArea,260,180,120,60,http://www.netscape.com">
```


HTML *RoundHrefButtonArea*: Since many buttons are round, not square, this parameter lets you transport to another URL only when an oval within a certain range is clicked. So, if you had a round button located between (260,180) and (120,60), and you wanted it to take you to the Netscape home page, you would type

```
<PARAM NAME=area1 VALUE="RoundHrefButtonArea,260,180,120,60,http://www.netscape.com">
```

HTML *AniArea*: This advanced area type lets you create animated buttons. For example, you can roll your mouse cursor over the picture of a telephone and suddenly the receiver will lift off the hook and the entire phone will shake. Create the animation as one large GIF image file. You must then specify where the top corner of each frame of the animation begins. For instance, if you have a three-frame animation of a telephone ringing, you may want to make it into a 128x384 image (*phone.gif*), where the top 128x128 square shows the first frame of the animation and the bottom two squares have the second and third frames. Assuming the original telephone button is located between (10,10) and (138,138), you can create the phone animation area as follows:

```
<PARAM NAME=area1 VALUE="AniArea,10,10,138,138,phone.gif,0,128,0,256">
```

To figure out the hot areas within your image, run the applet with only the *img* parameter. Create one area called a *ClickArea*, giving it the size of the entire image. For example if you have an image called *sample.gif* that is 256 pixels by 256 pixels, you would use the code:

```
<APPLET CODE=ImageMap.class WIDTH=256 HEIGHT=256>
<PARAM NAME=img VALUE="sample.gif">
<PARAM NAME=area13 VALUE="ClickArea,0,0,256,256">
</APPLET>
```

Now load up the Web page with this non-functional image map. Press down the mouse button, drag the mouse, and release the button, and the coordinate information will appear at the bottom of the screen.

For instance, the following markup creates the imagemap shown in Figure 18-6. If you roll over the circle, the words "This is a circle" appears in the status bar. If you roll over the square the *boom.au* explosion sound will go off. If you roll over the triangle it will highlight. Click on the circle to jump to the Netscape home page. Click on the square to jump to the Yahoo search site.

```
<APPLET CODE=ImageMap.class WIDTH=200 height=200>
<PARAM NAME=img VALUE="sample.gif">
```

```

<PARAM NAME=highlight VALUE="brighter30">
<PARAM NAME=area1 VALUE="SoundArea,260,180,120,60,boom.au">
<PARAM NAME=area2 VALUE="HrefButtonArea,260,180,120,60,http://www.yahoo.com">
<PARAM NAME=area3 VALUE="NameArea,260,180,120,60,This is a circle">
<PARAM NAME=area4 VALUE="RoundHrefButtonArea,260,180,120,60,http://www.netscape.com">
<PARAM NAME=area5 VALUE="HighlightArea,260,180,120,60">
</APPLET>

```

CREATING YOUR OWN APPLET

Some of the world's best programmers don't have to type so much as line. Instead, they know where to find existing routines and they know how to exploit them. Java is a full programming language. As such, you'd expect to find routines which help you perform most any programming task you have in mind. There are libraries for 3D graphics, for MIDI sound, for talking

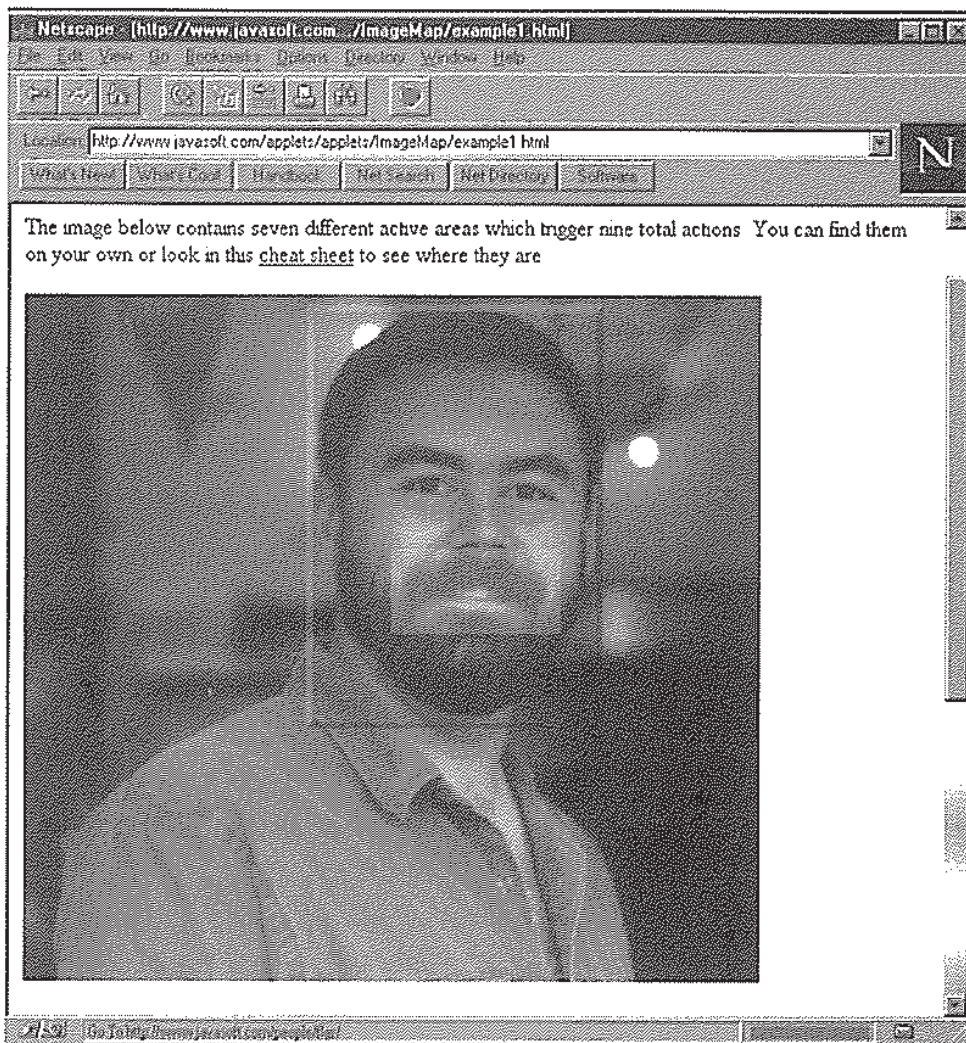


Figure 18-6 A Java image map

over the Internet, for video, etc. Instead of writing these libraries from scratch, they can be borrowed, bought, or licensed.


Half the job of creating a Java applet is to find a similar applet out there and then to change it to suit your needs. In many cases, all you'll need to change is a URL, a graphic, or a sound. Like any good programmer, half the job is finding libraries and the other half is suiting them to fit your needs.

There are many good books out if you're interested in learning Java programming. For example, check out the Waite Group's *Java Primer Plus*.


The Java Developer's Kit

If you're interested in compiling your own Java applets, download the Java Developer's Kit (JDK) at: <http://www.javasoft.com/JDK-1.0/installation.html>

Versions are available for the following platforms:

 Sun Solaris 2.3, 2.4, and 2.5 SPARC-based machines

 Microsoft Windows NT and Windows 95

 Macintosh System 7.5 (JDK Beta 1 release)

Installing the JDK: Windows 95

Click on the links until you've found a free site where you can download the JDK. The Developer's Kit is around 4 Megs, so you might want to make a sandwich or something while you download. When done, copy the resulting file to your hard disk's root directory (usually C:\).

The JDK works under the DOS command prompt. Go to DOS by clicking on the MS-DOS Prompt icon within the Programs group under the Start menu. Now you need to unpack the JDK. Use the *pkunzip* command. Type

```
pkunzip -d jdk
```

This will create the Java directory, along with several subdirectories.

You must now take the important step of updating your environment variables. Edit your AUTOEXEC.BAT file (in your root C:\ directory) with Notepad or any other text editor. There should be a PATH command in the file which looks something like this:

```
PATH=C:\;C:\WINDOWS;C:\WINDOWS\COMMAND
```

Add *java\bin* to your path, by typing a semicolon at the end of the path list and tacking it on, as follows:


```
PATH=C:\;C:\WINDOWS;C:\WINDOWS\COMMAND;C:\java\bin
```

You now need to use the CLASSPATH environment variable to point to the appropriate Java directory. Add the following line to your AUTOEXEC.BAT file:

```
SET CLASSPATH = java\lib\classes.zip
```

Installing the JDK: Mac


Download the Java Developer's Kit (it should be around 2.5 Megs). You can choose from the MacBinary or the hqx format. Uncompress the JDK. The popular Stuffit program can decompress MacBinaries, and the DeHqx program can uncompress the DeHqx format.

A folder will be created called "JDK-beta1-mac."

Installing the JDK: Sun

Download the JDK from the Sun Web site. The file's about 4.6 Meg long. The file is tarred and zipped. You can use the following Unix command to decode it:

```
zcat JDK-1_0-solaris2-sparc.tar.Z | tar xf -
```

 **NOTE:** The 1.0 Solaris release supports Sun Solaris 2.3, 2.4, and 2.5 SPARC-based machines.

A directory called *java/* will be created. You must now add *java/bin* to your path.

Writing the Applet

All Java programs end with the extension *.java*. Let's write a simple applet that prints something onto the Web page every time something special happens: either when the page is first loaded, when it is closed, or when a new page is called.

To begin writing a Java applet, create a new directory for it. This will contain your HTML pages, applet, and any other supporting files. Now create a Java source file called *Simple.java*.

Every Java file begins with a list of basic libraries that are needed to make the program run. There are special libraries for graphics, sound, time and date, and other common functions. For example, we'll start the simple applet with a line that imports the *java.awt.Graphics* library—a library that lets you print text and graphics to the Web page screen:

```
import java.awt.Graphics;
```

Every Java program is in a class of its own—literally. An applet, in fact, is a class function that can be called directly from the Web browser. You must name this class, make it public, and apply it to the existing Java applet class. Most every Java program begins with a similar line:

```
public class Simple extends java.applet.Applet {
```

You can then create sub-functions and other classes for your program to use. One of the most important methods is *init()*. This routine is automatically called every time the applet is first run. You can use this to define the basic look and feel of the applet, and to initialize any variables the applet may need to use. This is a good time to use the *resize()* method, where you can define the area on the Web page in which the applet's should be printed.

Another important method is *start()*, which is executed every time the applet is loaded or reloaded. The *stop()* method is called whenever the applet is stopped—for example, if someone quits the browser or jumps to another Web page.

Finally, the *destroy()* method cleans up any last-minute tasks before the applet is unloaded entirely from memory.

The main Applet class also has a *paint()* method. This powerful method can print any sort of basic information onto the Web page. The *update()* method redraws any text or graphics if they have been updated. Each class you import comes with its own custom methods. Also, most every action the user performs has its own built-in event. For example, there are events that are called for all sorts of movements and clicks of the mouse or keyboard: *mouseenter()*, *mouseExit()*, *mousemove()*, *mouseUp()*, *mouseDown()*, *mouseDrag()*, *keyDown()*, and *action()*.

The code for the full simple applet is as follows:

```
import java.applet.Applet;
import java.awt.Graphics;
public class Simple extends Applet {
    StringBuffer buffer = new StringBuffer();
    public void init() {
        resize(500, 20);
        addItem("Initializing!");
    }
    public void start() {
        addItem("Starting!");
    }
    public void stop() {
        addItem("Stopping!");
    }
    public void destroy() {
```

```

        addItem("Preparing for unloading!");
    }
    public void addItem(String newWord) {
        System.out.println(newWord);
        buffer.append(newWord);
        repaint();
    }
    public void paint(Graphics g) {
        g.drawRect(0, 0, size().width - 1, size().height - 1);
        g.drawString(buffer.toString(), 5, 15);
    }
}

```

When the applet is first loaded, the `init()` method will be called and generate the string: "Initializing!" This string is sent to the `addItem()` method that prints the string into the system (using the `System.out.println()` function), the buffer is updated and prepared with the string, and then the `repaint()` method is called. The `repaint()` method actually prints the buffer string onto the Web page, as specified with the `g.drawString(buffer.toString(),5,15)` line.

All this may seem incredibly complicated, but it actually fits together quite nicely. Every function within every method can easily be accessed using this object-oriented format. This gives the programmer absolute control over what the program does.

Compiling It

Compile your `.java` file using the Java compiler. Simply type:

```
javac sample.java
```

If everything was written correctly, a file called `Simple.class` will be created. If there are any errors, go to the indicated lines and correct them.

 **NOTE:** Java is case-sensitive. A variable named `SampleVariable` is completely different from one named `sampleVariable`.

If you're already on a Web server, then you can use your applet immediately. Otherwise, copy or upload the applet to a directory within your Web server. You're all set!

Using It

You can now add the `Simple.class` applet file to your HTML page. Be sure the applet is in the same directory as the Web page itself: Edit the HTML document and type the following when you want the applet to appear:

```
<APPLET CODE="Simple.class" WIDTH=150 HEIGHT=25>
</APPLET>
```


Now view your Web page. A gray rectangle, the size you specified in the above <APPLET> tag, is created. The words "Initializing!" and "Starting!" should appear in the status bar as the applet is initialized and started up. The applet is now running. If you go to another Web page, the applet will print, "Stopping!" as it halts. If you return to the applet page, the applet will print "Starting!" again as it fires back up.

Finally, if you quit Netscape the last thing you should see is the applet printing "Preparing for unloading!" as it cleans itself up and removes itself from memory.

Java Development Packages

Java is pretty difficult to compose from scratch. Luckily, there are a number of visual packages that make creating an applet almost as easy as dragging and dropping. Simply define what you want the applet to do and these packages will help you piece it together, error-free.

HTML JFactory: This visual package by Rouge Wave Software lets you design and test Java applets. JFactory comes with many predefined components, many of which you can just drag and drop together to form an applet. <http://www.roguewave.com/rwpav/products/jfactory/jfactory.html>

HTML JAGG: BulletProof.com's Java ODBC database engine. This package makes it easy to generate SQL statements from an applet. <http://www.bulletproof.com/jagg/>

HTML Nutmeg: Thought Inc.'s package uses a class library similar to Smalltalk's, allowing you to quickly assemble Java programs. Download a trial version at: <http://www.thoughtinc.com/nutmeg/>

HTML Symantec Cafe: This full-feature packages lets you create dynamic Web pages using Java. The environment is intuitive and has many features that seasoned programmers will enjoy. <http://www.symantec.com/lit/dev/javaindex.html>

WHAT NOW?

As complete as we've tried to make this book, we must admit that there's no way we could teach you how to program Java in just one chapter. If you like what you've seen so far, you should definitely check out a complete Java

guide such as the Waite Group's *Java Primer Plus*. If you're interested in more interactivity, but Java seems too complex for your needs, check out the previous chapter, Javascript. You'll learn how to harness Netscape's Javascript language, based on Java, to add interactivity and other cool effects to your Web page.

Flip to Chapter 23, Where to Place Your HTML Documents, for information about how to put files and documents onto your Web server.