Telecommunications: The Macintosh Modem Book

- Coverage of state-of-the-art high-speed modems, voice mail and faxing options
- Get online and increase compatibility with custom configurations and file transfer methods
- Comprehensive listings of bulletin board services, modems and cables
Telecommunications:

The Macintosh Modem Book

by Stephen Taylor
To Sam Taylor, ardent bookman and eminent Macintosh avant.
# Table of Contents

- Introduction ................................................................. 1
- Chapter 1: How Things Got to Be the Way They Are: A Very Brief History of Telecommunications 9
- Chapter 2: Getting Started with Modems ........ 17
  - External/Internal ......................................................... 17
  - Speed ................................................................. 18
  - Error Checking ......................................................... 19
  - Compression .......................................................... 21
  - Cables ................................................................. 23
  - Final Thoughts ....................................................... 27
- Chapter 3: Acquiring a Terminal Program ........ 29
- Chapter 4: Configuring Your Terminal to Your Taste ............ 39
  - Setting Your Settings ............................................... 39
  - Fonts ................................................................. 40
  - Which Serial Port? .................................................. 42
  - Keyboard Settings .................................................. 43
  - Screen Buffers ....................................................... 49
  - Capturing Text to Disk .............................................. 52
  - Download Files Destination ......................................... 56
  - Flow Control ........................................................ 57
  - Saving Settings ...................................................... 57
- Chapter 5: Configuring Your Terminal to Their Taste ............ 59
  - Terminal Program Settings ......................................... 61
    - Extended Characters ............................................. 61
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text Pacing</td>
<td>64</td>
</tr>
<tr>
<td>XON/XOFF</td>
<td>65</td>
</tr>
<tr>
<td>Wait for Echo</td>
<td>65</td>
</tr>
<tr>
<td>Delay Between Characters and Lines</td>
<td>65</td>
</tr>
<tr>
<td>Waiting for a Line Prompt</td>
<td>66</td>
</tr>
<tr>
<td>Text Pacing Expertise</td>
<td>67</td>
</tr>
<tr>
<td>Keyboard Settings</td>
<td>67</td>
</tr>
<tr>
<td>What your Return Key Sends</td>
<td>67</td>
</tr>
<tr>
<td>Backspace, Delete, Rubout, Etc.</td>
<td>68</td>
</tr>
<tr>
<td>Keyboard Buffers</td>
<td>70</td>
</tr>
<tr>
<td>Serial Port Settings</td>
<td>72</td>
</tr>
<tr>
<td>Data Rate</td>
<td>72</td>
</tr>
<tr>
<td>Data Bits</td>
<td>74</td>
</tr>
<tr>
<td>Parity Bits</td>
<td>74</td>
</tr>
<tr>
<td>Stop Bits</td>
<td>75</td>
</tr>
<tr>
<td>Duplexing/Echoing</td>
<td>76</td>
</tr>
<tr>
<td>Chapter 6: Commanding Your Modem: The Hayes Language</td>
<td>79</td>
</tr>
<tr>
<td>Talking Hayes</td>
<td>81</td>
</tr>
<tr>
<td>Result Codes</td>
<td>90</td>
</tr>
<tr>
<td>Stringing Commands Together</td>
<td>92</td>
</tr>
<tr>
<td>A Note on Hayes Compatibility</td>
<td>92</td>
</tr>
<tr>
<td>Initialization Strings</td>
<td>93</td>
</tr>
<tr>
<td>A Few Additional Commands</td>
<td>95</td>
</tr>
<tr>
<td>Chapter 7: Getting Online: A Sample Session</td>
<td>97</td>
</tr>
<tr>
<td>Preparing for Your Debut</td>
<td>97</td>
</tr>
<tr>
<td>Settings</td>
<td>99</td>
</tr>
<tr>
<td>Capturing the Session</td>
<td>101</td>
</tr>
<tr>
<td>Dialing Up the Service</td>
<td>103</td>
</tr>
</tbody>
</table>
Disabling Call Waiting .............................................. 104
The Service Answers and You Log On ...................... 104
Some Elementary Navigating 'Round the BBS ............ 107
Message Bases ...................................................... 109
More Navigation ................................................... 125

Person to Person Connections ......................... 132

Chapter 8: File Transfers I: A Cornucopia of File Formats ................................................. 133

Transferring Text Files ........................................ 133
About Macintosh Files ........................................ 136
Type and Creator ............................................... 136
Forks, Etc.: More on Binary Files ......................... 138
MacBinary ......................................................... 140

Compression Formats ....................................... 143

Chapter 9: File Transfers II: Protocols and the Mechanics ............................................. 149

Basic Downloading Mechanics ......................... 150
Transfer Errors ............................................... 159

Error-Checking File-Transfer Protocols ................ 160
XModem ......................................................... 162
XModem-1K .................................................... 162
YModem ......................................................... 163
YModem-g ....................................................... 164
Kermit ........................................................... 164
Sealink ......................................................... 165
ZModem ......................................................... 166
CompuServe B, B+, and Quick-B ......................... 167

Timeouts .......................................................... 168

More Transfer Options ....................................... 169
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hangup</td>
<td>259</td>
</tr>
<tr>
<td>Open capture</td>
<td>259</td>
</tr>
<tr>
<td>Close capture</td>
<td>259</td>
</tr>
<tr>
<td>Comments</td>
<td>259</td>
</tr>
<tr>
<td>Executing a ZTerm 0.9 Script</td>
<td>260</td>
</tr>
<tr>
<td>Sample Script</td>
<td>261</td>
</tr>
<tr>
<td>More Sophisticated Scripts</td>
<td>262</td>
</tr>
<tr>
<td><strong>Chapter 14: Using High-Speed Modems</strong></td>
<td>265</td>
</tr>
<tr>
<td>Modem Protocols</td>
<td>267</td>
</tr>
<tr>
<td>Modulation Protocols</td>
<td>268</td>
</tr>
<tr>
<td>Error-Control Protocols</td>
<td>268</td>
</tr>
<tr>
<td>Data Compression Protocols</td>
<td>269</td>
</tr>
<tr>
<td>Acquiring a High-Speed Modem</td>
<td>270</td>
</tr>
<tr>
<td>Still More Settings</td>
<td>271</td>
</tr>
<tr>
<td>Speed</td>
<td>271</td>
</tr>
<tr>
<td>Local Flow Control</td>
<td>273</td>
</tr>
<tr>
<td>Initializing Your High-Speed Modem</td>
<td>274</td>
</tr>
<tr>
<td>Modem Connect Responses</td>
<td>285</td>
</tr>
<tr>
<td><strong>Chapter 15: Voice Calls, Faxes, and Networks</strong></td>
<td>287</td>
</tr>
<tr>
<td>Voice Calls</td>
<td>287</td>
</tr>
<tr>
<td>Fax Modems</td>
<td>291</td>
</tr>
<tr>
<td>Networks</td>
<td>297</td>
</tr>
<tr>
<td><strong>Appendix A: Macintosh Bulletin Board Services</strong></td>
<td></td>
</tr>
<tr>
<td>Around the Country</td>
<td>301</td>
</tr>
<tr>
<td><strong>Appendix B: U.S. Government Bulletin Boards</strong></td>
<td></td>
</tr>
<tr>
<td>The Economic Bulletin Board</td>
<td>325</td>
</tr>
<tr>
<td>Other U.S. Government BBSs</td>
<td>327</td>
</tr>
<tr>
<td>Appendix C: Macintosh Modem Cables</td>
<td>331</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>-----</td>
</tr>
<tr>
<td>Obtaining Hardware Handshake Cables</td>
<td>331</td>
</tr>
<tr>
<td>Some Cable Anatomy</td>
<td>333</td>
</tr>
<tr>
<td>Hardware Handshake Cables</td>
<td>337</td>
</tr>
<tr>
<td>Making Your Own Hardware Handshake Cable</td>
<td>340</td>
</tr>
<tr>
<td>Appendix D: Brief Guide to Modems</td>
<td>341</td>
</tr>
<tr>
<td>Generic Modems</td>
<td>342</td>
</tr>
<tr>
<td>Global Village</td>
<td>342</td>
</tr>
<tr>
<td>Hayes</td>
<td>342</td>
</tr>
<tr>
<td>Intel</td>
<td>343</td>
</tr>
<tr>
<td>Mass Microsystems</td>
<td>343</td>
</tr>
<tr>
<td>Practical Peripherals</td>
<td>343</td>
</tr>
<tr>
<td>Prometheus</td>
<td>343</td>
</tr>
<tr>
<td>PSI</td>
<td>344</td>
</tr>
<tr>
<td>Shiva</td>
<td>345</td>
</tr>
<tr>
<td>Supra</td>
<td>345</td>
</tr>
<tr>
<td>Supra</td>
<td>345</td>
</tr>
<tr>
<td>Telebit</td>
<td>345</td>
</tr>
<tr>
<td>U.S. Robotics</td>
<td>346</td>
</tr>
<tr>
<td>Zoom Telephonics</td>
<td>346</td>
</tr>
<tr>
<td>Glossary</td>
<td>347</td>
</tr>
<tr>
<td>Index</td>
<td>367</td>
</tr>
</tbody>
</table>
One of the more beguiling notions that arose during the early days of personal computing was the concept of shareware. Microcomputers, as they were then called, were proliferating every which way, but mainstream software publishers were only just getting a feel for how the public wanted to use them. The result was that many people found themselves in possession of machines whose power was often more potential than actual. Hardware with all kinds of intriguing possibilities might be sitting on their desks, but the programs to exploit those possibilities were still in relatively short supply.

So a number of private individuals, some just dabblers and others truly inspired hackers, leapt in to fill the gap. Using programming tools that by today's standards were pretty rudimentary, they began writing hundreds—thousands is actually more like it—of the programs that personal computer owners craved but couldn't find. (Yes, they also wrote some programs that nobody craved, and others whose bugs would tie a computer into knots that called for an out-and-out steam cleaning, but that's a story for another time.) Lacking access to commercial channels of distribution, these independent
software authors created a distribution system of their own, and thus did shareware come into being.

The concept relied heavily on computer bulletin board systems—BBSs in the lingo. Largely as a public service to computer users (and here and there in hope of making the odd buck or two from membership fees), a number of people around the country had already begun hooking up their computers to phone lines and leaving them on for long periods, often 24 hours a day. Of course, the crucial nexus between phone line and computer was the modem, and BBS operators—system operators, or sysops in the jargon—and their patrons were the first heavy non-corporate modem users. The computers in question would run one or another piece of BBS software—host programs—what they’re called in the telecom world—and the public was invited to call in and post messages on the bulletin board. The messages themselves were a mixed bag. Some were available to anyone who logged onto the BBS, others were private and readable only by someone with the right password. In time, computer BBSs came to be a lot like CB radio. Users exchanged real information and casual wisecracks, posted “for sale” ads and dirty jokes, even left chapters of their sci-fi novels-in-progress for others to critique. By and large, however, the focus of all this telecom traffic remained the computers themselves. Just a few years ago modems were substantially more expensive than they are today, but having one gave you access to an endless trove of computer lore. Owning a modem meant that you were serious about computers.

When you’re online with a BBS, you can post messages in one of two ways. You can type to the BBS directly, in which case your software and modem send along each character as you type it, or you can prepare material in advance and then send it all at once. The second method—which is faster and cheaper because you’re not on the phone while you’re thinking—is commonly called uploading. The files you can upload to a BBS are by no means limited to English prose. You can post any kind of file you want. What these one-horse software developers did (and still do) was upload the programs they’d written to BBSs so that others could try them out.

Generally, each program was packaged with some documentation (the telecom-ese term is docs) and a plea for remuneration. A typical plea would go something like: "This program is being distributed under the shareware system. You may use it without charge for 30 days, and you may give it to a friend. If you like it and find it useful, please send $25 to John Jones at
When people found the listings of these programs in the libraries of a bulletin board, they'd sometimes instruct the BBS to send them along—i.e., they'd *download* the programs from the BBS to their own computers. Then they'd try them out, separate the gems from the junk, and stash any genuinely useful ones in their own personal software libraries. Very rarely, they'd even pay the shareware fees.

As shareware matured, the programs tended to fall into two broad categories. Some, as we've already said, aimed to fill the gaps left by commercial software publishers. But others challenged the commercial developers head-on. One of the best-known Macintosh programs in the second category was, of all things, a telecommunications program. For reasons that remain vague it was called Red Ryder, and for years it probably kept more Macs yodeling to more computers through more modems than any other Mac telecom program around. In its shareware form, it evolved through nine full versions and many more modifications. Its developer named his company FreeSoft, and, while technically speaking, Red Ryder wasn't free, even if you paid the shareware fee it was vastly cheaper than the commercial competition on a feature-by-feature basis.

This all happened against the mercantile mentality of the 1980s, so it felt more or less inevitable when Red Ryder's developer (who'd become something of a folk hero among Mac telecommunicators) went commercial. Although still a bargain at the price, version 10 of Red Ryder was sold through normal commercial channels. Later FreeSoft broke with its mainstream allies and marketed the program on its own. The name Red Ryder, however, remained enmeshed in contractual boilerplate, and the program was rechristened White Knight (perhaps an unfortunate choice). Today White Knight is one of the most widely used telecommunications programs for the Macintosh and one of my own personal favorites.

"Red Ryder" was a great name for a telecom program. Like "Apple," the name is literally impertinent: Red Ryder, the cartoon cowboy of the forties and fifties, has no more to do with telecommunications than apples have to do with computers. It all harked back to the funky sense of community that pervaded the sixties, when names were intentionally unconnected to the things they were naming and a shareware developer could still permit
himself to believe there was an honor system. Actually the cowboy hero Red Ryder harks back even further to a frontier era in which people rallied to put up a settler’s barn the way BBS users still often rally to solve each other’s computer problems.

Figure 1.1. Main window from Red Ryder v.10.3.

In fact, when something’s gone awry with my computer and I’m stumped, I usually compose a note about it, dial up a BBS that carries one of the national Macintosh echoes and post the note for all to see.

What’s an echo? It’s part of a system that allows bulletin board systems to share their message bases over a network extending all the way around the world. One such system called FidoNet offers net-mail forums on dozens of topics, many not related to computers. None, so far as I know, deals specifically with the building of frontier-style barns, but the one on home repair is getting more popular all the time. Red Ryder, the lonesome cowboy who solved people’s problems and then rode off into the sunset, the man in whose saddlebag no copy of Self magazine was ever tucked, would probably approve.

A final note on the subject of cowboys, particularly as it relates to being in love with one’s horse. For people of Red’s ilk, loving your horse wasn’t all that
That kind of relationship tends to be especially intense where the Macintosh is concerned. Fielding allegations that the Mac was just a toy, or that even if it wasn't a toy it still wasn't a serious computer, Mac owners used to plunge into battle with IBM enthusiasts like Montagues jousting with Capulets, their loyalty becoming fiercer with each encounter. Even now, when they discover one another at work or social gatherings, they can talk MacinBabble for hours at a time, usually to the utter boredom of whoever else is listening.

But whatever private passion may exist between a computer owner and his Macintosh, good relationships must eventually open outward or risk becoming too hermetic. The best couplings bear fruit and also require nourishment along the way. Computers need a dose of data every now and then, and they also need a way to get their reports, spreadsheets, and databases out into the world.

For many of those purposes, printed paper works just fine. And when connectivity is wanted there's always the floppy disk. Where greater speed or volume is called for, a good short-range solution is the LAN, the now much talked- and written-about local area network. Longer range, we arrive at that favorite instrument of lovers, the telephone. And from early on, Macs took to the phone with singular alacrity. Computers normally link up to phone lines through their serial ports, and while some machines used to come with no serial port at all—if you wanted one you had to buy a special board—every bare-bones Macintosh has always come with two. Mac telecommunications software therefore matured very fast, and several surveys conducted during the 1980s indicated that the percentage of Mac owners who used modems was higher than with any of the other popular computers.

Apple then came forth with the AppleFax modem, which essentially lets you print almost any document at all to someone else's fax machine. Modems also got cheaper and faster very quickly. Other fax modems—better ones
than Apple's—showed up, and the 300 bits-per-second non-fax standard escalated through 1200, 2400 and 9600 and is now hovering around 14,400. As BBSs and online services became less clique-y and appealed to a more general audience of telecommunicators, it dawned on many people that there's a great deal of interesting information to be had out there, and that a modem is a handy device for acquiring it. Soon Prodigy, a huge online service, was marketing itself through network television, boasting of an interface so non-technical that almost any idiot could use it.

Prodigy, it should be noted, often comes in for particular contempt among experienced telecommunicators. They'll criticize this or that aspect of it, but I think what really bugs them most is its very simplicity. If you go to France and find that the French have all started speaking English in an effort to encourage tourism, France isn't going to feel very much like France, regardless of how user-friendly it's become. Similarly, by adopting an interface that spares you from having to learn virtually anything at all about telecommunications, Prodigy can make you feel you aren't really telecommunicating. So if you want to start using a modem with your Mac, you can join Prodigy and learn next to nothing about what you're doing, or, the intensely curious reader that you no doubt are, you can forge ahead in this book.

What will we be covering? Well, partly to put the current state of Macintosh telecommunications into the perspective of history and partly to introduce some telecom vocabulary, we'll start by taking a quick glimpse at the art of sending messages over wires during the last century-and-a-half or so. Then we'll examine some of the basic telecom necessities—modems, cables, programs—and try to match them to your needs.

When you use a modem with a Mac, you normally wind up talking to at least three different entities. Yes, you connect with the party at the other end of the line, but before that you're usually called upon to say a few things both to your telecom program and your modem. Using the right program settings for each connection is half the battle, so we'll spend some time on how you go about configuring your settings. Modems prefer being spoken to in Hayes language (not among the Indo-European tongues), and we'll also spend some time on that. Then we'll get down to the basic business of computer telecommunications: typing back and forth and transferring files. Since files can be transferred in a variety of forms, we'll also talk about preparing files before you send them and what to do with them when you receive them. En-
route we'll look at any number of Macintosh programs which, while not telecom programs as such, can be of great assistance to the telecommunications process.

Budding telecommunicators often develop a lively interest in finding other parties they can contact with their modems, so we'll give some special attention to commercial online services and amateur computer bulletin boards. As your telecom proficiency grows, you'll probably want to make increasing use of the conveniences offered by your software. To that end, we'll explore the automation and scripting capabilities of some of the better-known Macintosh telecommunications programs. We'll also take a closer look at high-speed modems, and we'll venture only slightly far afield to investigate fax modems and also how to use your data modem to dial voice calls. When we've accomplished all that, several appendices will add some information that lends itself to being listed rather than narrated, and there'll be a glossary for quick reference to all the specialized terminology introduced within the book.

Enjoy!
Nearly everyone who approaches computer telecommunications for the first time is struck by the sheer quantity of new terminology encountered. Yes, they happen upon any number of familiar words—flow, handshaking, protocol, etc.—but in the dusky nether-world of telecommunications those everyday words are often apt to take on special meanings. Sometimes it will seem as though one can’t read about the subject without first mastering some sort of pseudo-English dialect made up partly of scientific nomenclature and partly of modem-junkie street slang.

We could attempt to make things easy here by trying to substitute ordinary language for telecom-ese wherever possible, but in the end we’d likely find ourselves in a worse morass than if we simply go ahead and learn some basic vocabulary. What’s the point of ducking the jargon only to be baffled when it turns up in the manuals and menus of every telecom program you’re apt to find? Copious nomenclature often comes along with tricky concepts, but as we venture into our subject you may discover that the nomenclature is actually a lot more daunting than the concepts behind it. And once you’re at ease with a couple of dozen common terms, you’ll probably be able to grapple with just about any telecom program you care to try your hand at.
An interesting thing about telecom terminology is that much of it came into use long before anyone even dreamed about the first computer. In its most basic sense, telecommunications is the transmission over wires of digital data—strings of ones and zeroes, nos and yeses, ons and offs—and it’s been going on full tilt for more than a century and a half. Nowadays the wires are telephone lines, but the process isn’t so different from traditional telegraphy, in which the ones and zeroes were dots and dashes. Modern telecommunications even more closely resembles the teletype technology of the recent past. It’s true that most teletype terminals have given way to computers by now, but a great many of the standards, conventions and nomenclature that arose with teletype—and even telegraph devices—still prevail today. In fact, any number of telecom applications for personal computers still call themselves terminal emulation programs. And the terminals they emulate are mainly teletype terminals. So, while it’s hardly necessary to master the entire history of telecommunications in order to telecommunicate, it may be worth a few moments of your time to see computer telecommunications in the perspective of what came before it. However forbidding the lingo of telecom may appear when approached from a cold start, it will probably make considerably more sense after you know why things are called what they’re called.

Let’s begin with the early nineteenth century, when sending a telegraph message was still almost magical. By current standards it was also ridiculously simple. You ran a wire from point A to point B. At point A you connected a battery in series with a switch. At point B you hooked up an ordinary electromagnet. When you closed the switch, the battery sent current through the wire and energized the electromagnet, which in turn deflected a needle. In time the switch took the form of a telegrapher’s key. When you held down the key, you sent a pulse through the wire and the needle would be deflected. A unique pattern of pulses was assigned to each letter of the alphabet. That done, you could send letters, words, and sentences (albeit very slowly) to whomever was watching the needle, so long as they also knew the pulse code. Quite a few prominent experimenters of the time—among them Carl Gauss and Andre Ampère—tinkered with telegraphy, and eventually a pair of Englishmen, Charles Wheatstone and William Cooke, hooked up the world’s first railway telegraph in 1837.
Another experimenter who'd been working with the telegraph was the American Samuel F. B. Morse, and by 1831 he'd developed what was to become the standard telegraph receiver of its time. What Morse did was replace the electromagnetically deflected needle with a scribe held against a moving strip of paper. As the paper rolled by, the undeflected scribe traced only a straight line. But its deflection by the electromagnet, which traced a blip on the paper, meant the sender was holding down the key—i.e., closing the switch. Morse then went on to create his dot-dash (short pulse/long pulse) method for encoding alphanumeric characters.

Note that Morse code is already a binary system. Even before electricity mutated into electronics, the on-off, dot-dash, one-zero method of encoding was in wide use. Collaborating with Alfred Vail, Morse soon improved his receiver to a point where it could print the dots and dashes directly as they were received, and it was to this new receiver that in 1844 he sent his famous query—"What hath God wrought?"—from Baltimore to Washington. Curiously, people who worked every day with Morse telegraphy became so facile with it that they no longer had to look at the moving scroll of paper in order to know what was coming in over the wire. The scribe would click with each deflection, and operators soon learned to interpret the clicks. Possibly to protect their special skills by creating a closed fraternity, most telegraphers came to abandon the scroll and rely entirely on their ears. And with the advent of radiotelegraphy much later, facility with purely auditory Morse code remained a vital skill well into the twentieth century.

The burgeoning popularity of telegraphy quickly made for inevitable traffic problems—crowded wires. At first only one signal could be sent in one direction along a given wire, but the efforts of a number of other experimenters led to the development of *multiplexing*—sending more than one signal along the same line. And it's here you might want to begin paying particular attention, since a number of the following concepts are a) still valid today and b) found in the menus and buttons of many Macintosh telecommunications programs. The first breakthrough was the introduction of *duplexing*, which allowed for simultaneous transmission of signals in both directions along a line. Duplexing was followed by *diplexing*, in which two signals could simultaneously be sent in the same direction. Then, in 1874, marrying diplexing to duplexing, crafty old Tom Edison created *quadruplexing*, in which two duplexed signals could be sent along the same wire at the same time—four signals altogether!
In an effort to send *still* more information over existing wires, inventors began tackling the problem of human limitation. Because an operator can key and interpret Morse code only so fast, people kept seeking ways to automate the entire telegraphic process. By the mid-nineteenth century, an apparatus appeared which could print telegraph signals on paper tape, followed almost immediately by another that punched *holes* in paper tape. Soon operators were preparing punched paper tapes on keyboards and feeding them to devices that could read the perforations and transmit them as Morse code over telegraph lines. Paper-tape readers were faster and more accurate than people, and as the nineteenth century came to a close automated telegraphy was pushing speeds of nearly 400 words per minute.

Meanwhile, multiplexing saw further development with the introduction of what was called the *time division system* of telegraphy. Divide a minute into its component 60 seconds and then apply an arbitrary rule: If there's current flowing during any given second, that signifies a one or dash or whatever else you like to name your binary units. If there's *no* current flowing during that given second, it signifies a zero or dot or whatever else you've named the *other* of your two binary units. Now go ahead and create a device which either does or doesn't send current every second, depending on whether it's supposed to send a dash or dot. Then create a second device that inquires every second if the current is on or off. If the current is on, the device assumes you're sending a dash; if it's off, you're sending a dot. As long as both of your devices are synchronized, the sender can transmit 60 dots or dashes every minute, and the receiver can *receive* 60 dots or dashes every minute.

Another designation for that process is *synchronous telecommunications*, and a French experimenter named Emile Baudot refined it to a point at which it could handle a great many more than 60 binary units per minute. By 1874, Baudot's work allowed as many as six telegraphers to share a single line at the same time. Synchronous methods of telegraphy still coexist with newer *asynchronous* methods (the White Knight manual continues to proclaim on its cover page: "An Asynchronous Telecommunications Program for the Apple Macintosh...").

It's Baudot, incidentally, whose name survives in the term *baud rate* or just *baud*, which is still often used to indicate the rate at which binary units are transmitted over a line. There's a fine distinction between *baud* and *bits per second*, but in practice the two terms are now used more or less interchangeably when describing the speed of modems.
Baudot also went on to create a new binary code that was gradually to replace Morse's. In Morse code, the various alphanumeric characters are encoded with two or more units; in Baudot's system the number of units is fixed at five. That is, each character is represented by exactly five binary units, no more, no less. Each of the five units can be a one or a zero. The number of unique combinations of five ones and/or zeroes is two to the fifth power, or 32, which means that Baudot's code could accommodate 32 characters altogether. A Macintosh byte, by contrast, contains eight usable bits and can therefore represent any of $2^8$, or 256, unique characters. A number of Mac word processors will give you the decimal equivalent of any character. If you convert one of those decimal equivalents to binary, you'll get the bit pattern—the pattern of ones and zeroes—with which the character is encoded in a Macintosh byte. In any event, after undergoing some minor changes, Baudot's five-unit code became International Telegraph Alphabet No. 2, which was used well into the twentieth century.

As yet we haven't mentioned the telephone, but Alexander Graham Bell did invent it in 1876, and it rapidly became quite popular. Some people fretted that it might wipe out telegraphy altogether, yet no such thing was about to happen. In fact, telegraphy wound up borrowing liberally from telephone technology. One way in which the two techniques differed was in the kind of signals they sent out over the wires. Telegraph devices sent synchronous pulses: a pulse means one, no pulse means zero. Telephones, on the other hand, sent continuous audio frequency signals, which are considerably more complex. Where the waveform of a direct current pulse signal might look like this:

![Figure 1.1](image-url)
Me saying "hello" produces an alternating current wave form that happens to look like this:

![Waveform Image]

*Figure 1.2*

The second signal carries vastly more information. It contains, in spoken form, the equivalent of the five characters that spell "hello," but it also contains enough information about the timbre of my voice to identify the speaker as yours truly and enough about my intonation to allow someone to draw conclusions about my state of mind at the moment I uttered the word.

The next generation of experimentors in telegraphy were eager to cram more information through the wires, so what they did was expropriate the audio signal for telegraphic purposes. And their labors promptly begat the then-new technology of *carrier telegraphy*. A steady alternating current voice frequency tone called a *carrier signal* or just *carrier* was transmitted between sender and receiver. Onto the carrier was superimposed the direct current pulses of the telegraph signal. That process of superimposition is called *modulation*—the telegraph pulses *modulate* the carrier. Actually, it's from that very process that the modem got its name. *Modem* is an acronym for "modulator-demodulator." A modem modulates an outgoing audio carrier when it sends bits from your computer out into the phone lines, and it demodulates an incoming audio carrier in order to extract the bits it passes along to your computer.

At first a given audio carrier traveling through a phone line provided a single telegraph channel. But one phone line can handle several audio signals at the same time, provided the frequencies of the signals aren't too close together. It's also possible to superimpose several low-frequency audio carriers onto one high-frequency audio carrier. At the receiving end, another device can separate the component carriers by passing the high-frequency signal through a series of audio filters. When the vacuum tube came along to make this kind of circuitry possible, a single phone line was soon capable of handling a couple of dozen high-speed telegraph channels.
The speed and volume of all this telegraph traffic soon outstripped the capabilities of earlier telegraph receivers, and most gave way to teletype terminals. A typical terminal looked like an electric typewriter armed with additional circuitry. It could print out messages as they arrived, and to send a message all you had to do was type it on the keyboard. For quite a long time, the limiting transmission speed of these devices was 300 bits per second. Beyond that speed, random pulses generated along the lines or in the circuitry itself made it difficult to decipher telegraph signals with any accuracy. Those random blips are what's known as noise, and, as transmission speeds increase, noise continues to be the bane of high-speed telecommunicators.

We've already said that a five-bit code was used to represent 32 alphanumerical characters. To those five bits, engineers introduced an additional bit in order to guarantee more accurate transmission. The new bit was called a parity bit. Let's say the engineers decide that their system will use even parity. (It could just as well use odd parity; the choice is arbitrary.) In an even parity system, each outgoing character must have an even number of ones. So, before a character is sent, a circuit scans the five bits that represent it. The circuit counts the ones and ignores the zeroes. If there's already an even number of ones, it sets the parity bit to zero, in which case there's still an even number of ones. But if there's an odd number of ones, it sets the parity bit to one, thereby assuring that the total number of ones associated with that character is still even. Now, when the character is received, the recipient's equipment again counts the number of ones it contains. If there's an even number of ones, that's fine. But if the number of ones is odd, that means all six bits didn't come through accurately. The recipient can then request a retransmission, or, if his terminal is smart enough, it can request a retransmission without bothering the operator at all.

As newer methods of error-checking keep arising, parity bits are falling into disuse. But you'll still see them mentioned in the settings options of virtually every Macintoshs telecom program. If you didn't already know what parity bits are, at least now you'll have some idea of what your program is talking about.

Another notion we've already broached is that telecommunicating can be synchronous or asynchronous. For a number of reasons, asynchronous methods became preferable, and when a sender and receiver weren't synchronized, it was necessary for the sender to tell the receiver when it was
sending data and when it wasn't. So two more bits were introduced into each character: a start bit and a stop bit. Essentially, the start bit says, "Here come the data bits," and the stop bit says, "No more data bits for this character." Stop bits are something else you'll come across when you fool around with the settings of Mac telecom programs.

Nowadays, moreover, the old five-bit method of encoding characters is no longer in broad use. Greater transmission speeds made it viable to send more bits, and the teletype community adopted the newer ASCII seven-bit standard. Two to the seventh power is 128, and the ASCII character set indeed contains 128 characters. Add a start bit, a stop bit, and a parity bit and you wind up with a total of 10 bits per character. That means that if you're telecommunicating at 300 bits per second, the equivalent rate is 30 characters per second (cps). For a while there was no great pressure to go faster than that since, owing to electromechanical limitations, teletype machines couldn't type much faster anyway.

But computers are something else entirely. Computers have memories to which they can write, and they can also write to magnetic tape, disk, hard disk, whatever. What's more, they do it at a much faster rate than 30 cps. So as fiber optics, satellite links, and solid-state repeaters continued to enable greater volumes of digital information to travel more and more rapidly over long distances, the venerable teletype became something of a dinosaur. More sophisticated terminals—teletype-computer hybrids like Digital Equipment's VT52, VT100, and VT102—arrived on the scene. However, now even those are getting rarer all the time.

In telecommunications, what mainly survives of the computer's antecedents are the conventions that governed their operation. Macintosh telecom programs do still emulate some common terminals. Most let you pick which emulation you want. For ordinary purposes the emulation you'll choose is TTY, which stands for teletype. To that extent, some part of the history of telecommunications will inform your actions. And if there's a shred of truth to the educator's maxim that context facilitates understanding, you now ought to have an easier time with the following chapters.
If you want to begin telecommunicating and you have a phone and a Macintosh, you already have two of the five elements most crucial to the process. The three additional things you'll need are a terminal-emulation program, a modem and a cable to connect your modem to your Mac.

The modem is the hardware intermediary between your computer and the phone line. And we're being careful to say "phone line" here and not "telephone," since there's no particular need for a telephone handset—what phone companies like to call your "instrument"—to telecommunicate. That isn't to say that a handset within easy reach of your computer won't be useful. Later on, for instance, we'll talk about using your modem to dial voice calls. We'll also discuss modems in varying degrees of depth, but right now let's lay down some basic guidelines.

External/Internal

If you're new to modems, you may or may not be aware that as physical entities they come in two flavors, external and internal. External modems are
self-contained units, each in its own case with its own power supply. As such, they are rarely computer-specific. When you're considering an external modem, you generally don't have to care whether or not it's for a Macintosh, much less for any particular kind of Macintosh. There are, however, a few exceptions, and the two most conspicuous ones can be recognized by the word "personal." The Apple Personal Modem has become something of a fossil by now, but the Hayes Personal Modem 2400 Plus for the Mac is a fairly new arrival. What makes both of them computer-specific isn't their innards but their cables. The Apple uses one unique unto itself, and the Hayes's cable is permanently attached.

Unlike externals, internal modems don't have cases or built-in power supplies. Basically they are circuit boards made to slide into an expansion slot and draw power from the computer's own power supply. Naturally enough, that makes them very much computer-specific. They're by far most common in the IBM-compatible world, but recently more and more internals have been turning up for Macs too, usually in the form of Mac II NuBus cards.

Internal modems have a main advantage and a couple of drawbacks. The advantage is lack of clutter: once you install one it's off your desk and out of the way. The drawbacks are in the realm of price and available options. An internal modem with a given set of capabilities is likely to cost somewhat more than its external counterpart, and there simply aren't as many available choices as to speed, compression, and error checking.

Speed

If there are still any modems kicking around that transfer data at rates lower than 300 bits per second (bps or baud), I haven't come across them. Nowadays working at 300 bps feels uncomfortably slow, and I'd strongly recommend against it. If someone gives you a 300 bps modem for nothing—which is fairly close to the going rate anyway—and you want to try it just to see what telecommunicating is all about, be prepared to send it into retirement once you get serious. By consensus, the low end of the speed spectrum these days is 2400 bps, and new modems that run more slowly than that are becoming almost impossible to find in stores and catalogs.

When you're deciding on speed, be wary of terms like effective speed or effective transfer rate. They usually refer to transfer rates obtained by
compressing the data before it's sent and then decompressing it when it's received. A first-class postage stamp lets you send a letter that weighs up to an ounce. You can send maybe 6 pages straight out of your typewriter, or you can find an office copier that reduces images by 50 percent and squeeze 12 pages onto 6. But either way you'll be sending six pages that weigh an ounce. Similarly, a 2400-bps modem that boasts an effective transfer rate of 4800 bps after compression is still sending at 2400 bps. And the advertised effective transfer rate is seldom actually attained, which is why effective transfer rates are usually preceded by an "up to," as in "up to 4800 bps."

Currently the most common modem speeds are 2400, 9600 and 14,400 bps, and prices for new modems rise pretty much in proportion to speed. If you expect to be just an occasional user, 2400 bps will probably do just fine. If you think you'll be transferring large files or large numbers of small files, consider a 9600-bps unit. The money you'll save on phone bills or online service connect-time charges will very soon cover the extra cost. If you aspire to be a power telecommunicator, by all means look into 14,400-bps modems, but remember that to move data at 14.4 Kbps, the modem at the other end also has to be built for 14.4 Kbps. At this writing, 9600 bps is the highest speed supported by the major online services. A few bulletin board services can handle 14.4, but they can sometimes be hard to find. A quick scan of a list of some 500 Mac-oriented BBSs indicates that around 2 percent will let you connect at 14.4 Kbps. So unless you're reasonably sure your co-communicators will be able to match your speed, you may wind up buying power you seldom use.

A final reason to stay away from 14,400 bps—possibly even from 9600 bps, for that matter—is that the speed may spoil you rotten, with the result that you'll feel compelled to spend more money than you really want to spend.

Error Checking

Some modems offer built-in, hardware-based error checking; some don't. At present there are two dominant protocols for hardware error checking: MNP Class 4 (usually just called MNP-4) and V.42. Of the two, MNP-4 is the older and more widely used. It was developed more-or-less unilaterally by Microcom, an American hardware and software company, and for a while it represented the cutting edge of personal modem-ology.
But then, in Geneva, a conference was convened by a U.N. agency called the International Telegraph and Telephone Consultative Committee (the venerable CCITT—and if you know enough French to reorder the nouns and adjectives, you can make the initials come out right) and the conferees came up with V.42. A number of modems offer both MNP-4 and V.42. And modems that offer MNP Class 5, which is a data-compression scheme, invariably offer MNP-4 as well.

Hardware- and software-based error checking are something else we'll examine more thoroughly in future chapters, but for now let's tackle the question of whether you need hardware error checking at all. The answer? Maybe. If the phone lines in your area are inordinately noisy, built-in error checking can be a help when you and your interlocutor are typing back and forth. The modems will transparently correct any errors, leaving you free to just type and read. It's true that most telecom software is able to check for errors when files are sent from one computer to another using a special set of rules known as a file transfer protocol. But sometimes, when all you're sending is a text file, you may not want to bother invoking a software-based protocol transfer, and when that's the case hardware error checking can be very helpful.

Oddly enough, hardware error-checking can also save you a little time if your phone lines are very clean. Let's say you've just developed a HyperCard stack and want to send it to your uncle Louie. The last file you laid on Louie was a 10-page essay about how the dehumanization associated with computers is seductively pleasurable. If a few characters here and there were garbled, trusty old Louie would still have caught your drift, so you sent the treatise as straight text without using any protocol at all. But the stack is something else entirely. If even one bit gets transmitted incorrectly, it's likely to crash Louie's computer when he tries to open it. So this time you're careful to use a software-based protocol. Let's say you pick YModem. If the phone lines between you and Louie tend to be fairly free of noise, here's where Louie can do something to make the transfer go faster. Software error-checking schemes all involve an ongoing encoded dialogue between the two communicating computers:

"Did that last batch of data come in okay?"
"Yep."
"Then go ahead and send the next batch."
"Okay, here it comes."
"Got it."
"Did it come through intact?"
"No, not this time. Better send it again."
"Okay, here comes the second try."

And on and on. All that *handshaking* between the two computers is what confirms the integrity of your data, but it also takes time and thereby nibbles away at transfer speeds.

If, however, you and Louie both have MNP-4 modems and you've both established what's called, in the parlance of MNP, a *reliable connection*, then Louie can instruct his telecom program to receive your HyperCard stack in a variant of YModem called YModem-g. YModem-g observes *some* of the conventions of YModem, but it omits most of the computer-to-computer tête-a-tête. The software still checks for errors, but, even if it finds one, it can't ask the other computer to retransmit the data and, therefore, has no recourse but to abort the whole transmission. On clean phone lines, YModem-g takes around eight percent less time than normal YModem, but if you're transferring a long file and the transfer aborts after 20 minutes, you haven't saved much of anything. If, on the other hand, the hardware inside your modem and the hardware inside Louie's modem are already checking the data, only data that's error-free will be passed along to the software, so the transfer won't abort. What it all boils down to is that hardware-based error checking lets you skip a certain amount of *software*-based error checking, which can often shave time off your file transfers.

Incidentally, if it's new modems you're looking into, the decision as to whether or not to go with hardware-based error checking applies almost exclusively to modems in the 2400-bps category. Nowadays nearly all faster modems have some form of error checking built in. And in a 2400-bps modem, error checking definitely won't change your life, but it's sometimes nice to have it anyway.

**Compression**

Another feature that many modems offer is hardware-based data compression. Here again, there are two prevailing standards, Microcom's *MNP Class 5*
Telecommunications: The Macintosh Modem Book

(commonly called—you guessed it—MNP-5) and the CCITT's V.42bis. MNP-5 advertises compression ratios of the order of two-to-one, while V.42bis is theoretically capable of shrinking a stream of data down to a quarter of its original size before sending it out. That means if you use V.42bis, you would need only 25 percent of the time it would take to send uncompressed data! The receiving V.42bis modem internally decompresses the data back to its original form, and the whole process is entirely transparent to you and your telecom program.

(If you have already begun to research modems and have encountered the terms V.32 and V.32bis, it may be worthwhile to point out here that those are not more primitive forms of V.42 and V.42bis. V.32 and V.32bis are modulation protocols—i.e., sets of rules that govern how a modem modulates a carrier tone—not error-checking or compression protocols. In fact, they are the modulation protocols observed by most current high-speed modems.)

Is hardware-based compression the best thing to have come along since "lite" mayonnaise? Not quite. The actual amount of compression you get is entirely dependent on the nature of the data. Let's say you send your uncle Louie a string of a thousand A's. That's 1,000 bytes of data. You might also send him the following string: \[A:1000]\]. Even with the brackets and colon, that's eight bytes. Assuming that Louie knows your compression scheme and will go ahead and expand \[A:1000]\] into a string of a thousand A's, you've succeeded in shrinking the data down to eight-tenths of one percent of its original size before sending it out, which is no small achievement. The trouble is, most data isn't quite that compliant. That very same compression scheme wouldn't save anything at all if you were to send Louie the text of this paragraph.

To be fair to Microcom and the CCITT, both MNP-5 and V.42bis are considerably more sophisticated than the scheme we've just developed. Both are especially effective with text. Binary files such as computer programs and graphics generally don't get compressed as tightly. Files that have already been compressed by one or another software-based compression utility are least tractable of all. In fact, when a modem running MNP-5 or V.42bis tries to compress a file that's already been compressed by StuffIt or Compact Pro (the two most popular Macintosh compression programs), the resulting data stream may actually turn out to be a tad larger than it was before.
One reason you want to be aware of all this is that the huge majority of Macintosh files available for downloading from online services and BBSs are already stuffed by Stuffit or compacted by Compact Pro. Software-compressed files take up substantially less space on the sender's hard disk, and they are also convenient for downloaders whose equipment doesn't sport all the niceties. Consequently, if you dial up an online service or BBS using MNP-5 or V.42bis, your file transfers actually may be somewhat less efficient than if you'd simply made a plain vanilla connection.

Then again, let's consider the novelists Joseph Conrad and Ford Maddox Ford. The two were good friends, and Conrad's erratic health and perpetually pressing deadlines moved the much more facile Ford to extraordinary acts of charity. Meddlesome to begin with, Ford volunteered to write whole chapters of several of Conrad's novels. Now—hypothetically of course—let's transport both men into the last decade of the twentieth century and give them Macs and modems, so they can easily send chapters back and forth. Because Ford was a technophobe and Conrad forever impatient for royalties, we can assume that neither would be much inclined to fool with Stuffit or Compact Pro before sending their files. Their binary Microsoft Word files would consist mainly of text, which submits readily to compression, and if the modems we give them are equipped with MNP-5 or V.42bis, they'll stand to save a fair amount of money and time during the course of composing a novel such as Lord Jim.

What's the bottom line? If you expect to be exchanging uncompressed files with someone whose own modem can perform hardware compression, you probably want hardware compression yourself. Otherwise it may not change your life any more profoundly than hardware error checking. And if the Macintosh on which you'll be doing your telecommunicating predates the Plus—that is, if it's a 128K, 512K, or 512K enhanced Macintosh—you probably won't be able to find a hardware handshake cable that's appropriate for an MNP-5 or V.42bis modem anyway.

Cables

Noting some exceptions, we've already said that external modems aren't computer-specific. What is computer-specific, though, is the cable that connects the modem's serial port to the serial port of your computer.
Virtually every general-purpose modem outputs serial data through a DB-25 connector, and the female of the species looks like this:

Figure 2.1. DB-25 connector, the serial port of nearly all general-purpose modems. Pin numbers apply when looking head-on at a female connector.

Macintosh computers each have two serial ports, one designated as the printer port, the other as the modem port. Only the printer port supports AppleTalk, but otherwise both ports function in much the same fashion. If you're not on an AppleTalk network, you can be as contrary as you like and print through the modem port while you telecommunicate through the printer port. The serial connectors are of two types, depending on a given machine's vintage. The most common type is the Mini DIN-8, which is found on everything from the Mac Plus on up:

Figure 2.2. Mini DIN-8 connector, the serial ports of Macs from the Plus on up. Pin numbers apply when looking head-on at a female connector.

Macs older than the Plus—the original 128K, the 512K, and the 512K enhanced—use DB-9 connectors for their serial ports. DB-9s look like truncated DB-25s:

Figure 2.3. DB-9 connector, the serial ports of the original Mac 128K, the 512K, and the 512K enhanced. Once again, pin numbers apply when looking head-on at a female connector.
So the very first thing that you need to be concerned with when you get a modem cable is to make sure it has the right connectors at either end. Early Macs need a DB-25 to DB-9 cable, later ones a DB-25 to Mini DIN-8.

That taken care of, are your worries over? Hardly. Not when 25 pins can be wired to 8 or 9 pins in just short of a zillion ways. Before MNP came into widespread use—and that isn't very long ago—there was a well-defined entity known as the Macintosh modem cable. It had a DB-25 connector at one end and either a Mini Din-8 or a DB-9 connector at the other. It could link just about any modem to a Mac, and serenity reigned over the land.

But with the advent of hardware-based compression, the ball game changed abruptly. Let's say my modem is receiving data at dreary old 2400 bps. Let's further say that the data is being compressed by the sending modem and that the nature of the data is such that it succumbs to two-to-one compression. Being aware that the incoming data is compressed, my own modem goes ahead and decompresses it; for every 1,000 bytes of data that come through the phone line into my modem, 2,000 are forwarded to my computer. Therefore, if my modem is to keep up with the incoming 2400-bps data stream, it has to send data into my computer at 4800 bps. Anticipating this very state of affairs, I've shrewdly instructed my terminal-emulation program to set the speed of my computer's serial port to 9600 bps (just to play it safe).

Now, somewhere in my computer's memory, my software has reserved some space to store incoming data. That particular chunk of RAM is what's known as a buffer. The buffer acts like a storage tank. As new data arrive at the inlet, the data already in the tank get written to disk or printed or maybe saved somewhere else, all depending on what I've instructed my telecom program to do with it. But the buffer is of fixed size, so if I don't do something soon with the data in the buffer, the buffer won't have any room to accept more data.

Should the buffer become full, it would be nice if the computer were able to emit a signal that says in essence, "Hey, stop pumping data until I can deal with what's already in the buffer." And indeed, a number of such flow control systems do exist. The most widely used software-based flow control system is what's known as XON/XOFF. XOFF (Control-S) means stop sending, XON (Control-Q) means start sending again. When my software wants a breather, it sends a Control-S character (a non-printing byte whose decimal value is 19).
When it's ready to accept more data, it sends a Control-Q (a non-printing byte whose decimal value is 17). This is what's known as software handshaking.

As long as what is being sent is pure text, XON/XOFF is a reasonable strategy for flow control. But when you're transferring binary files—applications, graphics, or precompressed data—there's always the chance that some 8-bit group within the file will have the decimal values 19 or 17 and screw up your flow control. That's seldom a problem when your modem and your computer's serial port are both running at 2400 bps. At that speed, your computer can almost always handle the data as fast as it comes in, in which case it won't need to resort to flow control in the first place.

At higher speeds, though, software flow control is notoriously unreliable, and another, hardware-based method of flow control is much to be preferred. We'll look at hardware handshaking more rigorously in due course, but for now let it suffice to say that a standard serial connection includes a number of lines (wires) in addition to those that actually transmit data. One is called the request to send (RTS) line, another the clear to send (CTS) line. By applying different voltage levels to the CTS line, a modem can tell a computer either to continue pumping data or to stop until it has finished dealing with the data it already contains. Similarly, by applying different voltage levels to the RTS line, a computer can say much the same thing to a modem.

If you have a 2400-bps modem capable of hardware-based compression, or if you have any 9600- or 14,400-bps modem, you definitely want to invoke hardware handshaking, which means you want a cable that can handle hardware handshaking. Unfortunately, the old standard Macintosh modem cable can't. Yes, you safely can use one if you are telecommunicating at no-frills 2400 bps. Otherwise you want to be sure to obtain a hardware handshaking cable that works with the modem you'll be using.

Lamentably, that's not always an easy matter. In past years it's been vastly easier to get a snazzy modem than a proper cable. Stores and even mail order houses were commonly selling modems with the old-style cables, and what's particularly tricky is that fancy modems will work with those cables—sometimes. They'll work, that is, until one day, under MultiFinder or System 7.0, another program operating in the background hogs so much processor time that your telecom program can't empty its buffers fast enough to make room for new data coming in. The new data will then overwrite—i.e., clobber—the data that's already there.
Recently, vendors have given signs that they're becoming aware of the situation. When you buy a cable, make sure the people you're buying it from know modems and Macs. Salesmen at generic computer stores often don't. Their profession has often—sometimes justly, sometimes not—been deemed a refuge for the technologically impaired. So get your cable from a reliable source or, if push comes to shove, think about making one yourself. If your supplier doesn't make you swell with confidence, check Appendix C of this book. It lists a few sources for hardware handshaking cables, and it also provides some guidelines for assembling a cable from parts available at many electronics stores.

Final Thoughts

Before we leave the subject, there are a few additional notions I'd like to touch on. If you're buying a new modem, you might as well get one that can also send and receive faxes. The extra cost is minimal, and when you send documents directly from your Mac they arrive looking much prettier than documents faxed from a fax machine. All the inexactitudes introduced by the optical scanner in an originating fax machine will be missing: no random dots, no blurs from tuna fish stains.

If the phone cord that comes with your modem isn't long enough, don't hesitate to get a longer one at the hardware store. Modems use standard RJ-11 phone cords, the same kind used by ordinary modular telephones.

If you'd like a more detailed feeling for what's out there in the way of modems, consult Appendix D. And if, once you lay your hands on a modem, you'll need to start telecommunicating instantly, think about buying one that's bundled with telecom software. Don't worry too much about how good the software is. You'll understand more about your software needs a month after you get started. We'll discuss terminal-emulation programs presently, but anything that functions—even the primitive telecom module in Microsoft Works—will get you going.
Acquiring a Terminal Program

Very basically, by serving as the software intermediary between your computer and your modem, a terminal emulation program turns your computer into a telecommunications terminal. It begins by giving you a window in which you can see what you are typing and what your modem is sending back to you. Beyond that, pretty much every terminal program gives you some array of essential features and less essential conveniences. Terminal programs that are small, old, and cheap are obviously going to be light on conveniences. Those that are big, new, and expensive are often so festooned with conveniences that they can seem burdensome and overwhelming. Mousing through myriad menus and buttons or flipping insanely through the manual of such a program in an effort to configure his terminal, a novice can sometimes find himself mired in so much unfamiliar nomenclature that he may opt to throw in the towel and go back to sending floppies in the mail in the manner of his ancestors.

If you don't have a terminal program at all, then, here are some legal ways to get one:

- Use one that comes bundled with a modem.
Download a shareware terminal program from an online service or BBS to your computer. (Yes, it's impossible if you don't already have a terminal program, but read on.)

Have someone else download a shareware program for you.

Get a copy of someone else's shareware program. (Shareware authors encourage the free distribution of copies.)

Use someone else's commercial terminal program on his computer (in which case you're not infringing on copyright laws) to download a shareware program for yourself.

Look for advertisements by companies that distribute shareware through the mail. Since they don't collect the shareware fees themselves, they charge only a few dollars per disk. You'll find their ads in the back of Mac-oriented magazines like *MacUser* and *MacWorld*.

Buy a commercial program.

If you are just getting your feet wet, or even if you've already done a limited amount of telecommunicating, I recommend that you begin with something simple. A number of modems are bundled with early versions of MicroPhone, one of the standard commercial Mac telecom programs. Over the years MicroPhone has become about as feature-laden as such programs can be, but any of the older bundled versions can serve as a good starting point. Another adequate beginner's program that often comes packaged with modems is MacKnowledge. Several Hayes modems are bundled with Smartcom EZ, a truncated version of Smartcom II that's also friendly to beginners. And if you have access to one of the integrated software packages like Microsoft Works or ClarisWorks, they too contain communications modules. The one in Microsoft Works is about as rudimentary as terminal programs get, but it will still do the job of downloading something more elaborate from a BBS or online service.
If you think you can arrange to get hold of one of the shareware terminal programs, you ought to be aware that the most popular of the bunch is ZTerm. Is going with the herd desirable? To Ibsen, no; to you, very possibly yes. In the event you run into problems you can’t solve by reading ZTerm’s documentation, calling a friend, or enquiring into this book, you can always turn to the Macintosh telecommunications echo on your nearest BBS. In recent years, of the many shareware terminal programs available for downloading, ZTerm has been by far the most discussed and analyzed. Still, some others you might want to try are FastComm, UATerm, TermWorks, Terminal, MultiXfer, Termulator, and Term-Plus. Each implements its own notion of what a Macintosh terminal interface ought to be, and one of those notions may just coincide with your own.
Figure 3.2. Termulator’s main window, also resizable. The user-definable macro buttons along the top facilitate easy dialing of commonly used numbers, sending commonly used sequences of characters, etc. Like ZTerm’s, the window displays elapsed time and, if you set the connect charges of your online service, also displays a running count of how many of your pennies have dropped into your service’s piggy bank at any time during a session.

Warning: This is as good a place as any to warn you about a mistake often made by inexperienced telecommunicators: If you are trying out several terminal programs and you use either System 7.0 or MultiFinder under System 6.x, don’t open more than one terminal program at the same time unless you know exactly what you’re doing! Terminal programs generally work fine while non-terminal programs are running. You can, for instance, upload or download large files in the background while you are writing your long-awaited memoirs using a word processor. But if two terminal programs are running, and they’re both set to use the same serial port, they will be competing for the same data. Any time a character comes in, it will be hit or miss as to which program gets it first, and if one program gets the character, the other may very well not. So, unless you have connected one modem to your modem port, another to your printer port, and have been careful to let each program know what you’ve done, experiment with your terminal programs one at a time!

If you do begin with a shareware terminal program, I recommend that you choose one that supports the ZModem file-transfer protocol. A file transfer protocol is the ensemble of handshaking signals, data formats, and error-
checking algorithms used by two connected terminals (for most purposes read "computers") while one is transferring a file to the other.

For quite a few years an older protocol, XModem, dominated personal computer telecommunications, and even now just about every terminal program supports it. Then YModem came along. YModem resembles XModem in many ways, but it offers the additional convenience of batch file transfer. That is, where XModem requires that you send or receive one file at a time, YModem lets you specify a list of files to be uploaded or downloaded without having to intervene each time a file goes in or out. YModem also offers the YModem-g variant, in which the ability to retransmit erroneous data is sacrificed to greater speed. ZModem adds still more benefits. It’s not quite as fast as YModem-g, but it’s faster than XModem or YModem-non-g, and it always allows for retransmission of erroneous data. It also has two nice features that its progenitors lack. Using earlier protocols, if you bomb in mid-transmission (because, let’s say, you lose your phone connection for some oddball reason), you’ve lost all the data that’s come in up to that point. The ZModem protocol, however, lets your terminal program save partial files, so that, when you reconnect to whatever service you were using and begin downloading the file that bombed, the download will resume at the point at which the bomb occurred. Even more convenient is the fact that ZModem doesn’t have to be manually invoked. If you’re online with someone who wants to send you a file and you’re using XModem, he has to tell you to
instruct your program to start waiting for an XModem receive. With ZModem it's all automatic. The other terminal tells your terminal that a ZModem transmission is about to begin, and your own program automatically puts itself into ZModem receive mode. This last feature is one of which beginners are inevitably fond.

When ZModem hit the streets, incidentally, ZTerm was pretty much the first Macintosh terminal program to support it, which is largely why it came to be so popular. By now, however, some other shareware terminal programs also support ZModem. Two general-purpose ones are Termulator and Terminal. One that offers less generality of use is MultiXfer, but it features the novelty of allowing both connected parties to exchange files in both directions while they chat back and forth online all at the same time. It's also freeware, which is even cheaper and less guilt-engendering than shareware.

![Figure 3.4. MultiXfer uses two windows, one for basic terminal functions and one for file transfers. Where most terminal programs don't let you do much else while you are transferring a file, MultiXfer lets you "chat" with your correspondent in the File Transfer window while the transfer is in progress.]
I've intentionally held off discussing the out-and-out purchase of a commercial terminal program because at this stage of our investigation it's difficult to prescribe one. If you're keen to make a major purchase, though, you ought to be aware that the field of commercial Macintosh terminal emulation applications is largely commanded by four major programs: White Knight, MicroPhone, Smartcom, and VersaTerm. (Actually, it would be more accurate to speak of them as program families. White Knight and its Red Ryder antecedents would make up one family, MicroPhone and MicroPhone II in their various versions another, Smartcom and Smartcom II still another, and VersaTerm and VersaTerm-PRO the last.) All four are very powerful, and each has a different sort of appeal.

For now, though, rather than offering specific recommendations I feel it makes more sense to scatter some observations. Major commercial telecom programs all support one or another form of scripting, i.e., the creation and execution of command-language instructions that control a program's execution of frequently used procedures. In the telecommunications libraries of Macintosh BBSs and the Mac forums of online services, White Knight and MicroPhone scripts seem by far the most numerous, with White Knight decidedly in the lead. Scripting is really a limited form of programming, and, if that's your bent, then one of those two programs should probably be your choice. (Speaking personally, I'd give the scripting edge to White Knight.) Then again, both MicroPhone II and White Knight have complex and rather quirky interfaces. Smartcom's command language is also quite powerful, and its interface is somewhat cleaner, which can make it a good choice for comparative beginners.

As for who likes what, serious amateur telecommunicators tend to gravitate toward White Knight, maybe because so many of them were weaned on Red Ryder during its shareware days. People who telecommunicate in offices and other professional situations, where technophobia is likely to eclipse the hobbyist's love of tinkering, often prefer MicroPhone. The difference between the feel of these two programs—make no mistake about it; the difference is considerable—is something like the difference between the feel of computing in the Macintosh environment on the one hand and the IBM-compatible environment on the other. White Knight is very Macintosh. To understand White Knight at its deepest levels you really need to understand the Macintosh at its deepest levels and know a thing or two about your
modem in the bargain. By contrast, MicroPhone tries to isolate you from the nuts and bolts of your modem and computer, much in the way that, say, WordPerfect for IBM-compatibles tries to protect you from the perils of DOS. Non-computer freaks who use PCs will talk about mastering WordPerfect or Lotus 1-2-3 a lot more frequently than they will talk about mastering DOS or the PC, since WordPerfect and Lotus 1-2-3 have already mastered DOS and the PC. Similarly, although MicroPhone II is every bit as complicated as White Knight, many of its complications are its own and not the Mac's per se. Essentially, in mediating between you and the intricacies of the Mac serial port, it translates the Mac's complexities into its own complexities. In the end, it's very much a case of six of one and half a dozen of the other; the complexities are unavoidable. It all comes down to whose complexities you want to wrangle with.

Figure 3.5. Just to provide some contrast between typical shareware programs and the big commercial terminal programs, here are four windows from White Knight, each with a different “status bar.” Users can choose whichever one suits their needs. At the upper-left is WK's general status bar. The upper-right shows its VT100 status bar, which, naturally enough, you would use if you were emulating a VT100 (eventually you'll probably want to try it). The DEC VT100 terminal has a numeric keypad and cursor movement (arrow) keys, neither of which are present on older Mac keyboards, so the window simulates them with buttons. At the lower-left is the buffered keyboard status bar, which you would use if you want the chance to edit a whole line before sending it out. The window at the lower-right shows WK's macro status bar, which, like the Termulator window in Figure 3.2, displays your user-configured macro buttons. A fifth option not shown is to use no status bar at all. The interface may appear more complicated than it is. All those options can prove to be convenient.
People who need to telecommunicate with mainframe computers and other specialized non-Mac terminals, as well as people who already have been trained in the use of such terminals, generally put their money on VersaTerm. Like quite a few other terminal programs, VersaTerm can emulate the DEC (Digital Equipment) VT100 terminal, but there are emulations and emulations, and VersaTerm’s is the most thorough one I’ve seen. Two more text terminals it emulates very well are the DEC VT220 and the DG (Data General) D200. It also does a nice emulation of the Tektronix 4014 graphics terminal. VersaTerm-PRO includes all the VersaTerm emulations, adds a Tektronix 4105 emulation, and is capable of scaling its graphics to fit large-screen monitors. Scripting is not VersaTerm’s strong point, and, as of this writing, they don’t support ZModem either. But for those who commonly find themselves connected to something other than a Mac or PC, VersaTerm and VersaTerm-PRO are often the terminal programs of choice.

Figure 3.6. The terminal window of MicroPhone II. This one is configured by a MicroPhone script, supplied with the program, to call GEnie, a popular online service. Like some windows you’ve seen before, it contains user-definable buttons. The typical MicroPhone window-cum-menus is deceptively simple. However, as you begin to pull down the various menus to explore the options they serve up, you may come to feel you are at the gateway to a labyrinth. Nonetheless, MicroPhone offers nearly every conceivable convenience, and its intricacies are worth mastering.
Conclusions? If none of this makes a whole lot of sense right now and you feel compelled to buy a program, fall back to a more common sense criterion and get one that the person at the other end of the line will be using too. There's always the chance that they know more than you do and can hold your hand when things get dicey. A great luxury during the early going is having two phone lines, one digital and one voice, so someone else can talk you through your troubles while you're both online. Otherwise, I stand by what I said before. Start simply, then let your own experience guide you in your choice.
Chapter 4

Configuring Your Terminal to Your Taste

Setting Your Settings

Somewhere in the menus of almost every Macintosh program there's an item called "Preferences" or "Settings." Occasionally there are both. If a program is designed for drawing or word processing, it's perfectly conceivable for someone to use such a program for years without really coming to grips with either item. A prudently chosen array of default settings or default preferences can often serve your purposes without ever being changed or even inspected.

Terminal programs are decidedly not that kind of animal. With terminal programs, settings come very close to being the whole ball of wax.

Because terminal programs have so many settings, they can't in good conscience give you a single window with buttons for every available choice. Even if it were crammed into one screen, the sheer clutter of it all would flout every principle on which the Mac's user-friendly interface is based. So what nearly every terminal program does is group its settings according to the predilections of its author. Of course, every author's personality has its own array of settings, and the rationale behind the interface will vary widely from
one terminal program to the next. A choice that appears under one program's Modem menu may be buried in a submenu of another program's Communications menu. MicroPhone and White Knight, for instance, each give you settings for all the telecommunications basics together with a wide variety of customizable conveniences. Yet their working environments are so different that at times you might find yourself wondering if they are both really terminal programs at all.

The point I'm working toward is that it would be nice if the rest of this chapter could be organized in a way that parallels the organization of the great majority of terminal programs. However, no such entity exists, and instead we'll try to take things in a sequence that makes some kind of sense a) on the printed page and b) to a telecommunications novice. When you have the time, you can explore on your own the terminal program you've decided to use and find out which of its menus and buttons implement the choices you want.

Most telecom settings fall into two broad categories. Some will apply regardless of with whom you are online, while others will vary according to the party at the other end of the connection. In this chapter, we'll look at the first group of settings—the ones in which your own personal preferences and the special characteristics of your computer shape your everyday use of your terminal program.

**Fonts**

The screen of a typical telecommunications terminal normally displays 24 lines of up to 80 characters each. Generally the characters are monospaced, which is to say that, unlike most Macintosh fonts, their spacing isn't proportional to their shape. Two common Mac fonts that are monospaced are Monaco and Courier. In proportionally-spaced fonts, such as Times or Geneva, for example, a period occupies much less horizontal space than a w. But 12 ten-point characters in the Monaco or Courier fonts always measure an inch across regardless of what letters, numbers, or punctuation marks they happen to be. With its big serifs, Courier is somewhat visually noisy for regular use as a terminal font. That leaves Monaco—sort of.

The small screen of all the classic Macs, from the original Mac 128K through the 512K, 512K enhanced, Plus, SE, SE/30, Classic, and Classic II, is
able—just barely—to display 80 nine-point Monaco characters across a
window that's been sized to fill the whole screen. The same window can
display 24 single-spaced lines, but when it does there isn't much room left for
buttons, sub-windows, status bars, or whatever other luxuries your terminal
program chooses to bestow on you. So, in furtherance of the cause of
terminal emulation, a number of programs contain their own internal
terminal fonts. Most often, the fonts are based on Monaco, but their line
spacing—called leading and pronounced "ledding" in the printing trades—is
sufficiently tight so that 24 text lines can comfortably share a small screen
window with some buttons, status bars, whatever.

Terminal programs that use an internal font generally default to nine-point
type, but some give you the option of using the same font in a larger, easier-
on-the-eyes version. A few also let you use any other font that you've
installed in your system. So, depending on the size of your screen and how
big you like your terminal window to be, you'll probably want to pick a font
and then set its size. When you get online, you'll notice that most people who
create fancy layouts or send columnar data do most of their formatting by
adding spaces. The formatter virtually always makes the assumption that his
correspondent(s) will be using monospaced fonts, so if you want incoming
columns to come out straight and centered headers to come out looking
centered, start by picking a monospaced font for your terminal.

Incidentally, terminal programs that put up resizeable windows usually let
you view a lot more than 24 lines at a time if your computer's screen has room
for them. But bear in mind that when you're receiving text, your program will
probably update its terminal window on a character-by-character basis. The
window will scroll automatically every time a new line comes in, and if you
have a fast modem and not all that fast a computer, the program may not be
able to update the window as rapidly as the data arrives. Scrolling a window
that displays 50 lines of big type takes significantly more time than scrolling
24 lines of small type, especially if you're relying on TrueType or Adobe Type
Manager to form the characters. What's liable to happen is that the effective
transfer rate of the data coming in may diminish as your terminal program
keeps attempting to refresh your screen. Fast screen updating is one of those
tele-macho points on which terminal programs are always challenging one
another, and if you are not sure how proficient your program is at screen
scrolling, be conservative when you set your font and window sizes.
Which Serial Port?

Your terminal program needs to know to which serial port your modem is connected. You will have to find out how your program accepts the setting, then tell it whether the modem is hooked up to your Mac's modem port or its printer port.

![Connection Settings](image)

Figure 4.1. One of VersaTerm-PRO's many settings windows. At the lower-right, you click on an icon to tell it which serial port you'll be using.

Look at Figure 4.1 for one of the many variations on serial port selection. The window displayed by VersaTerm-PRO has actually been put up by an entity called the Apple Modem Tool. Recently Apple created a set of communications resources collectively called the Communications Toolbox, which is built into System 7.0 and can be installed into later versions of System 6.x. System extensions that access the Communications Toolbox are called communications tools. Some programs let you use communication tools and some don't. VersaTerm-PRO likes you to use them liberally. We'll talk about communication tools again later, but for now think of them as miniature telecom programs to which larger telecom programs give you access. You configure the Apple Modem Tool just as you would any other telecom program, and in
VersaTerm-PRO it's accessed via the Settings menu. But just because the window is labeled "Connection Settings" don't jump to any rash conclusions. None of the options under the program's Settings menu actually says "Connection Settings" or even just "Connection." The window was called up by choosing a menu item called "Config...[Apple Modem]." The VersaTerms (PRO and non) may be a little less internally consistent in their use of nomenclature than some of the competition, but several other programs give them a good run for their money. Welcome to the dizzying world of high-end terminal emulators!

**Keyboard Settings**

For telecommunications purposes, there are essentially three Macintosh keyboards. The ADB keyboards used with every Mac from the SE on up have a Control key, arrow keys and a numeric keypad. The Mac Plus keyboard has a numeric keypad and arrow keys but lacks a Control key. Earlier Mac keyboards have no Control key, no arrow keys, and no numeric keypad.

Should any of this concern you? Possibly not; probably so. While an application is running, it can obtain information about your keyboard from the Macintosh system, and the terminal program you've chosen may be one of those that checks to see which keyboard you're using. If it's an ADB keyboard, you even may be able to get away with ignoring any keyboard settings until the need to know becomes more pressing. If you don't have an ADB keyboard, you'd best deal with your keyboard settings at the outset.

The reason keyboards are an issue at all is that the Control key, and sometimes the arrow keys, have special functions in telecommunications. As we said in Chapter 1, the ASCII (an acronym for American Standard Code for Information Interchange, usually spoken as "ass-key") character set consists of 128 characters with decimal values ranging from 0 to 127. ASCII characters whose decimal values are between 32 and 126 are all printing characters. ASCII 65 through 90 are the uppercase alphabet, ASCII 97 through 122 are the lowercase alphabet, ASCII 32 is a space, ASCII 42 through 47 are common punctuation characters, ASCII 48 through 57 are the digits 0 through 9, and so forth.

The ASCII characters whose decimal values are between 1 and 31, however, are control characters. Instead of printing, they control the terminal or
quasi-typewriter that your computer is emulating. For example, the last four control characters, decimal 28 through 31, move the cursor around in a manner corresponding to the four arrow keys on those keyboards that have them. ASCII 1 through 26 are usually called Control-a through Control-z (often denoted as ^A and ^Z), and if you have a Control key, you can send the characters simply by holding down Control and typing a character between a and z. ASCII 13 is a carriage return, which you normally send by hitting the Return key. If you have a Control key, however, you can also send a carriage return by typing Control-m. Control-h (ASCII 8) is a backspace, and typing an h with the Control key down is the same as hitting your Delete key (Backspace on earlier Mac keyboards). Similarly, Control-i (ASCII 9) is a tab and Control-j (ASCII 10) is the perennially pesky linefeed character. Unlike a typewriter, which normally advances the paper to the next line every time you return the carriage to the left margin, a telecom terminal divides that process into two stages, and each stage has its own control character. A carriage return instructs the terminal's real or simulated carriage to return to the left margin without advancing to the next line, and a linefeed character instructs it to go on to the next line, leaving the carriage just where it is. In the Macintosh non-telecom environment, those two functions have been reunited, but raw text produced by IBM-compatibles still has both a carriage return and linefeed after every paragraph.

The reason it's helpful to know all this is that a handful of control characters sometimes come into play when you're online. On the next page is a table of some of the more common ones.
Table 3.1 Terminal Control Characters

<table>
<thead>
<tr>
<th>Key</th>
<th>Decimal Value</th>
<th>What It Does</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control-g</td>
<td>7</td>
<td>The &quot;bell&quot; character. On a Mac it usually produces a system beep. If you are online with someone else, they'll hear a beep when you type it. Good for waking up someone who's asleep at the terminal.</td>
</tr>
<tr>
<td>Control-c</td>
<td>3</td>
<td>The interrupt character, usually used to interrupt some function being carried out by the terminal to which you are connected. If, for instance, you've asked some BBS to list all the files in its Mac library to your computer and then decide you haven't got the time, sending a Control-c may interrupt the listing and get you a prompt that gives you the option to resume or abort the listing.</td>
</tr>
</tbody>
</table>
| Control-l | 12            | This is the "clear screen" character. If the terminal programs that you and your correspondent are using are set to permit screen clears, this character tells both connected terminals to clear their screens and start with *tabula rasa*.
| Control-s | 19            | The XOFF character. It tells the terminal at the other end to suspend transmission temporarily. |
| Control-q | 17            | The XON character. It tells the terminal at the other end to resume transmission. |
| Control-k | 11            | Often used as an abort character to cancel the current transmission. |
| Control-x | 24            | Also used as an abort character. Typically, a BBS will tell you in one of its prompts which abort character it uses. |
| Control-m | 13            | Same as return. |
| Control-j | 10            | Linefeed. |
| Control-i | 9             | Same as tab. |
| Control-h | 8             | Same as Delete on the Macintosh ADB keyboard (or, on earlier Mac keyboards, Backspace). In telecom-ese it's normally called *backspace*. |
An alternate delete code sometimes called *rubout*. It's falling out of use, but one day you might go online with a system that still uses it. Regardless of whether your Mac keyboard says "Delete" or "Backspace," it normally sends a Control-h (i.e., ASCII 8) when you hit that key. So in the Mac keyboard realm, Delete and Backspace mean the same thing. But in the telecom realm, Delete and Backspace refer to two different characters. ASCII 8 is a backspace, ASCII 127 is a delete/rubout. To accommodate systems that prefer deletes, some terminal programs let you remap your keyboard so that when you hit Delete (or Backspace) you send an ASCII 127 and not ASCII 8. Refer to Figure 4.3 for an example of how you might configure such a remapping.

Now all these fascinating little facts may seem very nice if you indeed have a Control key, but a great many Mac keyboards don't. And if yours is one of those that doesn't, you probably have to tell your terminal program what kind of keyboard you're using, and what key to use to fake the Control key. Upon learning that you're not using an ADB keyboard, some terminal programs default to the assumption that you'll want your Option key to serve as the Control key, in which case you would send a Control-c by holding down the Option key and typing c. Other programs default to the Command key, and many leave the choice to you. Whatever the case, you will do well to make sure your terminal program is aware of your preference.
Chapter Four Configuring Your Terminal to Your Taste

Figure 4.2. In MicroPhone II you can call up this window by choosing Preferences in the Settings menu. The pop-up menu at the upper-right lets you assign any of three keys to serve as the Control key.

From time to time, a prompt from a system that you’re online with may ask you to press the Escape key (ASCII 27), and, once again, if you have one of the old non-ADB keyboards you won’t be able to find it. What a number of terminal programs do, therefore, is give you the option of mapping the Escape character to another key. You may never need to send an Escape in all your telecom experience, but then again, depending on with whom you connect, you may need to use it a lot. So just to be safe, if you do have an older keyboard and your terminal program supports remapping the Escape key, remap it. I recommend mapping it to the ` key (that’s accent grave) because a few programs default to it.

Incidentally, in Figure 4.3 on the next page take note of a unique White Knight feature. In frank acknowledgement that telecom settings can become too much of a morass to be navigated by the Mac’s normal command key system, White Knight offers double command key equivalents. As the Options submenu indicates, you can invoke the Key Mapping window by holding down the command key and typing OM.
Finally, there's the matter of the break. Along with the set of ASCII characters, your modem can send what's called a break signal, and every so often you may need to send one. With some systems, breaks function like Control-c interrupts. They also come in two lengths, short (233 milliseconds) and long (3.5 seconds). Some terminal programs assign a fixed key to each of the breaks; others allow the user, within limits, to create his own key assignments. Therefore, you want to find out how your terminal program handles
breaks, and whether you can exert your own preference on which keys send them. Figure 4.3 shows you how White Knight does it.

We'll wind up this first look at keyboard settings for now, but we'll be returning to them soon enough. True to our design, we've examined some that have to do with you and your computer. In the next chapter, we'll cover keyboard settings that vary with the system you're online with.

**Screen Buffers (a.k.a. Scroll Buffers)**

During any online session, as you type to your correspondent and your correspondent types back to you, the developing lines of prompts and responses, prose and verse, messages and replies, all scroll upward off the top of your terminal window and on into the enveloping ether overhead. Incoming information often flies by more quickly than you can read it, and inevitably you'll want to look at it again. To enable you to do that, just about every telecom program lets you scroll back your terminal window to any number of earlier points in the session. In order to be able to do that, it has to save the text that's flown by. The place in which it keeps all that information is an area of your Mac's memory known as the *screen buffer* (or sometimes *scroll buffer* or *scroll-back buffer*). Then, when you do scroll backward, your terminal program can feed the contents of its screen buffer into the top of your window, and, in a more leisurely fashion, you can see just what it is that's been going on.

Once you've scrolled back to an earlier portion of your session, most programs let you select text in your terminal window just as you would in a word processor. You can then copy it to the Clipboard and paste it into the Scrapbook or a word processor document or even back into the very same terminal program. Actually, pasting text back into your terminal program is something you'll probably do fairly often. If you log onto a BBS or online service expecting to download some files, you'd typically begin by asking the system either to list the files in one of its libraries or search a given library for files of a particular type. When the service complies, the list it gives you may scroll by much too fast for you to inspect it closely. What you would do then is scroll back to the beginning of the list and look at the file descriptions more carefully. You then decide which file you want to download, select it with your cursor so you capture its exact name, then copy it. Now, with the
file name in your Mac's Clipboard, you scroll down to the bottom of your window and tell the system you want to download a file. When it prompts you for the file's name, all you have to do is type Command-v or choose Paste from the Edit menu. Pasting text into a terminal window is normally identical in its effect to typing into the window directly.

Another thing that many terminal programs let you do is search for text stored in your screen buffer just as you would in a word processor. If something about great auks went by five minutes ago and you're mad for anything you can learn about auks, you can choose Find from whatever menu it's in, type auk into the Find window and then let your program scroll back to that point in the session where the system you're logged onto was talking about auks.

Why are we going over this in the context of settings? Because one day you'll search for auk and come up empty. You're sure it went by and that it was spelled correctly, but your terminal program tells you "Nix on auks." When that happens, it's usually because the screen buffer became too full. You see, when screen buffers fill up, terminal programs normally remove the oldest text from the bottom of the buffer to make room for more recent text coming in at the top. The auk lore you had in your computer may have soared away as swiftly as an auk.

In most cases, therefore, it falls to you to specify just how big your terminal program's screen buffer will be. Some programs let you do it with a setting that's internal to the program. As an example, look at Figure 4.4 to see how it's done in White Knight.
Other programs take a different approach and, using the memory available to them, allocate screen buffers as large as they can get away with. If the version of the Macintosh System you use is earlier than 7.0 and you don't use your terminal program under MultiFinder, the program will try to use all your application memory, which of course is a function of how much RAM you've got installed in your computer. When that's so, the only way you can jack up the size of your screen buffer is to delete some of the fonts and/or desk accessories in your Mac's System file or delete some of the RAM-gobbling INITs you ordinarily start up with.

Nowadays, however, most telecommunicators do work under System 7.0 or in MultiFinder under System 6.x, largely because it allows them to do something else during long file transfers and not have to stare at a progress indicator until the transfer is complete. And when you are working under MultiFinder or System 7.0, you're free to adjust the amount of memory available to a given program. Each program that's running has what's called a memory partition, and you can change the size of the partition any time you
are in the Finder and the program isn't up and running. Figure 4.5 shows more about how to do this.

Figure 4.5. You change a program's memory partition from the Finder. First, make sure it's not open, then click on it once to highlight it and choose Get Info from the Finder's File menu. The Info window shown above is from System 7.0, but Info windows in later versions of System 6.x aren't dramatically different. What you do is type the number of kilobytes you want to assign to the program into the "Current size" box. If the number falls short of the minimum partition the program needs, the Finder will warn you. If the number is large beyond all reason, the Finder will accept it, but when you try to open the program, you'll get a message telling you that neither MultiFinder nor System 7.0 can allocate that much memory to the program.

With most terminal programs, increasing the sizes of their memory partitions has the immediate effect of increasing the sizes of their screen buffers the next time you open them. When you do enlarge a partition, though, you have a few factors to juggle. If the partition is too big, after you've opened the program there may not be enough total memory left under MultiFinder or System 7.0 to open much else. If the partition is too small, the resulting size of the screen buffer may shrink to a point where it's just not practical to work with. Accordingly, you may have to play around some until you come up with a partition size that suits your everyday purposes.

Capturing Text to Disk

Screen buffers are nice, but, as we've just seen, they have their limitations. During very long sessions even the largest of them eventually begin dumping old text off the top to accommodate the new text pouring in. And because they're dependent on your computer's RAM, they're necessarily volatile. A
system crash or sudden power failure will likely wipe their contents out forever.

To cover yourself, you'll often want to—and now we're talking telecom-ese—turn on a *capture file*. With a capture file open and going, text that scrolls by your terminal window is saved not just to the screen buffer in RAM, but to disk as well. When you subsequently open a capture file with a word processor, you have a complete record of the session you were capturing from the moment that the capture file was started. Because it's written to disk—floppy or hard—the file's length is limited only by the amount of free space on the disk, and because it's not in volatile RAM it will be much less fussy about crashes, power outages, or whatever. Telecommunicators whose computers don't have very much internal memory often turn on capture files routinely at the beginning of every session. Then they don't have to worry about how much text is in their screen buffer. If they need to look at something, it's always there until they put it in the trash.

Now, associated with all true Macintosh files are a couple of four-character parameters called *type* and *creator*. Applications are of type APPL, MacPaint paintings are of type PNTG, and text files are of type TEXT. When a program creates a file, it also normally stamps it with its creator code. MacPaint's creator code is MPNT, MacWrite's is MACA, and MicroSoft Word's is MSWD. When a file is of type TEXT and its creator code is MACA, you can double-click on the file and, provided the MacWrite application is present in an open volume, the file will open in MacWrite. So what many terminal programs let you do is assign a file type to whatever text files they create. That way, if you like to use Microsoft Word, your capture files will be Word files right off the bat, and you won't have to go to the trouble of first opening Word and then opening capture files from *within* Word.
Figure 4.6. In Terminal, a shareware telecom program, the window above is invoked by choosing Other... from the Options menu. The first choice in the window lets you enter the creator code of your favorite word processor. Most of the other options shown are ones we've already discussed. By the way, if you prefer to telecommunicate in total silence, telecom programs usually let you suppress the beep set off by an incoming Control-g (see Table 3.1 on page 45). Terminal does it in this window.

Incidentally, the text-capturing prowess of a number of terminal programs goes well beyond straight file captures. Several let you save the entire current contents of their screen buffer as a text file, a process sometimes known as buffer archiving (not to be confused with compressing a file using a compression program like StuffIt, which is sometimes also called archiving). Additionally, some programs let you select any amount of text in the terminal window and save the selection as a text file. With either of those features, you can save earlier parts of a session to disk even if you didn't start a capture file when the session began.
Another thing to be aware of when you're dealing with terminal programs is that the notion of saving files embraces the additional notion of appending files. Quite a few programs, that is, offer the option of appending captured text to a text file that already exists. Depending on what program you're using, you may be able to append to the end of a file: 1) the entire contents of the screen buffer, 2) a selected portion of the screen buffer, or 3) all incoming text. If you go with the third option every time you get on line, you can even save your entire telecommunications history in one humongous text file.
As for settings per se, along with specifying the creator of your capture files, your program may want you to name a default folder in which to save them. When you archive the screen buffer in White Knight, for instance, it doesn't give you the chance to name the file. The program names it Archived Text, but it lets you decide where Archived Text is saved.

**Download Files Destination**

A fairly common occurrence in telecommunications is to download a file and then not be able to find it. To keep those mishaps to a minimum, your telecom program usually lets you specify a folder in which downloaded files will be stored automatically. The folder is often called the Receive folder, and Figure 4.8 gives an example of how to set it.

![Figure 4.8](image_url)

*Figure 4.8. In MicroPhone II, choose Select Receive Folder from the Transfer menu and get the window just above. Other programs may do it a little differently, but you'll get the general idea.*

After you've designated a receive folder, you'll at least know where to begin looking.
Chapter Four Configuring Your Terminal to Your Taste

Flow Control

As was explained in Chapter 2, flow control is the regulation of the flow of data between a computer and modem, also between one modem and another. The two dominant flow-control protocols are XON/XOFF and hardware handshaking (sometimes called RTS/CTS). If the speed of your modem is 2400 bps or less and it doesn’t support one of the hardware-based compression or error-control schemes like MNP-4, MNP-5, V.42 or V.42.bis, you’ll probably want to use XON/XOFF all the time. If you have a high-speed modem (in which category I include 2400 bps modems that do support MNP-4, MNP-5, V.42 or V.42.bis) you’ll undoubtedly want to use hardware handshaking eventually. However, using hardware handshaking requires that you configure your modem by sending it a series of modem commands. We’ll start on modem commands in the next chapter, and if you’re eager to have your modem begin pressing the flesh right now you can even jump ahead to the chapter on high-speed modems. On the other hand, you may count yourself a rank beginner at all this, and you may therefore be more eager to get online with someone or something, regardless of whether your equipment is performing to the max. If that’s the case, choose XON/XOFF for now and know that hardware handshaking is waiting in the wings. If you look back to Figure 4.1 you’ll see where to set the mode of flow control in VersaTerm-PRO. Other programs give you a roughly similar window or a checkable menu choice.

Saving Settings

Okay, so now you’ve taken the trouble to configure your terminal to your taste, you’re aware that there’s still some configuring to do because you still haven’t tuned your terminal to match the other party’s settings, and suddenly you wonder, “Am I going to have to go through this @#$!%&* process every time I get online?”

Relax. Just about every decent terminal program lets you save your settings. In fact, most let you save as many different terminal configurations as you like, each in its own separate file, each file named by you and saved in whatever folder on whatever disk you choose. Then, to restore the settings, all you have to do is load the settings file into the terminal program, and you’re ready to go.
Over the course of time you'll probably accumulate a number of settings groups. Even if you succeed in personalizing your terminal once and for all, there's always the other guy's requirements to account for. By and large, for instance, the major online services are notably friendly to the Macintosh. (Business being business, they have to be.) But CompuServe likes your terminal to be set for full-duplex transmission while GENie prefers half-duplex. So, when a group of settings turn out to work properly, save them to disk knowing they'll be around when you need them.

Finally, don't worry too much right now about mastering every last setting you can find in the menus of your terminal program or the manual that came with your modem. Recently, I logged onto the BBS run by NJMUG, the New Jersey Macintosh Users' Group. When you first connect with a BBS, it usually presents you with the day's opening announcement. NJMUG is one of the best Macintosh bulletin board services in the nation, and the people who operate it are notably long on telecommunications experience. When I logged on, the opening announcement apologized in advance for any glitches or inconvenience that users might encounter. Several days ago the BBS had put a new modem into service, and the sysop was still playing around with settings.
Chapter 5

Configuring Your Terminal to Their Taste

In the best of all possible worlds, every telecommunicator would have the same kind of terminal set to the same collection of settings and wired to the same type of breathtakingly fast, laughably inexpensive modem. New technology would be distributed to everyone simultaneously to guarantee complete compatibility, the universal interface would be surpassingly friendly, and at any given moment there would be one—only one—current edition of the user's manual, affectionately called "Docs." In the rare instance that something were to go awry, you'd need only flip through the punctiliously indexed and cross-referenced pages of Docs to find out exactly how to set things right.

Fortunately, such a world still isn't upon us. If it were, it would almost certainly be dreary, bureaucratized, and ominously glitchy, probably in the same way that air travel has become dreary, bureaucratized, and ominously glitchy since the pioneer days of aviation. Right now, though, there remain all kinds of parallels between the current state of digital telecommunications and those first hectic decades after the Wright brothers. New standards for modulation, data compression, and error checking keep appearing, and
inevitably the standards are followed by newer and faster modems that try supporting them. I say “try” because, despite the CCITT’s best efforts, the newer modems almost always turn out to be less than totally compatible with some of the older ones or even with each other. Bulletin board systems around the country hum constantly with complaints from serious amateur telecommunicators about how they can’t log onto this BBS or that online service or how they can’t connect with one or another party at 14,400 bps. Experimental modem initialization strings (we’ll talk about those soon) get traded back and forth, the contents of any number of modem S-registers get posted online in hope that others might be able to diagnose all manner of quirks, and now manufacturers who churn out modems (sometimes by the seats of their pants) are even confessing online to defects in their ROMs and bugs in their software. The evolving telecommunications subculture is still chaotic, capricious, and, therefore, eminently interesting. Like piloting some rattletrap biplane off the runway and into the air, connecting with a new BBS, service, or person for the first time can seem ever so slightly miraculous, even to the most grizzled power user.

Enjoy it while you can. Your next new modem just might be your last. ISDN is coming. Integrated Services Digital Network, a service offered by the major phone companies, obviates the modem altogether. There’s no audio carrier and therefore no modulation or demodulation. Instead of a modem, every participating terminal is fitted with an adapter that links it to one or all of the three digital channels carried by every ISDN line—two B-channels and a D-channel. Each B-channel can transmit digital data bidirectionally at 64,000 bps (that’s actual speed, not some seldom-achieved effective speed obtained by compressing data). The D-channel carries control signals or functions as a slower (16,000 bps) digital data channel. ISDN already exists, and it’s estimated that by the end of the decade the great majority of phones in America will be able to access it. (Don’t be silly; of course for a surcharge.) Recently both Apple Computer and a new firm in Japan called Advertnet announced ISDN NuBus boards for the Mac. Dubbed “Xpass-MC,” the suggested retail price of Advertnet’s board at this writing is $1,500. That makes it more pricey than a modem, but the Mac Plus I bought for $2,500 in 1986 can now be had for around $400 second-hand, and, in a swapfest, the 1200 bps modem I bought for $250 soon afterward would get me maybe half a Kit Kat bar at best.
Nevertheless, if you're still reading you're probably too eager to get online to wait the half decade or so before ISDN is in widespread use among individual telecommunicators. And even when it is, you can safely bet there'll be settings to contend with. Telecommunications without settings is no more imaginable than commercial aviation without those tiny packages of honey-roasted nuts.

Let's go ahead, then, with a few more settings you may want to think about before connecting. Because there's a kernel of truth in the notion that you have to learn to walk before you can run, you may want to learn to talk just a little bit to your modem before you talk to anyone else through your modem. (Even if you're using MicroPhone, which likes being the sole intermediary, you'd be wise to strike up at least a brief speaking acquaintance with the gadget.) Whereupon...

Terminal Program Settings

In the previous chapter most of the settings we looked at were intended to please you. Now we'll look at some whose purpose is to please the party with whom you'll be going online. To keep our nomenclature uniform, we'll call that party a service. The service may be a BBS, a commercial online service, a friend's computer-cum-modem, a corporation's private telecommunications service, whatever. As variable as all of those may be, many of the preferences of these services can be said to be fairly typical. We'll note which ones are and which aren't.

Once you finally do get online, you'll sometimes come across the abbreviation BTW. It stands for "by the way," and it cuts down on online typing. Any discussion of matters related to computing invariably leads to incidental observations of one sort or another. Starting now, as those come up I'll flag them with BTWs.

Extended Characters

Sometimes also called special characters, these are the Macintosh characters that take up where the ASCII set leaves off. Of the Mac's 256 possible characters, the 128 ASCII characters have codes from zero to 127. Starting with 128 and going up to 255, we're in less well-defined terrain. Different
Computer manufacturers often assign different characters to these codes, and with the Macintosh they even vary from font to font. The extended characters include many common foreign characters like ñ and é as well as symbols like ¿ and ¿. Most are invoked by holding down the Option key, sometimes in combination with the Shift key. Option-semicolon, for instance, gets you the character whose decimal value is 201, the familiar ellipsis character ..., which looks a little like three periods in a row but isn't.

**BTW**, a lot of people who use macro programs such as QuicKeys have trouble assigning a universal key to the familiar Save As... item that appears under most applications' File menus. The macro will work with some applications but not others. Often that's because in some applications the ellipsis after "Save As..." is a true ellipsis while in others it's just three periods, so the macro program doesn't always find the menu text it's looking for.

**BTW**, if you want to explore the extended characters of all your fonts, download a shareware utility called ASCII Chart. A desk accessory written by Jon Wind, it shows you all the characters in all the fonts in your system along with their decimal and hexadecimal codes and how

![ASCII Chart](image)

**Figure 5.1.** ASCII Chart's window for the Geneva font in 10-point size.
to type them. It's remarkably useful, and virtually every BBS or online service with a good Macintosh utility library keeps it on file.

Good terminal programs give you the option of using or not using extended characters. Most services you'll connect with won't use them, so you'll probably want to start out with your program set accordingly unless you know for sure the extended characters will be accepted. And in the event that a) you do elect to use extended characters, and b) your keyboard doesn't have a Control key, make sure to fake the Control key with something other than your Option key. That's because you'll be needing the Option key to send extended characters in the first place.

Finally, be aware that different terminal programs express themselves in different ways. In Figure 5.2, ZTerm says the same thing twice: "No Extended Chars" and "Strip hi bit." You need eight bits to encode 256 different characters but only seven bits to encode 128 characters. The high bit of an eight-bit Macintosh byte comes into play only when a character's decimal value is greater than 127. Otherwise it's always zero. MicroPhone II expresses its extended characters option by way of a button in its Terminal Settings window that's labeled Strip 8th Bit. In telecom-ese, you see, there are often several ways of saying the same thing. You may object that that's confusing, but all rich languages offer an abundance of synonyms.

<table>
<thead>
<tr>
<th>Terminal Settings for 'Local'</th>
</tr>
</thead>
<tbody>
<tr>
<td>☒ No Extended Chars (Strip hi bit)</td>
</tr>
<tr>
<td>☐ VT100™ Keypad</td>
</tr>
<tr>
<td>☐ Send RUBOUT for Backspace/Delete</td>
</tr>
<tr>
<td>☐ Don't drop DTR on exit</td>
</tr>
<tr>
<td>☐ Destructive Backspace</td>
</tr>
<tr>
<td>☐ Auto Line Feed</td>
</tr>
<tr>
<td>☐ VT100™</td>
</tr>
</tbody>
</table>

**Figure 5.2.** The Terminal Settings window from ZTerm. The button at the top applies to extended characters.
Text Pacing

At this moment in time I can think of four different ways to send text to a service you're connected to without using one of the error-checking protocols. You can type it, you can copy a block of text from somewhere else and paste it into your terminal program, you can create a macro or script that types something for you whenever it's invoked, or you can instruct your program to open a text file and send its contents. Whichever one you choose, only you know you've chosen it. As far as the receiving terminal is concerned, incoming text is incoming text.

When you type directly, the pace at which the characters go out is as slow and halting as your typing. (I don't mean to malign your keyboard skills, but a computer you're definitely not.) However, an attack of forethought may have led you to say, "Hey, what am I doing typing these sententious messages online while the meter's running when I can compose them beforehand and transmit them in a fraction of the time?" So you prepare a message using a word processing application, and then either you save it as a text file or you keep it handy for pasting into your terminal window.

Of course, your terminal program can send the message much faster than you can type, and now and then you'll run into a service that won't be able to handle the speed. If the speed at which you send the text exceeds the rate at which the service can receive it, the service may lose or garble some (or many) of your characters.

Ergo the text pacing settings offered by many terminal programs. You tune those to the service you're talking to, and while we go over them we'll look at MicroPhone's Text Transfer Settings window (see Figure 5.3). I'm using this as an example mainly because I like the way it groups the settings.
Chapter Five Configuring Your Terminal to Their Taste

**Figure 5.3.** Text pacing and related settings in MicroPhone II, version 4.0.

### XON/XOFF

With most services, XON/XOFF (or X-On/X-Off as MicroPhone would have it) will be the only form of text pacing you'll need. When either you or the service can't keep up with incoming text, an XOFF character is sent, and text transmission ceases until the next XON. It all happens transparently, and no intervention on your part is required. This is pretty much the fastest form of text pacing.

### Wait for Echo

With this much-slower form, your terminal waits until the service you're talking to echoes back to you each and every character you send before proceeding to send the next. It's extremely conservative, but it still won't work if the other service doesn't echo your text. Most BBSs do echo your text. CompuServe also does, but GEnie doesn't. As for one-on-one, friend-to-friend connections, by and large they don't echo your typing either. So despite its conservatism pacing-wise, the "Wait for Echo" option should be used carefully and rarely. Save it for services that absolutely require it.

### Delay Between Characters and Lines

Let's say you're online with some service that does echo back to you. (When that's the case you'll have a full-duplex connection; see later for more on
echoes and duplexing). You paste a five-line message into your terminal program, but what you see on your own screen is the original message minus some of its characters. The service has apparently lost some of your text and XON/XOFF alone doesn’t seem to be doing the job. Well, this is the time to think about introducing a delay between each character you send, possibly between each line you send. Which do you pick? If the lost characters were all at the beginning of your original lines, go with a between-lines delay. Start with maybe a quarter-of-a-second and see if that does the job. With a few systems you may even need a whole second, but there’s no point in wasting time if you don’t have to. If, on the other hand, the lost characters are distributed fairly randomly among your text, a between-characters delay may be the ticket. Five-sixtieths (5/60) of a second is a reasonable starting point, but chances are you’ll have to experiment.

BTW, if you still haven’t come across the tick, you may be interested to know that it’s an important unit of time maintained within the Mac system. It’s one-sixtieth (1/60) of a second, and it’s used frequently by Macintosh applications, especially terminal programs and HyperCard.

Waiting for a Line Prompt
This feature is more of a convenience than a necessity. Some services, particularly commercial online services like GENie and CompuServe, prompt you for each line of text when you’re sending E-mail to another member. Let’s say you’ve logged onto CompuServe. After going through its E-mail rigamarole, it prompts you for the first line of your message by typing:

1>

When you’ve typed your first line followed by a carriage return, you’re supposed to wait until CompuServe prompts you for the second line by typing:

2>

That’s fine if you’re composing each line as you go along. But if you want to paste several lines of pre-written text into your terminal program, they’ll probably go in so fast that CompuServe will lose some of your text while it sets up its line prompts. So, to keep you and CompuServe in synch, you could tell your terminal program to wait for the > character before sending each line. It’s a setting you can probably live without, but if you do use the E-mail
Chapter Five Configuring Your Terminal to Their Taste 67

features of online services, it's also a feature whose availability you'll want to keep in mind.

Text Pacing Expertise

Like many other aspects of Macintosh telecommunications, text pacing is something at which no one will ask you to become an expert. With the vast majority of services, XON/XOFF will take care of all your text pacing quite adequately. Does that mean you can even do without XON/XOFF? Often, yes. Many services can handle incoming text a lot faster than you can dish it up. Some, however, can also send it so fast that if you're receiving on an older, slower Mac—particularly one whose telecom program is running in the background under MultiFinder or System 6.x—it may be you who is creating pacing problems and not the online service. To ensure that all goes well in that event, at the very least use XON/XOFF while receiving.

Keyboard Settings

Although it's true that you set your keyboard settings through your terminal program, I'm grouping them in a separate category because that's what feels right to me. The same applies to the serial port settings starting on page 72. Everyone has a slightly different notion of where your terminal ends and your keyboard and/or modem and/or serial port begins. For now I ask you to live with my notion.

What your Return Key Sends

Most services will be content to have your Return key send only a carriage return. As we noted in the previous chapter, a carriage return only brings the "carriage" of a receiving terminal to the left side of the page without advancing to the next line. The line advance is triggered by a line feed character. Still, most services tack their own line feed onto received carriage returns and therefore don't require that the line feed come from you. Occasionally, however, you may encounter a service that requires a line feed. When that's so and you're not supplying line feeds, every line you send gets typed right on top of the preceding line, obliterating it altogether. So if you know that a service needs line feeds along with carriage returns, you'll want
to make sure that your keyboard sends them when you hit the Return key. Figure 5.4 shows where you’ll find the setting in White Knight.

**BTW,** if someone complains that everything you’re sending is coming in double-spaced, it’s almost certainly because you’re adding line feeds when you don’t have to.

![Figure 5.4. White Knight’s key-mapping options, reprised from the last chapter. We could use a similar window from another terminal program, but White Knight’s makes the best illustration. The relevant buttons for choosing what your Return key sends are just to the right of—sure enough—“Return key sends.”](image)

### Backspace, Delete, Rubout, Etc.

We said before that rich languages offer an abundance of synonyms. We also said in an earlier chapter that some Macintosh keyboards have Backspace keys while others have Delete keys. (If there’s one that has both I haven’t seen it.) As a rule, the older, pre-ADB keyboards have Backspace keys and the newer ADB keyboards have Delete keys. (For a while Apple championed the Backspace key and IBM the Delete key; IBM won that round.) The key is located just over the Return key, and regardless of which word is molded into it, when you’re not telecommunicating the key sends an ASCII backspace character. Decimal value 8. Simple enough.
BTW, ADB stands for Apple Desktop Bus and, starting with the SE and
continuing through later Macs, it's the ADB bus circuit into which you
plug such devices as keyboards, mice, joysticks, trackballs, and such
curiosities as ADB modems. Unlike normal modems that plug into one
of your serial ports (i.e., the modem and printer ports) ADB modems
plug into the ADB bus. If for one reason or another both of your
computer's serial ports are already used by other devices, ADB
modems may be something you want to investigate. Personally,
though, I advocate plain old serial modems. The power over your
serial port that you get with a good Mac telecommunications program
is something not easily relinquished.

BTW, even if both your serial ports are already taken, many vendors
will sell you a serial mini-DIN-8 AB switch—one good one is a mail
order house in Troy, Michigan called Lyben Computer Systems, tel.
(313) 649-4500. (No, this time the AB doesn't stand for much of
anything at all.) You also need what's called a straight-through mini­
DIN-8 cable. Straight-through cables resemble modem or printer
serial cables, but they're wired in the simplest possible way: pin 1 at
one end to pin 1 at the other, pin 2 at one end to pin 2 at the other, etc.
Let's say you're on an AppleTalk network, so your printer port is
unavailable for anything else. You sometimes need carbon copies, so
you're running a 9-pin dot matrix printer out of your modem port,
which leaves you without a free port for a modem. To connect a
modem using an AB switch, you'd run the straight-through cable from
your modem port to the input/output port on the AB switchbox. Then
you'd connect your dot matrix printer to the A port of the switchbox
and your modem to the B port, each with the same cables you'd use
normally. When you set the AB switch to A, your printer gets connect­
ed to your modem port. When you set it to B, your modem gets
connected. Obviously you won't be able to use the modem and dot
matrix printer at the same time, but that may be all for the best: it will
protect you from gadget overload.

A number of services, particularly those run on Digital Equipment (DEC)
terminals, don't like the ASCII backspace. What they want instead is the
rubout character, ASCII 127. So if you're online with some system that echoes
your typing, and you find that your Delete/Backspace key doesn't work properly, try setting your program to send rubouts. Of course, they won't always be called rubouts. They may be called delete, DEL, or even destructive backspace. Figure 5.4 shows how to choose that option in White Knight.

**BTW,** some ADB keyboards have keys for both varieties of backspace, and as often as not, just to heighten the confusion, both keys are labeled "delete." The ordinary Backspace/Delete key is above the Return key, while the destructive delete tends to be down among the cursor control (arrow) keys. Sometimes the destructive delete is marked with an X. Try it in a word processor and see what happens. In Microsoft Word, the normal Delete key deletes characters backward and the delete-with-an-X/rubout/destructive backspace key deletes them forward.

**Keyboard Buffers**

When you're typing into your computer, there's a major difference between a telecom terminal and a word processor. With a word processor, you can go back to the previous line, even the previous paragraph, and do some further editing. On the other hand, with most ordinary terminal emulations, once a line is sent it's out and gone. If you see an embarrassingly misspelled word and you've already sent the line, now your service will know just how illiterate you sometimes are. Even if you haven't sent the line and you're connected to a friend, he or she will see all your backspacings and respellings while you're doing them. Some people don't mind that sort of intimacy, but others prefer not to share each and every edit with the rest of the world.

Another scenario: You've joined an online service that holds real-time conferences. (On GENie they're held virtually every day.) A guru from Epiphany Software is holding court over some hot new Mac utility that promises to bring new meaning to your life, and you join the conference. So, however, have a couple of dozen other people, and half the time they're all sending messages at once. You try sending a message yourself, but part way through it someone else starts typing, and soon not even the conference moderator can make any sense of your question, pithy though you know it to have been.
To deal with situations like these, most terminal programs offer keyboard buffering. When keyboard buffering is turned on, your terminal stops sending what you type on a character-by-character basis. Instead, your typing goes into a keyboard buffer whose contents aren't sent out to your modem until you hit the Return key. Until then you can edit each line in total privacy. When you finally do send the line, it all goes out at once, leaving little room for someone else to barge in and interrupt your statement.

*Figure 5.5.* ZTerm's Keyboard Buffer window sitting just in front of its normal terminal window. When I'm satisfied with the window's contents, I'll type a carriage return, and the text it contains will be sent and appear in the terminal window. In ZTerm, keyboard buffering appears as an option under the Edit menu.

BTW, you might consider keyboard buffering to be the kind of personal convenience that more properly belongs in Chapter 4. The thing is, if you use it at all you'll almost certainly use it selectively, depending on with whom you're online. In cases where you're logging onto a service that asks mainly that you reply to prompts with a *y* or *n* (i.e., yes or no), there's not much point in invoking keyboard buffering.
Serial Port Settings
(a.k.a. Communications Settings)

Thus far we've enlarged on some of the settings categories we established in the last chapter. Now we'll start a whole new category—not just new, but crucial. As far as being on the same wavelength as your service is concerned, this is the single most important group of settings you'll deal with.

![Serial Port Settings](image)

**Figure 5.6.** White Knight again—this time its Serial Port Settings window. There are similar windows in other terminal programs, but this one puts everything up front at once. The settings to the right of Baud rate, Parity, Databits, etc. are in pull-down menus. The proceeding illustrations show the menus in their pulled-down state.

Data Rate

The ad for your modem said that when it connects to another modem it automatically negotiates the highest possible data rate of which your system and the service are capable. Maybe so, but in order for those negotiations to take place at all, both your system and the service have to be able to support those negotiations. Your service's modem has to be more or less as sophisticated as yours, and both modems have to be set to enter into the negotiating process. For now we aren't at all sure what the service's story is, and neither
have we set your own modem to initiate negotiations. That being so, I suggest you start out with simple, conservative settings. We'll get to the fancy stuff in due course.

What does all this mean in practice? The least risky setting is 2400 bps. There are still a few—very few—services around that won't run faster than 1200 bps, but telecommunicating that slowly ceases to be much fun. Of course, if your own modem tops out at 1200, that's the speed you're stuck with. And, yes, a service whose top speed is 14,400 bps will be able to handle all the lower speeds. It may not like handling them, but it will. If you and your service can both handle 9600 bps, there's every reason to go for it. If you're logging onto a commercial online service, check to see which of their telephone access numbers handle what speeds. Both GEnie and CompuServe have some access numbers that take 9600 bps and others that won't go higher than 2400 bps. The better Macintosh BBSs generally accept 9600 bps because, among other things, modem manufacturers give special deals to sysops.

BTW, if you do have a high-speed modem and you're not ready to try out hardware handshaking, don't shy away from 9600 bps just because you plan to download files. GEnie and most BBSs support the ZModem file-transfer protocol, which works perfectly well with
XON/XOFF (i.e., software-based) flow control. ZModem provides for special handling of the XON and XOFF characters so they're not mistaken for binary data, thereby eliminating the need for hardware handshaking. Just stay away from the older XModem and YModem protocols, with which XON/XOFF handshaking can cause big problems.

Data Bits

Eight data bits is the current standard in personal computer telecommunications. That's the number of bits per character. A few services operated by mainframes—those being computers so large that they often require their own room—prefer seven data bits, but even most of those can make do with eight in a pinch. The online services, BBSs, and friends' computers you'll connect with will generally want eight data bits, so use that number unless you know the service wants something else.

Parity Bits

When you select eight data bits, you won't want your bytes to contain any parity bits, so you'll select No Parity, which a few terminal programs call None. Seven-data-bit systems will want a parity bit, but it's up to you to find out if the service likes its parity odd or even. By and large, though, eight data bits and no parity will do just fine.
Stop Bits

In asynchronous telecommunications, which is what you'll almost always be doing, the stop bit tells the modems at either end of the connection where one byte ends and the next one begins. (That's as opposed to synchronous telecommunications, in which each byte is transmitted during a fixed-time interval; when that's so, there's no need to separate the bytes with stop bits.) Most Macintosh telecommunicators always use one stop bit and never give the matter another thought. In the rare instance in which you know a service likes another setting, terminal programs give you other options.

In telecom-ese the four settings we've just described are frequently combined into— I'm not sure exactly what to call it—a phrase. A typical one would read 2400-N-8-1. Data rate-parity-data bits-stop bits. It's a shorthand way of
packaging communications settings, and if you become any kind of telecommun­nicator you'll see it often.

**Duplexing/Echoing**

Some services *echo* what you type and some don't. What do I mean by echo? Simple. You send them an A, they promptly send you an A. You send them a carriage return, they send you a carriage return. Everything you type gets echoed back to you. When a service does that, you'll want a *full-duplex* connection. In full-duplex mode, information travels bidirectionally and simultaneously; that is, it flows between you and the service in both directions at the same time.

Other services won't echo what you type. Such services favor a *half-duplex* connection, where information flows in only one direction at a given time. Of the major online services, CompuServe requires full duplex, GEnie requires half duplex, and America Online gives you their own proprietary software so you don't have to care. Most Macintosh BBSs run one of three popular host programs: Second Sight, Red Ryder Host (Second Sight's immediate ancestor), and Hermes. All three like you to log in with a full-duplex connection. Individual-to-individual connections normally take place at half duplex.

Again, when you're online with a half-duplex service, your typing won't be echoed back to you. Consequently, if you're set for full duplex, you won't see what you're typing into your terminal window. The service will see what you're typing but not you. You'll see what the service is typing to you, but not what you're typing to it. That's because your terminal expects your typing to be echoed by the service, but the service isn't doing it.

To provide for that no-echo state of affairs, your terminal program gives you the option of echoing your own typing. That's called *local echo*—local because the echo comes from your end of the phone connection and not the service's end. In Macintosh terminal programs, *local echo* and *half duplex* mean essentially the same thing. Some programs use one term, some the other. Either way, when you're set for half duplex or local echo, its *your* terminal that will echo your typing into your terminal window.

In everyday practice, it's easy enough to tell if your duplexing is correctly set. If you and your service are both running the same kind of duplexing, your typing appears normal enough and so does your service's.

If your service is running at half duplex but you're set for full duplex, you'll
see its typing but not your own typing. That's because nobody's echoing. Half-duplex services don't echo, and your own terminal, expecting an echo from the service, isn't echoing either. When that happens, make sure to set for half duplex/local echo when you connect to that particular service.

If your service wants a full-duplex connection and you're set for half duplex/local echo, everything you type will show up twice in your terminal window. If you type \textit{hello} slowly, it will show up as \textit{hheelllllo}. If you paste \textit{hello} into your terminal so that it goes out very fast, it will probably show up as \textit{hello hello}. (When you paste text there may not be enough free transmission time between the characters for the echoing to take place one byte at a time.) So if you find yourself seeing double, postpone making an appointment with your ophthalmologist until you make sure your modem is set for half duplex when you connect to that service.

Baud rate: \texttt{2400}
Parity: \texttt{NONE}
Data bits: \texttt{8}
Stop bits: \texttt{1}
Duplex: \texttt{Full Duplex, Half Duplex, Echo Duplex, Null Duplex}

\textbf{Figure 5.10.} The duplex pull-down menu from Figure 5.6. White Knight is unusual in offering echo duplex and null duplex. Very few other terminal programs do. Even VersaTermPRO, which is something of a telecom techie's toyland, contains just a simple local echo menu entry that you check or uncheck. If you have White Knight and you want to host a full-duplex session with a friend, set your terminal to echo duplex. Then your friend can call you with his own terminal set for full duplex and everyone's typing ought to come out looking right.

In all communications, attunement is of primary importance. Telecommunications is no exception. When you've attended to this last array of settings, the chances are good that you and your service will be properly attuned. And even if you aren't, the results may be intriguing. A screen full of garbage characters in which there appears to be some dim kind of pattern can be every bit as interesting as reading the want ad for a used 40-meg hard drive that someone's posted on your local BBS.
Chapter 6

Commanding Your Modem: The Hayes Language

Your modem has two modes of operation: the *online mode* and the *command mode*. As long as it's powered up and turned on, it's always in one mode or the other, never both at once.

When you're online with another service, your modem is, naturally enough, in the online mode. In that mode your typing goes into your computer, your terminal program passes the characters along to the computer's serial port, the serial port sends them over to your modem, and your modem obligingly sends them out through your phone line to the service.

When you first power up your modem, however, it defaults to the command mode. It's not online with any service, no one's told it to dial up any numbers, and it simply sits there patiently awaiting your next order. You've already wired everything up, you've configured your terminal (presumably using some of the suggestions in the last two chapters), and now the ball is in your court. It's incumbent on you to say *something* to your modem, but exactly how do you begin?

Back in the days of the British raj, nearly every prosperous English household had a steward who received orders in English and transmitted
them to the domestic staff in Hindi or Bangali or whichever of India's many native languages the staff happened to speak. A pair of terminal programs springs to mind, because they're designed to function as just that kind of steward. One, as we've already said, is MicroPhone II, packaged as it is with enough settings files, each one copiously decked out with all manner of automated, button-actuated scripts, so that you may very well be able to get away with years of telecommunicating and never once confront your modem face-to-face. Then there's Smartcom II, which goes MicroPhone still one better. When you double-click on Smartcom II's icon so that it opens to its main window, no sooner do you begin typing to your modem than the thing protests in no uncertain terms!

“No, no, no!” it seems to tell you. “The sahib must not be burdened with the task of addressing his unworthy modem. It is for me to perform so pedestrian a chore. Of course, if the sahib absolutely insists—forward-looking equalitar-
ian that we all know him to be—then he may pull down the Connection menu and select Direct Connect."

If you like the idea of playing Col. Blimp to your computer, if you're reluctant to learn any more technology than you absolutely have to, or if speaking directly to your modem is otherwise objectionable to you, I recommend that you buy a copy of Smartcom II or MicroPhone II and proceed to the next chapter. Yes, you'll be unmasked as the sort of tourist who travels through a country without bothering to learn even a few words of the native language. Hard-core telecommunicators will patronize you every chance they get. But you understand your needs better than anyone else does, and countless people have lived full and happy lives without ever mastering a single phrase of the Hayes language.

**Talking Hayes**

It's my own feeling that everyone who uses a modem should know at least a few basic Hayes commands. Language labs get you started talking right off the bat, and I see no reason to do things differently. If you're a beginner, go through the following checklist. If all of its conditions aren't satisfied, you may not be able to get that first crucial conversation going.

1. Make sure your computer and modem are both powered up.
2. Make sure your modem is connected to an actual phone line. There should be a phone cord running from the modem to a telephone outlet in the wall or an equivalent connection.
3. Make sure there's a proper modem cable running between your modem and one of your Mac serial ports (either the modem or printer port).
4. Open your terminal program and make sure it knows to which serial port your modem is connected.
5. If you are using Smartcom II, anticipate what it will tell you when you try typing to your modem and choose Direct Connect from its Connection menu.
6. Regardless of what terminal program you're using, attend to all its basic settings. The last two chapters will take you through that. In most cases you won't have to fiddle much with settings, since the defaults of most terminal programs are intelligently chosen.
Now try your hand by typing:

```
AT
```

followed by a carriage return. Your modem should respond by typing “OK” back to you, and the resulting interchange should look something like what you see in the terminal window in Figure 6.2. If you’re set for half duplex/local echo, your typing will be “doubled” because each character will be echoed twice, once from your modem and once from your locally echoing terminal. A given character will be sent to the modem only once, however, so there’s no danger of its becoming confused.

![Figure 6.2](image)

*Figure 6.2. This is Smartcom II again, just to reassure you that it will indeed let you talk to your modem if you choose to. (Despite my teasing, it’s an eminently powerful program.) The first two ATs were typed with the local echo option turned off, the next two with local echo turned on. As you can see from the second pair of ATs, local echoing causes you to see double when you type to your modem, even though you’ve typed each character only once. Not to worry.*

What does your modem do when you send an “AT” to it? Actually, not much of anything. AT is sometimes called the *attention* command. When your modem is in the command mode and receives an AT, it replies with “OK” if it
understands you, thereby telling you that, yes, it's indeed in the command mode and, yes, it understands rudimentary Hayes language and, yes, it's getting the message. But an AT by itself neither changes its settings nor initiates any new actions.

One thing you might want to try early on is typing a lowercase AT—i.e., an "at"—to your modem and see if that's all right with it. In Figure 6.2 it's fine, and the modem responds with its usual "OK." A number of older modems accept only uppercase commands, however, and that's something about your modem you'll want to know right now. If your modem is comfortable with "AT" but responds to "at" either with "ERROR" or nothing at all, then you'll need to make sure to type all your modem commands in uppercase.

**BTW**, if you type an AT to your modem and it doesn't respond even though you've gone through the six-point checklist on page 81, you may have one of those very rare modems whose factory default is set so that it echoes its responses back to you only when you expressly tell it to. There's a modem command to handle that situation. You type:

```
ATE1
```

followed by a carriage return. ATE1 tells your modem to echo its responses. Almost all modems default to the "characters echoed" setting in any event, but you never do know for sure...

Now let's try dialing a number. In fact, let's begin with your own number. Yes, you'll get a busy signal, but let's go through it anyway. For purposes of illustration we'll assume the number of the line attached to your modem is 555-1234, but when you do this yourself substitute your real number. The Hayes command for dialing a number is either ATDT or ATDP. The nearly omnipresent AT prefix identifies the line you're typing as a Hayes command. The D tells your modem to dial the phone, and the T or P that follows tell it whether to dial the number with tones or pulses. If your phone line doesn't accommodate touch-tone phones, use P. Otherwise, use T, since it's the faster of the two.

So, starting on a new line, type ATD followed either by T or P followed by your phone number and then a carriage return. In your terminal window it should all look something like this:

```
ATDT555-1234
BUSY
```
Your modem has dialed the number 555-1234 with tones, discovered that the line is busy (as it always is when you dial yourself) and therefore typed back BUSY.

Now some observations on what just happened:

- If you leave the tone/pulse character out of the dialing string and you've only just turned on your modem—i.e., if you'd typed the string ATD555-1234 instead of ATDT555-1234—your modem still dials the number, but it uses its default tone/pulse setting, which with the vast majority of modems is pulse. Once you dial a number with tones, however, any further dialing strings that lack the tone/pulse character will be dialed with tones. That won't change until you reset your modem to its default settings.

- There are two main ways to reset your modem to its default settings. One way is to turn it off, then on again. Another is to type:

  \textbf{ATZ}

followed by a carriage return. When you do type an ATZ, your modem should respond with "OK."

- Your modem doesn't care about the embedded dash between the 555 and the 1234. The command ATDT5551234 produces the same result as ATDT555-1234. Most—\textit{but not all}—other non-numeric characters embedded in the phone number portion of a dialing string are also ignored, so you can safely surround area codes with parentheses or divide up long numbers with spaces.

- Since most non-numeric characters embedded in the phone number portion of a dialing string are indeed ignored, you can't directly dial phone numbers expressed in the old alphabetic prefix form. John O'Hara once wrote a bestseller called \textit{Butterfield 8}, but if you issue an "ATDT (212)BU8-1234" command to your modem (212 is the area code for Manhattan, where the BUtterfield 8 exchange is now just 288) you won't get the number you're looking for. So you
either have to translate alphabetic prefixes yourself or let your terminal program do it for you. Many terminal programs let you put letters into phone numbers, but then you have to let them dial the number for you.

A few non-numeric characters after the ATDT (or ATDP or just ATD) are not ignored. These characters are sometimes called **dial modifiers**, and they serve to give you additional control over the dialing process. In all but the rarest of cases you'll only need to know about three of them: the comma, the semicolon, and $W$.

**Comma** is the pause character, and when it's embedded in a phone number following an ATD your modem pauses before dialing the next digit. With most modems, the default length of the pause is two seconds. If you care to, you can change the length of the pause induced by each comma, but that involves resetting one of your modem's S-registers, which we'll save for a little later. An easier way to get long pauses is to type two or more commas in a row. Look at the next example to see how you might use it.

$W$ is the "wait for dial tone" character, and it can come in handy if you're dialing numbers with a credit card, which you might need to do if you're out on the road with a laptop and a modem. When you're using a credit card, several long-distance services require that you dial an 800 number, then wait for a computer tone, then dial the number you want to reach, then wait for another tone, then enter your credit card number. Sometimes—only sometimes—your modem will recognize one of these tones as a dial tone. Let's say I have a Sprint card and that Sprint's access number is 1-800-333-4444. Let's also say that I'm out on the road and I want to call my own home at 516-555-1234, and that my credit card number is 999 888 7777 6666. Sprint says I have to dial 1-800-333-4444, wait for a computer tone, then dial 0-516-555-1234, then wait for another computer tone, and finally dial 999 888 7777 6666. That's more digits than I care to dial
manually anyway. Even if my telecom program doesn't support automatic dialing, once I get the sequence right I can save it in my Note Pad desk accessory and paste it into my terminal window whenever I need it. I try the sequence manually once, and I discover that the wait for the first tone is very long, maybe eight or ten seconds. I could get that with four or five commas in a row, but instead I experiment with \textit{W}. I therefore create the following string:

\begin{verbatim}
ATDT1-800-333-4444 W 0-516-555-1234 W
999 888 7777 6666
\end{verbatim}

What I discover when I try it is that my modem recognizes the first tone as a dial tone, but not the second. That is, after dialing 1-800-333-4444 it patiently drums its little fingers and does nothing until it hears a tone, after which it goes on to send the 0-516-555-1234. However, it doesn't recognize the second tone as a dial tone and keeps waiting for one instead of going on to send my credit card number. So now I replace the second \textit{W} with a couple of commas, just to allow some time for the second tone to kick in. The resulting string looks like this:

\begin{verbatim}
ATDT1-800-333-4444 W 0-516-555-1234 , ,
999 888 7777 6666
\end{verbatim}

And when I try it, it actually works! I quickly store it in my Note Pad, and now, whenever I have to call home from Walla Walla, I no longer need to read the tiny print on the reverse side of my credit card or dial all those digits by hand and probably get them wrong anyway.

\textbf{Semi-colon (;)} is the "return to command mode after dialing" character, and it normally comes into play when you're dialing voice calls. Any time you place a call with your modem, it defaults to the assumption that you're calling another modem. It waits for a dial tone, dials the digits you've told it to dial, and then waits some more until it detects a data carrier at the other end of the line. Try dialing
some harmless local number with your modem—the weather number, say. In my part of the country its 976-1212. When I do that, what I get in my terminal window is either:

\text{ATDT976-1212}
\text{BUSY}

or

\text{ATDT976-1212}
\text{NO CARRIER}

What determines whether I get a BUSY or a NO CARRIER when I dial the weather number seems to be the timbre of the voice at the other end. When it's a man with a deep voice, my modem, imperfect creature that it is, thinks it's hearing a busy signal. When the weather person is a woman, my modem goes into its online mode, waits for a data carrier, fails to detect one after some number of seconds, and then hangs up. Either way, I never get to hear the end of the weather report on my modem speaker. What I want to do in this case is tell my modem not to go into the online mode after placing the call. Forget about carriers and their detection. All I'm looking for is the whole weather report. So, next time around, I append a semicolon to the end of my dialing string so that it now looks like this:

\text{ATDT976-1212;}

and, sure enough, my modem speaker gives me the weather report in full.

Most people's systems are set up so that an ordinary handset shares a line with their modem. If you place a voice call with your modem, you may want to use the handset and not your modem speaker. You might, for instance, actually want to talk to somebody when the call goes through. If you pick up your phone while your modem is off hook (i.e., while its own imaginary handset is also picked up), you won't be able to hear the other party very clearly. Yes, you'll hear them and they'll hear you, but the connection will seem to be very
poor. To get your modem to go on hook (i.e., off the line, to hang up), you give it an ATH0 command. (That’s ATH-zero, and with nearly every modem ATH all by itself will suffice.) Remember that ATH is a modem command, and therefore has to be followed by a carriage return. Remember too that it will only work if your modem is in its command mode, so make sure to put a semicolon (;—the “return to command mode after dialing” character) after the last digit of the number you’re dialing. Also, don’t issue an ATH until you’ve picked up your handset. If you do, your call will disconnect. Just as with any line equipped with more than one extension, once you hang up all of them, the active call is disconnected. You need at least one live extension to keep a phone call going.

When you dialed your own number and got a busy signal, the speaker in your modem probably let you hear the whole transaction as though you were holding a handset to your ear. (That’s if your modem has a speaker; the great majority do.) Depending mainly on their ages, most speaker-equipped modems give you a fair amount of control over what your speaker does and when and how it does it. On the next page is a table of some common Hayes commands relating to your modem speaker.
Table 6.1. Hayes Commands for Modem Speakers

<table>
<thead>
<tr>
<th>Command</th>
<th>What It Does</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATM0</td>
<td>Speaker always off.</td>
</tr>
<tr>
<td>ATM1</td>
<td>Keeps your speaker on until a data carrier is detected, then turns it off. (This is the setting that most modems default to.)</td>
</tr>
<tr>
<td>ATM2</td>
<td>Speaker always on (but it's masochistic to listen to the whoosh of a carrier during a long data transmission).</td>
</tr>
<tr>
<td>ATM3</td>
<td>Speaker off both during dialing and for as long as a carrier continues to be detected.</td>
</tr>
<tr>
<td>ATL</td>
<td>Sets your speaker volume to low.</td>
</tr>
<tr>
<td>ATL0</td>
<td>Also sets your speaker volume to low.</td>
</tr>
<tr>
<td>ATL1</td>
<td>Also sets your speaker volume to low. (No one ever said the Hayes language was without redundancies.)</td>
</tr>
<tr>
<td>ATL2</td>
<td>Sets your speaker volume to medium. (Most modems default to this setting.)</td>
</tr>
<tr>
<td>ATL3</td>
<td>Sets your speaker volume to high. (Tiny modem speakers often produce interesting squawks at this setting.)</td>
</tr>
</tbody>
</table>

Check with your modem's manual before trying all the speaker commands, as some modems support only a subset of them. For example, an old modem of mine—one that's been living in a foster home for some years—does have a speaker, but it supports only the ATM commands and not the ATL commands.

BTW, one hears differing accounts of how the Hayes language came into being, but the one I like best has young Zebediah Hayes, who'd go on to be founding patriarch of the Hayes telecommunications empire, stopping one day at a roadside diner and overhearing a puzzling conversation. First a patron asked the counterman, "FUNEX?" "SVFX," the counterman replied. "FUNEM?" the patron continued. "SVFM2." The patron smiled. "OKLFMNX."
Hayes found the exchange oddly haunting. Diners were well known for their kitchen codes—"down" for toast, "grade A" for milk and all that—but this language all of letters was entirely new to him, and, even if he didn't know what it meant, its efficiency impressed him so deeply that he felt compelled to invent one of his own. The rest was history. The Hayes language became all but universal, and soon nearly every modem made—by Hayes or anyone else—could understand it.

Then, years later, Hayes was standing on line in his own company cafeteria in Atlanta. A number of immigrants had settled in the city, and two Ruritanian cleaning women were just ahead of him. Hayes was preoccupied with balance sheets and new debentures, but he sprang to attention when one of the women asked the cafeteria attendant, "FUNEX?"

The attendant seemed bewildered, so the second woman offered help. "She is saying, 'Have you any eggs?'"

"Yes," the attendant replied. "We have eggs."

The second woman translated for the first. "He says, 'SVFX.'"

Nodding, the first woman then asked, "FUNEM?"

Again the second woman translated. "She says, 'Have you any ham?'"

"Yes," said the attendant, "we have ham too."

The second woman turned to the first and explained, "He is telling you, 'SVFM2.'"

The first woman appeared pleased. "OKLFMNX."

"Now I get it!" the attendant exclaimed. "You just said, 'Okay, I'll have ham and eggs.'"

The cleaning woman was happy, but Zebediah Hayes was even happier. The mystery that had plagued him all his life was solved at last.

**Result Codes**

You've no doubt noticed by now that if you talk to your modem in language it can understand, it talks back to you. When you give it a command it can execute, for instance, it normally responds with an "OK." When it encounters a busy signal it tells you "BUSY."
These responses are called *result codes*. You address your modem with commands; it replies by sending you result codes. Interestingly, *your* vocabulary is expected to be substantially larger than *its* vocabulary. I'm not inclined to count up the total number of Hayes commands, but take my word that there are lots. On the other hand, there are only 10 basic result codes. (A few additional codes called *extended result codes* have been appended to the original Hayes language to accommodate MNP and V.42 connections and the like, but we'll get to those later.)

Here's the ensemble of basic result codes in tabular form:

<table>
<thead>
<tr>
<th>Result Code</th>
<th>Short Form</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>OK</td>
<td>0</td>
<td>Command acknowledgment</td>
</tr>
<tr>
<td>CONNECT</td>
<td>1</td>
<td>Connection made at 300 bps</td>
</tr>
<tr>
<td>RING</td>
<td>2</td>
<td>Ring detected</td>
</tr>
<tr>
<td>NO CARRIER</td>
<td>3</td>
<td>Unable to detect a data carrier signal</td>
</tr>
<tr>
<td>ERROR</td>
<td>4</td>
<td>Error in command format</td>
</tr>
<tr>
<td>CONNECT 1200</td>
<td>5</td>
<td>Connection made at 1200 bps</td>
</tr>
<tr>
<td>NO DIALTONE</td>
<td>6</td>
<td>Unable to obtain a dial tone</td>
</tr>
<tr>
<td>BUSY</td>
<td>7</td>
<td>Busy signal detected</td>
</tr>
<tr>
<td>NO ANSWER</td>
<td>8</td>
<td>No answer</td>
</tr>
<tr>
<td>CONNECT 2400</td>
<td>10</td>
<td>Connection made at 2400 bps</td>
</tr>
</tbody>
</table>

The result codes are informative enough, but when you learn to write your own scripts for your telecom program they'll be indispensable. For example, one of your scripts might test for which result code you get after dialing the number of a service. If it's CONNECT 2400, the script might go on to look for one of the service's prompts. If it's BUSY, the script might wait a while and then try the number again.

As for the short forms of the result codes, I don't recommend you ever use them, but there's no harm in knowing they exist. The command ATV (or ATVO, and that's a *zero* after the *V*) tells your modem to respond with *short form* result codes. ATV1 tells it to respond with *long form* codes (sometimes called
verbose codes in the questionable literature of telecommunications). Virtually all modems default to using long form codes.

**BTW,** if for some perverse reason you don't want your modem to respond with any result codes at all, send it an ATQ1 command. (I think the Q stands for quiet.) If you do that and then want to undo it, issue an ATQ0 (that's a numeric zero) or just an ATQ. Your modem's default setting will almost certainly be to send result codes.

### Stringing Commands Together

Most modern modems let you string several Hayes commands together on the same line. That is, instead of typing AT plus the rest of the command plus carriage return, you type AT just once at the beginning of the line, then a string of commands without the AT. For instance, using commands we've already gone over, you may be able to type:

```
ATM2L3DT976-1212;
```

and then a carriage return. The M2 following the AT instructs your modem to keep your speaker on all the time, the L3 sets your speaker volume to its highest level, the DT976-1212 dials the weather number in many parts of the country with tones as opposed to pulses, and the semicolon tells your modem to return to the command mode when it's finished dialing and not look for a data carrier.

A good many older modems don't support command strings, so you may have to consult your modem manual if you have one or simply try out some command strings. If they work, fine. If not, the following sequence will yield the same result:

```
ATM2
OK (from your modem)
ATL3
OK (from your modem)
ATDT976-1212;
```

### A Note on Hayes Compatibility

As "universal" as the Hayes language has gone on to become, it remains the case that, as with any language, you have to take some account of whom
you're speaking to. Hayes Microcomputer is just one of many companies that manufacture modems, and different makers implement the language in different ways. That's especially true where hardware-based compression and error checking are concerned. Not so very many years ago, error checking and compression were left entirely to your software. A modem simply wasn't called upon to do as much as today's modems. Flow control and other forms of handshaking were handled by your terminal program, a modem was pretty much a modem, and true Hayes compatibility was relatively easy to support.

Eventually, though, modems started to diverge, especially as competing manufacturers began to introduce proprietary protocols. Telebit came out with PEP, U.S. Robotics with HST, and Hayes with its own V-series protocol. Each of those required special modem commands, and soon the Hayes language had developed any number of modem-specific dialects. Some are very much worth knowing. Many U.S. Robotics modems, for instance, will display in your terminal window a summary of all the commands they understand and even help you with their use. (USR owners should explore the AT$, AT&$, ATD$, and ATS$ commands.)

In this chapter we've stayed close to the original inner core of the Hayes language, and thus far Hayes compatibility ought not to be a problem for you. But if you find yourself talking more and more to your modem, you'll certainly want to read its manual. The gizmo may turn out to have all kinds of capabilities that you (and I) don't know about.

**Initialization Strings**

Even if you're a heavy user of your terminal program's automated features, and thereby keep a safe distance between yourself and the Hayes language, one area that may require some passing knowledge of at least a few key phrases is that of **initialization strings**. Initialization strings are command strings that you send to your modem at the beginning of a session, and what they do is configure your modem for whatever it is you're about to do.

Let's take ZTerm as an example. It has sufficient automation to take care of all your dialing, but it likes to be sure that your modem is configured in the way it wants. To do that, it sends your modem an initialization string as soon as you fire up the program. After all, earlier that day you may have used some other communications application that left your modem configured in a way
ZTerm doesn't want. For instance, ZTerm expects verbal (a.k.a. "verbose," a.k.a. "long form") result codes from your modem, and some other program might have set your modem so that it sends numeric, short form result codes. So when you first use ZTerm, the default initialization string it issues to your modem is "ATE1V1\M". The E1 makes sure your modem echoes commands back to your computer, and the V1 instructs your modem to use long form result codes. \M stands for Control-M, which, as we've explained before, is another way of saying carriage return. (That's a caret before the M, and you get it by holding down the Shift key and typing 6.)

Sometimes you may also want to send your modem an initialization string that's unique to a given service you like to call. One of the BBSs I frequently log on to may use a primitive, bare-bones modem, while some other one may offer me all the compression and error-checking benefits of MNP-5. When I call the first service, the only initialization string I really need is an "ATZ\M" to set my modem to all of its default settings. With the second service I might initialize with "ATX3S11=55&D0\C1\J0\N3\Q3\G1\V1%C1" before I dial. For now, we'll skip explaining what all those commands mean. Suffice it to say that their primary purpose is to make sure that my modem and my service's modem negotiate a connection that gives me as many of the benefits of MNP

![Figure 6.3](image)

Figure 6.3. In ZTerm, it's in this window that you can enter an initialization string for a particular service. The string in the "Pre-dial init" box is the same MNP string I quoted just above. The box just isn't long enough to show all of it. You can call up this window by choosing "Connection" under ZTerm's Settings menu.
as we can both accommodate. ZTerm, as well as a number of other terminal programs, lets you enter a different initialization string for every number in your program's "phone book."

**A Few Additional Commands**

While we're on the subject, let me leave you with a few more common Hayes commands. First, a couple of rare birds—rare because they don't take the usual AT prefix and aren't followed by a carriage return.

**A/** is the *repeat last command* command. When you type A/ to your modem, it repeats whatever it did last. If you dialed the weather by issuing an ATDT976-1212; and want to dial it again for whatever reason, A/ will do the trick.

**+++** is the only command I know of that's used when you're *not* in the command mode. It's called the *escape* command, and its purpose is to switch your modem over to command mode when it's currently in the online mode. So if you *are* in the online mode and you want to issue a command to your modem, first type three pluses in a row and wait about a second. You'll know you've arrived in the command mode when your modem sends you an "OK."

Be aware that +++ doesn't always work on the first try. If your modem is handling a lot of data while you're in the online mode, some of it may insinuate itself between your pluses so that it won't be apparent to your modem that three *sequential* pluses have been typed.

**ATO** is the *return to online mode* command. Obviously enough, after you've typed a +++ to get into the command mode and then issued whatever command you chose to issue, you'd use ATO to return to the online mode. That's the letter O, by the way, and not a zero.

**ATA** tells your modem to answer the phone. Because you're answering the phone with your modem, it will look for a data carrier. If it doesn't detect one after a few seconds, it will tell you "NO CARRIER" and hang up. You'd use this command if you expect someone to call you in order to make an individual-to-individual connection.

Most modems made nowadays also support auto-answering, and those that do usually have an LED on the front panel labeled "AA." When the LED is lit, your modem will answer the phone after a pre-set number of rings. The ring count is saved in what's called an *S-register*, of which your modem has
28. Typically, a modem's S-registers are labeled from S0 to S27, and the auto-answer ring count is stored in S0. (Yes, that's a zero.) When S0 contains a zero, auto answering is disabled. Whereupon the following commands:

   **ATS0=0** puts a zero in the S-register labeled S0, which turns off auto answering. This is the default setting of most modems.

   **ATS0=1** puts a 1 in S-register zero (S0). It tells your modem to answer the phone after one ring. Actually, the number following ATSO= can be anything from zero to 255. (It would be interesting to meet the person who uses this command to its fullest and sets his modem to answer after 255 rings.)

Another S-register you may want to try is S11, which controls tone-dialing speed. Most phone companies support a dialing speed much faster than the 95 milliseconds-per-tone that the average modem defaults to. Register S11 will accept values between 50 and 255 (milliseconds), and I usually issue an **ATS11=SS** before dialing a number with my modem.

In this chapter we've only skimmed the surface of the Hayes language, but even if you haven't done that before, you now ought to be able to read your modem manual with a bit less puzzlement and more comprehension. Despite the automation that may be built into your terminal program, I strongly suggest that you experiment with at least some of the Hayes commands, if for no other reason than to get a feel for what it is that's going on behind the scenes.
Chapter 7

Getting Online: A Sample Session

Preparing for Your Debut

Let it be known right now that this chapter is primarily for beginners. Seasoned modemers can scan it briefly or skip it altogether without fear of missing anything terribly vital. Beginners will probably do well to pay it some attention if for no other reason than that over the years a fair number of inexperienced telecommunicators have told me that going online makes them oddly tense. Whenever I'd try getting a bead on exactly why that was, the clues always seemed to point toward one (or both) of two root causes: parsimony and stagefright.

The thing is, nearly everyone was raised to be frugal with the telephone. Authors of terminal programs know that, and, in what they imagine to be a convenience, most put some sort of timer in their terminal windows so users can watch the seconds tick by during a given session. It's true that during the course of any common voice call you can watch the seconds tick by simply by glancing at your wristwatch or looking up at the old clock on the wall, but usually the sheer sonic content of even the homeliest voice call is enough to anesthetize your parsimony. Besides, when you or the person you've called
are talking, there's at least the sense that your phone time is being put to some kind of elementary use. With data calls it's different. The service at the other end types you a longish message—maybe about some interesting new files that were recently uploaded to its libraries—and you oblige by reading it, possibly even thinking about it. And while you're reading nothing's happening! Not only is no one saying a blessed word; no one's even typing a blessed word! Before you're halfway through the message your upbringing asserts itself and tells you, "Stop wasting time on the telephone!" Then your superego climbs on board by reminding you that every second on the phone represents a particle of debt you'll soon be required to make good, and just to keep up the pressure the service at the other end chimes in with "If you do not press a key in the next two minutes you will be logged off."

The stagefright part is understandable enough. Services keep prompting you incessantly; their constant insistence that you make decisions can lead to online performance anxiety. And there's always the chance—for beginners a very good chance—that the next prompt will contain some exotic piece of nomenclature that you simply won't know how to respond to. What's worse, many of the prompts will remind you in passing that you're running out of time. BBSs that use the Second Sight host program typically preface their major prompts with "30 minutes left," which soon becomes "20 minutes left" and "15 minutes left." So not only is your own terminal timing your call; the service at the other end is timing it too. And even if some online menu specifies a key you can type to get help, half the time the help text is even less decipherable than the prompt it's supposed to help you with. Meanwhile you've accomplished next to nothing, your time limit is almost up, and instead of hanging off the service in the proper fashion, you want nothing more than to hang the phone up and be done with it—except by now you're so flustered that you can't remember how to hang up your modem and you wind up reaching for the switch somewhere behind it and turning the darn thing off.

To make your online debut go more smoothly, then, let's do some elementary preparation and then go through a moot first session with a Second Sight BBS.
**Settings**

We'll be choosing a bulletin board system from the list of Macintosh-oriented BBSs in Appendix A. Those which use Second Sight (or its immediate predecessor, Red Ryder Host, which is close enough) will have the phrase "Host: 2nd Sight/RRH" in their descriptions. The description may include the BBS's top speed, but it also may not, so for now let's play it safe and stick with 2400 bps. We won't be downloading any files this first time around, so high speed isn't all that vital anyway. It's the rare speed-reader who can read English prose at 240 characters per second while it's scrolling up a terminal window on a Macintosh screen.

Second Sight likes a full-duplex connection, eight-bit bytes, no parity, and one stop bit, so our basic terminal settings will be 2400-8-N-1, full duplex. If your terminal program lets you choose an emulation, pick the simplest Teletype emulation, TTY. If it doesn't offer TTY—and several terminal programs don't—go with VT-52 or VT-100, as both those emulations will respond to Second Sight in more or less the same way as TTY. Set your flow control for XON/XOFF. Inasmuch as this will be a maiden voyage, we won't be presenting your system with the kind of challenges that require hardware handshaking. (If you're using White Knight, leaving the hardware handshake button unchecked will cause XON/XOFF flow control to take place automatically.)

Now make sure your terminal window can handle 24 rows and 80 columns of type. In some programs, you do that simply by resizing the terminal window with the Mac's standard resize button in the lower-right-hand corner of the window. In other programs, you specify window size directly. In still others, you can do it either way. A settings window from MicroPhone II is shown in Figure 7.1 on the next page.

Back when all Macs had the same nine-inch screen, users didn't have much latitude if they wanted to configure their terminal windows. Things are considerably different now. Monitors of all shapes and sizes abound, and MicroPhone II's way of grouping "Rows," "Columns," and "Font Size" in the same part of the settings window lets you play with all of them to get the terminal window you want. If your terminal program lets you pick your font, choose a monospaced font like Monaco or Courier (Monaco looks cleaner).

From an earlier chapter, you'll remember that "Strip 8th Bit" is the same thing as "No extended characters." You won't be needing extended characters
for this session, so the eighth bit will indeed be stripped. Since typing extended characters requires using your Option key, that key will now be free to serve as the Control key. So if you have a non-ADB keyboard (and therefore no Control key) this is the time to tell your program that you’d like to have the Option key be your control key. You’ll need a control key, because we’ll be sending a control character or two during the session.

![Terminal Settings](image)

**Figure 7.1.** By grouping settings together in a logical way, MicroPhone II’s Terminal Settings window reminds you of the little things that sometimes need attention.

MicroPhone’s “New Line” button is its way of asking if you want line feeds to be inserted automatically after carriage returns. Second Sight doesn’t need the line feeds, so it’s nix on “New Line” if you’re using MicroPhone II. If you’re using another terminal program, it will have its own way of asking the same question. The answer remains no. As for what the Backspace key sends, Second Sight expects a plain old backspace.

Many terminal programs let you decide what kind of cursor appears in your terminal window, and MicroPhone II is one of them. This is a question of aesthetic or ergonomic preference more than of telecommunications, so feel free to experiment.

If you’re wondering what Auto Wraparound means, it’s just a way of telling your terminal window how to handle text. If your window is 80 characters wide and a 90-character line of text comes in, your program can either put the
last 10 characters on a new line or just not display them at all. With Second Sight, you’re unlikely to need Auto Wraparound, so it doesn’t make much difference if it’s checked or not. (A number of other terminal programs call this option “Wrap text.” Whatever your program calls it, don’t expect it to do the fancy text wrapping that you get from a word processor. Terminal programs seldom bother about such niceties as breaking lines between words; if you’ve got an 80-column window and a 90-column line, they break the line after 80 columns even if it means splitting a word into two pieces.) When MicroPhone does break a line, it can also insert a carriage return between each part of the break. You won’t want that, so don’t check it. It’s not especially crucial in any event.

If you’re curious about Answerback, it’s purpose is to accommodate remote terminals that use a custom log-on sequence. As part of their log-on procedures, some terminals send a ^E (that’s Control-e) and then expect a text reply (i.e., an answerback). Second Sight has no such elevated expectations.

One last thing: A number of terminal programs will let you instruct them to ignore full-screen clears. You’ll remember from Chapter 3 that ^L is the clear screen character, and every now and then a Second Sight BBS may send one. For now I recommend that you not let your screen be cleared. (Incidentally, the same option will be presented to you in a different way by the BBS itself, but more on that in a moment.)

Virtually all terminal programs let you save your settings. We’ll call the group of settings we’ve just made “Tyro BBS Settings” and save them where they can be found for the next go-around.

**Capturing the Session**

For your online debut and even several times thereafter, it will probably be wise to record everything that’s gone on between you and the service. Things do happen pretty fast online, and you may want to review the day’s events at your leisure. In times past, the way most people did this was by turning on their printers and printing out their sessions on a real-time basis. However, that was when only the rare user owned a hard drive, and capturing a session of any length to a floppy disk almost guaranteed you’d run out of disk space in the middle of a session.
BTW, I'm aware of all the allusions to the good old days of telecommunications that keep coming up, but my motive in discussing them is more than sentimental. Many current terminal programs have retained many of the same features they had five years ago. You may not want to use them now, but one day some oddball situation may arise, and it's worthwhile knowing that those antiquated features are still accessible.

In those same bygone days, 300 bps modems were all the rage, and a printer wouldn't bog down a session all that unacceptably even if it were asked to keep up with what was happening in the terminal window. Nowadays, real-time printing of a session is generally an obsolete practice. What you'll want to do instead is capture the session to disk and then, when it's over, either print out the capture file or look at it on screen. Just about every terminal program lets you turn on a capture file. When you do that, you're prompted to create a new file, name it, and save it somewhere. The file is empty when you first create it, but as your session rolls along, every line of text that appears in your terminal window is also saved to the capture file. The file itself will be in plain-text format, readable by any word processing program or by TeachText, Apple's own no-frills text reader that comes bundled with all its recent versions of the System. Most hard disks are fast enough so that capturing a session has no appreciable effect on the speed of text transfer.

Some terminal programs offer filtering options that influence the capture process. At a minimum, they'll automatically delete the so-called garbage characters—line feeds and other control characters that might be sent by the service. For now I'd recommend that you not filter anything at all. It will be interesting to see just what control characters do come in.

Warning: From time to time, you may want to print something selectively during a session. ZTerm, for instance, has a Print Selection option under its File menu which works just the way you'd expect: you select the text you want to print and then go ahead and print it. White Knight gives you a pair of buttons in its general status bar:

Figure 7.2. White Knight's "capture-screen-to-printer" and "capture-screen-to-disk" buttons.
When you click on the capture-to-printer button, the current screen—not the whole capture buffer—is sent to your printer. Similarly, clicking on the capture-to-disk button sends the current screen to a plain-text file on disk. Most other programs offer printing options too, but use them carefully. Try them out before you go online! Quite a few older terminal programs—and remember that some of the versions of MicroPhone that come bundled with new modems are actually quite old—make the assumption that you'll be printing only to an ImageWriter. The ImageWriter is usually accessed through the ImageWriter printer driver that you choose with the Chooser DA, but programs can also send text to the printer in much the same way as they send text to a modem: through a straight serial connection unmediated by any printer driver at all. A number of older terminal programs (and old versions of current terminal programs) do just that, and if you have, say, a DeskWriter or LaserWriter hooked up to your printer port, those printers won't know how to handle the text that's been sent to them and may well crash your Mac in mid-session.

Dialing Up the Service

For now let's not bother with your terminal program's auto-dialing features. It's easier to see what's happening when the dialing process takes place out in the open. I've already made the above settings in my terminal program, and now, just to make sure my modem hasn't retained any unwanted settings from another session, I reset it to its defaults:

```
ATZ
OK
```

Now I'll dial up a Second Sight BBS in New Jersey called Dragon's Cave. I type:

```
ATDT908-469-3450
```

In my neck of the woods it's not necessary to put a "1" before the area code, but if it were necessary I'd have typed ATDT 1-908-469-3450. I could have left out some or all of the spaces and/or dashes, and I could just as well have typed ATDT (908) 469-3450. The spaces, dashes, and parentheses are ignored by the modem. If my line didn't support tone dialing, I'd have typed ATDP908-
to tell my modem to dial the number with pulses—the same way a rotary dial phone would do it.

Disabling Call Waiting

If my line supported call waiting, I'd have disabled it for the duration of the call. As a general rule, you'll want call waiting disabled for all your data calls. That's because the little beep you get when your line is busy and another call comes in can sabotage your data connection. It may produce garbage characters at best, and sometimes you may lose a connection altogether. In most parts of the country you disable call waiting from a touch-tone phone by dialing *70 and then the phone number. (The asterisk corresponds to the star button on the touch-tone keypad.) From a rotary dial phone, you disable call waiting by dialing 1170 followed by the phone number. Note that while these codes are in common use, they aren't in universal use. So check with your local phone company just to be sure.

In any event, if I did have call waiting on my modem line, I'd have dialed ATDT*70,908-469-3450. When an asterisk appears in a phone number following an ATDT, the modem generates the tone associated with the star button. As for the comma after the *70, you'll recall that it gives rise to a pause of approximately two seconds. That's to give your phone company's system a moment to digest the request to disable call waiting. (Don't worry: as soon as your modem hangs up, call waiting becomes active again.)

The Service Answers and You Log On

Now, if all goes well, when you type a carriage return after your dialing string you'll hear your modem dial the number, then you'll hear the service's phone ring and pick up. The brief whooshing tones that follow will be data carriers, but after a second or two your modem will stop sending all this activity to your speaker and, in your terminal window, you'll see:

CONNECT 2400

And now nothing happens. That's because Second Sight, like many host programs, expects you, the caller, to type a carriage return when you're connected. So let's get on with it and hit Return. The BBS then responds with:

Connection made at 2400 baud.
Dragon's Cave BBS
FidoNet 1:2605/602  
Maclist 6:6001/2  
VNET 45:580/102

If you don't complete the validation survey, your account is deleted at midnight!  
REAL NAMES only, please!  
Your first name?

The numbers next to “FidoNet,” “Maclist,” and “VNET” are Dragon's Cave's node addresses in those networked message bases. That's so you know from the beginning what message networks it belongs to. As for “Your first name?” it's a prompt. Prompts typically conclude with question marks. The ball is in your court now, so you type your first name followed by a carriage return. You're then prompted for your last name and, when Second Sight can't find your name in the BBS's user log, you're prompted for a confirmation. On your screen the whole transaction looks like:

Your first name? John  
Your last name? Smith  
Searching userlog...  
John Smith, is that correct (Y or N)? y  
Where are you calling from?

You respond to that last prompt with your town and state (again followed by a carriage return), and Second Sight comes back with:

Now enter a password you would like to use. IT MUST BE AT LEAST 4 CHARACTERS LONG AND UP TO 8 CHARACTERS LONG. Lower and uppercase letters are the same, and you may also use numbers and symbols like '=' and '*'. REMEMBER THIS PASSWORD!  
What is your password?

You make up a password, “dingbat,” say, and type it followed by a return. Dragon's Cave asks you to type it again, and this time its prompt and your response look like:

Type it 'ain to be sure I've got it correct--> *******

From now on you'll never see your password on your screen again. For reasons of security, it will always be echoed back to you with asterisks.
Now you'll get Second Sight's line feed test:

```plaintext
== Linefeed Test ==

How many rows of periods do you see (1 or 2)? 2
```

We've responded to the last prompt with "2." That's because we don't need Second Sight to send us a line-feed character every time it sends a carriage return. If we did need the line feed, the second row of periods would have come out directly on top of the first row, and we'd have seen just one row of periods. We'd accordingly have answered "1" instead of two, and from that point forward Second Sight would send line feeds after each of its carriage returns.

Understand that these are all first-time procedures. The BBS is developing a group of settings for us and will apply those settings when we call again. We can change them later if we wish, but otherwise we won't see these tests again. That is, we won't see them until we call another Second Sight BBS.

After the line feed test comes the clear screen test:

```plaintext
Hello World
== Clear Screen Test ==

This test tells me how to clear your screen. If you would rather have me never clear your screen, you can enter the letter 'X' to the question below.
Is there still a line on your screen above this that says 'Hello World' (Y,N,X)? x
```

I mentioned earlier that Second Sight would let you tell it how to handle screen clears, and this is where it does it. You can see from our response to the prompt that we've elected never to have the screen cleared. There's too much risk of losing information at which we may want to take a second look.

Now some user preferences. First:

```plaintext
Do you want this system to prompt you to 'Press CR to continue...' after each 'screen' of information sent to your terminal (Y/N)? y
```

I recommend that for your first several sessions you take your medicine one screen at a time. No point in rushing things. Later, when much of what you see has become old hat, you can go back to your user preferences (we'll soon see how to get to those) and change them to suit your developing tastes.
Finally we get:

==== 'Hot Menus' Preference ====

'Hot Menus' means that menu choices will be reacted to immediately when you press a command selection key (without having to enter a carriage return afterwards). The main advantages are that this is much faster than having to wait for the whole menu to be printed out, and only one keypress is needed to select a command. The main disadvantage is that if you are prone to 'noisy' phone lines when accessing this system, spurious characters generated by phone line static may be interpreted as random command selections. Do you wish to have the 'Hot Menus' feature used (Y or N)? y

My phone lines tend not to generate much random noise, so we went ahead and chose the hot keys option. Some non-Second Sight host programs give you hot keys automatically; others don't offer this option at all. With many prompts, the hot keys option spares you the need to type a carriage return after your reply, but that's not true of all prompts. It's never true when your reply can be of variable length. If the service prompts you for your name, for instance, and you reply with "Max," the service has no way of knowing that you were planning to type "Maximillian" and can therefore make no assumptions about where the carriage return will go. In general, hot keys only work with prompts that list all acceptable replies, for instance the prompts that end with (Y or N)? But if you become familiar enough with the service to know that such a prompt is coming, you can type your reply before the prompt actually appears in your terminal window. When you do that, Second Sight won't bother to complete whatever it's typing. It will assume you know what you're doing and immediately take action in accordance with your reply.

Some Elementary Navigating 'Round the BBS

Now that you're logged on, there's a good chance that your Second Sight BBS will send you something like this:

Welcome to the Dragon Cave BBS!!
This BBS is dedicated to the exchange of information and files relating to the Apple Macintosh series of computers.
However, users of ALL types of computer users are welcome here!
To become a member of this BBS, you MUST complete the validation survey.
This is a simple and painless process, which is rather standard in the BBS world. It gives the sysop some basic information about his/her users and assists in ensuring that everyone is fair and maintains a single account on the BBS.
Here are the few rules I would like everyone to follow:
No uploading of any commercial or pirated programs. Doing so results in the immediate deletion of your account. If you see a file that has been improperly posted, leave the sysop E-mail indicating so! Nude images and other pornographic material will not be posted on this BBS, and will not be credited to a user as an upload.
Please upload all files using the Compactor utility, as this saves disk space and time (since I will have to stuff files if you do not). The current version of Compactor is in a self-extracting archive in the Compression programs section of the Library (file download sections).
Please do not post your messages entirely in uppercase letters. THESE MESSAGES MAKE IT SEEM TO OTHERS THAT YOU ARE SHOUTING AT THEM (see what I mean!?).
No racial comments or foul language of any type will be tolerated in the message areas. If you have a strong opinion, fine, but make it known in a civilized fashion.
I request that all users be active participants - in the message areas, as well as in uploading files. If you do not have a Macintosh, you are still welcome here; please contribute the best you can to appropriate “Special Interest Group” (SIG) message areas.
As a general disclaimer, I am not responsible for any consequences as a result of using files that are uploaded by users. I cannot be absolutely sure if all of them are bug-proof and “safe” programs. If you run into any programs which you are having real problems with, please post a Public Message about them.
Above all, have fun and call often!
Welcome, John Smith, from Anywhere USA!
Connected on 04/22/92 at 09:59:09
You are caller number 17357
No new messages have been sent to you since your last call.
Press CTRL-S to Pause, or any key to resume.
Reviewing the text we just received, if you plan to use the BBS at all, complete the validation survey and join the BBS. The survey will be presented to you as a series of prompts, and you can be reasonably sure you won't be asked too many embarrassingly intimate questions. The sysop will want to know your real name, address, and phone number, maybe also your age, but that's about as much personal information as you'll be required to disclose. To remain a member in good standing, you may (or may not) be required to maintain a given upload/download ratio. That is, you may have to upload a file for every 10 or so files you download in order to keep from losing your downloading privileges. You may also be required to kick in a few bucks toward the BBS's expenses. Those expenses can be considerable, and membership fees and other mandatory contributions are normally quite small.

Dragon's Cave may be a tad more prudish than some other BBSs as to the matter of nude images, but if it's soft porn graphics you want, as long as you're willing to invest the phone time and dial up enough services, you can probably find what you're looking for. Dragon's Cave seems to like only Compactor (now called Compact Pro) for compressing whatever files you might want to upload to the BBS, but most BBSs will also accept files compressed with Stuffit. The warning about messages sent entirely in uppercase letters applies to telecommunicating in general. SAVE ALL-CAPS FOR EMPHASIS! And, beyond Dragon's Cave's warning about racial slurs and foul language, try to avoid losing your temper. Along with being just plain offensive, online eruptions of rage and indignation (they're called flames in the telecom world) have a way of looking particularly witless when they scroll past someone else's screen.

Message Bases

As for the last two lines in Dragon's Cave's message to first-time callers, Second Sight automatically checks to see if any messages, local or networked, have been left for you on the BBS. Since you're only now just logging on, you're not likely to have any mail. But to demonstrate that the system actually works, in the course of preparing Appendix C for this book I wanted to list some mail-order sources of mini DIN-8 connectors for people who like to make their own modem cables. In many parts of the country, they can be very hard to find in stores. I posted a message asking for help with mail-order
sources, and a few days later, instead of "No new messages have been sent to you since your last call" I got:

The following 6 messages have been sent to you:
Msg #48796 in *Macintosh Hardware Topics* Created on 04/15/92 at 02:22:52
To: Stephen Taylor, From: Gary Schmidt
Subject: Mini DIN-8s
Msg #48411 in *NJMUG Echo* Created on 04/14/92 at 01:31:51
To: Stephen Taylor, From: Liz Kerr
Subject: Mini DIN-8 connectors
Msg #47761 in *Macintosh Hardware Topics* Created on 04/12/92 at 12:08:16
To: Stephen Taylor, From: Rich Felleppa
Subject: Re: Mini DIN-8s
Msg #46959 in *Macintosh Hardware Topics* Created on 04/12/92 at 11:50:11
To: Stephen Taylor, From: Dickens
Subject: RE: Mini DIN-8s
Msg #46855 in *Macintosh Hardware Topics* Created on 04/12/92 at 06:52:29
To: Stephen Taylor, From: Bill Tomkins
Subject: Mini DIN-8s
Msg #46798 in *NJMUG Echo* Created on 04/12/92 at 23:10:34
To: Stephen Taylor, From: Alice Price
Subject: RE: Mini DIN-8 connectors
PLEASE: Promptly read and delete all private messages no longer needed!

The replies came from all around the country, and when I checked out the information, much of it was eminently useful.

"Press CTRL-S to Pause, or any key to resume" simply means that if text goes scrolling by too fast for you to read it, hold down your Control key and type an s. (For non-ADB keyboard owners, remember that we've asked our terminal program to make the Option key behave like a control key for the duration of the session.)

Almost all BBSs send you a marquee adorned with some primitive graphics at the beginning of a session. The following is Dragon's Cave's.
After that comes the message of the day (which I'm presenting only in part):

Press CTRL-S to Pause, or any key to resume.

NEWS AND COMMENTARY FROM THE SYSOP

Last Updated: 01/07 (10:30pm)

“BBS IN A BOX” IS BACK!!! THIS IS A CD-ROM WITH 620MEGS OF FILES TO D/L!

The latest volume (IV) of “BBS in a Box” is now online, offering 8,000 files - 620MegaDs compressed, -> 1.1 GIG decompressed <-> to download.

For info on the CD, choose option “B” off the Main Menu.

The File Transfer Menu ...

Press any key to continue...

Note that announcements of any length are usually preceded with a line telling you what control characters you can use to slow down the flow of information from the BBS. We press any key to continue and get:

Press CTRL-S to Pause, any key to resume or CTRL-C to cancel...
Happy Birthday...
In 1816 Charlotte Bronte, author of "Jane Eyre"
In 1838 John Muir, naturalist.
In 1916 Anthony Quinn, Actor/Sculptor
In 1918 Ella Fitzgerald, jazz singer
In 1926 Elizabeth II, Queen of England

Quite a few BBSs begin each session with Michael E. Connick's Daily Bulletin V2.0. It's a daily online almanac that lists about two pages of birthdays and events in history for whichever day you call. Just this once, however, we'll interrupt the procession of birthdays with a ^C. When we do that, the BBS goes on and sends its main menu:

--- DRAGON'S CAVE BBS MAIN MENU ---

--- File Transfers
--- Public Messages
--- Read the OTHERNETS Conference
--- Information and News
--- Ask for Validation
--- Read about the COPERNICUS Point Software!
--- How to order Copernicus
--- "BBS IN A BOX" - NOW ONLINE HERE!!
--- User Utilities
--- Goodbye, Logoff this BBS
(27 minutes left) Command (F,P,*,I,A,R,H,B,%U,G) ?

Since the prompt following the menu is the kind that ends with all allowable options in parentheses, if you've chosen Second Sight's hot keys feature you won't have to wait for the whole menu to appear on your screen before you reply to the prompt.

Now let's explore. First we'll type <p> to examine the public message base. Like most host programs, Second Sight accepts its one-letter user responses in either uppercase or lowercase. (In computer-ese, we'd say that Second Sight isn't case-sensitive.) And <p> is telecom-ese for "the letter p." From now on we'll be using that notation—bracketing a character or sequence of characters between less-than and greater-than characters—to describe our prompt responses.
<p>brings us to Dragon’s Cave’s public message base menu:

--- Public Message Base Functions ---

--- Local’ Public Message Bases
--- ‘Networked’ Public Message Bases (Echos)

--- What are ‘Local’ vs. ‘Networked’ Public Bases?

--- Choose Sections (Local & Echos) to Combine
--- Scan Combined Sections with Option to Read
--- Read Combined Sections with Option to Read

--- Local Electronic Mail
--- On-Line Games
--- Zap Back to Main Menu
--- Download Sections (26 minutes left)

Command (L,N,W,1,2,3,E,*,Z,!) ?

If I want to find the messages for me that were listed above, I now have a pair of options. Second Sight will let me read messages either from one message base at a time or from several message bases combined. After giving my message listings a quick once-over, I see they come from two different echoes, Macintosh Hardware Topics and the NJMUG Echo, both of which are Macintosh-related. Combining message bases for subsequent reading can be just a trifle time-consuming, so first I try the shorter way and type an <n>, which takes me to the echo (i.e., networked public message base) messages:

--- Networked Public Message Menu ---

Networked Public Message Sections are ‘Echos’

--- Macintosh “General Topics” Echos
--- General Interest/Non-Computer Related Echos
--- Macintosh Support Echos (TABBY, RRH, MANSION, etc)

--- Back to the Public Messages Menu

--- Local Electronic Mail
--- On-Line Games
--- Zap Back to Main Menu
--- Download Sections (25 minutes left)

Command (1,2,3,B,E,*,Z,!) ?
If the Macintosh Hardware Topics and NJMUG echoes can be reached this way, it will probably be by typing <I> for "Macintosh 'General Topics' Echos [sic]," so that's what I do. And lo! I get yet another menu:

======== Macintosh "General Topics" Echos ========
Echos on general Macintosh computer-related topics

<1> Macintosh Developers/Advanced Programming (MACDEV)
<2> Macintosh Novices Programming (MACNOV)
<3> Macintosh HyperCard Discussions (MACHYPE)
<4> Macintosh Advertisements, Buy/Sell (MAC_ADS)
<5> Macintosh Networking/Telecommunications (MACNETCOM)
<6> Macintosh File Listings (MACFILES)
<7> Macintosh Virus Prevention (MACVIRUS)
<8> Macintosh Games & Entertainment (MAC_GAMES)
<9> Macintosh Hardware Topics (MACHW)
<A> Macintosh Software Topics (MACSW)
<C> PC-Mac Connectivity Echo (PCMAC)
<D> Macintosh Questions & Answers (MACQA)

======== Back to the Echos Menu ==========

<B> Back to the Echos Menu

<E> Local Electronic Mail <*> On-Line Games
<Z> Zap Back to Main Menu <I> Download Sections
(25 minutes left) Command (1, 2, 3, 4, 5, 6, 7, 8, 9, A, C, D, B, E, *, Z, !)?

I see that Macintosh Hardware Topics is listed in the menu, but the NJMUG echo—BTW, NJMUG stands for New Jersey Macintosh Users Group—isn't listed.

If this were real life, I'd probably just go ahead and type a <9> and retrieve at least some of my messages while I'm already this close. But because this is, after all, a didactic session, let's go all out and type a <B>, which takes us back to the Echoes Menu, then another <B> to go even further back to the public message base menu. Remember that if I've activated hot keys and type each <B> one right after the other, I'll get the menu I'm looking for almost instantaneously. I'll reprise it now so you don't have to turn pages:

======== Public Message Base Functions ========

<L> 'Local' Public Message Bases
<N> 'Networked' Public Message Bases (Echos)

<W> What are 'Local' vs. 'Networked' Public Bases?
This time, let's go another route and combine the two message bases from which we want to read messages. That way you can see what some of the national networked bases are. A BBS can choose to carry as few or as many message bases as it cares to. One of the reasons I picked Dragon's Cave as an example is that it carries so many different bases. When I respond to the prompt above by typing a <1>, Second Sight comes back with a prompt for every message base on the BBS:

Dragon Chat (Local Chat) - add to combined section list (Y/N,Q)? N
Macintosh Files - add to combined section list (Y/N,Q)? N
Second Sight Support - add to combined section list (Y/N,Q)? N
Mac Novice Programming - add to combined section list (Y/N,Q)? N
NY-NJ Metro Mac Users - add to combined section list (Y/N,Q)? N
Tabby Support - add to combined section list (Y/N,Q)? N
Macintosh Developers - add to combined section list (Y/N,Q)? N
Macintosh Advertisements - add to combined section list (Y/N,Q)? N
Mac Virus Topics - add to combined section list (Y/N,Q)? N
Macintosh Networking - add to combined section list (Y/N,Q)? N
Mac Hypercard Programming - add to combined section list (Y/N,Q)? N
On-Line Games Support - add to combined section list (Y/N,Q)? N
Wrestling Match Reports - add to combined section list (Y/N,Q)? N
Macintosh Q&A - add to combined section list (Y/N,Q)? N
PC-Mac Connectivity - add to combined section list (Y/N,Q)? N
Mansion Support - add to combined section list (Y/N,Q)? N
Macintosh Games - add to combined section list (Y/N,Q)? N
North New Jerseyy Users - add to combined section list (Y/N,Q)? N
Macintosh Hardware Topics - add to combined section list (Y/N,Q)? Y
Macintosh Software Topics - add to combined section list (Y/N,Q)? N
Dragon's Cave Points - add to combined section list (Y/N,Q)? N
NJMUG Echo - add to combined section list (Y/N,Q)? Y
NJ For Sale Conference - add to combined section list (Y/N,Q)? N
Local BBS Advertisements - add to combined section list (Y/N,Q)? N
Mac FREQs for Points - add to combined section list (Y/N,Q)? N
WireTap TeleCom Support - add to combined section list (Y/N,Q)? N
Home & Garden Chat - add to combined section list (Y/N,Q)? N
Crafting - add to combined section list (Y/N,Q)? N
Macintosh Communications - add to combined section list (Y/N,Q)? N
MacWoof Support Echo - add to combined section list (Y/N,Q)? N
Willowbrook Saga RPG - add to combined section list (Y/N,Q)? N
Points Conference - add to combined section list (Y/N,Q)? N
Offline Readers - add to combined section list (Y/N,Q)? N
Consumer Report - add to combined section list (Y/N,Q)? N
Home Repair Tips - add to combined section list (Y/N,Q)? N
Macintosh System 7 Echo - add to combined section list (Y/N,Q)? N
Role-Playing Games - add to combined section list (Y/N,Q)? N
Start Fleet Battles Game - add to combined section list (Y/N,Q)? N
OtherNets Conference - add to combined section list (Y/N,Q)? N
First Class BBS Echo - add to combined section list (Y/N,Q)? N
UNKNOWN - add to combined section list (Y/N,Q)? N
Changes are <P>ermanent, <T>his call only, <C>ancelled
(P,T,C)?

Public Message Base Functions

<L> 'Local' Public Message Bases
<N> 'Networked' Public Message Bases (Echos)

What are 'Local' vs. 'Networked' Public Bases?

Choose Sections (Local & Echos) to Combine
Scan Combined Sections with Option to Read
Read Combined Sections with Option to Read

Local Electronic Mail <*> On-Line Games
Zap Back to Main Menu <!> Download Sections

Command (L,N,W,1,2,3,E,*,Z,!) ?

You can see that I answered "no" (i.e., <N>) to the prompt for every base except Macintosh Hardware Topics and the NJMUG Echo. You can also see that you can quit the process at any time by typing <Q>. But mainly you can see how many message bases there are—and those are by no means all extant bases; they're just Dragon's Cave's unusually generous selection. When I'm done with the list I tell the BBS that my changes—that is, the selection of echoes I'll be reading—apply only to this call, as on another day I may want to look at a different assortment of message bases. When I've made my choices, I'm presented yet again with the Public Message Base menu, and this time I type <3> to go to where I can read messages on the two bases I've chosen. The BBS now sends:

Read messages:
<N>ewly added messages since last call
<B>egin after highest message number last read
<F>orward chronologically
<R>everse chronologically
<I>ndividually

Command (N,F,R,I,B) ?

I want to read my messages individually, so I type <!> and the BBS responds with:

Message numbers range from 1 to 51264
Enter message number:
I now enter 46959, the number of one of the messages, and Dragon's Cave comes back with:

Searching...
Msg #46959 in *Macintosh Hardware Topics* Created on 04/12/92 at 11:50:11
To: Stephen Taylor, From: Dickens
Subject: RE: Mini DIN-8s
ST> Does anyone know a good mail order source for Mini DIN-8 connectors?
ST> I need to wire some custom Mac cables. When I need a custom cable, I find one that isn't being used and adapt it!
But try:
   Electro Products/ Seattle, WA
   1-800-423-0646 (Ordering)
   Custom Cabling, cable supplies
They seem to carry what I need and they're cheap too!!
Dickens - Tabby 3.0b3
* Origin: 4th Dimension BBS/Mansion 617/494-0565 Boston, MA (1:101/450)
<D>elete, <R>eply, <N>ext, <C>ontinuous, <Q>uit,
RETURN = Next (D,R,N,C,Q)?

The prompt explains that "RETURN = Next." It also says <N>ext, which is BBS shorthand for "type <N> for next message." We conclude that typing a return at that prompt is equivalent to typing an <N>, and sure enough it is! The carriage return causes us to be prompted for another message number:

Message numbers range from 1 to 51264
Enter message number:

I then enter the number of another of the messages that are waiting for me and continue until I've read all six messages.

BTW, those lines in Mr. Dickens's message that begin with "ST>" are quotes from the message I'd originally posted. There are several shareware quoting utilities that let heavy message base users do that. If it's something you think you might like to do yourself, you can download such a utility from the file libraries of many BBSs. The next chapter deals with downloading.
BTW, as for the content of the above message, I checked with Electro Products, and they wound up being listed as a parts source in the appendix, for which I thank Mr. Dickens. However, adapting pre-wired cables can be tricky. You need to be sure that the cable you begin with has lines coming from all the pins you'll be wanting to wire up. Otherwise you can chop up a perfectly good cable and discover there's no wire connected to pin 3, say, even though you plan to connect pin 3 to something else. Yes, you can test a cable with a continuity checker, but you can also stay on the safe side and restrict your adaptations to straight-through cables in which all the mini DIN-8 pins have wires soldered to them.

BTW, from the "Origin" line at the bottom of the above message, you'll notice it came from Boston. I received other replies to my original postings literally from points all across the country, ranging from eastern Long Island to New Jersey to Minnesota to Arkansas out to Texas, Arizona, and California. Each of the messages sent to me by the BBS concludes with this prompt:

<D>elete, <R>eply, <N>ext, <C>ontinuous, <Q>uit,
RETURN = Next (D,R,N,C,Q)? Q

After I'm done reading all them, I type a <Q>, and the BBS returns me to its Public Message Base menu. The <C>ontinuous option doesn't apply when you're reading messages individually (as we are right now) but I could also choose to reply to a message right then and there by typing <R>. (That's what "<R>eply" means.) When I do that I'm prompted as follows:

.........1........2........3........

Subject (40 characters max.): Mini DIN-8s?

As a default, Second Sight presents the subject of the message I'm replying to, but it also gives me the opportunity to alter it. As a further convenience, it types a "ruler" (the line of periods and numbers) just above the prompt so I can see if I'm observing the maximum of 40 characters allotted to message subjects. I elect to leave the default subject as is by typing just a carriage return, and now I'm prompted with:

You may now type in up to 200 lines of text.
Lines will automatically wrap around at the 75th character.
To stop entering, type a carriage return on a blank line.

1> Many thanks for the information, especially since Macintosh cabling connectors can sometimes be tough to find. Rest assured that the info will be put to good use.

4> <C>ancel, <S>ave, <L>ist, <I>nser, <D>elete, <A>d, <E>dit (C,S,L,I,D,A,E)? C

Unlike the message functions of a number of the online services, Second Sight performs automatic, word-processor-style word wrapping. If you type past the 75th character, as I did when I typed "connectors" on line 1, the host program breaks the line between words and puts the overlong word on the next line. As you can see, I typed a carriage return on a blank line following my message, and when I did that I got a prompt asking what I now want to do with my message. When I choose <S>ave by typing <S> the BBS tells me:

Saving message #55149...
Message size 164 bytes...
Flagging Dickens that a message is waiting...
Press carriage return to continue...

Soon—probably in the middle of that night—Dragon's Cave's computer will link up automatically with another node (i.e., another member BBS) on the network. The network software will pick up a packet of new messages from the node and also drop off a packet of its own new messages. Eventually, moving from node to node, my message will make its way to Boston, where, presumably, Mr. Dickens will check in with the 4th Dimension BBS and read it.

Incidentally, if you're just poking around a BBS's message bases to see what's going on, you can do things a bit more simply. If you're a hardware freak and want to know what sort of concerns are on your conferees' minds, you can go to the following Public Message Base menu on the next page.
-------- Public Message Base Functions --------

<L> 'Local' Public Message Bases
<N> 'Networked' Public Message Bases (Echos)

<W> What are 'Local' vs. 'Networked' Public Bases?

<1> Choose Sections (Local & Echos) to Combine
<2> Scan Combined Sections with Option to Read
<3> Read Combined Sections with Option to Read

<E> Local Electronic Mail  <*> On-Line Games
<Z> Zap Back to Main Menu  <!> Download Sections
(23 minutes left) Command (L,N,W,1,2,3,E,*,Z,!) ?

and just type an <N>, which will bring you to:

-------- Networked Public Message Menu --------

Networked Public Message Sections are 'Echos'

<1> Macintosh "General Topics" Echos
<2> General Interest/Non-Computer Related Echos
<3> Macintosh Support Echos (TABBY, RRH, MANSION, etc)

<B> Back to the Public Messages Menu

<E> Local Electronic Mail  <*> On-Line Games
<Z> Zap Back to Main Menu  <!> Download Sections
(23 minutes left) Command (1,2,3,B,E,*,Z,!) ?

Now, if you type a <1> you'll get to:

-------- Macintosh "General Topics" Echos --------

Echos on general Macintosh computer-related topics

<1> Macintosh Developers/Advanced Programming (MACDEV)
<2> Macintosh Novices Programming (MACNOV)
<3> Macintosh HyperCard Discussions (MACHYPE)
<4> Macintosh Advertisements, Buy/Sell (MAC_ARTS)
<5> Macintosh Networking/Telecommunications (MACNETCOM)
<6> Macintosh File Listings (MACFILES)
<7> Macintosh Virus Prevention (MACVIRUS)
<8> Macintosh Games & Entertainment (MAC_GAMES)
<9> Macintosh Hardware Topics (MACHW)
<A> Macintosh Software Topics (MACSW)
where you can type a <9> and go right to:

------------------------- MACHW Echo -------------------------
Topics related to Macintosh Hardware

<P> Post a new public message
<S> Scan public messages
<D> Delete a public message
<R> Read public messages
<F> Find a user's valid name for message addressing

-------------------------

<B> Back to Computer-Related Echos Menu
<Z> Zap back to Main Menu

(21 minutes left) Command (P,S,D,R,F,B,Z) ?

As usual, the menu concludes with a prompt, and typing <R> to read messages brings you to:

Read messages:
<N>ewly added messages since last call
<B>egin after highest message number last read
<F>orward chronologically
<R>everse chronologically
<I>ndividually

(21 minutes left) Command (N,F,R,I,B) ?

You can see from the first two options that Second Sight keeps track of the messages read by every user of the BBS. If we read messages "<F>orward chronologically" and start with message #1, we may wind up seeing some pretty old messages. We don't have enough information about what might be a good message number to start with, so let's choose "<R>everse chronologically." Incidentally, the chronology in question pertains not to the dates on which the messages were actually posted but to the dates on which the BBS received them. If the network delivers an old message to the BBS last night
and a much newer message to the BBS last week, the newer message will show up in the BBS's chronology as being older than the old message.

Nonetheless, we'll type an <R>, and Second Sight responds with:

Message numbers range from 1 to 50614
Enter starting message number:

To work backward from the newest message on the BBS, we then type <50614>, whereupon the BBS types back:

Searching...
Msg #50614 in *Macintosh Hardware Topics* Created on 04/19/92 at 16:11:55
To: Sam Lewis, From: Kevin Gray
Subject: RE: Three Blind Mice
IW> Having to use a PowerBook trackball, I'm suffering from mouse-withdrawal
IW> symptoms, and thinking about alternate possibilities of a CORDLESS mouse.
I've used a cordless mouse for a while, Ilbert. The two major problems that I encountered were:
1. Battery Life: unless you used rechargeable batteries (which you have to replace even more often than alkaline) it will keep you poor. I found that I was replacing my nicad batteries every two to three days. The mouse that I used had an auto shut down feature that shut itself off after five minutes of inactivity to preserve battery life.
2. Infrared reception: The cordless mice that I've seen all use an infrared device that must be within a certain radius of the mouse to pick up the signals from the mouse. So, while they are called a cordless mouse, they really aren't cordless at all...only the "mouse" portion that you roll around on your desktop.
The infrared receptor posed a problem for me since it has to be in front (or within approximately 45 degree radius) of the remote mouse. In addition, the mouse must point to this device for it to work. I found that my mouse habits had me pointing the mouse in odd angles, thus the receptor was not always picking up the signals from the mouse...therefore no cursor movement on the screen. This is annoying and possibly deadly if you're playing a fast game.
After about a month I sold the cordless mouse, although I liked its feel very much. I don't think this type of mouse is an acceptable solution for you.

- Copernicus II (100)
  * Origin: => Wiltype * London/Ontario/Canada *
  <= (1:12/109.555)
  <D>eleate, <R>eply, <N>ext, <C>ontinuous, <Q>uit,
  RETURN = Next (D,R,N,C,Q)?

Either an <N> or a carriage return takes us one message further back in the message base:

Searching...
Msg #50613 in *Macintosh Hardware Topics* Created on 04/19/92 at 09:50:11
To: All, From: Sam Lewis
Subject: Three Blind Mice?
Three Blind Mice ............etc., and she cut off their tails with a carving knife!
Having to use a PowerBook trackball, I'm suffering from mouse-withdrawal symptoms, and thinking about alternate possibilities of a CORDLESS mouse.
Anybody out there have knowledge or hands-on experience with this species of rodent?
Magazine reviews, recommended buys? And are they compatible with 170 PowerBook's configuration? Do any others of you miss your mice? Any feedback (cheesy or otherwise!) appreciated.
- Copernicus II (0117)
  <D>eleate, <R>eply, <N>ext, <C>ontinuous, <Q>uit,
  RETURN = Next (D,R,N,C,Q)?

And there's the message that gave rise to the one before. At this point we can keep reading backwards into the base either by typing a carriage return after each message or, since we're capturing all incoming and outgoing text, we can just hit <C> and let the messages be sent to us continuously. If, while they're rolling by, we suddenly decide to cut them off, all we need do is type a ^C—old mister interrupt. That will cause Second Sight to prompt us with:

<D>eleate, <R>eply, <N>ext, <C>ontinuous, <Q>uit,
RETURN = Next (D,R,N,C,Q)?
after the last message that scrolled by. We can then take whatever action we care to, including typing <Q> to quit the message base.

More Navigation

Now let's proceed with our BBS tour. You may have noticed that all of Dragon's Cave's submenus let you return to its main menu by typing <Z> for "Zap Back to Main Menu." Let's do it.

---------- DRAGON'S CAVE BBS MAIN MENU ----------

---------------------------------
<F> File Transfers
<P> Public Messages
<* Read the OTHERNETS Conference
<E> Electronic Mail (member-to-member)
<I> Information and News
---------------------------------
<R> Read about the COPERNICUS Point Software!
<H> How to order Copernicus
---------------------------------
<B> "BBS IN A BOX" - NOW ONLINE HERE!!
---------------------------------
<U> User Utilities
<0> On-line Games
---------------------------------
<G> Goodbye, Logoff this BBS
(20 minutes left) Command (F,P,*,E,I,R,H,B,%U,O,G) ? E

Note that if you're a first-time user, the above menu will look a little different. Instead of the line that reads "<E> Electronic Mail (member-to-member)" there'll be a line that says "<A> Ask for Validation." That's how you reply to the aforementioned validation survey (i.e., by typing an <A>). Note too that these menus will differ in their contents from one Second Sight BBS to the next. It's certainly true that all Second Sight BBSs have a distinct "look and feel," just as, say, Hermes BBSs have their look and feel. But each BBS remains free to customize its menus to its taste, and the variation can be considerable.
Exploring some more services of our chosen BBS, we can type an <E> and go to:

------------------- Local Electronic Mail -------------------
Mail sent in this form is “member-to-member”

----------<R> Read your local electronic mail
----------<S> Send electronic mail to other users
----------<L> Look for all messages on this BBS addressed to you
----------<F> Find a user’s valid name for message addressing
----------<E> Electronic mail to the Sysop

----------<N> “Networked” Electronic Mail (“NetMail”) Functions

----------<P> Public Messages Section
----------<Z> Zap Back to Main Menu
(19 minutes left) Command (R,S,L,F,E,N,P,Z) ?

where we can (<R>) read our local mail if we have any, (<L>) scan all the message bases on the BBS for mail addressed to us, (<E>) send a note to the sysop of this BBS, (<F>) find the exact name of a user to make sure any mail we send him or her gets properly addressed, (<S>) send mail to members of this BBS, or (<N>) gain access to the BBS’s networked E-mail functions, through which we can send E-mail to members of other BBSs. On that last count, it’s not exactly cricket to use the auspices of FidoNet to conduct a major correspondence with someone halfway around the world. Moving messages from one node to another involves a good amount of long distance telephone time, even if most of it does happen in the middle of the night when rates are low. Phone costs are borne by the participating BBSs, and excessive use of inter-BBS E-mail for personal reasons is tantamount to mooching.

With that understood, take another look at the main menu. Typing an <I> brings us to:

------------------- Information Files -------------------

----------<A> Access Levels & Stats for Users
----------<D> Display Callerlog (previous day)
----------<C> Calendar of Events
----------<L> List all Callers to this BBS
----------<M> Make a voluntary $$ donation to the Sysop
----------<V> Voluntary Donators - THANKS!
Most BBSs offer menus something like this one. This is where you can find out more about the particular BBS you're calling, and typing one of the bracketed letters in the menu will cause the BBS to print the relevant information to your screen. Once again, if you're capturing text to disk, you can read it or print it out when the session is over.

Typing a <U> back at the main menu prompt will take you here:

-------- User Utilities --------

<N> New Password
<C> Change Terminal Preferences
<Y> Yell for System Operator to Chat
<S> Show Your Statistics

<E> Local Electronic Mail
<F> File Transfers
<O> On-line Games
<P> Public Message Sections

--------

<Z> Zap Back to Main Menu

(18 minutes left) Command (N,C,Y,S,E,F,O,P,Z) ?

As you can see, it's here that you can change your password or any of the other terminal preferences we went through when we first logged on. We won't show all of them right now, nor will we trek up every last tributary of the river we've been exploring. Nonetheless, the meat and potatoes of most Mac bulletin board systems are their file transfer sections, so let's take a preliminary peek. File transfers aren't really Dragon's Cave's strongest point, but for now that doesn't make much difference.
Let's type an <F> and have a look anyway:

------------------ File Transfers Menu ------------------
Upload Files directly to the appropriate DL section

CALENDAR: What is it & why use it?

------------------ File Download Sections ------------------
< C> Compactor: What is it & why use it?

------------------ Local File Download/Upload Sections ------------------

------------------ "BBS in a Box" CD-ROM File Download Sections ------------------

------------------ Difference between "local" files and CD-ROM files ------------------

------------------ Zap Back to Main Menu ------------------
(18 minutes left) Command (C,L,B,D,Z) ?

Dragon's Cave offers file transfers from two different sources, its own local
download/upload section and "BBS in a Box," a CD ROM distributed by the
Arizona Macintosh Users Group. "BBS in a Box" contains hundreds of mega-
bytes of Macintosh shareware that a BBS can make available to its members.
The local download/upload section, on the other hand, is supported by the
BBS's own users and sysop. For now that's where we'll go by typing <L>:

------------------ File Download Sections ------------------

------------------ Communications ------------------
------------------ Fonts ------------------
------------------ HyperCard Stacks ------------------
------------------ FKEYs/DA's ------------------
------------------ Programming Files ------------------
------------------ Utilities ------------------
------------------ Net-Related Files ------------------
------------------ Mac II Files ------------------
------------------ V-NET Files ------------------
------------------ Demo Software ------------------
------------------ Games ------------------
------------------ INITs/Cdevs ------------------
------------------ Music/Sound ------------------
------------------ Textfiles/Miscellaneous Stuff ------------------
------------------ Art, Graphics, Film ------------------
------------------ Red Ryder Host Support Files ------------------
------------------ Role-playing Files ------------------
------------------ Compression Programs ------------------

------------------ Back to File Transfers Menu ------------------
------------------ Zap back to Main Menu ------------------

Communications is the subject of this enquiry, so we'll type a <C> and get:

------------------ Communications ------------------

------------------ New files added since your last call ------------------
------------------ List available files ------------------
------------------ Search for characters in file names
<G> Get file’s description and other information
<D> Download a file from this BBS to your computer
============================================
<U> Upload a Communications File to this BBS
============================================
<B> Back to Downloads Menu
<Z> Zap back to Main Menu

(17 minutes left) Command (N,L,S,G,D,U,B,Z) ?

And now we'll list the files in Dragon's Cave's local communications library by typing <L>. This is what we get:

Press CTRL-S to pause, CTRL-Q to resume, or CTRL-C to cancel.

Name Size Date Accesses File Type
-- -- ---- --- --- ----
ASLtalk.SIT 28119 09/12/88 8 Macintosh Document
Terminal program in a DA, no xfers, supports 19,200 bps.

<D>ownload, <N>ext, <Q>uit, <C>ontinuous, RETURN = Next

Actually, ASLTalk is something of a rare find. You seldom see it in BBS file libraries any more. Back in the days before System 7.0 and even before MultiFinder first appeared (there I go again), the only way you could run two programs at the same time on a Macintosh was to have at least one of them be a Desk Accessory. And of all the varieties of software that you'd like to be able to run in the background, telecom programs come out near the top of the list. Even now, as we're about to ask for a continuous listing of Dragon's Cave's telecom-related files, we might very well want to work on something else while the list comes in. ASLTalk can't upload or download files, but there was another DA called BackDown that could, and if you had both DAs installed you could access all the basic telecom functions in the background and still have a major application running. That's history now, and since we're saving downloading for the next chapter we can skip the <D>ownload option in the prompt following ASLTalk's description. Instead, let's type <C> for a continuous listing of files. This is what we get:

Broadcast.SI 14025 03/26/89 12 Macintosh Document
Put this in your system folder and you can send messages to other users via AppleTalk. Go to the Chooser and click on Broadcast and a user to receive the message and you can send a short (3 line) message with a description icon
(choice of 12). Cheaper than MicroSoft Mail but no mail or whistles.

Charm.SIT 86963 07/08/89 3 Macintosh Document
Charm, a telecommunications program that supports the SEAlink protocol.
This application is in development. Docs are in Italian!

EasyCom.SIT 11852 11/23/88 26 Macintosh Document
Make MicroPhone have all the features of SmartCom.

FreeTerm.SIT 35432 07/02/88 12 Macintosh Document
PD telecommunications program

InstallStuff 104374 08/25/88 7 Macintosh Application
Installs Stuffit 1.5 by overlaying your earlier version.

MacArc.SIT 27605 07/02/88 22 Macintosh Document
De-arcs IBM format files on a Mac disk.

SETCLOCK 2.3 12476 11/17/91 3 Macintosh Document
New version of SetClock utility - calls a number via modem and sets clock exactly.

MACEXPRESS 8246 12/11/91 8 Macintosh Document
Call the MacExpress BBS (201) 235-1558 at 2400/1200 bps!
For more info, download this file!

FIRSTCLASS 206394 01/17/92 0 Macintosh Application
This is the interface for a great Mac BBS. Call 215-584-4756. It's a Phili number but they have lots of files, news and conferences. The great thing about this BBS is that it uses a Graphic interface like American Online.

AUTO_CLOCK.CP 112942 02/10/92 3 Macintosh Document
AutoClock 1.1.2. Calls the US Naval observatory and sets the Mac clock. Much better than Set Clock, and is compatible with System 7. Includes an init which, when used under System 7, sends an Apple event to the application to set the clock at any given interval (e.g. every day, month, hour, etc.)

ZTERM 189596 02/11/92 3 Macintosh Document
Finally! The newest release of Dave Alverson’s fantastic shareware communications program, ZTerm! Version 0.9 adds scripting, phone book, and lots of smaller features.

MWoof12.CPT 175029 03/02/92 4 Macintosh Document
MacWoof 1.2, macintosh point program.

SerialT.cpt 14567 03/02/92 3 Macintosh Document
Serial Toolbox, for use with MacWoof 1.2

WK 11.13 205567 04/15/92 0 Macintosh Document
For those of you who have WhiteKnight 11.12, here are the
patches for updating WK to 11.13. You will need the PUP updater to use the patches.
Press carriage return to continue...

I should mention that I've edited a lot of old files out of the list. Programs uploaded in 1988 are likely either to have been superseded by newer versions or to have fallen into disuse for one or another reason. But I've tried to show you enough of the list to tease you with possibilities of greatly enriching your trove of telecom software.

Thus teased, let's take a last nostalgic look at the main menu:

--- DRAGON'S CAVE BBS MAIN MENU ---

<F> File Transfers
<P> Public Messages
<*> Read the OTHERNETS Conference
<E> Electronic Mail (member-to-member)
<I> Information and News

--- Copernicus Software! ---
<H> How to order Copernicus

--- "BBS IN A BOX" - NOW ONLINE HERE!!! ---

<U> User Utilities
<O> On-line Games

<G> Goodbye, Logoff this BBS
(16 minutes left) Command (F,P,*,E,I,R,H,B,%,U,O,G) ?

The <R> option causes a blurb about a program called Copernicus to be printed to your telecom window. If you're about to become a serious BBS user you'd do well to consider buying it. It's capable of doing pretty much all the BBS navigating that we've just done manually, except it does it automatically and unattended. You can set it to call a BBS when phone rates are at their lowest, retrieve all your messages, post both new messages and replies to messages you've already read, download some files, read a few message bases and then log off—all while you're fast asleep.

And speaking of logging off, let's do it now by typing <G>. Try never to log off a service simply by hanging up your modem in the middle of a session. Yes, Second Sight can handle it, but some host programs become temporarily deranged by *phonus interruptus.*
Person to Person Connections

When you connect with a friend or business associate, things are a whole lot simpler. You can use the same settings we used for our first BBS call with one important exception: prearrange that both parties are set for half duplex/local echo instead of full duplex. If your number is 123-4567, all the other party has to do is type:

ATDT1234567 (followed by a return)

to his modem. If your modem is turned on, connected to your computer, and also connected to the phone line, and if you have a terminal program up and running on your Mac, the word RING should appear in your terminal window when the phone rings. Now all you need do is type:

ATA (followed by a return)

and your modem should answer. If your modem supports auto-answering and you type:

ATS0=1 (followed by a return)

to your modem before the other party calls, you won't even have to bother with an ATA. ATS0=1 (that's ATS zero =1) sets your modem to answer the phone after one ring. (Yep, ATS0=2 sets it to answer after two rings and so forth.) You'll both hear some phone rings and carrier tones in your modem speakers, and then you'll both see

CONNECT 2400

on your screens. Someone should start typing around now just so both parties get some reassurance that the connection actually exists.

During ordinary voice calls, most of us use a shared system of hints and signals to let each other know when we're in talk mode and when in listening mode. When speaking, we say "uh-huh" or "yeah?" or inflect our voices upward or downward. With a person-to-person data connection, two successive carriage returns usually mean that one party is done typing and the other is invited to respond. Beyond that, both of you are on your own.
Chapter 8

File Transfers I: A Cornucopia of File Formats

Transferring Text Files

If you dial up a BBS and start exploring, the chances are excellent that even before you explicitly try to upload or download anything, you'll be a party to the elementary transferring of files. In the last chapter, just after we logged onto the Dragon's Cave BBS, we were sent this introductory message:

Welcome to the Dragon Cave BBS!!
This BBS is dedicated to the exchange of information and files relating to the Apple Macintosh series of computers. However, users of ALL types of computer users are welcome here!
To become a member of this BBS...

We observed that Dragon's Cave was operated by a host program called Second Sight, and what Second Sight was doing then was sending us a text file. The Dragon's Cave sysop had prepared the file beforehand and saved it somewhere on the BBS's hard disk. He then configured Second Sight so that, once new callers have keyed in a few choices about how they'd like to use the
BBS, the host sends them the file that contains the welcome message. And since we started a capture file before we began the session, the text in the Dragon's Cave welcome message file was transferred to our capture file. Putting it another way, Dragon's Cave uploaded its welcome message file to our computer. Putting it a third way, we downloaded Dragon's Cave's welcome message file to our computer. We didn't explicitly request the download, but Dragon's Cave insisted, and, as first-time callers, we were in no position to object.

That kind of file transfer is sometimes called text transfer or straight text transfer or ASCII transfer or ASCII dump or text dump. The sending computer emulates a human being at the keyboard and sends the text file very much as though it were being typed.

Now and then you'll probably want to transfer text files yourself. If you intend to post a message on a BBS, for instance, you may decide to compose the message before you go online. You'd write the message using a word processor and save it as plain text. Then you'd log onto the BBS, find your way to whatever message section you want to post in, and when you're prompted for your message, you'd ask your own terminal program to send a text file. Some programs have a Send Text (or something very similar, like Send ASCII) option under their File menus. Others group text transfers with all the other kinds of file transfers. In Smartcom II you send text with this button:

When you click on it, you get Smartcom II's version of the Mac's Standard Files dialog box, shown in Figure 8.1.
Hewing closely to the metaphor that likens sending a text file to typing text online, Smartcom II calls the sending of text files autotyping. You pick the file you want to send, and then, in the Standard Files window, you click on Type. Personally, I'm not wild about the practice of giving proprietary names to common telecom procedures. If a service ever prompts you to send a text file, it means you'll have to remember that sending it is the same as autotyping it, and you can be pretty sure that no one's ever going to prompt you to autotype anything. Still, you get the idea.

MicroPhone II, meanwhile, is one of the terminal programs that groups text transfers with file transfers of other kinds. You select “Send” under the Transfer menu and get their version of the Standard Files dialog box, shown in Figure 8.2 on the next page.

If you're sending straight text, you have to be sure to choose ASCII/Text from the Protocol submenu. That's something you can easily forget to do, and it's why my ideal terminal program interface would make the sending of text files a separate option from protocol transfers. (White Knight has a separate Send TEXT File... option under its File menu, but it has other interface problems of its own.) In fact, sending plain-text files is sometimes called sending no-protocol, because no system of inter-terminal handshaking other than plain old XON/XOFF is involved.

Nonetheless, when you send or receive text files, you can be said legitimately to have entered the realm of file transfers. Of course, if you try to send
files that you’ve created with your word processing program and you don’t save the files in plain-text (i.e., ASCII) format, you won’t get very far, however much you may protest that the files do in fact contain text. Any time you try to open a non-plain-text file with your terminal program in hope of sending it as text, the Standard Files dialog box either will not show the file at all or its name will be grayed out. A few text processors such as the venerable McSink and MiniWriter DAs, save their files only in plain-text format, but full-featured word processors like Word and MacWrite normally save files in proprietary formats of their own. You can tell them to save what you’ve written as plain text (usually by using Save As... as opposed to Save), but in the bargain you’ll lose whatever custom formatting you’ve applied.

About Macintosh Files
Type and Creator

The distinction between plain text (or, if you prefer, ASCII) files and all other kinds is one you’d be well advised to understand when you begin uploading and downloading files. In the everyday language of Macintosh telecommunications, all other kinds of files are called binary files. Does that mean plain
text files aren't binary? Not at all. All Macintosh files are technically binary; in fact, all files of all modern digital computers are binary. What it does mean is that every 8-bit byte of information in a text file is understood to correspond in some reasonably meaningful way to a character in the Macintosh character set. By contrast, a binary file may contain large binary numbers—not decimal digits but true binary numbers still in ones-and-zeros format—spread out over several bytes, and it may also contain single-bit on/off switches and other data encoded according to schemes that no one but the programmer understands. If such a binary file were to be loaded into a text editor, it could well show up as a collection of nonsense characters.

To guard against that occurrence, every Macintosh file contains in its file header two 4-character codes that describe the nature of the file. One is the type code, the other the creator code. Apple likes to protect you from this fact on the grounds, I think, that it represents more high technology than is good for you. If you select a file in the Finder and choose Get Info, all you're told about its type or creator is whether the file is a document, an application, a system extension, or whatever and, if it's a document, the name of the application that created it. In order to learn more about files, you need some extra software, and if you don't already have this software, one way to get it is to download it from a BBS or online service, which, miracle of miracles, happens to be just what we're discussing now.

The 4-character type code for plain-text files is TEXT (all the letters in uppercase). Files of type TEXT are universally recognized by Mac text processors and word processors and also by telecom programs that have been asked to send a text file; files of type text aren't. Microsoft Word's "normal" file type is WDBN (for Word binary?). MacWrite's normal file type is WORD, and MacWrite II's normal file type is MW2D. File types tell programs what file format to expect. When MacWrite opens a file of type WORD it knows where to look, not just for the text you've typed into it, but also the format and page layout information you've assigned, and the program can then arrange the text so it looks the way you want it to.

Even though Apple tries to insulate you from all this, you can use any number of utilities to obtain type and creator information about a file. If you happen to have Apple's own ResEdit lying around—it's distributed free of charge through BBSs and online services—you just choose Get File/Folder Info... from ResEdit's File menu.
All halfway-serious Mac telecommunicators eventually download the shareware file compression utilities Compact Pro and/or Stuffit Classic. Both those programs routinely include type and creator information when they list files.

DiskTop, the popular Finder-like Desk Accessory, supplies type and creator information as a matter of course, but only if you choose Preferences... under the DiskTop menu. Then, when you’re shown DiskTop’s three Level buttons, click on Technical.

Among Mac modemers, McSink may be the most popular shareware text-editing DA, largely because a number of its features are particularly useful for massaging text before it’s sent and after it’s received. If you select File Utilities under McSink’s File menu and then slide over to the File Info... submenu, you can get type and creator information for any file you choose.

Probably the smoothest way of all is to use either of a pair of applications you can find on most Mac BBSs and online services: Finder Info, which is freeware, and GetInfo, which is shareware. GetInfo only works with System 7.0, but it’s the easier of the two to use because it’s a drag-and-drop utility. You just drag a file icon over GetInfo’s icon, and as soon as you let go, GetInfo promptly tells you all about the file. Be careful, though. Both applications not only let you see the type and creator of every file; they also let you change them. So if you do use them, make it a point to understand exactly what you’re doing.

Forks, Etc.: More on Binary Files

Macintosh files are divided into two parts. Maybe because Mac files can be said to fork off in two directions, each part is called a fork. (Actually, if the nomenclature were used precisely, the whole file would be the fork and each part would be a tine.) One part is known as the data fork, the other the resource fork. Stuff you type into your word processor or numbers you enter into your spreadsheet are data. Things like program code, icon bitmaps, and dialog boxes are all resources. Consequently, because an application file consists mainly of program code, dialog boxes, and the like, it will normally have a big resource fork and a small or nonexistent data fork; a word processing or spreadsheet document, which consists mainly of user-supplied data, will normally have a large data fork and a small or nonexistent resource fork.
The presence of forks is one feature of Macintosh files that distinguishes them from, say, IBM-compatible DOS files. A second distinguishing feature of a Mac file is its file header, the collection of information that contains a file’s name, type, creator, creation date, the date on which it was last modified, the size and location of each of its forks, its Finder icon, and so forth. Those and a number of other parameters comprise a file’s Finder information, and, while it’s true that other files ordinarily have headers too, the headers of non-Mac files aren’t formatted in the same way. As an obvious example, Mac files are permitted to have much longer names than DOS files, which means that more bytes have to be allotted for file names in the Macintosh file header.

Now, when you transfer a file from one or another service, you naturally want it to appear in your own computer with its name and both its forks in the right place, its type and creator properly assigned, its creation and modification dates intact, and all the rest. That would be easy enough if all computer files in the telecommunications universe were formatted according to Macintosh specifications. The format is rigidly defined, and your terminal program would know exactly where to look for the information it needs to set up your file properly. The problem is that type-less, creator-less, short-named DOS files, not to mention files created on any number of other non-DOS computers, store information about themselves in a number of different formats.

An early solution to this problem was called BinHex. BinHex is the name of both a program and a file format, and (back when glaciers were still receding from New England and the northern Plains states) Mac files were almost always transferred in BinHex format. First, using the BinHex program, the sender would convert from Mac to BinHex format before sending the file. Then, on receiving the file, the receiver would convert from BinHex back to normal Mac format. That way data and resource forks got put in the right places, and names, icons, types, creators, and other Finder information were conveyed properly. The trouble was that BinHex was intrinsically awkward; it had a lot of overhead—space overhead and also chronological overhead. BinHex files tended to be significantly larger than the Mac-format files from which they were created, and the conversions were annoyingly time-consuming. These days you don’t often see BinHex files posted on BBSs, but every now and then some dusty antique of a BinHex file will manage to turn up, and you may as well know what they are. BinHex files can usually be recognized by their suffixes—.HQX, .HEX, or sometimes .HCX—and diligent downloaders
usually keep some appropriate software around in the rare event that they might have to decode a BinHex file. The BinHex program itself is readily available as shareware, and at this writing the most recent version is 4.0. (Since demand is inordinately light, the appearance of a further upgrade is improbable.) The BinHex DA, Stuffit Classic, and Compact Pro are some other shareware programs that can hex and un-hex files, but I wouldn’t advise treating the acquisition of a BinHex utility as a matter of any great urgency.

MacBinary

Nowadays the predominant Macintosh file transfer format is called MacBinary. Conversion to and from MacBinary format is much faster than with BinHex, and terminal programs normally do it transparently and automatically. Overhead is low (only 128 extra characters), and you can be confident that no matter what kind of computer sent you the Macintosh file you've downloaded, its Finder information will be properly restored and both its forks will be correctly placed. Once upon a time, when non-Macintosh computers listed Macintosh files in their file libraries, files in MacBinary format would have the suffix .BIN appended to their names. Nowadays that practice is much less common. Instead, by a kind of gentleman’s agreement, most services put all Mac files except TEXT files into MacBinary format and don’t bother with the suffix. The file header tells your terminal program whether or not the file is MacBinary, so when you download it you don’t have to take any special action. Your terminal program recognizes it as a MacBinary file and performs the MacBinary-to-Macintosh format conversion as a matter of course.

Under normal circumstances, the MacBinary format isn’t used for the transfer of TEXT files. How come? Well, mainly because the text in TEXT files may be of interest to people who don’t use Macs, and MacBinary is meaningless to non-Macintosh computers. If a DOS-user downloads a MacBinary file, his computer may become profoundly confused. When we talked about settings, we noted that one setting nearly every terminal program asks you to supply is the creator of received TEXT files. The MacBinary file header normally contains a creator code that’s restored to a file while it’s received, but non-MacBinary TEXT files have no such code, and all Macintosh files, TEXT files included, like to have a creator code of one sort or another, preferably the creator code of a word-processor or text processor.
Incidentally, although we may be inclined to think of TEXT files as consisting of nothing but pure, unvarnished data, some TEXT files do have resource forks. The McSipk and BBEdit text processors, as well as those that come along with THINK C, Mac Pascal and MPW, add a small resource fork to their TEXT files in which they store such parameters as font, font size, tab settings, window placement, and file size. When TEXT files are sent or received in MacBinary format, their resource forks and the parameters they contain are preserved intact. But when TEXT files aren't MacBinarized before they're sent, their resource forks and the contents therein are lost entirely. Also lost are the file's original name and its creation and modification dates. What that means is that there may be times when you want to override your terminal program's default settings, which invariably boils down to "Use MacBinary for everything except TEXT files," and invoke the MacBinary format for TEXT files too. But if you go ahead and do that, remember that MacBinary format is meaningful only to other Macintosh computers. It will leave other computers addled and disoriented.

There are still other occasions when you may want to override your terminal program's MacBinary default settings. Let's say you have a business associate named J.P. Morgan who uses an IBM-compatible computer, and from time to time you and J.P. need to exchange spreadsheets. J.P. uses the DOS version of Lotus 1-2-3 for doing spreadsheet work and you use Microsoft Excel. At first glance, there's no major compatibility problem, because Excel can open 1-2-3 documents and is also able to save its own documents in 1-2-3 format. There's also no particularly serious telecommunications problem when J.P. sends you a 1-2-3 spreadsheet. It's unlikely that J.P. will be able to convert the 1-2-3 spreadsheets into MacBinary format before sending them along, but not to worry. Yes, even though J.P.'s files are really binary, your terminal program won't recognize them as MacBinary and will therefore save them as text files. As long as you open them from within Excel and don't try double-clicking on their icons, everything should work out fine. Problems could arise, however, when you send your spreadsheets to J.P. Although you've done J.P. the service of saving your Excel spreadsheets in 1-2-3 format, they remain binary files. And when you send any binary file with your terminal program, your default settings will normally instruct your program to put the file into MacBinary format before sending it. That, in turn, will cause major problems for poor J.P.'s PC-compatible computer, which, even
on the best of days, would probably be unable to tell a MacBinary file from a bitmapped image of a hole in the wall. Consequently, to further accommodate old J.P., you'd have to override your default settings and not send the file in MacBinary format despite the fact that it's a binary file.

A general rule is that TEXT and non-Macintosh files are sent without MacBinarization while all others are sent using MacBinary. A second general rule is that the first general rule sometimes needs to be broken.

BTW, the original MacBinary format evolved in 1987 to MacBinary II, but terminal programs are generally pretty good at figuring out which is which without bothering you about it.

It's also worth re-stating that in non-MacBinary file transfers, a file's resource fork and Finder information will be lost. That means that when you receive any non-MacBinary file you won't get its creation and modification dates and you'll also be asked to name it. That's because there's no finder information and hence no "native" file name.

![File Transfer Options](file transfer options.png)

**Figure 8.3.** White Knight's File Transfer Options window lets you tell the program when to use MacBinary and how to handle files that aren't MacBinary.
Compression Formats

When a service makes files available for downloading, it usually has them ready on a hard disk. You request a file; it sends the file from its hard disk to your computer. If the service maintains a file library of any size, that translates into significant storage space on disk. If you *download* a file of any size, that translates into a long phone call for you and a long time during which no one else can use the BBS phone line that's feeding you the file.

To reduce both the disk space needed to store them and the transmission time needed to send them, services ordinarily *compress* their files. The compression I'm talking about here is software-based, and it takes place quite apart from any hardware-based compression that your modem, or a service's modem, might be capable of. When you download a compressed file, you have to decompress it with the appropriate utility before you can use it. However, the transmission time saved by transferring files in compressed format usually well exceeds the extra time it takes to decompress the file. Compression programs are becoming more and more efficient all the time, and savings of over 50 percent—sometimes as much as 90 percent—are not uncommon.

Compressed Macintosh files are sometimes called *archives*. They can usually be identified by the suffixes at the end of their names. On the next page, then, are the suffixes associated with the best-known Macintosh compression utilities and some other information worth knowing.
### Table 8.1. The Most Common Macintosh File Compression Formats

<table>
<thead>
<tr>
<th>Suffix</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>.SEA</td>
<td>This stands for self-extracting archive. If you don’t have any decompression utilities, this is the suffix you should look for first. .SEA files come already bundled with the program code that’s needed to decompress them, which means you aren’t required to have any decompression utilities on hand. Technically, self-extracting archives are applications with smallish resource forks (mainly the decompression code) and large-ish data forks (the compressed data). When you double-click on a self-extracting archive, the compression code extracts the original file from the archive and writes it to your disk. There are a number of file compression utilities (sometimes called archiving utilities) that can create self-extracting archives, but when all you’re doing is decompressing downloaded files, you really don’t have to care a whole lot about what program created the archive.</td>
</tr>
<tr>
<td>.CPT</td>
<td>This suffix identifies files in Compactor or Compact Pro format. You can decompress these files with either of a pair of freeware utilities: Extractor and Stufflt Expander. You can also decompress them with Compact Pro itself, which is shareware. Compact Pro used to be called Compactor, but I suggest that you not download Compactor if you see it listed in some service’s file library. The formats of .CPT files have changed slightly over the course of time, and while Compact Pro can extract files that were compressed by Compactor, Compactor will be flummoxed by files compressed by Compact Pro.</td>
</tr>
<tr>
<td>.SIT</td>
<td>This identifies archives created either by Stufflt Classic, which is a shareware program, or by Stufflt Deluxe, which is distributed commercially by Aladdin Systems, Inc. Both those programs can decompress .SIT files. The freeware Stufflt Expander can do it too. In its many incarnations, Stufflt has probably become the most popular compression program among Macintosh telecommunicators.</td>
</tr>
<tr>
<td>.DLX</td>
<td>This suffix is sometimes used to identify files compressed specifically with Stufflt Deluxe. Stufflt Classic and Stufflt Expander can decompress them, too.</td>
</tr>
<tr>
<td>.PIT</td>
<td>This identifies Packlt files. Some years ago Packlt was the reigning Macintosh compression program. It’s very seldom used today, but there are still a fair number of old Packlt files around, and one day you may encounter one that piques your fancy. You can unpack Packlt files with the Packlt utility itself (it’s shareware), but because Stufflt, either Classic or Deluxe, also decompresses .PIT files, there’s not much point in keeping Packlt around. To unpack Packlt files with Stufflt, choose UnPack... from Stufflt’s Other menu.</td>
</tr>
</tbody>
</table>
This identifies files compressed by DiskDoubler, a commercial compression program distributed by Salient Software. Naturally DiskDoubler itself can decompress .DD files, but for decompression only Salient offers a freeware utility, DDExpand, through BBSs and online services.

Files with this suffix were produced by Apple's own DART utility, which creates bit-for-bit images of floppy disks. The image files are compressed and written to a hard disk. Then, if you ever want to make an exact reproduction of the original floppy, DART will decompress the .DART file and transfer the disk image to your floppy. When Apple distributes System software or such felicities as System 7.0 Tune-up through BBSs and online services, it usually posts the material as floppy-disk images. DART itself is available as freeware from BBSs and online services.

Files with this suffix are also floppy-disk images, but they were created with another Apple utility, also freeware, called DiskCopy. Unlike DART, DiskCopy doesn't compress the disk images it creates (although you can certainly go ahead and compress them with any good compression program). On the other hand, DiskCopy is usually packaged with MountImage, a Control Panel that mounts DiskCopy files just as though you'd inserted the original disk into your floppy drive. And MountImage v.1.2b1 or later lets you mount up to eight disk images at once, which is much more convenient than having eight floppy drives, and which also lets you do extremely fast System installations without performing a single floppy swap.

Now let's deal with all this compression stuff pragmatically. If you have a modem and all you expect to do is decompress the files you've downloaded, you don't have to buy a thing. BBSs and online services generally limit the files in their Macintosh libraries to one or two compression formats, and, in nearly every case, the allowable formats will be those of StuffIt and Compact Pro. And since StuffIt Expander \(a\) can decompress files in either format, \(b\) is widely available on BBSs and online services, and \(c\) also happens to be freeware, it takes care of the bulk of your decompression needs. What's more, like most other freeware and shareware decompression utilities, it's always posted in file libraries as a self-extracting archive, which means that, even though the file arrives compressed, it will be more than happy to decompress itself upon double-clicking.
Later, as you begin to download larger numbers of files (and maybe even Apple system software updates), you may want to acquire copies of DDExpand, DART, and DiskCopy. If you like downloading years-old Mac files just to see how Macintoshing looked in bygone days, download the shareware version of StuffIt (that's StuffIt Classic) so you can decompress PackIt files and de-hex BinHex files.

Eventually you'll also want to be able to compress files yourself. Compact Pro is ideal for that: its shareware fee is very reasonable, it offers very tight compression, and it gives you the option of creating self-extracting archives in the bargain. Stuffit Classic, whose shareware fee is also very reasonable, has many of the same features. Still later, when you become a monomaniacal compressor, you'll probably want to buy StuffIt Deluxe, which comes along with compression optimizers that tailor file compression to the type of file that's being squished together, offers a wide variety of compression trade-offs between speed and tightness, and even allows encryption of files so they can be opened only with a password.

It's safe to say that StuffIt and Compact Pro are currently the most-used Macintosh compression/decompression programs, and even though their interfaces aren't all that complicated, we shouldn't end this chapter without having a look at their main windows, shown in Figure 8.4.

Each window displays the contents of an archive, which is a single Macintosh file that contains other files in compressed form. Both archives were created by compressing the shareware terminal program Termulator along with the ancillary files that accompanied its shareware distribution. Notice from the amount of disk space saved that Compact Pro and Stuffit Classic were just about equally adroit at compressing the files. Note too that each file is listed with its type and creator code. Applications are always of type APPL. Documentation for the program was distributed in both MacWrite and Microsoft Word formats. You can see what the type and creator codes for those files are. The two additional files are Termulator documents and therefore have the same creator code as Termulator itself. When you extract (i.e., decompress) files from these archives, you can do it one file at a time or all at once. After extraction the archive remains intact, so if you lose or trash a file you've just extracted, you can go back and extract it all over again. The tool bar below the StuffIt Classic window provides an alternative interface to the program's menus.
Compact Pro always works in the background, so under MultiFinder or System 7.0 you can be decompressing one archive while you’re downloading still more files with your terminal program. Decompressing in the background does slow things down a bit (sometimes more than a bit) so Stufflt lets you choose between allowing background tasks or not. You select “Preferences” from Stufflt’s Edit menu and use the “Allow Background Tasks” button.

One day, when you’re comfortable with your modem and you’re also feeling expansive about phone bills, you may very well decide that your telecom experience so far has focused too parochially on matters Macintosh. You’ll dredge up some interesting-sounding DOS-based BBS or navigate over to the PC areas of your online service and start to poke around the file libraries. Suddenly you’ll come across a file whose description promises some information you’ve been trying to track down for years. Only two things

Figure 8.4. Windows from Compact Pro and Stufflt Classic respectively.
stand between you and instant download: the file name ends in .ZIP, an alien suffix, and the file description points to its having been created by some unfamiliar entity called XYWrite. Could that possibly mean that it's an IBM-compatible word processor document that's been archived with a non-Macintosh compression scheme? That's just what it means. Must you therefore pass it up, delicious-looking though it appears? Not at all. The proliferation of file formats and the alchemy that transmutes one format into another make up one of computing's truly never-ending stories. A few chapters hence we'll plunge into the dreaded domain of DOS.
If you have a Mac, a modem, a phone line, and anything more than a passing interest in computing, the presence of thousands upon thousands of Macintosh files in the file libraries of amateur bulletin board services and the commercial online services may turn out to be an enticement you simply can’t continue to ignore. Nearly every new day brings a fresh handful of shareware utilities that promise to make the hours you spend in front of your computer significantly more productive, and some of those utilities even duplicate the functions of commercial programs costing several times the shareware fee. More and more free upgrades to commercial software are appearing in file libraries too, and if you own any of the software yourself, downloading the upgrades can be a fast and easy way to keep your precious programs up-to-date. Also, enough computer games are available for downloading so that it may become feasible to postpone buying your kid the Super Nintendo or Sega Genesis he craves—provided, of course, that you permit him or her to go anywhere near your computer.

But our subject is telecommunications, and those very same file libraries are rife with extremely useful telecom software: not just terminal emulation
programs of remarkably high quality but compression and decompression programs, file massagers of every type and stripe, lists of BBSs old and new, almost anything you could ever want to stash away on your hard disk to facilitate the telecommunications process. Sure, some of those files are duds, but others verge on being patently essential. We’ve discussed and even praised some of them already, and we’ll talk about still more of them soon enough. Right now let’s begin to put our hands on some.

**Basic Downloading Mechanics**

Say you’ve just called CompuServe, and in the course of bopping around that service’s labyrinthine highways and byways you decide to see what’s currently to be had in the way of interesting software doodads for the Mac. Many of the best new trifles make their maiden appearances in CompuServe’s Macintosh System Forum.

At the next prompt from CompuServe, regardless of what it is, we type:

```
   go macsys
```

and receive this reply:

```
Welcome to Macintosh Systems Forum, V. 2G(41)
Hello, Steve Taylor
Last visit: 08-May-92 15:12:33
Forum messages: 125364 to 126373
Last message you've read: 114399
Macintosh Systems Forum Menu
  1 INSTRUCTIONS
  2 MESSAGES
  3 LIBRARIES (Files)
  4 CONFERENCING (0 participating)
  5 ANNOUNCEMENTS from sysop
  6 MEMBER directory
  7 OPTIONS for this forum
Enter choice!
```
It's the file libraries we want to look at, so we type a 3 and get:

Macintosh Systems Forum Libraries Menu
0 UPLOAD HERE & Help
2 Using System/Finder
3 INITs/cdevs
4 DAs/FKEYs
5 Fonts
6 Utilities
7 System Hardware
8 PowerBook/Portable
9 Disks/Storage
10 Printers/Output
11 Monitors
12 Other Hardware
13 Early Macs
14 Macintosh II Apps.
15 System Conflicts
Enter choice!

Let's be clear that what I'm about to say is very much a matter of opinion, but we've declared we're after software doodads, and for my money the most interesting ones being written by amateurs these days are INITs and cdevs (or, if you like, System Extensions and Control Panels). So we proceed to type a 3 again. CompuServe responds with:

Macintosh Systems Forum Library 3
INITs/cdevs
1 BROWSE Files
2 DIRECTORY of Files
3 UPLOAD a File (FREE)
4 DOWNLOAD a file to your Computer
5 LIBRARIES
Enter choice!
Incessantly curious folk that we are, we opt for browsing and type a 1, after which we can dispense with the next three annoying CompuServe prompts by typing carriage returns:

Enter libraries (e.g. 1,2,4 or ALL)
or <CR> for current library:
Enter keywords (e.g. modem)
or <CR> for all:
Oldest files in days
or <CR> for all:

Then, one by one, starting with files posted most recently and going all the way back to the earliest primitive INITs and cdevs still moldering in CompuServe’s vast software cellars, the service lists all the files in the chosen library. Here’s a sample listing:

[71052,1562]
TOMINI.CPT/Bin Bytes: 33280, Count: 649, 09-Apr-92(08-May-92)
Title : TOMINIT II 1.0b4
Keywords: TOM TEAR OFF MENUS COOL TOT TEAR-OFFS TOMINIT is reborn! TOMINIT II allows menus to be torn off in any application. Can tear hierarchials as well. Great for large monitors. Save yourself from those long trips to the menubar. Tear off titles (TOTs) as well, miniature tear off menus. TOMINIT II works with all system configurations above 6.0. TOMINIT II has been heavily tested with System 7.0.1 and System 7 Tune-Up. This version makes major internal changes needed to stability, adds new cool color icons & fixes billions of bugs. Enjoy.
Dearchive with EXTRAC.BIN
Press <CR> for next or type CHOICES!

For now, don't concern yourself with the relative merits of TOMINIT II 1.0b4, which does happen to be a useful little control panel. Instead, let’s see what we can learn about it that’s directly relevant to telecommunications. First, it seems to have both a name and a title. The title TOMINIT II 1.0b4 indicates that even though the program's author avows that his program has been
heavily tested, he still doesn’t feel it’s a truly finished product. The “b” in its
version number means it’s still in beta testing, usually the last stage of testing
before nominal completion. CompuServe’s name for the file is TOMINI.CPT/
Bin, indicating (as readers of the last chapter will quickly recognize) that it’s
in MacBinary format (the Bin suffix) and that it’s been compressed with
Compact Pro (the .CPT suffix). The file description goes on to add what you
already know: that Extractor (available on CompuServe’s library as EXTRAC.BIN)
will de-archive (i.e., decompress) the file. You can also see how many people
have downloaded the file so far (that’s the “count”), the date on which it was
posted in the library and (very important) its size in bytes. Some of the files
in CompuServe’s libraries run to half a megabyte and more, which can make
for long, expensive downloads. At 33,280 bytes, TOMINIT can be said to be
petite.

All that being so, let’s take a chance on TOMINIT. We could type the whole
word CHOICES as CompuServe suggests, but the service doesn’t really care if
you respond in upper- or lowercase, and ch will be enough to convey a sense
of what you want. So we type ch and get:

Macintosh Systems Forum Library Disposition
1 READ this file
2 DOWNLOAD this file
3 DESCRIPTION
4 RETURN to library menu
Enter choice or <CR> for next !

and then we type 2 to download the file. CompuServe responds with:

Library Protocol Menu
Transfer protocols available -
1 XMODEM
2 CompuServe B+ and original B
3 DC2/DC4 (Capture)
4 YMODEM
5 CompuServe QB (B w/send ahead)
6 Kermit
0 Abort transfer request
Enter choice !
The first six options are all file-transfer protocols. During any session—a session being the time from logon to logoff—CompuServe will ask only once what protocol we want to use for downloading. If we download some more files during the same session it will assume we're using the same protocol we used the first time. Which protocol do we use? Well, the choice depends on a slew of things. We certainly wouldn't want to choose Kermit if our terminal program doesn't support it. (Many don't!) CompuServe B+, original B, and QB are all proprietary to CompuServe itself, and some terminal programs don't support those either. I've said before that ZModem is my all-around protocol of choice, but that one isn't supported by CompuServe. So for now let's pick reliable old (literally old, reasonably reliable) XModem, which almost every Macintosh terminal program can handle easily. We type a 1, upon which CompuServe informs us:

File TOMINI.CPT, 33280 Bytes, Lib 3
Starting XMODEM send.
Please initiate XMODEM receive
and press <CR> when the transfer is complete.
At this point we instruct the terminal program to begin receiving XModem. ZTerm lets you initiate XModem receives in two slightly different ways:

![ZTerm's two ways of initiating an XModem download](image)

**Figure 9.1.** ZTerm's two ways of initiating an XModem download. XModem joins the other protocols on ZTerm's Receive Files submenu, but owing to its senior status I've also given it a menu entry of its own. ZTerm lets you select a protocol, such as XModem, to appear on the File menu.

White Knight's receive commands are also under its File Menu. In Smartcom II you invoke a receive by selecting a button...

![Receive](image)

which in turn gets you a highly customized version of the Mac's Standard Files Dialog Box, shown in Figure 9.2 on the next page.
MicroPhone II’s “Receive...” command is under its Transfer menu, and when you choose it you also get a version of the Macintosh SF Dialog Box, shown in Figure 9.3.
When we tell our terminal program to start receiving, CompuServe begins to query us for an XModem acknowledgment, but it's up to our terminal program to do the acknowledging. Once the program goes into its XModem receive mode, it and CompuServe go back and forth a couple of times in XModem-ese, and after they're satisfied that they're both talking the same language and have negotiated the terms of the transfer, the file is transmitted, packet by packet (i.e., block by block, chunk of bytes by chunk of bytes) from CompuServe's system to our computer's hard disk.

During the transfer most terminal programs maintain a status window on the screen. This is White Knight's:

![File Transfer Status Window](image)

**Figure 9.4. The file transfer status window from White Knight.**

As we examine the window, we see that despite whatever names or titles CompuServe may list for the file, White Knight will extract its original name, TOM INIT II 1.0b4.cpt, from the MacBinary header and assign that name to the file. We can also see at any moment how much of the transfer is complete in both numeric form and as a wedge of pie. And because our terminal program knows how much data has come in since the transfer began, it can compute the current transfer rate in characters per second (total characters received so far divided by the number of seconds since the transfer began). Once it has a transfer rate, it can also estimate how long the whole transfer

---

**Chapter Nine File Transfers II: Protocols and the Mechanics**

157
will take. The size of the entire file comes in at the beginning of the transfer as part of the file's MacBinary header. Subtract the number of bytes received from that, and you know how many bytes to go. Divide the number of bytes to go by the transfer rate in bytes per second, and you get the number of seconds to go. Of course, that's assuming the whole transfer proceeds at the same rate that's prevailed so far.

File transfer status windows also typically show the efficiency of the file transfer under way. To understand what's meant by transfer efficiency, suppose you've made a 2400 bps connection with a service. That's 2400 bits per second. A character—or byte if you prefer—normally consists of eight bits plus start and stop bit. That's 10 bits per character. Therefore, 2400 bits per second translates into 240 characters per second, so if you've received 240 characters for every second since the transfer first began you'd be transferring at 100 percent efficiency. XModem has more overhead than some of the newer protocols, so the efficiency of XModem transfers tends to hover around the low 90 percent area.

**BTW,** if XModem transfer efficiency is in the low 90s, how come Figure 9.4 shows an efficiency of only 23 percent? Well, here's the way it is: When I called up CompuServe, I cheated. Yes, I know I said we'd hold back on MNP, V.42bis and the like for now, but I snuck in an MNP Level 5 connection in the vain hope that CompuServe might finally have started to hardware-compress its outgoing data. We've talked about in-modem hardware-based compression before, and we noted that if you have a 2400 bps connection in which there's a chance you may be receiving hardware-compressed data, you had better set your serial port to a speed much higher than 2400 bps. Data may come into your modem from your phone line at 2400 bps, but 2400 compressed bits may optimistically turn into as many as 9600 bits after your modem finishes decompressing them. That being so, even though I connected at 2400 bps, my serial port was set at 9600 bps, and it's the latter figure that White Knight used to perform its efficiency calculation. (220 cps divided by 960 bps gives 23 percent.) If my serial port had been set for 2400 bps, the window would show a more accurate rate. Look at ZTerm's transfer status window in Figure 9.5. I performed that transfer at a no-frills 2400 bps, and the 82 percent efficiency is right
in line with GEnie's normally anemic sending speed during its peak hours between 6:00 and 8:00 p.m.

Transfer Errors

If you examine the transfer status windows of most terminal programs you'll probably see some mention, either tacit or explicit, of transmission errors. In Figure 9.4 you can see that White Knight keeps track of the cumulative number of transmission errors during a file transfer. ZTerm does the same thing in gentler terms by keeping track of "retries" (see Figure 9.5). However your program chooses to put it, what it's trying to tell you is that a number of packets of information didn't arrive intact when they were first received. How did it know there were errors? Keep reading. Why were there errors? It could be any number of things. The most likely culprit is noise on the phone lines, spurious pulses or signals picked up anywhere along the route between you and the service. Noise could also have been generated within either of the communicating modems, or maybe you picked up your phone during the transmission and coughed or cursed or simply slammed down the phone hard.

Whatever the causes, don't jump to any dire conclusions. Just because there were transmission errors during a transfer, you don't have to worry
about the integrity of the file you received. Error-checking protocols normally allow both terminals participating in the transfer to talk to each other while the transfer is under way. If the receiving terminal detects an error in a packet (i.e., a block of data) that's come in, it can ask the sending terminal to retransmit it. Each of those retransmissions is what ZTerm refers to as a retry and what White Knight somewhat more alarmingly calls a transmission error. No matter what your program calls them, as long as the transfer concludes in normal fashion, the file you received should correspond bit for bit with the file that was sent, regardless of how many errors occurred during the transfer. The only time a file won't arrive intact is when some packet of data in the file refuses to come through without errors even after repeated retransmission. When that happens—and it's very rare that it does—the sender or your own terminal program will abort the transmission. So either you get the file in error-free form or you don't get it at all.

Error-Checking File-Transfer Protocols

We just went through a CompuServe download, so now for variety's sake let's try one on GENie. When you tell GENie you want to download a file it prompts:

Enter download request or <H>elp?

We type in a file number—actually what I normally do is paste a file number into my terminal window after copying it from somewhere else. In this case, let's pick GENie file 23093 by typing 23093. We could also have typed (or pasted, or even autotyped, Lord knows) the numbers of several files at that prompt, in which case our reply to the prompt might have looked something like: 23096,23098,23103,23114. GENie, along with some other services, will let you download files in batches, so if you already know exactly which files you want, you're spared going through the same rigamarole with each file. For now, though, let's stay with a simple one-file request. We enter a single file number and GENie then proceeds to tell us more about the file and asks us if we wish to go ahead with the download:

**************************************************************************
Number: 23093 Name: STUFFIT LITE PUBLIC BETA
Address: ALADDINSYS Date: 920504
Approximate # of bytes: 352128
Number of Accesses: 98  Library: 25
Description:
StuffIt Lite 3.0 will ship later this year as the upgrade
to StuffIt Classic 1.6, our current shareware compression
product. Many of our users have expressed an interest in
seeing StuffIt Lite prior to shipping. Attached is a
public beta release of the StuffIt Lite 3.0 application and
some of the extras that will ship with StuffIt Lite 3.0.
This package is being distributed by Aladdin Systems, Inc.
and Raymond Lau. This package must be distributed in its
entirety.
Keywords: StuffIt, Lite, 3.0, Compression, Fast, Aladdin,
Ray Lau
---------------------
File: STUFFIT LITE PUBLIC BETA
is a BINARY File.
Press <RETURN> to skip, <D>ownload, <L>ist, or <Q>uit.
?

We type a d, confirming that we want the file, and now GENie asks us to make
a choice:

Select Download Protocol
1. XMODEM
2. XMODEM (w/1K blocks)
3. YMODEM
4. ZMODEM
Enter # or <RETURN> to skip?

You'll notice that GENie's online protocol menu isn't vastly different from
CompuServe's. The services offer two protocols in common (XModem and
YModem), and the most notable difference is that only GENie supports
ZModem while only CompuServe supports its own proprietary B, B+, and QB
protocols. Just about all the other services you're likely to connect with will
have online protocol menus of their own, so let's run down the best-known
protocols to help ensure that the choices you make are good ones.
Developed by Ward Christensen in 1977, XModem was the first important error-checking protocol intended specifically for small computers. It's considerably less popular than it used to be back when its name was virtually synonymous with error checking, but it continues to be supported by more services and more terminal programs than any of the other protocols. In its original form, it calls for files to be divided into 128-byte packets. The data in each packet, regardless of what it signifies, is literally added all together by the sender, and the resulting *checksum* is sent along with the packet. When the receiving terminal gets the packet it recomputes the checksum. If the sender's checksum is the same as the receiver's, the packet is deemed to have been transferred intact. If the receiver's checksum is different from the sender's, the receiver requests that the sender retransmit the packet.

Original XModem worked well enough, but it wasn't perfect. Its main flaw was its checksumming scheme, which every now and then would let a bad packet slip by. Soon an enhanced error-checking procedure called CRC (for *cyclic redundancy checking*) was added to the XModem protocol as an optional alternative. CRC is substantially more reliable than checksumming, and most Macintosh terminal programs give you the choice of using it or not. In most cases programs can be set to try CRC first and then fall back to checksumming if CRC isn't supported by the remote system. These days, however, nearly all remote services use CRC when they send XModem, so in practice the fallback seldom takes place.

One thing to remember about XModem is that in order to function properly it requires that your communications parameters be set to no parity, eight data bits, and one stop bit.

**XModem-1K**

Another of original XModem's drawbacks was its speed—more accurately its lack thereof. After every 128-byte packet is transmitted, each of the two communicating parties has to stop and chat about what's just taken place. The receiving system has to tell the sending system either that the current packet has come in with no errors and that therefore it's okay to send the next one (known in the trade as a *positive acknowledgment* or *positive ACK*) or that the current packet has arrived with errors and needs to be retransmitted.
These handshakes are transparent to the user, but the extra overhead created by all that crosstalk caused file transfers to progress in sluggish fits and starts. Why was each packet only 128 bytes long? Noise. Not only were older modems inefficient at squelching normal phone-line noise, they also contributed additional noise of their own. If packets were made much larger than 128 bytes, the probability of their arriving without errors plummeted sharply. And retransmitting 128 bytes is slow enough. Retransmitting eight times that because one measly bit gets lost, and doing it over and over again during a single file transfer, is out-and-out impractical.

Things are different now. Noise-suppression circuitry has become much more sophisticated, and phone lines themselves are a little cleaner. All that translates into the far greater likelihood that a 1K packet (actually 1024 bytes) will make the crossing with all its bits in the right place. Ergo XModem 1K, which is basically XModem CRC with larger packets. Since 1,024 is 8 times 128, the communicating systems have to perform only an eighth as many handshakes during a file transfer as they do with original XModem, and even on phone lines that are only fairly clean transfers become appreciably zippier.

**YModem**

YModem sometimes means slightly different things to different systems. True YModem allows for batch transfers—the transfer of several files with only one request. When we looked at GEnie's way of prompting you for downloads, we noted that you could specify the names or numbers of a batch of files, each file name/number separated by a comma. If that's what we had done, GEnie wouldn't have offered XModem as an option, but it would have let us use the YModem protocol. True YModem is a variant of XModem 1K that made its debut in a telecom program called YAM (for “Yet Another Modem”) written by Chuck Forsberg. In a YModem file transmission, a file's name, size and modification date are sent in an extra block of data that goes out just before the file itself. This allows files to be transmitted in batches, and it's why the protocol is sometimes called YModem Batch.

One complication—really just a minor one—is that some systems choose to differentiate between YModem and YModem Batch. When you encounter such a system, you're usually safe in assuming that YModem without the batch is nothing more than XModem 1K, while YModem Batch is what your
terminal program understands to be YModem. And yes, if you're using YModem Batch, it's perfectly reasonable for a batch to consist of only one file.

YModem-g

YModem-g is what's known as a *streaming* protocol. Very limited handshaking, many fewer fits and starts. It's a variant of YModem, and it's the fastest protocol I've used so far on a Macintosh, ever so slightly faster than ZModem. YModem-g allows for the transfer of files as a continuous stream of 1K packets, and after a packet goes out it doesn't expect any acknowledgment from the receiver about whether the packet came across intact or not. Essentially the only crosstalk between the sending and receiving systems that can take place during a YModem-g transfer is a request from the receiver that the transfer be aborted. Which, by the way, is exactly what the receiving system does if it detects a CRC error, as there's no provision for retransmission of packets.

Why would you use YModem-g if it can't resend bad packets? Well, if you know your phone connection is immaculately clean you might consider giving it a try. Sometimes person-to-person connections that use newish local lines are pleasantly uncontaminated by noise, and in those circumstances it may turn out that YModem-g aborts infrequently enough to make it practical. But by far and away the principal use for YModem-g is with reliable connections—MNP Class 4 or V.42 connections in which the participating modems are performing hardware-based error checking. In that case software error checking is redundant anyway. If the remote service and your terminal program both support YModem-g and the remote modem and your own modem both support MNP-4 or V.42, YModem-g is a fast and sensible way to go.

Incidentally, with many services you don't have to say anything about the "g" part of the protocol to the service. All you do is tell *them* to send YModem and ask your terminal program to receive YModem-g.

Kermit

Kermit is a once-popular protocol developed at Columbia University that you may never need to use. It was originally designed for file transfers between mainframes and minicomputers, and many of the best shareware telecom programs, ZTerm among them, don't even bother to support it. Only a very
few older mainframes at corporations and universities insist on Kermit, and I suggest that you avoid it unless you absolutely can't. Why? It's not particularly fast, it probably means buying a commercial telecom program that you may otherwise not need, and take a gander at some of the options:

![Kermit Options window](image)

*Figure 9.6. VersaTerm PRO's Kermit Options window. The corresponding window in White Knight is equally fraught with options. Refer to the appropriate manual when you're making your Kermit choices.*

As you can see, Kermit is almost a universe unto itself. I've tried it with CompuServe and made it work, but CompuServe's proprietary protocols are significantly more user-friendly.

**Sealink**

The Sealink protocol is yet another XModem variant designed for use with special purpose phone systems like packet-switching networks and satellite relays. You won't find too many terminal programs that support it, but if a remote service insists on it you can sometimes get away with setting your own terminal for XModem.
ZModem

Owing to a number of significant improvements over earlier protocols, ZModem is rapidly becoming the protocol of choice among telecommunicators around the country. It was developed privately for Telenet by Chuck Forsberg at Omen Technology, and basic ZModem soon came into the public domain. The protocol continues to be refined by its developers—the enhanced version is called ZModem-90—but thus far the refinements have remained proprietary. Even in its public domain form, however, ZModem directly addresses some of the flaws of the older protocols and also adds some unique new features of its own.

One thing that distinguishes ZModem right off the bat is its efficiency. Even though it provides for complete software-based error checking—16- or 32-bit CRCs check both data blocks and handshake messages—it’s capable of operating in full streaming mode just like YModem-g, with no time-consuming terminal-to-terminal crosstalk between packets. The big difference is that YModem-g doesn’t let the receiver request that a bad packet be resent. ZModem does. In other words, crosstalk is elective. It happens only when it’s needed.

Another of ZModem’s advantages is crash recovery. If a file transfer is aborted due to modem disconnection or because you accidentally clicked the cancel button in your transfer status window or the sender suddenly experiences a power loss or any number of other reasons, when the transfer is restarted it takes up where it left off. The receiving terminal saves the data that was received up until the crash, and once the transfer is reinitiated the receiving terminal can tell the sending terminal: “Hey, I already have 62 percent of that file, so just send me the last 38 percent.” At present ZModem is the only widely available protocol that can do that.

BTW, there will be times when even ZModem won’t be able to bail you out of a mid-transfer crash. If the receiver suddenly loses power or experiences certain kinds of system crashes, his terminal program may not be able to properly close the file in which the transferred data was being kept. When that occurs the receiver’s partial file is essentially lost forever, and the new transfer will have to begin from square one.

Another nice feature of ZModem is that when the sender initiates a transfer, the receiver doesn’t have to do anything at his end. If the receiver’s
terminal program correctly implements ZModem, it will recognize the character sequence that kicks off a file transmission and promptly go into ZModem receive mode. Many Mac terminal programs have a "Receive ZModem" command in one or another menu or window, but if all is going well it's a command you should never have to use.

Two more ZModem benefits. First, all file transfers are batch transfers. A single file is a batch of one. As with true YModem (i.e., YModem Batch) every file is sent with its name, size and the date and time at which it was last modified. Second, ZModem "quotes" the two XON/XOFF characters $S and $Q. That is, it successfully distinguishes between, on the one hand, the $S and $Q flow control characters intended to halt and restart the flow of data and, on the other hand, any $S or $Q characters that might turn up randomly as bit patterns in the data being transmitted. This makes it viable to use XON/XOFF flow control all the time, even if you have a high-speed modem that would ordinarily insist on hardware handshaking. As a consequence, it also makes it viable to perform high-speed file transfers without a hardware handshake cable.

Is all this a sales pitch for ZModem? In many ways it is. I'd certainly suggest that ZModem become your default protocol wherever you can use it. We've seen that GEnie supports it, as do Macintosh BBS host programs like Hermes and Second Sight. Most PC-based host programs also support it, and I strongly advise that you use a terminal program that supports ZModem, too.

**CompuServe B, B+, and Quick-B**

At this writing CompuServe still doesn't support ZModem, maybe because its own proprietary file-transfer protocol, CompuServe B+, is comparably efficient. We've been paying some attention to the genealogy of our protocols, so you may (or may not) be interested to know that CompuServe B+ was begat by CompuServe Quick-B, an interim protocol begat by the original CompuServe B. When you tell CompuServe you want to download a file and CompuServe lists its available protocols, I suggest you pick B+. Of course, that's assuming your terminal program supports the CompuServe proprietary protocols. (The commercial biggies all do; among the shareware terminal programs that also support ZModem, ZTerm, and Terminal support CompuServe's proprietary protocol as well.) Like ZModem, CompuServe B+ is
equipped to trigger auto-receives. In other words, your terminal program recognizes a sequence of characters sent by the transmitting terminal (the sequence is designed so that there's little likelihood you'd type it for any other reason) and shifts into receive mode without any direct intervention from you. Your terminal program does have to cooperate, and some Mac telecom programs that otherwise support CompuServe's protocols don't support the auto-receive portion. When they don't, you initiate a CompuServe B+ or Quick-B receive just as you'd initiate an XModem receive. Check with your program's documentation before you go online with CompuServe.

**Timeouts**

Sometimes a remote service is so busy that, quite apart from the protocol you're using, it can't send your file at the normal tempo. GENie and CompuServe have a great many members, and it can happen that during peak usage periods their computers simply can't keep up with the demand. Similarly, a friend who lives just around the corner may be sending you a file, but he may be running under System 7.0 or MultiFinder and some other application may be hogging so much CPU time that his terminal program takes an unusually long time to get at the next block of data in the file it's been asked to send. The resulting momentary pauses during file transfers are called *timeouts*, and most protocols contain rules governing how long to wait before resynchronizing the transmission—i.e., before the sending system gives up waiting for an acknowledgment and tries to resend a given block or the receiving system gives up waiting for a given block and tries to request a resend. For average use, a good day-to-day default setting for the timeout period is five seconds. Have a look at Figure 9.7 to see how to make (or inspect) the setting.

Because systems like GENie and CompuServe can occasionally introduce still longer delays during peak traffic periods, you may want to increase the timeout period to as much as 30 seconds for those services. (That's why terminal programs let you save different groups of settings; you often need to use a different ensemble of settings for each service that you call.)
More Transfer Options

Let's go ahead and use Figure 9.7 as a springboard for discussing some additional transfer options. When it's necessary to use XModem, unless you have information to the contrary start by going first class and using XModem 1K. These days services that don't support XModem's 1K block option are very much in the rare bird category. If you have a conspicuously noisy connection (your file-transfer status window will tell you that by indicating that an inordinate number of blocks have been retransmitted) you may want to drop back to XModem CRC with only 128-byte blocks. That way you won't be bogged down by the continuous retransmission of big 1K packets. As we've said, when most terminal programs are asked to use XModem CRC, they're capable of falling back to the classical checksum method, but if you know for sure that some service uses only checksumming you can set your program accordingly. As for White Knight's "supercharged" XModem, its really a way of obtaining the streaming benefits of YModem-g when you're not explicitly offered YModem. I suggest you don't try it unless you have a reliable connection (MNP-4 or V.42). One teeny little error will bomb out the whole transfer.
We've been saying "nowadays" a lot because terminal programs often try to accommodate telecommunications practices that are all but obsolete. For example, when you're sending files using the YModem protocol, White Knight has the ability to send 1K blocks only if requested. Nowadays YModem transfers almost always use 1K blocks. White Knight also makes the retention of partial ZModem receives an option. Most other programs do it automatically, and I suggest that if you use White Knight you always keep the button checked. And while we're on the subject, if you attempt to resume a previously aborted ZModem download, make sure the partial file is in your current receive folder. That's where terminal programs expect to find them, and if they don't find them there they normally begin the transmission from the beginning of the file.

Background File Transfers

If you're operating your Mac either under System 7.0 or under MultiFinder with System 6.x, there's certainly no reason to sit around twiddling your thumbs while you're transferring files. Even if you have no non-telecommunications-oriented tasks to do, you may at the very least want to de-archive some files you've just received or compress some files you're about to send—er, I mean to upload. Your terminal program will be able to take care of business even while it's running in the background, but there are a few considerations you ought to be aware of. We've mentioned some of them before, but this time let's focus specifically on protocol-governed file transfers.

First off, know that things like managing an active serial port and computing CRCs do require a certain amount of central processor time. If your Mac is very fast and your modem very slow, the time won't be terribly significant. But if the reverse is true your Mac may not have all that much time to spare for other tasks. When other programs let you specifically choose whether or not to permit background tasks to take place (the various StuffIts are prominent members of that category) please go ahead and let them do so. Certain other applications—the word processor Nisus comes to mind—hog so much foreground time that your terminal program's normal background functioning can slow to a crawl. You can guard against file transfer trouble by setting for long timeouts, but if you're sending a file to a friend, you won't be doing him any favors if you inadvertently reduce the transfer rate so sharply
that he has to wait forever for the file to come in. That's not to mention the fact that you'll be using more expensive telephone time (and, if you're downloading from a service, more expensive online time) to complete the transfer.

The bottom line is: pay attention! Check your transfer status box every once in a while to see how fast the transfer is progressing. Get a feeling for which background operations slow your terminal program down the most and which ones don't have much effect. And always remember Cardinal Rule Number One: No more than one terminal program open at the same time, unless you're absolutely positive that each one uses a different serial port from the moment it's opened!

**Uploading**

If you look back at Figure 9.1 on page 155, you'll see that ZTerm's send commands are grouped together with its receive commands. The same is true, give or take the odd discrepancy, with most other terminal programs. When you're uploading files to a remote service, its prompts will usually make it clear just what you have to do. Among the host programs we've talked about so far, Hermes has received the shortest shrift, so let's see now what a Hermes-hosted BBS sends when you tell it you want to upload a file. In the next example we'll be uploading the DA ScrapIt to the MacsDelight BBS on Long Island. We've already told it we want to do that, and Hermes asks for confirmation:

```plaintext
Upload 'ScrapItDA1.3.sit' to DA's?
```

When we type a `y` Hermes goes ahead and types "es," so on screen the line looks like:

```plaintext
Upload 'ScrapItDA1.3.sit' to DA's? Yes
```

Then Hermes says:

```plaintext
Please enter a one line description.
```

...and we say:

```plaintext
Update to ScrapIt DA 1.1. Best shareware Scrapbook replacement.
```
Hermes then asks:

```
Protocol (?=list):
```

...and we proceed to type ? (a question mark) whereupon it gives us:

```
Q: Abort Transfer(s)
0: Don't Transfer
1: ASCII
2: Z-Modem
3: Z-Modem Batch
4: Xmodem
5: Xmodem 1K
Protocol (?=list):
```

We type 2 (although we could have chosen any of the other protocols as long as our terminal program supported them) and now Hermes comes back with:

```
Zmodem v1.0.1 ^X to abort:
**80100000023be50l
```

As you can see, Hermes offers a last chance to cancel the transfer by typing ^X. The next line of characters is actually a string of ZModem control characters that were sent to the screen. You'll often see a line of them when you perform ZModem transfers yourself. They're intended for our terminal program and not for us, so they don't require that we take any action. All we have to do now is instruct our program to send ZModem (or whatever other protocol we've chosen). When you ask your terminal program to send a file, you'll get its version of the Mac's Standard Files Dialog Box, and from it you'll select the file or files to be sent. Different programs do have different ways of piling up batches of files for batch transmission, so that's one area in which it's wise to read your docs or your manual as the case may be. For ZModem sends, ZTerm does it simply and smoothly, as shown in Figure 9.8.

When the Standard Files window first comes up, the Start button is grayed out because there are as yet no files in the batch-to-be. As soon as you select a file and click on the Add button, the Start button becomes active. You can now add as many files as you like to the batch. When you're done, you click on Start, and ZTerm begins sending the files. From that point on it's all in ZTerm's hands. You either watch the whole thing happen or you do something else in the background until the program beeps you to let you know the transfer is over.
Sending Files Person-to-Person

Uploading to a friend's computer isn't so very different from uploading to a remote service. The main difference is that instead of being drawn from a fixed repertoire of prompts, the typing back and forth that usually precedes and follows file transfers will probably be improvised. When you're sending files to non-Macintosh computers, you'll no doubt be careful to warn your terminal program not to convert your files into MacBinary format unless they're being passed along to another Mac down the line. And this time you're the boss. You have to announce when it's time for the remote party to begin receiving, and on making the announcement you have to get ready to begin sending (except that ZModem receives, being auto receives, don't require that the receiver take any special action). It also falls to you to specify what protocols to use. And by now you ought to know enough to choose them halfway wisely.
If you've read Chapter 8, you've probably become—how shall we say?—sensitized to the multiplicity of forms that a downloaded file can take. Yes, your terminal program will transparently convert MacBinary files to their non-MacBinary, “true” Macintosh format. Even so, it's the rare downloaded file that you can use as is. Many of your downloaded files will have to be decompressed at the very least, and sometimes decompression turns out to be only the first step in the process of hammering a file into a form you can use. Table 8.1 in Chapter 8 lists the most popular Macintosh compression formats together with some programs that can handle them. But what if you want to download a file that wasn't compressed on a Mac? What if, for instance, you log onto a PC-oriented BBS and discover that one of the files available for downloading is your ex-lover's diary, but it's written in WordPerfect 5.1 and compressed with the DOS compression program PKARC? Does that mean you'll have to pass up the peekaboo chance of a lifetime?

Not in the least. The situation may be serious, but there are a number of ways to deal with it. So, starting with the simplest of file-massaging proce-
dures and working up to others that are fairly sophisticated, let’s examine some of the things you can do to make your downloaded files more tractable.

**Docs**

When you download an application from a service, it’s typically bundled with some sort of documentation file. Sometimes the docs are plain text (a file of type TEXT), but just as often the file format will be that of a common word processor. Word processor files permit the writer to scatter graphics—for example, pictures of Macintosh windows and menus—throughout the text, but plain-text files can contain text and nothing else. The acknowledged lingua franca for documentation files is MacWrite. Not MacWrite II, mind you, but original MacWrite. That’s because almost every common Macintosh word processing program can open MacWrite files. (Most can also save files in MacWrite format.) So my first suggestion is that if you plan to download applications, make sure to have either MacWrite itself or a MacWrite-compatible word processor on hand for opening doc files. Word, WriteNow, or MacWrite II will do just fine.

When you do open a MacWrite file with another application, don’t expect to be able to double-click on its icon. You’ll have to open the compatible word processor first and then, from within that word processor, open the MacWrite file. The word processor will proceed to convert the MacWrite file into its native format.

**BTW,** if you’re a Microsoft Word user and find yourself opening MacWrite files in any quantity, there’s a handy utility posted in the Mac libraries of many BBSs and online services. It’s a tiny program called MacWrite to Word, and it lets you double-click on MacWrite files just as though MacWrite itself were on your hard disk. You put it in the same folder as your Microsoft Word application, and when you double-click on a MacWrite document it opens MacWrite to Word, which in turn opens Microsoft Word and then causes Word to open and convert the MacWrite file. Once it's done all that, MacWrite to Word even has the decency to quit immediately to give you back the RAM it occupied.
If a documentation file is in plain-text format, you can open it with whatever general-purpose word processor you happen to have, although you may still have to open the file from within your application. One built-in feature of System 7.0 is that if you double-click on a plain-text file and don’t have the application that created it, the Finder gives you the option of opening the file with TeachText, Apple’s own mini text-processor. And almost everyone has about 12 redundant copies of TeachText lurking in the darker recesses of their hard drive.

Every now and again, however, you’ll come across a non-plain-text docs file whose format a) is that of some Macintosh word processor which you don’t own and b) is incompatible with the word processor you do own. Recently I’ve seen several such files in Word 4.0 and MacWrite II format, and there’s no guarantee that files with less-familiar pedigrees won’t come along as well.

What do you do then? You cope. You remember that Word 5.0 can open and convert MacWrite II files. Most word processors will usually let you open any file at all—at least the data fork of any file at all—if you hold down the Option key and select Open from the File menu. When you do this, the word processor doesn’t preserve the file’s original formatting and may display various “garbage characters” that correspond to formatting information embedded in the file. But most of the time it’s possible to edit out the garbage and retrieve the salient text.

Another approach to that situation is to buy a copy of MacLink Plus or at least the MacLink translators, which the latter can be used with Apple File Exchange. (Apple File Exchange is distributed with Apple’s copies of the System and can also be downloaded from CompuServe, GEnie, America Online, and many other BBSs.) The core purpose of MacLink Plus, which we’ll examine more closely later in this chapter, is to convert between Macintosh and non-Mac formats, but it’s equally proficient at Mac-to-Mac translations.

Text Files

*When in the Course of human Events, it becomes necessary for one People to dissolve the Political Bonds which have connected them with another, and to assume among the Powers of the Earth, the separate and equal Station to which the Laws of Nature and of Nature’s God entitle them, a decent Respect to the Opinions of Mankind requires that they should declare the Causes which impel them to the Separation.*
In the foregoing paragraph there's one carriage return after the final period and no invisible embedded control characters. If, however, you'd logged onto a commercial online service and downloaded a plain text file that consisted of those very words, the chances are quite good that the file would look a little different, probably more like this:

When in the Course of human Events, it becomes necessary for one People to dissolve the Political Bonds which have connected them with another, and to assume among the Powers of the Earth, the separate and equal Station to which the Laws of Nature and of Nature's God entitle them, a decent Respect to the Opinions of Mankind requires that they should declare the Causes which impel them to the Separation.

The slanted lines at the beginning of each line except the first are linefeeds (normally, they are little boxes but they are non-printable symbols so slanted lines were substituted), and there's also a carriage return after every line. What's going on? Well, what you're seeing is the universal text format observed by most services (for universal read IBM-compatible). One thing that's not universal about the format is line length. On that score it's generally user's choice, which means that when you open the file with your word processor it could easily look like this:

When in the Course of human Events, it becomes necessary for one People to dissolve the Political Bonds which have connected them with another, and to assume among the Powers of the Earth, the separate and equal Station to which the Laws of Nature and of Nature's God entitle them, a decent Respect to the Opinions of Mankind requires that they should declare the Causes which impel them to the Separation.

...which is pretty unsightly.

What to do? Well, one by one you could delete each of the linefeeds and replace each carriage return with a space, but for a document of any length that technique can be unacceptably tedious. You could speed things up by globally deleting the linefeeds in your word processor (by replacing the
linefeeds with nothing at all) and then globally replacing all the carriage returns with spaces. Of course, you'd end up with one big paragraph, which is fine if the document you're massaging is only one paragraph long, but not fine if it's not.

There is, however, one helpful convention in the universal text format—at least it's helpful to Mac users—provided the convention is carefully observed. It's that true paragraphs are normally separated by a pair of carriage returns, so that the incoming document might look like this if it contained more than just a single paragraph:

```
When in the Course of human Events, it becomes
/necessary for one People to dissolve the Political
/Bonds which have connected them with another, and to
/assume among the Powers of the Earth, the separate and
/equal Station to which the Laws of Nature and of
/Nature's God entitle them, a decent Respect to the
/Opinions of Mankind requires that they should declare
/the Causes which impel them to the Separation.
/
/Mind you, this isn't to say that we're ready to
/dissolve any Bonds right now, although we certainly
/are thinking about it.
/
```

In Microsoft Word, for instance, that convention lets you globally replace carriage returns with spaces without combining all the lines of the original text into one humongous paragraph. First, you replace all the linefeeds with nothing at all. (Don't worry about having to type a linefeed into Word's Find what: box, although you can do it easily enough by typing Control-j; all you need do is copy a specimen linefeed and paste it into the box.) Then, when you replace each carriage return with a space, check the Match Whole Word Only button before you go ahead and Replace All, as shown in Figure 10.1 on the next page.
When Match Whole Word Only is checked, only single carriage returns qualify for replacement with spaces, and after you click on Replace All, the resulting text comes out looking like this:

When in the Course of human Events, it becomes necessary for one People to dissolve the Political Bonds which have connected them with another, and to assume among the Powers of the Earth, the separate and equal Station to which the Laws of Nature and of Nature's God entitle them, a decent Respect to the Opinions of Mankind requires that they should declare the Causes which impel them to the Separation.

Mind you, this isn't to say that we're ready to dissolve any Bonds right now, although we certainly are thinking about it.

Now there's only one carriage return after each true paragraph, the linefeeds are all gone, and the text is ready to be formatted and/or printed according to your liking.

With all that said, there's still an easier way to do what we've just done. Download a copy of a shareware text processing DA called McSink. Its name derives from the fact that its text massaging features include everything but the kitchen sink. Countless Macintosh telecommunicators swear by it.
If I'd opened my sample text file with McSink instead of Word, I'd have been able to remove all the linefeeds with a single stroke of the mouse. All I'd have had to do is choose Strip Linefeeds from McSink's Convert menu. (See Figure 10.3 below.) Then, to get rid of the carriage returns after every line, I could have chosen Form Paragraphs from McSink's Format menu, whereupon, just about instantaneously, my text would be transformed into the two-paragraph format.
A further glance at Figure 10.3 will reveal that McSink can also add linefeeds and wrap text either to a specified length or the width of a window, which can come in handy if you're ever called upon to put some text of your own into the format that online services prefer. If you plan on using any of the online services, I strongly recommend that you get hold of McSink or its commercial version, Vantage.

BTW, another very useful shareware program that performs many of the same functions is called Add/Strip. Add/Strip is better at some things than McSink and vice versa. But any Macintosh telecommunicator worth his or her salt should have at least one of the two.

Working with DOS Files

In case you haven't heard, DOS is the all-but-universal operating system in the IBM-compatible world. Although plain-text files created in a DOS environment aren't all that different from Macintosh text files, non/plain-text files are likely to be a lot different. For one thing, if you download a DOS file, your terminal program will probably be alert enough not to do a MacBinary conversion, since the MacBinary format is next-to-meaningless outside a Macintosh setting. For another thing, compressed DOS files will almost certainly have been compressed with DOS compression programs, which means they won't be in familiar Macintosh compression formats like those of Stuffit and Compact Pro. And for a third thing, even when they're not compressed, DOS files simply aren't Macintosh files!

What it all comes down to is that you can almost always get to the data in DOS files, but in order to do it you generally have to work them over. Let's look at some of the strategies you might adopt.

Decompressing DOS Files

You'll remember from Chapter 8 that a file in compressed format is normally called an archive, and that an archive can contain several compressed files. Regardless of whether a compression utility is operating from a Macintosh or PC platform, it's almost certainly going to be capable of gathering up several files, compressing each of them, and then saving all of them in a single archive.
The most common DOS compression formats are ARC, ZIP, and LZH, and
DOS archives in those formats will have .ARC, .ZIP or .LZH file extensions
appended to the ends of their names. LZH is probably the least-frequently
used of the three, and at this writing I don't know of a good Macintosh utility
that can decompress LZH archives.

**BTW,** if you've downloaded an LZH archive and absolutely have to see
what's inside it, I suggest you take it to a friend who has a DOS-based
computer and decompress it there. If the friend has a modem, you can
send the file over. If the friend has a 3-1/2-inch floppy drive and you
have a Macintosh SuperDrive, Apple File Exchange will allow you to
reformat one of your own floppy disks according to DOS specifications
("Erase Disk" under AFE's File menu lets you put floppy disks into
DOS format) and will then let you copy your file to the DOS disk. If
your friend has the DOS program called LHA, he can then use it to
dearchive your file on his computer. If he *doesn't* have LHA but the file
is one you simply can't do without, you'll have to download LHA as
well and take it (or send it by modem) along with your LZH archive.
(Most services that post LZH archives also post a recent version of
LHA, which is likely to show up in a file library under the name
LHAx.EXE, where the lowercase *x* is a version number. The .EXE suffix
is DOS's way of telling you that the file is *executable* on a DOS
computer, and LHAx.EXE will probably be a self-extracting file that
metamorphoses into a usable version of LHA upon execution.) Once
your friend has used LHA to dearchive your file, he can copy the
decompressed version(s) to your disk or send it back using a modem.
If he takes the floppy route, you have to use Apple File Exchange or an
equivalent program to copy the file back to your hard disk, as DOS
disks don't ordinarily show up on your desktop. Of course, if your
friend has neither a modem nor a 3-1/2-inch floppy drive, you're out
of luck.

**Warning:** Yet another reminder about MacBinary! When you're using
your modem to send DOS files to a DOS-based computer, make sure
to turn MacBinary off. PCs are befuddled by it!

**BTW,** I was careful to say that DOS disks don't *ordinarily* show up on
your desktop. There are programs that can coax them to do just that.
One such utility is called DOS Mounter from Dayna Communications. DOS Mounter supports formatting of DOS diskettes and also lets you copy DOS files to and from DOS diskettes using the Finder's normal click-and-drag technique. If you're using System 7.0, Apple's own utility, Macintosh PC Exchange, performs many of the same functions and a few more to boot. Yes, old-but-by-no-means-venerable Apple File Exchange is distributed free of charge, but price is one of the very few things that recommend it. It's cumbersome and non-intuitive, and if you have to work with DOS disks I suggest you shell out a few bucks for DOS Mounter or Macintosh PC Exchange. AFE should be allowed to die the ghastly death that it deserves.

ARC and ZIP archives are easier to handle directly on a Mac. StuffIt Deluxe can extract both AR Ced and ZIP ed files, and several shareware utilities can too. To decompress ARC archives on a Mac there's ArcMac and ArcPop.ARC, both of which can be downloaded from most commercial online services and good Mac BBSs. ArcPop.ARC is a kind of subset of ArcMac, which also lets you create ARC archives. ArcPop.ARC is strictly a de-ARCing utility, but it's much simpler to use. If decompressing DOS ARChives is all you plan to do, I'd recommend you not bother with ArcMac. To unpack ZIP archives on a Mac you can use either of two shareware utilities, MacUnZip or UnZip. Like ArcMac and ArcPop.ARC, you'll find them posted in the file libraries of most good Macintosh-oriented BBSs and online services.

**BTW**, as with the Mac itself, some DOS archives are self-extracting. You'll remember that self-extracting Macintosh archives are normally called sea files. In the DOS world they're usually called sfx files. Whatever they're called, if you have to decompress a self-extracting DOS file, you'd normally do it on a DOS computer. The program code that performs the extraction is IBM-compatible code, not Macintosh code.

**Opening DOS Files**

Okay. Now, by hook or crook, you've decompressed the files in your DOS archive. Maybe they weren't compressed to begin with, so you were able to skip that step altogether. You may be collaborating on a book with a friend who uses a DOS computer, in which case your collaborator is probably giving you DOS files and just as probably sparing you the need to decompress them.
It's interesting that when Mac users and PC users exchange files, the responsibility for converting files almost always seems to fall with the Mac user. DOS people love to claim ignorance about things like converting DOS files to Mac files on a PC, even though DOS-based conversion utilities like LapLink Mac are perfectly ready to take on the job.) So now you've got these uncompressed files, but you also have a problem: they're still DOS files. Now what?

Well, if they're plain-text files, you can open them with your word processor and massage them at will. If they're binary files, you'll want to know which PC program created them or at least what format they're in. More often than not, DOS file names are tagged with an identifying suffix, so below is a list of some common suffixes and the DOS formats to which they correspond.

**Table 10.1. Common DOS File Suffixes**

<table>
<thead>
<tr>
<th>Suffix</th>
<th>DOS Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>.BMP</td>
<td>Windows bitmap</td>
</tr>
<tr>
<td>.CSV</td>
<td>Comma-separated numerical values</td>
</tr>
<tr>
<td>.DBF</td>
<td>dBASE (a DOS database application)</td>
</tr>
<tr>
<td>.DCA</td>
<td>IBM-revisable-format text (an interchange format for word processors)</td>
</tr>
<tr>
<td>.DOC</td>
<td>Word for Windows/MultiMate (word processors)</td>
</tr>
<tr>
<td>.DOX</td>
<td>MultiMate version 4.0 (word processor)</td>
</tr>
<tr>
<td>.EPS</td>
<td>Encapsulated PostScript</td>
</tr>
<tr>
<td>.PCX</td>
<td>PC PaintBrush (graphics format)</td>
</tr>
<tr>
<td>.PM4</td>
<td>PageMaker PC</td>
</tr>
<tr>
<td>.RFT</td>
<td>See .DCA</td>
</tr>
<tr>
<td>.SYL</td>
<td>Multiplan PC or SYLK spreadsheet format</td>
</tr>
<tr>
<td>.TAB</td>
<td>Tab-separated text</td>
</tr>
<tr>
<td>.TSV</td>
<td>Tab-separated numerical values</td>
</tr>
<tr>
<td>.TXT</td>
<td>Plain text</td>
</tr>
<tr>
<td>.W42</td>
<td>WordPerfect version 4.2 (word processor)</td>
</tr>
<tr>
<td>.W50</td>
<td>WordPerfect version 5.0 (word processor)</td>
</tr>
<tr>
<td>.W51</td>
<td>WordPerfect version 5.1 (word processor)</td>
</tr>
</tbody>
</table>
If you're lucky, you'll have a Macintosh application that can open and convert your DOS file all by itself. And in that regard I'd like to buck the prevailing trend and put in a good word for Microsoft. There's not a whole lot of love lost between Microsoft and the Mac-oriented press these days, especially as Microsoft goes on trying to engorge the digital world with a voracity equaled only by the tomato that ate the Bronx. One thing I personally dislike about Microsoft is the window that appears on my screen every time I open Excel—the one that more or less threatens that someone will come to my house and break my knees if I let anyone else use the program. But it does have to be said that when it comes to PC-to-Mac compatibility, Microsoft does as good a job as anyone around. Word 5.0 for the Mac can open files created by Word for Windows versions 1 and 2, by Word for DOS and by WordPerfect versions 5.0 and 5.1, and it can also open DCA/RFT interchange-format files. Word 4.0 for the Mac can open Word for DOS files, and it's bundled with Apple File Exchange translators for a couple of popular versions of WordPerfect. Word 5.0 can also save files in any number of PC formats, which is a great help when you have to pass your files along to your hypothetical PC-based collaborator.
Excel 3.0 for the Mac is equally PC-friendly. Look at the formats in which it can save documents, shown in Figure 10.5.

And of course it can read files in all those formats as well. A growing number of other Mac programs (PageMaker is a notable example) are also capable of opening and converting DOS files, so when you have a PC file you need to look at or work with, the first thing you should do is check out your software and see what kind of alien files it can handle.

If your meat-and-potatoes applications won’t fill the bill, I suggest you reach for your wallet and buy DataViz’s MacLink Plus, whose armamentarium of file translators is still unequaled. It comes bundled with DOS Mounter, so if the DOS file you want to translate is on a DOS-formatted floppy, you can just slip the floppy into your SuperDrive and it will mount to your desktop just as though it were a Mac disk.
Figure 10.5. In Excel 3.0, you get this window when you click on "Options" in the Save As... window. Excel can read and write files in these formats.

When you open MacLink, you get a window like this:

Figure 10.6. The window in MacLink Plus in which you choose your translators.
Listed on the left in Figure 10.6 are the PC formats that MacLink can handle. (I've scrolled randomly to the middle of the list; it's actually quite long.) My particular copy of MacLink can also handle files created on some other computers too. If I hold down the mouse on IBM PC Formats it turns into the following pull-down menu...

...which gives me access to these other file formats.

Let's say the file I want to translate was created by the DOS word processor XYWrite III. I highlight XYWrite III in the list on the left, and the list on the right promptly reduces itself to show only the Mac formats into which MacLink can render XYWrite III files. I decide that I want my XYWrite III file to be translated into Macintosh Word 4.0 format, so that's what I highlight on the right, as shown in Figure 10.7.

Figure 10.7. Choosing your translation in MacLink Plus.
Then all I need to do is tell MacLink what file to translate and where to put it. I do that by clicking on Select Files, which gives me the window shown in Figure 10.8.

![Figure 10.8](image)

In MacLink Plus the lists on the left and right work like ordinary Standard Files dialog boxes. You use them to find a file you want to translate and to select a destination for the translated file.

The DOS file I want to translate is in a folder called Alien Files on my hard disk, and I want the translated version to show up in another folder called Translated Files. I've selected both of them in the window, and now all that's left to do is click Convert. MacLink Plus takes care of all the rest.

You should be aware that no matter what means of translation you decide to use, it's unreasonable to expect every last filigree of the original formatting to show up in your translated files, especially if your source files are very elaborately formatted. Nevertheless, file translation has improved enormously over the past few years, and with the help of programs like MacLink, what used to be a tedious, distasteful process has become eminently painless. Largely because of the enhanced connectivity brought on by computer telecommunications, the once-provincial Macintosh community has joined the wider world, and now there's nothing but your own questionable sense of propriety to keep you from knowing what your ex-lover wrote in that diary.
Chapter 11

More About Online Services and BBSs

Right from the outset, there are two very gross differences between a commercial online service and an amateur BBS.

When you call a popular BBS with your modem, the line is often busy. Even the most prosperous multi-phone-line BBSs don't have that many lines, and sometimes you have to wait hours—days, even—to log on. When you call a commercial online service, you can almost always connect on the first call. Access is usually instantaneous.

That's the upside. The downside difference is that every commercial online service charges you a fee, usually a monthly base rate plus some number of cents per minute of connect time. Some amateur BBSs also charge a fee, but even if they do it's almost always a small annual membership that guarantees you a generous amount of connect time per day.

More differences? Well, commercial online services offer gateways to still other services. (A gateway is simply a means by which, after you've logged onto one service, you can use that service to gain access to a second service.) GEnie, for instance, offers a gateway to Dow Jones Retrieval, which (for an additional fee) lets you search what's probably the best online financial database there is. CompuServe, GEnie and America Online all offer gateways.
to EAASY SABRE, the American Airlines' travel service that gives flight information about virtually all commercial air traffic in the world and also lets you book online reservations and buy package tours. (That's provided you're willing to put up with EAASY SABRE's not-very-self explanatory online menu system; most times it's easier to call a travel agent.) Amateur BBSs almost never give you gateways.

If you want to download shareware, the commercial online services have the most. Ambitious BBSs like AMUG (Arizona Macintosh Users Group), BMUG (Berkeley Macintosh Users Group) and NJMUG (New Jersey Macintosh Users Group) have a lot, but CompuServe has very nearly all. America Online tends to have it all eventually, but CompuServe usually gets the hot new programs first. Like AOL—at least this is my observation as a frequent GEnie user—GEnie also runs a little behind CompuServe, and its shareware listings aren't quite as abundant. Still, even GEnie beats the BBSs, and it's also the least expensive of the three. The thing about the BBSs is that, for the most part, they depend on their own members to upload files to their file libraries. And from where do their members get the files to upload? Mainly from GEnie, CompuServe, and America Online.

Another thing: The many amateur BBSs that are plugged into FidoNet, a networked message system that we'll soon discuss more thoroughly, let you send electronic mail to destinations all around the world. But FidoNet is slow, and its reliability is less-than-optimal. The commercial online services offer instant worldwide E-mail, their reliability is very good, and their E-mail systems subject you to far fewer restrictions than does FidoNet. We keep using book collaboration as an example, so, let's say you live in New York and your collaborator lives in Los Angeles. GEnie has local access numbers in both those cities. If you call GEnie after 6:00 p.m. and upload a draft of the chapter you've just completed as plain text, you incur no additional fees beyond GEnie's $4.95 base monthly rate and the price of a local call. Evenings and weekends, unlimited use of GEnie's plain-text E-mail service is free. Now a couple of hours roll by, and soon it's 6:00 p.m. in L.A. Your collaborator logs onto GEnie, and for the price of a local call he downloads the chapter you've posted without incurring any additional fee either. There are no long distance charges in either direction, and your collaborator doesn't have to wait by the phone so he can man his modem when you call. What's more, for only a small surcharge you can mail binary files too, which means that, if your chapters
have embedded graphics or lots of fancy formatting, you can send them as true word processor documents. With amateur BBSs, conducting a collaboration in that fashion is prohibitively cumbersome.

Still another thing: product support. Quite a few vendors of Mac-related equipment and software offer online support through the commercial services. Of the three major services, CompuServe is probably the best because it's also the biggest, and vendors can therefore reach the largest audience. Periodically, members of the technical support staff of participating firms check the message bases, answer questions and try to deal with user complaints (and sometimes even with user encomiums). If, for instance, you're logged onto CompuServe and type \texttt{GO MACAVEN}, you get this menu:

\texttt{Mac A Vendor Forum Menu}
\begin{itemize}
  \item 1 INSTRUCTIONS
  \item 2 MESSAGES
  \item 3 LIBRARIES (Files)
  \item 4 CONFERENCING (0 participating)
  \item 5 ANNOUNCEMENTS from sysop
  \item 6 MEMBER directory
  \item 7 OPTIONS for this forum
\end{itemize}

Enter choice!

And if your choice is 2 for MESSAGES, you get this menu:

\texttt{Mac A Vendor Forum Messages Menu}
\begin{itemize}
  \item Message age selection = [All]
  \item 1 SELECT (Read by section and subject)
  \item 2 READ or search messages
  \item 3 CHANGE age selection
  \item 4 COMPOSE a message
  \item 5 UPLOAD a message
\end{itemize}

Enter choice!

If you now type 1 for SELECT, you can access message bases for all the vendors listed below:

\texttt{Mac A Vendor Forum Sections Menu}
\begin{itemize}
  \item Section names (#subjs/# msgs)
  \item 1 Forum Business (14/24)
  \item 2 Portfolio Systems (31/82)
\end{itemize}
Moreover, those are just the vendors in CompuServe's Mac A Vendor Forum. There are more in CompuServe's Mac B Vendor and Mac C Vendor Forums:

<table>
<thead>
<tr>
<th>Mac B Vendor Forum</th>
<th>Mac C Vendor Forum</th>
</tr>
</thead>
<tbody>
<tr>
<td>GCC Technologies</td>
<td>Aladdin Systems</td>
</tr>
<tr>
<td>Altsys Corp.</td>
<td>Baseline Publishing</td>
</tr>
<tr>
<td>Olduvai Corp.</td>
<td>Inline Design</td>
</tr>
<tr>
<td>Software Ventures</td>
<td>CoStar</td>
</tr>
<tr>
<td>ChipSoft</td>
<td>Iomega Corp.</td>
</tr>
<tr>
<td>T/Maker Inc.</td>
<td>Softsync, Inc.</td>
</tr>
<tr>
<td>Working Software</td>
<td>Farallon</td>
</tr>
<tr>
<td>Microseeds Pub.</td>
<td>Alysis</td>
</tr>
<tr>
<td>Hyperpress Pub.</td>
<td>Salient Software</td>
</tr>
<tr>
<td>Radius Inc.</td>
<td>Mirror Technologies</td>
</tr>
<tr>
<td>Deneba Software</td>
<td>Virtus Corp.</td>
</tr>
<tr>
<td>SuperMac Technology</td>
<td></td>
</tr>
</tbody>
</table>

Altogether, CompuServe gives you a whole lot of vendors to complain to, and that's something no amateur BBS can do. Furthermore, it all happens out in the open. When you deal directly with a vendor on the phone, you don't know how many other people are having the same problem. In the vendor forums, all the complaints—yours and everyone else's—are there for the whole telecom community to see, so that by scanning a vendor's message
base you can often tell which products are making trouble for users and which ones are behaving decently. The bottom line is that, wonderful as they are, amateur BBSs can do only so much, and eventually you might give some thought to becoming a member of one of the commercial services. To that end, let's review the three online services most popular in the Macintosh community.

**CompuServe**

Often referred to as CIS (for CompuServe Information Services), CompuServe is the Big Daddy of the online services. It has the most members, the most forums, the most files, and also the biggest price tag. When you log on, you quickly get CompuServe's TOP menu, so called because it's from there that CompuServe multitudinous forums all eventually branch:

```
    CompuServe  TOP
  1 Access Basic Services
  2 Member Assistance (FREE)
  3 Communications/Bulletin Bds.
  4 News/Weather/Sports
  5 Travel
  6 The Electronic MALL/Shopping
  7 Money Matters/Markets
  8 Entertainment/Games
  9 Hobbies/Lifestyles/Education
 10 Reference
 11 Computers/Technology
 12 Business/Other Interests
Enter choice number!
```

On CompuServe you have the option of branching your way, category by category, to the forum you want to visit or simply typing `go` followed by enough of the destination forum name to identify it uniquely. For instance, typing `go macave` is enough to get you to the Mac A Vendor Forum. And while we're on the subject, it's worth pointing out that in addition to their message bases, CompuServe's participating vendors also maintain file libraries where
they often post software upgrades and product information. Here's an example:

Mac C Vendor Forum Library 13
Mirror Technologies
1 BROWSE Files
2 DIRECTORY of Files
3 UPLOAD a File (FREE)
4 DOWNLOAD a file to your Computer
5 LIBRARIES
Enter choice!

If I type a 1 to browse some of Mirror Tech's files, I see (with the option to download) such items as:

[75300,3204]
PRPRC.TXT/Asc Bytes: 6150, Count: 8, 06-Jun-92
Title: Mirror Product Pricing Information 6/5/92
textfil
Keywords: MIRROR
Mirror Technologies product pricing information effective 6/5/92. This is a Word 5.0 doc.
Press <CR> for next or type CHOICES!

[75300,3204]
PRDTEC.TXT/Asc Bytes: 5257, Count: 10, 30-May-92
Title: Mirror Product Technical Specifications 5/1/92
Keywords: MIRROR TEXT
Mirror Product Technical Specifications as of 5/1/92.
Press <CR> for next or type CHOICES!

...and so forth.
Instead of a forum name, you can also type the name of a group of forums. Here, for instance, is what you get if you type go mac:

Macintosh/Apple MACINTOSH
1 Applications Forum +
2 Communications Forum +
3 Community Clubhouse Forum +
4 Developers Forum +
5 Entertainment Forum +
Chapter Eleven  More About Online Services and BBSs

All of CompuServe's Mac-related forums are listed in the menu. What do the pluses mean? They refer to CompuServe's rate structure. CompuServe gives you a choice of two different ways to spend your money. Under their standard pricing plan, a $7.95 monthly membership fee gets you unlimited connect time with an assortment of CompuServe's general interest services. The assortment includes news, sports and weather services (Associated Press Online, Hourly News Summaries, Sports, Entertainment, Business News, Today in History, AccuWeather Maps, National Weather Service); reference library services (Grolier's Academic American Encyclopedia, Consumer Reports, Peterson's College Database, Healthnet); shopping services (The Electronic Mall, Shopper's Advantage, A Discount Shopping Club); "Money Talks" services (Basic Current Stock Quotes, Issue/Symbol Reference, Mortgage Calculator); online games and entertainment services (Science Trivia Quiz, The Grolier Whiz Quiz, ShowBizQuiz, CastleQuest, Black Dragon, Classic Adventure, Enhanced Adventure, Hangman, Ebert's Movie Reviews); communications exchange services (CompuServe Mail, Classified Ads [read-only], Practice Forum, Directory of Members, Ask Customer Service); travel and leisure services (Travelshopper airline, hotel, rental car information and reservations, plus other travel information, Department of State Advisories, Visa Advisories, Easy Sabre); and so forth. Still, numerous and varied as all those services are, they represent only a fraction of CompuServe's expanding universe. Forums marked with a plus aren't covered under the blanket of the standard pricing plan, and using them incurs a surcharge.

BTW, with CompuServe's alternative pricing plan you're charged $2 for monthly membership, which gives you free unlimited use of only the online Membership Support services (that includes the Electronic Mall, through which you can buy almost anything, computer-related or not, from major retailers such as Sears and minor retailers whose
names you've never heard before). All other usage is billed at connect rates of $6.30 per hour for 300 bps, $12.80 per hour for 1200 and 2400 bps and $22.80 per hour for 9600 bps, plus any applicable network and premium surcharges. Surcharges? Some forums on CompuServe—certain financial databases are one example—will cost you more to use, although if you make a killing on the market you can probably deduct your CIS fees on your tax return. And as regards phone networks, you aren't limited to using CIS's own. Companies like U.S. Sprint and MCI sell access to networks such as TYMNET and P.C. Pursuit, and sometimes it's ultimately cheaper to access CompuServe through a local network number than a true CompuServe access number that's more distant from your home or office. You'll have to do the math yourself, but CompuServe will provide the relevant information.

At any rate, just for the hell of it type a 2 at the CompuServe menu quoted just above and visit...

Computing Support      MACCOMM
Welcome to Mac Communications Forum, V. 2G(41)

Once again, you could have arrived here from any prompt simply by typing go maccom.

Mac Communications Forum Menu
1 INSTRUCTIONS
2 MESSAGES
3 LIBRARIES (Files)
4 CONFERENCING (0 participating)
5 ANNOUNCEMENTS from sysop
6 MEMBER directory
7 OPTIONS for this forum
Enter choice!


Mac Communications Forum Messages Menu
Message age selection = [New]
1 SELECT (Read by section and subject)
Chapter Eleven More About Online Services and BBSs

2 READ or search messages
3 CHANGE age selection
4 COMPOSE a message
5 UPLOAD a message
Enter choice!

Typing 1 will give you an idea of the Mac communications subjects under discussion:

Mac Communications Forum Sections Menu
Section names (#subjs/# msgs)
0 Library Questions (12/26)
1 Forum Bus./General (20/53)
2 CIS Navigator (42/121)
3 Term Programs (7/15)
4 Communication Utils (34/100)
5 Hardware (64/221)
6 FAX (57/183)
7 Local Area Nets (42/130)
8 Talking To PCs (18/47)
9 BBS Systems (14/61)
10 Using CompuServe (5/20)
11 Hot Topic (1/1)
Enter choice(s) or ALL!

Now type a 5 to learn what members are saying about the state of telecom hardware for the Mac:

Mac Communications Forum Subjects Menu
Subject (# msgs)
Section 5 - Hardware
1 Supra Problem (22)
2 PP & PBX (4)
3 More Supra Problems (1)
4 Network Dial In/Out (20)
5 DoveFaxPlus? Yes/No (1)
6 Supra Problems (1)
7 Possible Supra Fix! (1)
8 Hayes products announced (11)
9 Prometheus Home Office (1)
Supra has had a hard time meeting demand for its new V.32bis modem. But how hard a time is the public having? Type 1 and find out. (Some names have been changed to protect the innocent.)

Barry,
We will be shipping the Macintosh package of the SupraFAXModem V.32bis starting tomorrow. We have had delays because we were waiting for the fax software to be updated to work correctly with the new modems. We have it now and will be shipping a lot of packages out over the next few weeks. MacZone should start seeing some by the end of this week.
John
Supra Corp.

There are 3 Replies.
Press <CR> for next or type CHOICES!

Gary Reeves seeing -some- of the modems?
Since we ordered ours ages ago, I really don’t know what this means. If it means another 3-4 weeks waiting or longer, we’re going to cancel the order. If it means that MacZone can get to our very early order immediately we’ll let it ride.

Supra advertised several months too early on this product and we ordered based on discussions on CIS, what MacZone told us, and the advertising.

There are 2 Replies.
Press <CR> for next or type CHOICES!

#: 67030 55/Hardware
    28-May-92 18:03:17
Sb: #66959-Supra Problem
Fm: Emilio Vasquez 100042,444
To: Gary Reeves 76254,321 (X)

Gary,
Three days ago I contacted MacZone via CIS mail asking them when I could expect them to ship the Supra FaxModem V.32bis modem if I ordered it “today”. I have just received a reply: 1 to 2 months! I don’t know if that helps but I assume you must be well ahead in the queue!
There is 1 Reply.
Press <CR> for next or type CHOICES!

#: 67059 55/Hardware
    29-May-92 07:49:12
Sb: #67030-Supra Problem
Fm: Gary Reeves 76254,321
To: Emilio Vasquez 100042,444 (X)

Update
I was told that Supra was shipping the new modems (new chip set) this week. I called them yesterday. Their backlog must be horrendous.
Press <CR> for next or type CHOICES!
Gary,
I can't tell you how many orders MacZone has, or where you stand on their list. I do know however that we are shipping a ton of the Macintosh V.32bis packages out of here and that many dealers should be able to clear out a lot of orders in the next week or so. It is unfortunate that we started our advertising a little earlier than we should have. Development took longer than anticipated and then it took longer to get the supporting fax software packages to work with them. However, I think that you'll find it worth the wait.
John
Supra Corp.
There is 1 Reply.
Press <CR> for next or type CHOICES !

Had enough? Just type off at the prompt. A soulless computer will thank you for visiting the Mac Communications Forum. Then it will give you a little information about when you logged off and how long you were on and disconnect you. Finally your modem will chime in with NO CARRIER, and the problems of Supra Corp. and its potential customers will recede until the next time you log on and check the message base.

Summing up, CompuServe is a vast, complex, intriguing edifice. It's immensely useful once you know the territory, but it can sometimes be like Kafka's Castle if you don't. You can type find followed by a word or phrase at any prompt to help you get around, and almost every time you do you'll discover things on CompuServe that you never knew were there. If you're interested in exploring further, buy a CompuServe membership kit from a software retailer (the kits are often included free of charge with new modems) or call CompuServe directly at 1-800-848-8199.
For better or worse, GEnie is CompuServe's second fiddle. Its membership and roster of services are somewhat smaller, and after 6:00 p.m., when the rates go down and GEnie's hardware begins to groan under the heavy traffic, everything you try to do tends to take a little longer. Despite all that, GEnie can satisfy most people's online needs perfectly well, and it doesn't cost a whole lot either. For $4.95 a month you get unlimited 2400 bps after-prime-time use (that is, after 6:00 p.m. on weekdays, all day Saturday, Sunday and holidays) of a reasonable assortment of general-interest services, and most services not in the basic assortment are billed at only $6 per hour. Yes, the prime time rate of $18 an hour is rather stiff, and yes, the 9600 bps prime and non-prime rates of $30 and $18 per hour respectively are even stiffer, but with a slow modem and a little self-discipline you need never come to grips with them.

These are the category headings for GEnie's "basic" services, the ones you can use without limit as part of your $4.95 monthly membership:

<table>
<thead>
<tr>
<th>GEnie*Basic Services</th>
<th>Page 8001</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. [•] GEnie Users' (GENIEUS)</td>
<td>2. [•] Aladdin Support BB's</td>
</tr>
<tr>
<td>3. [•] GEnie News, Index &amp; Information</td>
<td>4. [•] Send/Read GE Mail</td>
</tr>
<tr>
<td>5. [•] User Settings/Billing Info</td>
<td>6. [•] Entertainment Services</td>
</tr>
<tr>
<td>7. [•] Travel Services</td>
<td>8. [•] Money Matters/Personal Fin.</td>
</tr>
<tr>
<td>9. [•] Hobby &amp; Leisure Services</td>
<td>10. [•] Education Services</td>
</tr>
<tr>
<td>11. [•] General Interest Service</td>
<td>12. [•] Classic Games</td>
</tr>
<tr>
<td>13. [•] News, Sports and Weather</td>
<td>14. [•] Shopping Services</td>
</tr>
<tr>
<td>15. [•] About GEnie Services</td>
<td>16. [•] Surveys from GEnie</td>
</tr>
<tr>
<td>17. [•] GEnie Info Library</td>
<td></td>
</tr>
</tbody>
</table>
You've already looked at CompuServe's TOP menu; now you can compare it to GEnie's:

<table>
<thead>
<tr>
<th>GEnie</th>
<th>*TOP Page 1</th>
<th>GE Information Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. [*] GEnie Basic Services</td>
<td>2. [*] GEnie Information</td>
<td></td>
</tr>
<tr>
<td>3. [*] Billing and Setting Information</td>
<td>4. Communications (GE Mail &amp; Chat)</td>
<td></td>
</tr>
<tr>
<td>5. Computing Services</td>
<td>6. Travel Services</td>
<td></td>
</tr>
<tr>
<td>7. Finance &amp; Investing Services</td>
<td>8. Online Shopping Services</td>
<td></td>
</tr>
<tr>
<td>11. Career/Professional Services</td>
<td>12. Business Services</td>
<td></td>
</tr>
<tr>
<td>13. Leisure Pursuits &amp; Hobbies</td>
<td>14. Education Services</td>
<td></td>
</tr>
<tr>
<td>15. Entertainment Services</td>
<td>16. Symposia on Global Issues</td>
<td></td>
</tr>
<tr>
<td>17. Research &amp; Reference Services</td>
<td>18. GEnie Explorer RT</td>
<td></td>
</tr>
<tr>
<td>19. Leave GEnie (Logoff)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Enter #, ,H.elp?

Services marked with an asterisk are basic and can be used without limit during non-prime time. Also, some of the categories not marked with an asterisk contain basic services too. As with CompuServe, it's from the TOP menu that you start to navigate your way around. Of course, if you know the name of your destination, you can go there directly. Unlike CompuServe, you don't type go and then the name or part thereof. To get to GEnie's Macintosh RoundTable—GEnie's RoundTables correspond to CompuServe's forums—you just type mac and any prompt. To read your mail or send a missive to another member you type mail:

<table>
<thead>
<tr>
<th>GEnie</th>
<th>MAIL GE Mail Page 200</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. [*] Display Queue of GE Mail Letters</td>
<td></td>
</tr>
<tr>
<td>2. [*] Read GE Mail</td>
<td></td>
</tr>
<tr>
<td>3. [*] Read (List) All Letters in your Mailbox</td>
<td></td>
</tr>
<tr>
<td>4. [*] Read (List) Letters From Specific User</td>
<td></td>
</tr>
<tr>
<td>5. [*] Read (List) Letters From Specific Date</td>
<td></td>
</tr>
<tr>
<td>6. [*] Compose and Send GE Mail Online</td>
<td></td>
</tr>
<tr>
<td>7. [*] Upload a Text Letter</td>
<td></td>
</tr>
<tr>
<td>8. [*] Search GE Mail Directory</td>
<td></td>
</tr>
<tr>
<td>9. [*] GE Mail Command Mode</td>
<td></td>
</tr>
</tbody>
</table>
10. [*] About Attached Files
11. Send (Upload) Attached Files
12. Receive (Download) Attached Files
13. [*] Send FEEDBACK to GEnie

Enter #, <P> revious, or <H> elp?

Like CompuServe, everything but sending and receiving binary files falls under the umbrella rate, and also like CompuServe there's a RoundTable for most species of home computer:

GEnie COMPUTING Computing on GEnie
1. [*] GEnie Computing News (920421) 2. GEnieLamp Newsletter
3. Apple/Macintosh RoundTables 4. Atari RoundTables
5. Commodore & Amiga RoundTables 6. IBM PC/Tandy RoundTables
9. Special Interest Group RTs 10. COMPUTE Magazine
11. Computing Conference Center 12. PC Catalog

Enter #, <P> revious, or <H> elp?

From the travel menu you can see that GEnie rounds up several of the usual suspects:

GEnie TRAVEL Travel Services
1. American Airlines EAASY SABRE
2. Traveler's Info. Services RT
3. Adventure Atlas
4. OAG Elect. Edition Travel Service
5. Aviation RT
6. Destination Florida RT
7. California RT
8. Canada RT

Enter #, <P> revious, or <H> elp?
The investing area contains a fairly impressive array of resources, including unlimited closing stock quotes for no extra charge (note the asterisk) after 6:00 p.m.:

<table>
<thead>
<tr>
<th>GENie INVESTING Personal Finance &amp; Investing Services</th>
<th>Page 600</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. About Financial Products</td>
<td>2. [*]GENie Closing Quotes</td>
</tr>
<tr>
<td>3. Dow Jones News/Retrieval</td>
<td>4. The Investment ANALYST</td>
</tr>
<tr>
<td>5. Charles Schwab Brokerage</td>
<td>6. Investors' RT</td>
</tr>
<tr>
<td>7. News &amp; Special Features</td>
<td>8. GENie QuikNews Clipping Service</td>
</tr>
<tr>
<td>9. GENie Loan Calculator</td>
<td>10. Business Services</td>
</tr>
<tr>
<td>11. [*]TRW Credentials Service</td>
<td>12.</td>
</tr>
</tbody>
</table>

Enter #, <P>revious, or <H>elp?

Shopping services include a GENie Mall that doesn't list as many vendors as CompuServe's but functions pretty well just the same. The News, Sports & Features department includes an online clipping service, and of course there are the dreaded online, multi-player games:

<table>
<thead>
<tr>
<th>GENie MPGAMES GENie Multi-player Games</th>
<th>Page 801</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. MPGAMES RT</td>
<td>by Washburn</td>
</tr>
<tr>
<td>2. Games RoundTable</td>
<td>by Scorpio</td>
</tr>
<tr>
<td>3. Stellar Warrior</td>
<td>by Kesmai</td>
</tr>
<tr>
<td>4. Stellar Emperor</td>
<td>by Kesmai</td>
</tr>
<tr>
<td>5. Air Warrior</td>
<td>by Kesmai</td>
</tr>
<tr>
<td>6. Dragon's Gate</td>
<td>by AUSI</td>
</tr>
<tr>
<td>7. NTN Trivia</td>
<td>by NTN</td>
</tr>
<tr>
<td>8. QB1</td>
<td>by NTN</td>
</tr>
<tr>
<td>9. A-Maze-ing</td>
<td>by Kangaroo Koncepts</td>
</tr>
<tr>
<td>10. RSCARDS</td>
<td>by Factory Programming</td>
</tr>
<tr>
<td>11. GemStone III</td>
<td>by Simutronics</td>
</tr>
<tr>
<td>12. Orb Wars</td>
<td>by Simutronics</td>
</tr>
<tr>
<td>13. Galaxy I</td>
<td>by Mark Jacobs</td>
</tr>
<tr>
<td>14. Federation II</td>
<td>by Federation</td>
</tr>
<tr>
<td>15. Diplomacy (tm)</td>
<td>by Avalon Hill &amp; AUSI</td>
</tr>
<tr>
<td>16. Software Club</td>
<td></td>
</tr>
</tbody>
</table>
17. **LiveWire Chat Lines** by BONBON

Enter #, <P>revious, or <H>elp?

I'm not about to lead you all around GEne, but just to demonstrate that it too has its more secluded corridors, look at GEne's Issues department:

<table>
<thead>
<tr>
<th>GEne</th>
<th>ISSUES</th>
<th>Page 528</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>California RT</td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>Deutschland (Germany) RT</td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>disABILITIES RT</td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>Japan RT</td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>Jerry Pournelle RT</td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>Military RT</td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td>PC-Van</td>
<td></td>
</tr>
<tr>
<td>9.</td>
<td>Religion &amp; Ethics RT</td>
<td></td>
</tr>
<tr>
<td>10.</td>
<td>Space &amp; Science RT</td>
<td></td>
</tr>
<tr>
<td>11.</td>
<td>Canada RoundTable</td>
<td></td>
</tr>
<tr>
<td>12.</td>
<td>Family RoundTable</td>
<td></td>
</tr>
<tr>
<td>13.</td>
<td>Public Opinion Online</td>
<td></td>
</tr>
<tr>
<td>14.</td>
<td>Communications Services</td>
<td></td>
</tr>
<tr>
<td>15.</td>
<td>Career &amp; Professional Service</td>
<td></td>
</tr>
<tr>
<td>16.</td>
<td>Travel Services</td>
<td></td>
</tr>
<tr>
<td>17.</td>
<td>Education &amp; Reference Services</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Enter #, &lt;P&gt;revious, or &lt;H&gt;elp?</td>
<td></td>
</tr>
</tbody>
</table>

What about the Mac? If you do type mac at any prompt, you get the following menu (preceded by the day's Macintosh-related announcements):

<table>
<thead>
<tr>
<th>GEne</th>
<th>MAC</th>
<th>Page 605</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Macintosh Bulletin Board</td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>Macintosh Real-Time Conference</td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>Macintosh Software Libraries</td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>About the RoundTable</td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>RoundTable News (920609)</td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>Newsbytes News Network</td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td>[*]GEneLamp Macintosh Newsletter (920601)</td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td>[*]ENTER The Outbound Systems Contest</td>
<td></td>
</tr>
<tr>
<td>9.</td>
<td>[*]Feedback to the Mac RT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Enter #, &lt;P&gt;revious, or &lt;H&gt;elp?</td>
<td></td>
</tr>
</tbody>
</table>

The Macintosh RoundTable is structured much like all the other RoundTable menus on GEne. Like them it offers a bulletin board for messages, occasional real-time conferences (GEne is particularly big on real-time conferences) and software libraries. Right now let's investigate the Mac-related messages.
They're organized by category, and once you're in the bulletin board you can get a list of all the categories by typing `cat`:

<table>
<thead>
<tr>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Welcome to the GENie Macintosh User Group</td>
</tr>
<tr>
<td>2 Games/Diversions</td>
</tr>
<tr>
<td>3 Communications and the Macintosh</td>
</tr>
<tr>
<td>4 Commercial Software</td>
</tr>
<tr>
<td>5 Freeware/Shareware</td>
</tr>
<tr>
<td>6 Online Product Support</td>
</tr>
<tr>
<td>7 The Macintosh</td>
</tr>
<tr>
<td>8 Mac Graphics</td>
</tr>
<tr>
<td>9 Press Releases</td>
</tr>
<tr>
<td>10 Open Window</td>
</tr>
<tr>
<td>11 Electronic Music - MIDI</td>
</tr>
<tr>
<td>12 Word Processing/DeskTop Publishing Software</td>
</tr>
<tr>
<td>13 But Seriously, Folks..</td>
</tr>
<tr>
<td>14 Database Applications</td>
</tr>
<tr>
<td>15 Computer Aided Design/Engineering</td>
</tr>
<tr>
<td>16 Printers, Plotters, Digitizers</td>
</tr>
<tr>
<td>17 Disk Drives and Backup Systems</td>
</tr>
<tr>
<td>18 Modems and Local Area Networks</td>
</tr>
<tr>
<td>19 Macintosh User Advertisements</td>
</tr>
<tr>
<td>20 The Mac II Family - II, IIX, IIcx, IICI, IIFX, LC, IISI, &amp; SE/030</td>
</tr>
<tr>
<td>21 Apple's HyperCard</td>
</tr>
<tr>
<td>22 *** System 7.0 ***</td>
</tr>
<tr>
<td>23 Deneba Software - Product Support</td>
</tr>
<tr>
<td>24 Berkeley Systems - Product Support</td>
</tr>
<tr>
<td>25 GCC Technologies - Product Support</td>
</tr>
<tr>
<td>26 Software Ventures - Product Support</td>
</tr>
<tr>
<td>27 Fifth Generation Systems - Product Support</td>
</tr>
<tr>
<td>28 Mirror Technologies - Product Support</td>
</tr>
<tr>
<td>29 Beagle Bros. - Product Support</td>
</tr>
<tr>
<td>30 Aladdin Systems, Inc. - Product Support</td>
</tr>
<tr>
<td>31 Sigma Designs - Product Support</td>
</tr>
<tr>
<td>32 Now Software - Product Support</td>
</tr>
<tr>
<td>33 OutBound Systems - Product Support</td>
</tr>
</tbody>
</table>
You are currently attending Category 1

Starting with category 23 you'll notice a series of vendor support sections, but those aren't the only ones. (Yes, GEnie's features aren't always arranged with airtight, unexceptionable logic, but you begin to develop an affection for its haphazardness.) Category 6 is called Online Product Support, and because the next step down the hierarchy from categories is topics, you can go to category 6 and get a topic list:

<table>
<thead>
<tr>
<th>No.</th>
<th>Subject</th>
<th>Messages</th>
<th>Status</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>About Online Product Support...</td>
<td>1</td>
<td>Closed</td>
<td>CMUG [Brian Ebarb]</td>
</tr>
<tr>
<td>2</td>
<td>MicroNet Technology (SCSI Data Storage)</td>
<td>49</td>
<td>Open</td>
<td>MICRONET [Charles]</td>
</tr>
<tr>
<td>3</td>
<td>MacAvenue Technical Support</td>
<td>57</td>
<td>Open</td>
<td>MACAvenue</td>
</tr>
<tr>
<td>4</td>
<td>Hiller Sales &amp; Supply [Hiller Sales]</td>
<td>8</td>
<td>Open</td>
<td>J.HILLER1</td>
</tr>
<tr>
<td>5</td>
<td>GDT Softworks Inc. Support [GDT - Mike]</td>
<td>88</td>
<td>Open</td>
<td>M.BLACKSTOCK</td>
</tr>
<tr>
<td>6</td>
<td>Survivor Software Ltd: MacMoney [Mike]</td>
<td>78</td>
<td>Open</td>
<td>MFARMER</td>
</tr>
<tr>
<td>7</td>
<td>C.M.S. On Line Support</td>
<td>77</td>
<td>Open</td>
<td>FBARTON [Bart]</td>
</tr>
<tr>
<td>8</td>
<td>Applied Engineering</td>
<td>10</td>
<td>Open</td>
<td>APPLIED. ENG [Brian]</td>
</tr>
<tr>
<td>9</td>
<td>TurboMouse rusting problems</td>
<td>12</td>
<td>Open</td>
<td>M.KANKY</td>
</tr>
<tr>
<td>10</td>
<td>Alysia Software Tech Support</td>
<td>41</td>
<td>Open</td>
<td>R.SIGMON [Rodney]</td>
</tr>
<tr>
<td>11</td>
<td>BMUG Helpline Online GEnie</td>
<td>82</td>
<td>Open</td>
<td>RANDY.SIMON</td>
</tr>
<tr>
<td>12</td>
<td>LinksWare and Nexus support</td>
<td>8</td>
<td>Open</td>
<td>LINKSWARE</td>
</tr>
<tr>
<td>13</td>
<td>Affinity Product Support</td>
<td>80</td>
<td>Open</td>
<td>AFFINITY</td>
</tr>
<tr>
<td>17</td>
<td>Supra Corp</td>
<td>34</td>
<td>Open</td>
<td>SUPRATECH</td>
</tr>
<tr>
<td>18</td>
<td>Xerox Imaging Systems, Inc. - DATACOPY</td>
<td>58</td>
<td>Open</td>
<td>J.HUGHES17</td>
</tr>
<tr>
<td>20</td>
<td>Portfolio Systems on-line support [Mac+AppleToo]</td>
<td>90</td>
<td>Open</td>
<td>W.KUBECK</td>
</tr>
<tr>
<td>22</td>
<td>Advanced Gravis</td>
<td>38</td>
<td>Open</td>
<td>GRAVIS1</td>
</tr>
</tbody>
</table>

It seems that some vendors get a whole category to themselves (maybe the ones who check in with their message base every other day or so) while others get just a topic within a category (possibly the ones who read only the favorable messages). Still, no vendor support message base that features
Advanced Gravis, makers of what's arguably the best Macintosh joystick in existence, can be safely ignored.

In the GEnie bulletin boards, just about any action at all concludes with a reprise of this menu:

1. CATEGORIES 10. INDEX of topics
2. NEW messages 11. SEARCH topics
3. SET category 12. DELETE message
4. DESCRIBE CAT 13. IGNORE category
5. TOPIC list 14. PROMPT setting
6. BROWSE new msgs 15. SCROLL setting
7. READ messages 16. NAME used in BB
8. REPLY to topic 17. EXIT the BB
9. START a topic 18. HELP on commands

Enter #, <Command> or <HELP>?

(The 6 just before the question mark in the prompt reminds you that you're in category 6.)

Notice that Topic 17 is reserved for Supra Corp., and I can't help wondering how Supra is doing here on GEnie as opposed to CompuServe. So let's type REA 17 and see what you get:

**********Topic 17 Fri Dec 13, 1991
SUPRATECH at 23:01 EST
Sub: Supra Corp
Product Support for Supra's Macintosh products
34 message(s) total.
**********

Category 6, Topic 17
Message 5 Sat Dec 14, 1991
J.ARACH [Jack] at 14:03 EST

Does the 9600 and the 9600bis modems have the problems with the Rockwell chipset? If so, when do you expect to have new modems shipping without the problem? Also, what are the suggested prices? You list the 2400 modem as shipping but have a 1/15/91 date for the 9600/9600bis modems? Do you mean 1/15/92? I am in the market for a 9600 baud
modem, my school is starting to provide 9600 baud service, and if I can cut my long distance bill for calling the computer from $40 to $15 dollars a month, the modem might pay for itself within 15 months.

... Mike K.

-----

Category 6, Topic 17
Message 6 Sat Dec 14, 1991
ANSWERMAN [Tim] at 15:34 EST

I've been using Supra's 9600 MNP5 modem for 4 months now and have had not ONE problem with it. I highly recommend them whenever I'm asked which 9600 bps modem to buy.

-TIM

-----

Category 6, Topic 17
Message 7 Sun Dec 15, 1991
GWISSE [Gordie] at 05:23 EST

I'm also very interested in purchasing the soon-to-be-released Supra V.32bis fax/modem. BTW, there's an ad on page 44 of the 12.10.91 MacWeek that shows what the new Supra modems look like. I was pleasantly surprised - they're attractive. (Actually, they look very similar to a Hayes modem.)

Tim - a few questions...
1) The price is listed as $479.95 including the cable and Mic II v1.6. Can Mic II v1.6 be upgraded through Software Ventures to the current version? Do you know the price?
2) Several friends say they never heard of Mic II v1.6 - was it a special version for modem manufacturers to distribute?
3) Can it be purchased for $400 without the cable/software?

-----

...and on and on. The whole thing has a far more positive tone than Supra's area on CompuServe, but don't rush to conclude that GEnie's membership is rife with Pollyannas. Because no dates were specified, the list of messages that GEnie gave you began with the oldest ones on the bulletin board, and December 1991 may have brought an interlude of peace to Supra. Newer messages confirm that things got more tumultuous in '92.
Loyal telecommunicators that we are, type 7 for Telecommunications & Networking and then ask for a directory of files. That will get the following:

<table>
<thead>
<tr>
<th>No.</th>
<th>File Name</th>
<th>Type</th>
<th>Address</th>
<th>YYMMDD Bytes</th>
<th>Access Lib</th>
</tr>
</thead>
<tbody>
<tr>
<td>23515</td>
<td>SUPRAFAXMODEM V.32BIS</td>
<td>X J.BRUNI</td>
<td>920608 5248</td>
<td>18</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>Desc: MicroPhone II 4.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23491</td>
<td>STUFFIT EXPANDER V1.0.1</td>
<td>X ALADDINYS920604 73856</td>
<td>272</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Desc: Update of ESSENTIAL expanding tool!</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23432</td>
<td>APPLELINK SIGNUP2.SIT</td>
<td>X BILLR</td>
<td>920530 8192</td>
<td>109</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>Desc: 'Link is available to US only</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23221</td>
<td>APPLELINK SIGNUP.SIT</td>
<td>X BILLR</td>
<td>920509 11648</td>
<td>423</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>Desc: 'link is available to all, get it.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23072</td>
<td>MACWOOF1.5.1.CPT</td>
<td>X GARNER.M</td>
<td>920503 249472</td>
<td>39</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>Desc: &gt;Fabulous&lt; FidoNet Point for BBSs!</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23016</td>
<td>MP II GENIE 4.0.1B1.CPT</td>
<td>X MACSPECT</td>
<td>920501 31232</td>
<td>203</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>Desc: GEnie Script 4.0.1b1 for Microphone</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22986</td>
<td>BBS ONE CONFERENCE</td>
<td>X MICHAELBEAN</td>
<td>920427 1920</td>
<td>13</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>Desc: Info on BBS ONE Conference in Denver</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22976</td>
<td>IDOD 3.4.1.CPT</td>
<td>X BOB.DANIEL920426 74240</td>
<td>222</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Desc: GEnie Batch Downloader/Front End!!!!</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22972</td>
<td>EFFECTOR V.2 N.8.SIT</td>
<td>X GMUG.MAN</td>
<td>920426 16384</td>
<td>61</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>Desc: Electronic Frontier Foundation News</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22952</td>
<td>HAYES MODEM TOOL 1.0.2</td>
<td>X GMUG.MAN</td>
<td>920424 48256</td>
<td>298</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>Desc: Easy access to advanced Hayes comman</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

...and much more. To stop the listing, you send a break, which, by prearrangement with GEnie, can be a true break or some other break character like ^C. The files are listed starting with the ones most recently uploaded, and among the hundreds of telecom files in GEnie's Mac RoundTable you'll find terminal programs, Communications Toolbox tools, modem drivers for MicroPhone II, and scripts for programs such as White Knight and MicroPhone II.

If you want to join, call GEnie at (800) 638-9636.
America Online

You've heard of doing it all with mirrors? AOL does it all with windows, proliferating piles of them that stack up on your screen like cards in a gin rummy game. As for mastering the art of telecommunications or obtaining the perfect terminal program, forget it. AOL provides its own proprietary software; using it entails little more than clicking on some buttons and typing a few words. No, you can't use AOL's software with anything beyond America Online, but if terminal programs and all their various settings are too much for you, this may be the way to go.

AOL's terminal program opens to this window:

![Welcome to America Online!]

Screen name: STaylor2

Sign On  Help

When you click Sign On it shows you this window as it carries out the logon process automatically; the only thing you have to type is your password:
Then you get the window from which your session begins:

Normally you'd get to where you're going by clicking on Departments, but right now click on the icon for new members in the upper-right-hand corner. It takes you through a window-by-window tour of the service. To see a small selection of America Online's departments, browse some of the following windows:
BTW, MacWorld's department on AOL is a little like MacUser's forum on CompuServe. To get to the latter, you type **go zmac** when you've logged onto CompuServe.
BTW, Microsoft also has its own RoundTable on GEnie and its own forum on CompuServe.
And what would we do without...?
Telecommunications: The Macintosh Modem Book

To see the telecom programs that AOL lists, you double-click on Communications Programs in the window above and get—sure enough—another window:

Highlight 3RD CLASS Electronic Mail 1.0 and click on Get Description:
You feel that 3RD CLASS Electronic Mail 1.0 is worth a download? All you have to do is click on Download File, and AOL responds with a window asking you where to place the file:

![Download window]

When you click on Save the download begins and a progress indicator comes up to show you how it's going. There are no protocols to choose and no settings to fuss over. The proprietary software does it all by using (and sometimes overusing) the familiar Macintosh interface.

Is this your cup of tea? If it is, America Online's rates aren't bad at all. The monthly membership fee is $5.95, and connect time goes for $5 per hour evenings and weekends and $10 per hour during business days. You can sign up by calling 800-827-6364.

**Amateur Bulletin Board Systems**

Although amateur BBSs can't begin to provide as many services as their commercial counterparts, they're still enormously useful. From them you can download Mac shareware and freeware and also get help with your computer problems. Many BBSs list information on a broad assortment of topics, and most will let you read and post want ads. The ones that participate in FidoNet will also let you send and receive mail all around the world. And through FidoNet you can locate still other BBSs, even special-purpose ones that concentrate on law or general medicine or news on AIDS.
To get started with a BBS, I suggest you choose a few from the list in Appendix A. Try to find one that links to FidoNet. Once you've logged onto one, you can use Echomail to request information about other BBSs or just about anything else you want. To get a feel for navigating through the Echomail section of a Mac-based BBS, look at Chapter 7. And since FidoNet is really the most outstanding feature of many amateur BBSs, let's examine it in a little more detail.

**FidoNet**

FidoNet is an amateur personal-computer network that spans the globe. It came into being in 1984 with the advent of a PC-based BBS system called Fido, which contained procedures for linking multiple systems via modems and phone lines. At first there were only a handful of participants, but today FidoNet links several thousand individual systems and also includes a sub-network called Alternet. Since Fido itself is an IBM-compatible program, other software had to be written to allow Macs to join the network. Most Macintosh FidoNet systems use programs called Tabby or Copernicus.

FidoNet's principal function is to transmit Echomail and Netmail. What's the difference? Netmail is private and usually shows up in a BBS's private-message section. If you receive a Netmail message at your local BBS, only you can read it. Echomail is posted for public scrutiny in an Echomail conference devoted to a given subject. The conferences are called echoes for short, and a participating BBS can elect to carry as many echoes as it wants.

Every BBS that fully participates in FidoNet is called a *node* and is given a *node number* (or *node address*). In Chapter 7, when we logged onto Dragon's Cave BBS, it told us that its FidoNet node number is 2605/602. In turn, any node can choose to service satellite systems called *points*. Points get *point addresses*. If Dragon's Cave were to let me be one of its points, I'd be able to say in the lingo that I'm a point off Dragon's Cave. If I'm the *sixth* point off Dragon's Cave, my point address would be 2605/602.6, and anyone who logs onto a FidoNet BBS can send me mail by addressing it to 2605/602.6.

How does my mail actually get to me? By common consent, FidoNet's member systems have designated the hour between 4:00 and 5:00 a.m. Eastern Standard Time as National Mail Hour (a fact worth knowing if you're an insomniac and feel especially alone at 4:30 in the morning). When the hour
begins, all FidoNet nodes have to be up and running, and it's during that hour that the nodes exchange FidoNet mail, which is why you can't call into FidoNet BBSs in that hour-long period. Mail passes from node to node, and, if everything's working properly, it eventually gets where it's supposed to go, even if it sometimes takes a few days to travel across the country.

To send private Netmail to someone else, you need to know the node number of a BBS that your recipient logs onto regularly. If you want to send a note to Joe Blow who belongs to Dragon's Cave, you'd address it to Joe Blow, 2605/602. That's Joe Blow followed by comma followed by space followed by node number. Most Mac BBSs will prompt you for the address for your note and assemble it so that it meets FidoNet's specs.

To send public mail, you participate in an Echomail conference (or simply echo). Each of FidoNet's conferences has an uppercase area name. The Macintosh networking and telecommunications conference is called MACNETCOM, and the Macintosh HyperCard echo is called MACHYPE. When you post a message in a given echo, the message soon shows up on all the BBSs that carry the echo, wherever they may be.

If you look back to Chapter 7, you'll see what Echomail messages look like on a Mac-based BBS and also see Dragon's Cave's list of conferences. While you're doing that, you may notice that each message concludes with two lines more or less like:

- Tabby 3.0b3
  * Origin: 4th Dimension BBS/Mansion 617/494-0565 Boston, MA (1:101/450)

In FidoNet-ese, the first line is called the tear line, the second line the origin line. The tear line tells you what software was running on the computer that the from which message came. The origin line gives the name and node number of the BBS at which the message originated and also accommodates a phone number and sometimes a theme phrase like "Loads 'o Files."

All told, FidoNet is at once a major folk phenomenon and an important technical achievement. I encourage you to get to know it and to use it well and often.
Chapter 12

Automating Telecom Programs I: Macros and Auto-dialing

Macros

John Doe is a hopelessly inept typist. One day someone tells him about a little-known bulletin board system that calls itself the All Thumbs BBS and functions as a support group for the incorrigibly clumsy. John gives it a try and likes what he sees, so he begins to log on every week or so. The trouble is, when prompted for his name, he always winds up typing Jhon Doo or Jnoh Deo. The service gives him plenty of chances to correct his typos, but, like most other BBSs, when it prompts him for a password it expects him to get his act reasonably well together. It will tolerate a few mistakes, but half a dozen incorrect passwords and he's automatically logged off. (To prevent intruders from experimenting until they get your password right, most other BBSs will log you off after three incorrect passwords, but ATBBS makes special allowances owing to the unusual nature of its clientele.)

John Doe would love nothing more than to have an easy way of getting his name and password straight on the first try. Fortunately for John, the authors of some Macintosh terminal programs have addressed his problem. What
these programs do is let him define and invoke *macros*. What are macros? At the most basic level, they're user-defined sequences of keystrokes. Associated with each sequence is a *single* keystroke or a button. Once you define the sequence, you type a macro key, make a menu choice, or click the button for a particular macro, and the terminal program types the defined sequence into your terminal window while observing whatever text-pacing preferences you've set up.

To see how it all works, let's examine how ZTerm handles macros. Apart from the fact that I'm fond of ZTerm as an all-purpose terminal program for people whose needs aren't too specialized, it implements its macro features cleanly and simply.

ZTerm lets you define as many macros as you want. You define them in sets of 10. To invoke any of the 10 macros in a set, you hold down the Command key and type a number from 0 to 9, or you choose a macro from ZTerm's Macros menu. Look at Figure 12.1 to see what I mean. At any given time, only one set of 10 macros is available for use. If you want more than 10 macros, you must define another set.

![Figure 12.1. The Macros menu from ZTerm.](image)

To define a macro in the current set you choose Edit Macros... from the menu, which brings up the window in Figure 12.2.
You can see that John Doe has defined two macros for use with the All Thumbs BBS. He's assigned the label Name to the macro that types his name and the label Password to the one that types his password. Then, with all the time in the world to get his name and password right, he's entered a string of characters for each of the two macros. What's the ^M at the end of each string? It's a carriage return, Control-m. You'll no doubt remember from Table 3.1 that now and again it behooves you to send a control character, and that typing an m with the Control key down is equivalent to typing a carriage return. Mac programs normally obey a number of standard conventions, and if John had actually hit the return key after typing Doe, his action would be functionally equivalent to clicking on the OK button in the window, which would close the window instead of entering a carriage return after his name. You can put as many control characters as you like in a macro string, but you have to use the caret system to denote them. (Control-c is ^C, Control-x is ^X and so forth.) Refer to Table 3.1 to see what the various telecom control characters do.
Once John has defined his macros, ZTerm's Macros menu undergoes a minor change. Instead of showing the default names for the 10 macros in the set (Macro-1, Macro-2, etc.) the menu now displays the labels that John has assigned to his macros:

![Fig 12.3. ZTerm's Macros menu with labels for Macros 1 and 2.](image)

From here on in, when the All Thumbs BBS prompts John for his name or password, he doesn't have to type anything at all. Instead, he can pull down the Macros menu, select the macro he wants, and let ZTerm do the typing. Later along, when he's feeling more confident about his keyboard, John may even decide to toss caution to the winds and actuate his macros with Command keys. As you can see from Figure 12.3, Command-1 will type his name and Command-2 will type his password.

ZTerm also gives John the opportunity to define other sets of macros that include frequently typed strings for other services. If he chooses Macro Sets... from the menu, he gets another window:
In that window he can click on New and then name and create a new set of 10 macros. He can also rename or delete an existing set. And if he prefers that another set of macros—one he’s already named and defined—appear in his Macros menu, he can use the pull-down menu next to Macro Set: to choose a different set. When he clicks on the Select button, the macro set he’s chosen from the pull-down menu will be the active set.

Using macro strings is about the simplest kind of automation you’ll find in Macintosh terminal programs, but if you’re a rank beginner I recommend you try them out. With the advent of a more complicated automation scheme called scripting, some Macintosh telecom programs have stopped offering old-style macro capability in their newer versions. But macros are still around, and another program that supports them in a somewhat souped-up form is White Knight.

Where ZTerm associates its macro strings with command keys and menu entries, White Knight primarily uses buttons. Earlier in the book (in Figure 3.5 to be exact) we saw that White Knight’s terminal window could take several different forms, each with a different status bar. To see your custom macro buttons in White Knight, you choose Status Bar from the Local menu and then slide over to the Macro Keys submenu. That causes your terminal window to look something like this:

Figure 12.5. White Knight’s terminal window with Macro Keys status bar.

John Doe may never be a Horowitz of the computer keyboard, but let’s stay with him as our exemplary telecommunicator. We see that he’s created three macro buttons, one to call the All Thumbs BBS, another to type his name, and a third to type his password. From the size of White Knight’s macro buttons,
you can see there's room in the status bar for up to 10—two rows of five. Also take note of the number—it's the number 2 in Figure 12.5—immediately to the right of the macro button area. The arrows above and below the 2 are scrolling arrows. The 2 denotes the fact that we're currently displaying the second set of up to 10 macro buttons. Clicking on the upper scrolling arrow would show us the first set; clicking on the lower arrow would bring up the third set. White Knight's macro status bar may be able to accommodate only 10 macros, but it actually lets you have 30 macro buttons active at a given time.

How did John define his buttons? He chose Macro Keys from White Knight's Customize menu and then, in the submenu, he slid over to Edit. This window appears:

![Select Macro Key To Edit](image)

**Figure 12.6.** White Knight's window for picking the macro button you want to create or edit.

The macro key numbers from 0 through 9 in the window correspond to the equivalent positions in the macro status bar. You can see from the buttons John has clicked that he's chosen to define macro number 1 in set number 2, which from Figure 12.5 would be his Name macro. Now he clicks on Edit and gets the window in Figure 12.7.
You can see that John has once again given the label Name to the macro that types his name and that he's once again defined the macro so it types his full name followed by a carriage return. He might have chosen to have his macro button invoke a script (Get Procedure File) or bring up a file in which he's saved a set of macro keys. Figures 12.6 and 12.7 will give you some notion of the multiplicity of options available for handling macros in White Knight. The program's manual is eminently readable (even though the index could do with some improvement) so we'll skip most of those right now because what's most important here is that you understand in general terms what macros can do and how you go about defining them.

Incidentally, if you own an all-purpose macro program like QuicKeys or Tempo or even Apple's own MacroMaker, you can use it to define macros, in which case you can ignore your telecom program's macro provisions altogether. This can be especially valuable if you've taken to using a terminal program—MultiXfer is one example—that has no built-in macro capability. I'll use QuicKeys for purposes of illustration, mainly because I like it and recommend that every Mac user own it. In QuicKeys, John Doe might have created the pair of text macros in Figure 12.8 on the next page.
To see how John configured one of the macros, let's double-click on Password to display the macro's contents, shown in Figure 12.9.
There's John's admirably honest password, to which he's given the name Password and assigned the keystroke Command-Option-P. *Take note of one very important thing:* This time there's no \^M following the word "Klutz." If you look at the position of the cursor in Figure 12.9, you'll see it's on the line just after the password itself, which indicates that John has typed a true carriage return. One way that general-purpose macro programs differ from terminal programs is that they don't make the assumption that \^M stands for carriage return and won't translate that two-character sequence into a carriage return. If John had put \^M after Klutz, QuickKeys would have typed a caret and then an M. So even if John isn't too accomplished as a typist, he does appear to know his software. And now, when he's running MultiXf er while QuickKeys is active, all he has to do to type his password is hold down the Command and Option keys and type p.

Regardless of what program you use, however, simple macros have a fundamental shortcoming: They can't respond to what's happening in your terminal window. They can't see, for instance, that you've been prompted for a password and then just go ahead and send one. It remains for you to recognize the prompt and then execute the macro that responds to it. A true script, on the other hand, can see what's happening in your terminal. It can detect a particular prompt and then send an appropriate response. We'll see how it's done in due course.

**Auto-dialing**

Earlier in the book we noted that one remote service might prefer a full-duplex connection at 2400 bps, while another service might like half-duplex at only 1200 bps. We wound up devoting an entire chapter to the way you arrive at an ensemble of settings congenial to a given service. The process of making all those settings can be tedious and time-consuming, and when you've done it once and made the settings work you probably won't look forward to doing it every time you call the service.

To spare you the tedium, most terminal programs offer one or another form of *auto-dialing*. What is auto-dialing? It's a technique in which a service's phone number is coupled with its modem and program settings and then stored away for future use. When you want to connect to the service, your terminal program retrieves the service's phone number together with its
settings, sets itself accordingly, sets up your modem with whatever initialization strings it needs, then goes ahead and dials the number, waits for a connection and finally hands you back your terminal window. You configure the whole thing once, and after that whenever you want to connect to the service all you need to do (depending on your program) is click a button, type a key, or make a menu selection.

Let’s return to ZTerm to examine the mechanics of auto-dialing. Yet again, ZTerm does it all with simplicity and clarity, and after we go over the essentials we can see how several other terminal programs manage the same basic task.

ZTerm maintains a single file in which it stores your directory of phone numbers along with the settings that go with them, and it always calls the file ZPhoneList. ZPhoneList is a ZTerm document, and if you double-click on it it will open ZTerm the same way that double-clicking on a MacWrite document opens MacWrite. Under ZTerm’s Dial menu, the first entry is Directory; when you choose it you get the Directory window shown in Figure 12.10.

![Dial Directory Window](image)

**Figure 12.10.** ZTerm’s directory of services that can be auto-dialed.

To create a new entry, you click on New, which gives you a window that lets you name the service and make its basic connection settings. If you select a service you’ve already configured, clicking on Connection gets you the same window. You’d do that if you want to change the settings for a service you’ve already configured. The existing settings would appear in the appropriate
boxes, where you can alter them as needed. Here's what the window looks like for a new service:

![Figure 12.11. The ZTerm window in which you enter basic autodialing information for a service.](image)

You can see from Figure 12.11 that you enter the service's name and phone number in the first two boxes. The Pre-dial init: box lets you enter an initialization string for your modem—if your modem needs any special commands before it connects or, for that matter, if you prefer to configure your modem in a way that's special to the service in question. In the Chapter 6 on the Hayes Language, I mentioned that I can tell my modem to dial its tones faster than the 95 millisecond-per-tone rate to which most modems default. I do it by sending ATS11=55 plus a carriage return to the modem. If I wanted to accomplish that and nothing else before I call a particular service, I'd enter ATS11=55^M in the Pre-dial init: box, and ZTerm would then send ATS11=55^M to my modem before dialing the phone number.

Many services, whether they're amateur or commercial, assign you an account number, and nearly all of them require a password. If you enter your account number and password in the appropriate boxes, ZTerm will know where to find them if you ever get around to writing a script that uses them. (And we will get around to writing a ZTerm script. ZTerm's rudimentary—okay, let's be honest and call it primitive—scripting capability makes it the...
ideal platform from which to launch our maiden venture into script-writing for telecom programs.) The service's data rate, data bits, parity and stop bits are all set via the pull-down menus, and its duplex/echo and handshake options are set through the checkboxes. If any of that still puzzles you, refer to Chapters 4 and 5 on configuring your terminal.

And speaking of your terminal, after you've clicked OK in the window in Figure 12.11, the service you've just defined will appear in alphabetical order in the directory in Figure 12.10. If you then highlight it and click on Terminal, you'll get a window that allows you to make still more terminal settings for the service you're defining, as shown in Figure 12.12.

Regarding the first option in the window, we've already talked about extended characters and high bits, and I suggest you elect to "Strip hi bit" unless you're pretty sure that extended characters are used by the service you're setting up. A fair number of PC-based services are capable of sending what are known as ANSI graphics. (Just in case you're curious, ANSI is the acronym for the American National Standards Institute.) To handle them (if they can), Macintosh terminal programs invoke a terminal emulation that involves the use of a special graphics font along with a provision for interpreting character sequences that do things like change foreground and background
colors. If you're connecting to a PC ANSI-BBS and want to see how ANSI graphics come out on your screen, check the appropriate button and also leave the No Extended Chars button unchecked, as extended characters will be needed to define the ANSI graphics. Otherwise stay with ZTerm's VT-100 emulation. It will behave pretty much like TTY (Teletype), and if you're connecting to a UNIX or DEC system you'll probably gain the additional benefit of terminal compatibility. If you are connecting to a DEC system, you may also want to use the VT100 keypad (refer to the program's documentation to see how it's configured) and also elect to Send RUBOUT for Backspace/Delete.

Once you're done, click OK and you'll get back to the Directory window. If, in that window, you highlight the service you've just defined, you now have some further options. Look at Figure 12.10 again. If you click Set you'll immediately set your terminal to the settings you've just created. Mark and UnMark flag the highlighted service for access from a menu that we'll look at momentarily. Delete deletes the highlighted service from the directory. Dial dials the service with the settings you've assigned to it and also takes you back to your terminal window. QDial does the same thing, except that you remain in the directory. Manual gives you the following window...

![Enter the number to dial](image)

...from which you can dial a number manually using whatever settings you've applied most recently.

Once you've defined a service, it's added automatically to ZTerm's Dial menu, shown in Figure 12.13.

When you select a service from the menu, ZTerm configures itself to your settings, sends your modem the initialization string you've entered for the service, then dials the service's phone number and, if you connect, turns the reins over to you. It also looks for a script associated with the service you've selected, but that's for the next chapter.
Auto-dialing in Other Programs

Of the several reasons I used ZTerm to help demonstrate auto-dialing, one is that it's nicely usual. It couples a service's phone number with the settings and modem initialization strings that go with the number so that when you auto-dial it everything should come out right. And seen from the perspective of Macintosh telecommunications, ZTerm does it all in the usual fashion. There is, however, one respect in which ZTerm isn't usual: it keeps all its phone numbers, initialization strings, and program setups in one file—the one that ZTerm always calls ZPhoneList. By contrast, most other telecom programs create a separate settings file for each service. With those programs you enter a phone number, set your settings, and enter your initialization strings (if you need to), then you save the whole collection as a settings file for that particular service only.

To inspect the approach that some other programs take to auto-dialing, let's look at how you'd set up MicroPhone II to auto-dial a service. (Incidentally, a "service" needn't be a commercial online service or even an amateur BBS; it can also be good old uncle Louie down the block.) If you pull down MicroPhone II's Phone menu and pick Create Service... you get the window in Figure 12.14.
Chapter Twelve Automating Telecom Programs I: Macros and Auto-dialing

Macros and Auto-dialing

Create Service

Service Name: GENie
Phone Number: 324-5002
Dialing Mode: Tone

☐ After Connecting, Do Script:

Figure 12.14. The window in MicroPhone II in which you enter a service's name and number for auto-dialing. If you look at the grayed-out checkbox, you'll see that you can execute a script after connecting. If you'd created such a script, its name would appear under the checkbox, and the checkbox would be active.

After you've entered the name and number of your service and tell the program whether you're dialing with tones or pulses, you go on to set your settings for the service by choosing the appropriate options under MicroPhone II's Settings menu:

Figure 12.15. You get at MicroPhone II's settings through this menu. Each menu entry calls up a window of buttons and/or text fields.
We've talked enough about settings to be able to skip most of the particulars, but there's one settings window that you ought to have a look at:

![Communications Settings Window](image)

Figure 12.16. MicroPhone II's Communications Settings window gives you access to modem drivers as well as the Communications Toolbox.

When you buy MicroPhone II, the program comes bundled with a number of script modules called modem drivers. A modem driver serves as a software interface between MicroPhone II and the particular modem for which it's written. That is, the driver for a given modem (if it's written correctly) knows what modem commands are needed to configure that modem. If it's a driver for a standard MNP modem, for instance, it knows how to set up the modem for MNP operation and also how to set it up for non-MNP operation. As a result, all you need to do is click some buttons. The driver will then translate your button settings into the appropriate initialization strings.

In addition to the drivers bundled with the program, there are other downloadable MicroPhone II drivers posted in the software libraries of online services and BBSs. (As new modems with ever-newer quirks continue to appear, new drivers are written.) The principle advantage offered by these drivers is that they free you from having to use the Hayes language (although I strongly suggest you master its rudiments) at the same time that they spare you the need to learn all of the idiosyncratic ways in which your particular modem responds to the Hayes language (that part is nice).

For instance, standard MNP modems accept a special set of AT commands
(ones that begin with / and &) which configure them for use during MNP connections. So if you own a 2400-bps MNP modem, you could choose to learn all about MNP initialization strings or, as in Figure 12.17, you could select MicroPhone II's standard MNP driver. Then the only other things you'd have to remember are some special aspects of MNP operation: to set the handshaking method to "Hardware" (but only if you have a proper hardware handshake cable) and to set the speed of your serial port to four times the nominal speed of your modem (9600 bps for a 2400-bps modem). You take all those actions in MicroPhone II's Communications Settings window, and when you then choose Dial Service... from the Phone menu, MicroPhone II sends the following initialization string to your modem before it dials the number:

\texttt{AT E1 Q0 V1 X4 &C1 &D0 \60 \J0 \N3 \Q3 \V1 S0=0}

The string works perfectly well with the Practical Peripherals PM2400SA MNP/Level 5 modem that I sometimes use, and as you watch the whole process take place in your terminal window you'll even see the program send your modem the

\texttt{ATDT324-5002}

that does the dialing.
If all goes well, the next step is to save your settings. The settings you've just made were for GEnie, so you'd probably call the settings file GEnie Settings. To auto-dial GEnie in the future, you'd either open MicroPhone II and then, when you're in the program, go on to open your GEnie Settings file, or you could just double-click directly on the GEnie Settings settings file. Then you'd pull down MicroPhone II's Phone menu and pick Dial Service... to bring up the window shown in Figure 12.18.

![Figure 12.18. The window in MicroPhone II in which you choose a service to auto-dial.](image)

If, from Figure 12.18, you've concluded that MicroPhone II lets you define more than one service per settings file, you're absolutely right. *But all the services you create have to use the same group of settings.* If you were to define another service in your GEnie Settings file, for instance, the same terminal and modem settings would apply to it as already apply to GEnie. So if you do decide to define multiple services in a MicroPhone II settings file, be sure that they all use identical settings.

**The Communications Toolbox**

Let's have still another look at the MicroPhone II Communications Settings window. Its pull-down Method menu is in Figure 12.19.
Along with MicroPhone II's own standard method of connecting, there's a list of tools. What tools are those? They're communications tools that happen to reside in my System Folder. They're not a part of MicroPhone II per se, and if I remember correctly, not a single one of them was bundled with the MicroPhone II software. As far as MicroPhone II is concerned, it's simply agreeing to serve as an intermediary, an interface between me and my collection of communications tools. Confusing? Let's take things from the beginning.

Not so very long ago, in an effort to impose some uniform standards on Macintosh telecommunications, Apple created a set of system software accessories called the Communications Toolbox. The Toolbox is built into all versions of System 7.0, and if you can lay your hands on Apple's Communications Toolbox Installer you can also install it into later versions of System 6.x. The software routines in the Toolbox aren't directly accessible to the lay user, but you can gain access to them through communications tools. That is, communications tools are user interfaces to the Communications Toolbox. Apple has predefined what form communication tools have to take (just as it's predefined what form control panels or desk accessories have to take) but anyone can write them as long as they conform to Apple's specifications. If
you're using System 7.0, you place communications tools in the Extensions folder inside your System Folder just as you would any other system extensions. In System 6.x, which doesn't use an Extensions folder, you put communications tools directly into your System Folder.

And then what? Well, if you double-click on a communications tool, not a whole lot happens. Communications tools may be interfaces to the Communications Toolbox, but now you need an interface to your communications tools. As you can see from Figure 12.19, MicroPhone II can function as just such an interface. At this writing White Knight won't interface with communications tools and neither will ZTerm. The VersaTerms and Smartcom II will.

Do you need communications tools to telecommunicate? No, not at all. Good telecom software can provide all the requisite conveniences without making any use of the Communications Toolbox. Do you need access to communications tools? Maybe. If you feel it's important to locate yourself somewhere near the cutting edge, you probably want the option of using communications tools whether you actually access them or not. If you want to use all the many niceties of VersaTerm or VersaTerm PRO, communications tools will be essential. Are communications tools superior in their performance to tool-less telecom programs? At present generally not, but tomorrow someone may write a communications tool that blows the tool-less competition right out of the water. All of which is to say that right now the whole thing's up for grabs and that the only general-purpose communications tool I'm really fond of is the Hayes Modem Tool.

As long as we're talking about Hayes, let's examine Smartcom II's way of using communications tools. (Incidentally, Smartcom II has opted to call them connection tools; communications tool is Apple's term for the same thing.)

When Use Connection Tools isn't checked in Smartcom II's Connection menu, the menu looks like Figure 12.20.
This is the menu you’d use to configure a service for auto-dialing when you’re not planning to access the Communications Toolbox. You click on this icon...

...to bring up this window:

...in which you enter a service’s phone number and how you want to dial it. Then, pretty much as we’ve been doing with ZTerm and MicroPhone II, you set your settings and save them in a settings file. From that point on, whenever you open the file, Smartcom II will set itself to those settings, and when you click on the phone icon it will bring up the above window with the
number you first entered. If you click OK, Smartcom II will auto-dial the service in the way you've come to expect.

If, however, you check the Use Connection Tools option in the Connection menu, the menu mutates to another form:

![Connection menu](image)

**Figure 12.21.** Now that Use Connection Tools is checked, you can compare this form of Smartcom II's Connection menu with the one in Figure 12.20.

Gone are the Originate and Answer options. Gone are the Choose Port and Direct Connect options. Why? Because, as you'll soon see, some of those same options will be offered by the Hayes Modem Tool, and in just about all programs that let you use communications tools, the choices you make within the tool override the choices you've made in the program that gives you access to the tool.

Now let's check out the Hayes Modem tool itself. You bring up its window by choosing Configure Tool in the Connection menu. You'll get a fairly good idea of its essential features by scanning the series of illustrations on the next few pages.
Chapter Twelve Automating Telecom Programs I: Macros and Auto-dialing

**Figure 12.22.** The Hayes Modem Tool's Connection window. It will look the same regardless of what program you use to access the tool. You can see that the Connection icon is highlighted. This is the window to which the tool defaults when you first access it.

**Figure 12.23.** The Hayes Modem Tool's Modem window. Within it you can configure your modem without using a single phrase of the Hayes language.
Connection Settings

Method: Hayes Mode...

Connection Settings

Method: Hayes Mode...

Figure 12.24. The Hayes Modem Tool's Port window. This is where you choose your serial port and make some more connection settings.

Figure 12.25. The Hayes Modem Tool's V-series window. Most of the settings apply to Hayes V-series modems, but if you have one you're in luck. If you don't have one, you can easily disable the functions that are V-series-specific.
Chapter Twelve Automating Telecom Programs I: Macros and Auto-dialing

Figure 12.26. The Hayes Modem Tool’s Setup Strings window. Here’s where you can enter strings to initialize your modem at various times before and after a connection.

Figure 12.27. When you choose Open Connection from Smartcom II’s Connection menu, the Call Progress window appears. Each of the five functions shown in the window is highlighted as it takes place.
One thing you might have noticed when you looked at the Connection menu in Figure 12.21 is that only one tool, probably the apple of Hayes's eye, was displayed at the top of the menu. Does that mean it's the only communications tool you can use in Smartcom II? Not by any means. Communications tools observe a convention which prescribes that each tool give you access to any other tools you've installed in your System Folder. You pull down the Method menu, and voila!—there before you are all the tools that MicroPhone II showed us earlier, as shown in Figure 12.28.

Communications tools are available for a variety of connections: to modems, to networks, even directly to other computers. Some come bundled with software, some are available through online services, others can be obtained through Apple itself. And because Apple intended communications tools to abide by a specific set of guidelines, the basic operation and configuration of all such tools is pretty much the same regardless of the tool you're using.
If you do decide to use them, remember that most programs will let you save the tool you've selected in a settings file along with all its settings. The settings you make in a communications tool almost always override the settings in the program that's letting you access the tool in the first place. Incidentally, the Hayes Modem Tool is available for downloading as freeware through online services and BBSs. To my taste Apple's own communications tools (e.g., the Apple Modem Tool and the Serial Tool) are rather limited, although the newest version of the Apple Modem tool is essential for use with Apple's own internal modem for the PowerBooks. However, Abelbeck Software, the people who make the VersaTerms, have come up with some snazzy tools, and others from still new sources should start appearing fairly soon. You may be none the worse for ignoring communications tools today, but I suspect they're the wave of the future in Macintosh telecommunications.

Not so many years ago, scripting was the wave of the future in Macintosh telecommunications. Now that you're a master of macros and an ace at auto-dialing, let's go on to the next chapter and write a simple script.
If you're only a very occasional modem user, you'll probably never need any more automation than the macro and auto-dialing features we just discussed. In fact, you may be better off avoiding further automation. Automation can create a sense of distance between you and the nuts and bolts of whatever it is you're automating, and if you only use your modem every week or so you won't want to feel estranged from its principles of operation every time you turn it on. If, however, you find yourself using your modem fairly frequently, you'll eventually want to explore some of the Macintosh terminal programs that support a higher level of automation than macros and auto-dialing. At this most advanced level, your terminal program essentially surrenders control to a series of instructions called a script, and the script operates the program as though you were controlling it yourself.

In saying that, I was tempted to use the verb "run" a couple of times—as in "the script runs the program," for instance—but I avoided it deliberately. That's because when a computer is running a program, the program can also be said to be running (i.e., controlling) the computer, and when a terminal program is running a script, what the script is really doing is running (i.e.,
controlling) the terminal program. In computer-esque, the verb “to run” often has this bi-directional quality, and it’s from the context that you figure out in what direction the verb is pointed. I’ll try to be as clear as I can about that, but when your computer is running a terminal program and your terminal program is running a script, the script is controlling your computer. To that extent, the script is—horror of horrors!—a computer program.

Does all this mean that writing scripts is tantamount to writing computer programs? Technically, yes. Does it therefore mean that if you want to write a few scripts you’ll have to learn the entire instruction set for the Motorola 680x0 series of CPU chips? Thankfully, no. The Motorola CPU instruction set constitutes what’s called a low-level language—a language that lets you talk to your computer directly. It’s actually the only language your computer really understands, but there are any number of software assistants that stand ready to receive your orders in a higher-level language, translate them into a lower-level language, and then pass them along to your trusty central processing unit.

One such assistant that’s particularly familiar to Mac users is HyperCard, which supports a high level language called HyperTalk. If you script a HyperCard button as follows:

```plaintext
on mouseUp
  type "Hello"
end mouseUp
```

...then whenever you let go of the mouse clicker while the pointer is inside the button (that’s what `mouseUp` means in HyperTalk), the word “Hello” will be typed at the last text insertion point. HyperTalk is therefore a general-purpose high-level programming language whose statements the HyperCard program interprets and executes whenever certain events occur. When the mouse clicker pops up while the pointer is inside your button, that’s the event that causes the "type 'Hello'" statement to be executed. The scripting languages of most terminal programs are special-purpose high-level programming languages. Some telecom scripting languages are nearly every bit as powerful as HyperTalk, but the events to which they’re most keenly sensitive are events that take place in and around your serial port.

Typically, full-blown scripting languages, such as the ones supported by White Knight, Smartcom II and MicroPhone II, allow for many dozens of
different statements, each with its own syntax and often with several possible variations. To cover all of them would require not just a chapter but a whole book—and a fairly hefty one at that. Once again, however, trusty ZTerm comes to the rescue. For a couple of years ZTerm managed to retain its popularity among Mac telecommunicators mainly on the strength of its support for the ZModem file transfer protocol. Beyond that it was a pretty good terminal program as shareware software goes, but one feature it never did support was scripting. And then, just as ZTerm's loyal users were preparing to abandon hope, ZTerm 0.9—the long-promised scripting version—came along.

Was the Mac telecom community elated? No, not entirely. ZTerm 0.9 did support scripting, but not a lot of scripting. The script statements you could make were adequate for a complete service logon, which is to say that you could write a script that entered your password and account number automatically. But they weren't powerful enough to let you do anything really complicated. You couldn't, for instance, write a script that called some service at four in the morning when rates are low, logged you on, loaded a list of file names, went to the service's file transfer section, downloaded the files you'd specified and then logged you off. (A popular White Knight script called "I Dream of Dialing"—it's usually listed in shareware libraries as IDOD—does that and more for GEnie.) Still, if the overall response to ZTerm 0.9's scripting power tended toward the tepid, my own response is rather more enthusiastic. To cover scripting in a chapter, we need a telecom program whose scripting capability has training wheels, and ZTerm's limited but perfectly functional scripting power fills the bill quite nicely.

Suppose, however, that even though you don't know anything about scripting right now, you plan to do a lot of it in the future and also intend never to get within a mile of ZTerm. Will everything that follows be of no earthly use? Far from it. When push comes to shove, scripting is scripting. Script languages may differ from one terminal program to another, but the scripts themselves all try to accomplish pretty much the same thing. So regardless of what program you expect to use, I suggest you follow along just to see how scripting works.
Scripting for ZTerm 0.9

This is ZTerm 0.9's entire vocabulary of script commands:

- Wait
- Send
- Beep
- Display
- Hangup
- Open capture
- Close capture

Yes, the list is pretty short, but that makes it all the easier (and clearer) to review the syntax for each command.

**Wait**

The Wait command can take any of three forms:

- **Wait for prompt "Howdy do"**
  
  This form of the Wait command causes ZTerm to wait until the string "Howdy do" comes in from the remote service before going on to the next statement. The string can be surrounded either by straight or curly double quotes ("string" or "string") or straight or curly single quotes ('string' or 'string'). When the string arrives, ZTerm goes on to execute the next statement. Obviously you'd use this statement to test for prompts from a given service.

- **Wait for quiet 3 seconds**
  
  This form of the Wait command causes ZTerm to wait until three seconds go by during which there's no incoming data before going on to the next statement. You can, of course, replace the 3 with another positive integer.

- **Wait 5 ticks**
  
  We've already noted that a tick is 1/60th of a second. This form of ZTerm's Wait command is unconditional. Regardless of what is or isn't arriving at the serial port, it causes the program to wait the specified amount of time before going on to the next statement. As with the "Wait for quiet" statement just above, the unit of time can be either seconds or ticks.
Send

Send "Zebediah^M"

This statement does just what it says. In the example, it sends the string "Zebediah" followed by a carriage return just as if you’d typed it yourself. ^M, as we’ve noted several times elsewhere, is Ctrl-m, which happens to be a carriage return. You might use this statement to respond to a prompt for your first name. In a script, it might look as follows:

Wait for prompt “First name?”
Send “Zebediah^M”

Then, when a service sends First name? to you, you’ll send Zebediah^M to it.

Send $account “,” $password “^M”

A single Send command can send several strings one after the other, and it can also mingle the strings with a limited number of variables. Look at the window just below:

![Figure 13.1. ZTerm's main settings window. You fill it in once for every service in your directory.](image)

We crossed paths with it before when we talked about settings, and we noticed that for each service it lets you enter an account number and password. The account number for a given service is stored in a variable
(really just a storage compartment) whose name is $Account. The password for that service is stored in another variable called $Password. Accordingly, if your password for the service is Excelsior and you've entered it in the window in Figure 13.1, you can say either:

Send "Excelsior"

or

Send $Password

as both statements will do the same thing. The statement:

Send $account "," $password "AM"

mingles variables and strings. It sends your account number followed by a comma followed by your password and then a carriage return.

While we're on the subject, here's a table of all of ZTerm's built-in variables:

Table 13.1. Variables for ZTerm 0.9 Scripting

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Account</td>
<td>Service's account number as entered in the window in Figure 13.1.</td>
</tr>
<tr>
<td>$Password</td>
<td>Service's password as entered in the window in Figure 13.1.</td>
</tr>
<tr>
<td>$Service</td>
<td>The name of the current service.</td>
</tr>
<tr>
<td>$Date</td>
<td>The current date in the format MM/DD/YY.</td>
</tr>
<tr>
<td>$Time</td>
<td>The current time in hours, minutes, and seconds followed by AM or PM.</td>
</tr>
<tr>
<td>$Day</td>
<td>The current day of the month as an integer.</td>
</tr>
<tr>
<td>$dd</td>
<td>The current day of the month as an integer with zero fill (the fifth day of month appears as 05).</td>
</tr>
<tr>
<td>$mm</td>
<td>The current month as an integer.</td>
</tr>
<tr>
<td>$year</td>
<td>The current year.</td>
</tr>
<tr>
<td>$yy</td>
<td>The last two digits of the current year.</td>
</tr>
</tbody>
</table>

Beep

Beep

Beep 4 ticks
As you might expect, this statement gives rise to a beep. If your System is set to use the Mac's simple beep (the setting is made in your Sound control panel) you can also specify the length of the beep in ticks.

Display

Display "This script began running at " $Time

The format of the Display statement is the same as that of the Send statement, but the strings and variables you specify are sent only to your own terminal screen, not to the remote service.

Hangup

Hang up

This command hangs up your modem (in modem-ese, it causes your modem to go on-hook).

Open capture

Open capture $Service "Log " $Date

The Open capture command creates a plain-text file in the same folder as the ZTerm application. The file records everything that goes on in your terminal window after you initiate the Open capture. The command format is the same as Send and Display, which is to say that the capture file will be named according to the specified series of strings and variables (as long as you stay within the Mac's 31-character limit for file names). In the example just above, if you're calling a service named GEnie on January 1, 1993, the capture file would be assigned the name "GEnieLog 1/1/93."

Close capture

Close capture

The Close capture command simply closes the current capture file, so that the contents of your terminal window are no longer being saved to the text file you created with Open capture.

Comments

To help remind yourself of what you were doing when you originally wrote
a script, you can include as many comments as you like among the commands. Any line (that is, any paragraph) beginning with two hyphens (-- is the standard HyperCard label for comments) or with a number sign (# is the comment label used by MPW, Apple's software development system for the Mac) or with a semicolon is taken to be a comment and not a script command. Comments aren't executed. They're simply a convenience to the reader.

Executing a ZTerm 0.9 Script

Once you've written a script, you normally store it as a plain-text file in the same folder as the ZTerm application. It can then be executed in one of three ways:

- If you 1) define a service using the window in Figure 13.1 and 2) give the text file containing your script the same name as the service followed by a period and the three characters zts, the script will automatically be executed upon connecting with the service. If, for instance, you dial a service that you've named GENie and in the same folder as the ZTerm application you have a plain-text file called GENie.zts, ZTerm will try to execute the contents of the file as a script every time you dial up GENie.

- If you choose Run a Script... from under ZTerm's Macros menu, you'll get a simple version of the Mac's Standard Files dialog box in which you can select and open a plain-text file. If the plain-text file happens to be a ZTerm script, ZTerm will execute it forthwith.

- In the preceding chapter we saw how to configure ZTerm macros that send strings of characters. You can also run a script from a macro. Just make the macro string ® (type an r while holding down the Option key) followed by the name of a script that's already stored in the same folder as the ZTerm application. If you write a script and save it as a plain-text file called DoThisAndThat in the ZTerm folder and then assign the string:

®DoThisAndThat
to a macro, executing the macro will in turn cause the script
DoThisAndThat to be executed.

Sample Script

Inasmuch as we keep mentioning GENie, let's write a script that automatically logs us onto GENie without our having to trouble our pretty heads about account numbers and passwords every time we connect. When you dial GENie and establish a connection, GENie likes you to type hhh. And when GENie says type it means type, not send. It expects a tick or two of quiet time between each h, just as there'd be if you were actually typing. When GENie does receive your hhh, it prompts you with U#= for "user number equals." You're then supposed to respond to the prompt with your account number followed by a comma followed by your GENie password followed by a carriage return. While we're at it, let's also have our script log the whole session to capture a file.

To automate all that, we create a plain-text file called GENie.zts and save it in our ZTerm folder. As for the script itself, here's what it might look like (including embedded comments that explain what's happening):

```plaintext
-- Script to log onto GENie (this statement is ignored
-- because it begins with "--" which identifies it as
-- a comment. At this point we've already connected to
-- GENie, which now expects an hhh. We'll start by pausing
-- a couple of seconds to make sure GENie is ready for our
-- typing.
wait 2 seconds
-- Now we send the first h followed by a mini-interlude of
-- quiet to simulate real typing.
send "h"
wait 5 ticks
-- Same thing for second h
send "h"
wait 5 ticks
send "h"
-- Now we'll wait for GENie's prompt.
wait for prompt "U#"
-- We could just as well have waited for "#=" or even
-- just "=" since any subset of characters that uniquely
-- identifies the prompt will do. Now we'll send our
account
-- number followed by a comma followed by our password
-- followed by a carriage return.
```


send $account ";" $password " Amp"
-- Now we beep to alert ourselves to the fact that
-- we're online.
beep
-- Now we start our capture file. We'll make sure to
-- put the time into the file name so that if we call
-- GEnie again on the same day, the capture files for
-- each session won't have duplicate names.
open capture $Service " Log " $Date " " $Time
-- If we've called GEnie on the afternoon of February 1,
-- 1991, the Open capture statement just above would
-- result in our capture file being called:
-- GEnieLog 2/1/91 3:46:45 PM
--
-- Now let's inform ourselves that we're connected by
-- saying so inside our main terminal window.
display " Connected at " $time " AM"
-- The above statement would send " Connected at
-- 3:46:46 PM" to our own window but not to GEnie.

This script would then be executed every time we auto-dial GEnie through
ZTerm. The comments may be a little prolix, but there are only 11 real
statements in the script, and once the eleventh statement is executed we're on
our own. The capture file keeps capturing the contents of our terminal window
until we choose Stop Capture from ZTerm's File menu or quit the program.

More Sophisticated Scripts

Of all the procedures for which you're likely to want a script, basic service log-
ons pretty obviously fall at the head of the list. But suppose you want to do
something a little more sophisticated. Suppose that when you call GEnie you
often forget to notice what time it is. Before 6:00 p.m. on weekdays, GEnie
costs three times as much per minute of connect time, and it tells you so by
sending "PRIME TIME Rate in Effect." What if you want your script to do one
thing if GEnie's prime time rate is in effect (like beep five times and put up a
dialog window giving you the option of staying on or logging off immediately)
and another thing if it's not?

To accomplish that you'd do best to go to another program with a more
powerful scripting language. White Knight, for instance, would let you say:

ALERT1 PRIME TIME Rate/JUMPTO BigBucks
White Knight's comment lines begin with a left parenthesis. Starting here, White Knight will look for the incoming phrase "Prime time rate." If it doesn't find it, it goes on to execute the next command, which, in this example, waits for GENie's first prompt.

If White Knight DOES find the phrase "PRIME TIME Rate," instead of executing the next statement it will jump to the command just after the line labeled ":BigBucks" below.

PROMPT Enter #, <H>elp, or <CR> to continue? (Starting here we'd respond to GENie's first prompt and automate whatever else we want to automate.)

:BigBucks
(Beep five times to warn you that you're shelling out all that dough.)

BELL
BELL
BELL
BELL
BELL
(Etc., etc.)

Yes, in White Knight you say BELL instead of Beep and PROMPT instead of Wait for prompt, and strings aren't surrounded by quotes the way they have to be in ZTerm. But the salient thing is that White Knight lets you test incoming data in several different ways and then, depending on what conditions are met, it lets you take several different courses of action. Of course, the fact that White Knight's JUMPTO command enables you to jump all around your script can make for logical pathways so tortuous that once you've written a script you can't make head or tail of it. This is certainly one reason why the authors of MicroPhone II opted for what's called a structured command language. (HyperTalk is also a structured language.) Structured languages limit the amount of jumping you can do even if they let you test for every condition you can think of. Essentially, they demand that if a condition you're testing for is satisfied, you deal with it now and not jump off to some outlying byway of your script.

BTW, my own feelings about structured languages are very mixed. Yes, structured languages are probably good for you, but they're good
in the way that uncooked vegetables are good. Unstructured languages are the deep-fried food of computer programming, and sometimes you've just gotta have 'em.

White Knight is also unique among the popular commercial terminal programs in that it compiles your scripts. Where ZTerm directly executes a script as a series of text statements, White Knight converts your text scripts into binary packages of compressed code called procedure files, which it executes when called upon to do so. (During compilation, White Knight checks the syntax of all your statements and also does what it can to ensure that all your logical pathways can be traversed.)

As for recommending a terminal program to serve as the definitive platform for your most intricate scripts, I'm not altogether sure I want to cast a vote. Mind you, I'm under no pressure to remain unbiased—actually I like being biased—but White Knight, MicroPhone II, and Smartcom II all have their respective merits. I think I'd tell someone who already has a little programming experience to go with White Knight. The feel of its scripting language is reminiscent of some other programming languages, and for that reason the learning curve ought not to be too steep. MicroPhone II's scripting language is very much sui generis—extremely powerful, but not quite like anything I've ever seen. Smartcom II's Autopilot language (yet again, Smartcom has its own name for everything) most closely resembles plain old English, and reading a Smartcom II script is almost like reading ordinary prose. You may not find as many interesting Autopilot scripts listed in BBS and online service download areas as you would White Knight or MicroPhone II scripts, but Hayes maintains its own BBS (1-800-874-2937), and there are always a few good scripts available there.

If you can somehow put your hands on the manual for each terminal program, give yourself a sample read-through. That way you won't just see what all the scripting commands are, you'll also see what the manual is like. Choosing a script language you can live with should be based to some extent on finding a manual you can live with. And start by reading as many scripts as you can get yourself to swallow. Books and manuals are fine, but in the end scripting, like programming in general, is best learned from example.
Chapter 14

Using High-Speed Modems

Amateur telecommunicating has recently become a whole new ball game. The appearance on the hardware market of inexpensive modems capable of raw speeds—speeds not enhanced by data compression—on the order of 14,400 bps changes everything. Not long ago 9600 bps was a critical dividing line. Below it, generic 2400 bps modems could be had for a song—and often a pretty simple tune at that. Between 2400 and 9600 bps there was, and is, a void. Then, starting at 9600 bps, prices increased several-fold. Only the most serious telecommunicators—mainly sysops, BBS fanatics, people who transferred data in the course of business, and the idle rich—sprang for 9600. Everyone else made do with less.

The high-end modem field was dominated by three manufacturers. Even before use of the CCITT V.32 modulation protocol for 9600 bps became widespread, U.S. Robotics offered 9600 bps in its Courier HST modem. HST (for High Speed Technology) was U.S.R.'s proprietary modulation protocol, and the original Courier could make a 9600 bps connection only with another HST modem. At around the same time—we're talking about the mid-1980's—Telebit brought out its TrailBlazer, which used a proprietary protocol called
PEP (for Packetized Ensemble Protocol). Then Hayes introduced its V-Series 9600, which used still another proprietary protocol called Express 96. All three modems could attain 9600 bps or more, but their protocols weren't compatible, and a modem of a given brand could make a high-speed link only with another modem from the same manufacturer.

That began to change when it became increasingly clear that V.32 and its 14,400 bps successor V.32bis would eventually displace the various proprietary modulation protocols in much the same way that VHS video cassette format has ousted Sony's Betamax. U.S. Robotics responded by introducing its Dual Standard Couriers, which supported both HST and the CCITT's protocols, and Telebit and Hayes promptly followed suit. (See Appendix D for more on these companies' current offerings.)

Now Zoom and Supra (along with that faceless entity, Generic, Inc.) have entered the high-speed modem picture in a big way. Bought "naked" (without software or cable), the Zoom VFX V.32bis can be had from several discount mail-order houses for under $300, barely more than a 2400 bps MNP modem cost just a couple of years earlier. And that's 14,400 bps on top of 9600 bps send/receive fax! The SupraFaxModem V.32bis sells for a few dollars more, but in its Macintosh packaging it's bundled with the STF fax software, which a fair number of heavy fax users have come to prize (to the extent that you can prize any current fax software for the Mac). The end result is that a set of performance specifications which once cost around a thousand dollars per unit have suddenly become available for a few hundred dollars per unit, an outlay that a serious modemer can expect to amortize in reduced telephone costs over just a few months.

Are the cheap V.32bis modems as good as their standard-brand counterparts? I'd have to say they're not. CompuServe and some of the other big online services have been abuzz with complaints. Too many of the first units shipped haven't worked at all; others have "worked" but not performed as advertised. Still, low-end manufacturers do tend to honor their warranties, and Supra has made a particularly valiant effort to man its user support areas on CompuServe and GEnie, sometimes in the face of markedly unfriendly fire. All such new offerings are bound to have their bugs, and as you read this the first cries of outrage will have abated some. The majority of those first complaints had to do with modems not being able to negotiate connections with certain kinds of other modems, and for the most part the bugs were
neither electrical nor mechanical. Their origins were in the firmware.

Firmware? It's the computer code that's stored in ROMs, read-only memory units. (Since ROMs differ from RAM in that they can be read from but not easily erased or written to, the code can't be purged from memory the way that software can, but it's not wired permanently into the hardware either; so, being neither soft nor hard, it's firm.) Actually, every Mac contains at least one firmware-laden ROM, and it's firmware that runs your Mac when you first boot up. The firmware tests your RAM, puts up the happy Mac face (or the sad Mac face if your computer fails the tests), then makes sure there's a System File on your startup volume before turning the reins over to the software. Like the Macintosh itself, most high-speed modems contain firmware-bearing ROMs. Their firmware negotiates connections with remote modems and participates in error checking and data compression and decompression as well as any number of other functions.

In fact, for our present purposes I've decided to include among the ranks of high-speed modems any 2400 bps units that support protocols like MNP, V.42, and V.42bis. Maybe their speeds aren't all that dazzling, but the fact that they can negotiate a connection—that they can ask the remote modem "Do you also support V.42bis or only V.42 or MNP and, if so, what level of MNP?"—gives them enough in common with the big boys to earn a place in this chapter.

**Modem Protocols**

We've touched on modem protocols before, but this is a good time to visit them again. A protocol, you'll remember, is a set of conventions that governs some aspect of a modem's operation. Suppose, for instance, that two modems connect and that they're both capable of data compression and decompression. There are several different compression schemes around, and if one modem compresses data according to scheme A before sending it out only to have the other modem try to decompress the data according to scheme B, the likelihood of achieving a successful transmission begins to plummet toward the infinitesimal. If they both observe the same compression protocol, however—if, in other words, they both abide by the same set of rules—the chances of the data getting through intact become extremely good.
Three aspects of the operation of high-speed modems are protocol-dependent: modulation, error checking, and data compression. So for each of those functions, let’s run through the dominant protocols.

**Modulation Protocols**

It’s safe to say that the CCITT protocols V.32 and V.32bis have become worldwide standards for high-speed modem modulation. V.32 covers both 4800 and 9600 bps, full- and half-duplex, on voice-quality lines. V.32bis encompasses all of V.32 and adds 7200, 12,000, and 14,400 bps. It also includes a handshaking scheme in which two connected modems can rapidly change transmission speeds to adjust to a sudden increase (or decrease) in phone line noise. V.32 has a similar provision, but a mid-transmission speed change that can take as many as 10 seconds during a V.32 connection might not take much more than 100 milliseconds with V.32bis. All V.32 and V.32bis modems have top raw speeds of 9600 and 14,400 bps respectively. If they can’t attain those speeds, they aren’t V.32 or V.32bis modems.

**Error-Control Protocols**

It’s also safe to say that V.42, once again a CCITT protocol, has become the worldwide standard for in-modem error checking. In a V.42 connection, the sending modem divides the outgoing data into fixed length strings called packets and sends a CRC value along with the packet. (Look back at the description of the software-based XModem file transfer protocol in Chapter 9 to refresh yourself on CRC.) The remote modem then recomputes each packet’s CRC value and compares it to the one it just received. If the values differ, the remote modem tells the sending modem to resend the packet.

A similar procedure is built into MNP Level 4, an older error-checking protocol, and V.42 also includes (and is therefore downwardly compatible with) MNP-4. But if a V.42 modem connects to another modem, it will first try to negotiate a LAP-M connection. LAP-M stands for link access procedures for modems, and on noisy lines it yields better performance than MNP-4. If, in the course of negotiations, the V.42 modem discovers that the other modem supports MNP-4 but not LAP-M, it can fall back to MNP-4 to accommodate the other modem. V.42 can therefore be seen as a super-protocol that supports two sub-protocols, LAP-M and MNP-4.
It's important to emphasize that in-modem error checking protocols are \textit{not} the same thing as software-based error checking protocols like XModem or ZModem. ZModem's error checking takes place only during a ZModem file transfer, not when you and the remote service are typing back and forth. \textit{Modem-based error checking takes place throughout a given session}, regardless of what your software is doing. It neither knows nor cares if you're transferring a file, keying in replies to your E-mail or responding to a prompt. Every character you receive or send is checked by the participating modems. That's why a connection between two modems that's being overseen by an error-checking protocol is called a \textit{reliable link}.

\textbf{Data-Compression Protocols}

With in-modem data compression, another CCITT protocol, V.42bis, can now be said to have become the worldwide standard. MNP-5, an older compression protocol that's still in widespread use, is supported by nearly all V.42bis modems, but it's not a mandatory component of V.42bis in the way that MNP-4 is integral to V.42. So if two compression-capable modems connect and one of them supports V.42bis, they'll normally try to establish a V.42bis connection first. Only if that fails will they fall back to MNP-5. V.42bis is preferred, because it usually makes for tighter compression (up to 4-to-1, as compared with MNP-5's real world maximum of 2-to-1), and because it handles noisy lines and software-compressed data more efficiently. And since the V.42bis compression protocol can only work in conjunction with the V.42 LAP-M error-checking protocol, you can safely conclude that any V.42bis modem also supports V.42. (You can conclude with equal safety that any MNP-5 modem also supports MNP-4.) What you \textit{can't} conclude is that a V.42bis modem supports V.32 or V.32bis. Current standards for CCITT compression protocols are independent of any modulation protocols.

It's worth pointing out here that, as tight as V.42bis compression may be, it rarely beats the better software-based compression schemes for sheer efficiency. If you were to send the same file twice to the same service, first as is using V.42bis (so that compression is left entirely to the modem) and then compressed with, say, StuffIt Deluxe's "Better Compression" scheme but \textit{not} using any in-modem protocols (so that compression is accomplished only by your software), the software-compressed file would probably take measurably less time to transmit than the hardware-compressed file. On the other
hand, the software-compressed file would need to be compressed by you before it's sent and decompressed by your uncle Louie after he receives it, and both those procedures consume computer time even if they don't eat up telephone time. On the third hand, software-compressed files occupy a lot less room on your hard disk. Given all those factors, do you need in-modem compression at all? Maybe yes, maybe no. It all depends on how you like to use your modem, your time, and your income.

**Acquiring a High-Speed Modem**

Now that the prices of new high-speed modems have come down so sharply, prices of used high-speed modems have begun to follow suit. And while there's nothing intrinsically wrong with buying a good used modem, you ought to be aware that in some respects a used modem may have more in common with a new modem that's been sitting on some dealer's shelf for a couple of years than with a new modem that was shipped from the factory last week. To wit, the features you can expect to find in a modem of recent manufacture may not be present in a modem made a few years ago.

With 2400-bps modems, it's all fairly simple. V.42bis modems also support V.42, which means they also support MNP-4 and probably MNP-5. In the 9600-bps domain, however, things get a lot more complicated. Some older 9600 bps modems don't support any of the CCITT protocols. They may use proprietary protocols for everything—modulation, error checking (if they support it), and data compression (if they support that). (Several older Telebits and Couriers achieve 9600 bps only through the PEP and HST proprietary protocols, which limits your high-speed connections to other modems of the same stripe.) Personally, I'd advise steering clear of any such equipment. Nowadays, if you're going the 9600 bps route, there's every reason to have V.32 at the very least. Some V.32 modems—the original Hayes SmartModem 9600 is an example—don't offer built-in error checking or compression. Of course you can live without either, but you may not want to. Many older MicroTech 9600-bps modems support V.32 for modulation, but for error checking and compression they use various of the Microcom proprietary protocols (the MNPs all the way up through MNP-9) and not the CCITT protocols.
As regards fax capability, it's seldom available in older high-speed modems, but newer models—especially the cheaper ones, oddly enough—very often have it. And even with fax it's worth keeping your eye peeled. Some modems can only send faxes, while others offer send/receive. Some modems send and receive faxes at 9600 bps, while others send at 9600 but receive only at 4800 bps.

Given the many combinations of features you can find out there, I suggest that before you buy any high-speed modem, new or used, you thoroughly inspect its specifications. If the would-be seller of a used modem can't provide a reliable set of specs, get them from the manufacturer. Nowadays it's better to have CCITT protocols than any of the various proprieties, which will almost certainly begin to vanish in the fullness of time. I'm guessing that the proprietary protocol that hangs on the longest will be MNP, and an MNP-4 connection is as reliable (if not as noise-compliant) as one that's regulated by V.42. Still, right now the CCITT is in the driver's seat, and there's little point in having a Betamax video cassette player when nearly all the tapes at the local rental library are VHS.

**Still More Settings**

**Speed**

Several times we've mentioned the negotiations attempted by high-speed modems when they connect, and it's because you can't always predict the outcome of these negotiations that you have to give special attention to your terminal program's speed settings. We'll be talking about two different speeds here, so let's be clear about our terms. *Carrier speed* refers to the raw speed at which data travels between two connected modems. *Serial port speed* is the speed at which data travels between your modem and your Mac. When a pair of modems start shaking hands, you won't always know beforehand what carrier speed they'll agree on. But when you instruct your terminal program to set your serial port to a particular speed, that will be the speed at which data travels between your modem and your computer.

The prime consideration in all this is that you want your serial port speed to be as high or higher than the greatest speed at which your modem is liable to send data to your computer. If your serial port is set to a lesser speed than
that, you may wind up not taking full advantage of your modem's capabilities. If, for instance, your modem is capable of sending data to your computer at 19,200 bps but your serial port is set at only 9600 bps, your serial port will become a bottleneck.

If data compression never existed, a V.32bis modem would still be capable of attaining carrier speeds of up to 14,400 bps. It might negotiate a lower speed if it finds itself trying to talk to a slower modem or even to another V.32bis modem over a noisy line, but it also might not negotiate a lower speed, so if you're using a V.32bis modem your serial port should be set to a minimum of 14,400 bps in order to use your modem most efficiently. In general, notwithstanding compression, your serial port ought to be set at least to your modem's maximum carrier speed (14,400 bps for V.32bis, 9600 bps for V.32), and it's in your terminal program that you make the setting.

But now let's bring in compression. We've said that in-modem data compression can achieve ratios approaching 4-to-1, which means that at a carrier speed of 9600 bps your modem might try to pass incoming data along to your computer at 38,400 bps. So now, if you're still going to use your modem most efficiently, you need to tell your terminal program to set your serial port to 38,400 bps.

It may appear to follow from what we've just said that if you're running a V.32bis modem at its top speed of 14,400 bps you'd want to set your serial port to 57,600 bps, but here's where we run athwart some caveats. Most terminal programs are happy to let you set your serial port to 57,600, but a number of modem manufacturers report that speeds as high as that have made for glitches at the computer end. They claim that at speeds greater than 38,400 bps, members of the Mac II family haven't always been able to service their modem ports with ironclad reliability. Also that at speeds above 19,200 bps, Macs in the SE, Plus, and Classic class have exhibited similar problems. Such glitches tend to be sporadic, but you may want to take those warnings into account when matching your serial port speed to your high-speed modem.

If the foregoing were boiled down into a rule, it would go something like this: set your serial port to your modem's maximum carrier speed if the modem doesn't support compression (no V.42bis or MNP-5) and to four times your modem's maximum carrier speed if it does support compression—but don't go higher than 19,200 bps if you have a Mac Plus, SE, Portable, Power-
Chapter Fourteen  Using High-Speed Modems

Book 100, or Classic, and don't go higher than 38,400 bps if you have a Mac Classic II, LC, II, IIx, IICx, or PowerBook 140 (and possibly even a IIsi, IICi, or PowerBook 170). Clear enough?

Local Flow Control

Since it's possible for data to flow from your high-speed modem to your Mac at a rate that exceeds your terminal program's ability to handle it—or from your Mac to your modem at a rate that exceeds the modem's sending ability—there arises the need for some form of local flow control on the serial port to ensure that neither your computer nor your modem becomes overwhelmed. (The most common symptom is data loss; newly arriving data simply overwrites older data that hasn't yet been processed.) If you've read Chapter 4 or Appendix C, you'll know that flow control can be hardware-based or software-based. With software-based flow control—also called XON/XOFF flow control—the XOFF character serves as a red light: it's sent either by the computer to the modem or the modem to the computer when data starts coming in more rapidly than it can be processed. Whoever receives the XOFF character stops sending data until it receives an XON character, which serves as a green light. With hardware-based flow control, also known as RTS/CTS flow control, it's the RTS (request to send) and CTS (clear to send) lines in the RS-232 serial connection between the modem and the Mac that serve as red and green lights. By varying the voltage on those lines, either the computer or the modem can similarly halt or restart the flow of data.

For high-speed modems, RTS/CTS handshaking is the preferred method, which means that a Mac Plus or better is the preferred computer (because earlier models don't support RTS/CTS flow control) and a hardware handshaking cable running between modem and Mac is the preferred cable (since garden variety modem cables don't support the RTS and CTS lines). So if you have a high-speed modem, a hardware handshaking cable, and a Mac Plus or better, you'll normally instruct your terminal program to use hardware handshaking.

What if, however, you have a high-speed modem but you don't have any of the other preferred items? What if your computer is a Mac 512 and/or your modem cable is comparably antiquated? Does that mean you're up the creek? Fortunately, it means nothing of the kind. All it means is that you'll have to observe a few extra precautions and take care of a little extra business. First
off—I’ll begin with the medicine that’s hardest to swallow—you’ll have to read your modem manual to learn which of its commands turns on XON/XOFF flow control. (Later in this chapter I’ll try to point you in the right direction.) Yes, I know how unappealing it is to read your manual, but some of the newer ones aren’t as hideous as modem manuals used to be, and rumor has it that when the Khmer Rouge take prisoners of war they no longer soften them up for interrogation by making them read modem manuals for several hours at a stretch. Once you know the command that enables XON/XOFF flow control, you’ll have to make sure it gets sent to your modem every time you initialize it for a high-speed connection. This means you’ll either type it each time or, more sensibly, you’ll create a settings file or auto-dial setup that includes the command in the initialization string it sends the modem for a given service.

**BTW,** those of you who own CompuServe Navigator, a special-purpose commercial terminal program designed for heavy CompuServe users, should be aware that it doesn’t support hardware handshaking even if your cable and computer do support it. When using Navigator with a high-speed modem, you should likewise initialize the modem with a “use XON/XOFF” command.

The extra precautions you need to take concern your choice of software protocols when you transfer files. The thing you want to avoid is the appearance of XON and XOFF characters in the data stream, which is perfectly possible even though no one ever intended to send them. A binary file is theoretically capable of containing all possible bit patterns, and now and then some of those bit patterns may turn out to correspond exactly to the patterns that define the XON and XOFF characters. The Zmodem software protocol guards against such occurrences by quoting XON and XOFF—flagging them so they’re treated as data and not as flow control signals—and the CompuServe B+ protocol does much the same thing. But XModem and YModem don’t do it, and for that reason they should both be scrupulously avoided when you’re using XON/XOFF flow control.

**Initializing Your High-Speed Modem**

We discussed elementary initialization strings in Chapter 6 on the Hayes Language and again in Chapter 12 on automating your terminal program, but
these discussions were mostly outside the high-speed modem context. Putting together an initialization string for an old-style, non-protocol-driven modem barely qualifies as child's play. Putting one together for a high-speed modem may require enormous patience leavened by a touch of genius. Virtually all modems uniformly understand basic Hayes language, but the special commands for high-speed operation are neither uniform nor basic, and local dialects abound. Consequently, the following recommendations will necessarily be guidelines, not recipes.

What most complicates the initialization of high-speed modems is the sheer number of variables you may want to control depending on the service with which you hope to connect. Do you want your modem to negotiate all the protocols of a connection—speed, error checking and compression—or do you want to lay down the terms of the connection in advance? Some remote modems may not like the way your modem negotiates when it's given free rein and may disconnect you every time you call. It may sometimes be necessary to preset the speed of the connection and let your modem negotiate only the error checking and compression protocols. Do you want to set your modem for hardware handshaking, software handshaking, or no handshaking at all? And what about your modem's factory settings? Is there more than one group of factory settings, and if so which one should you use at what time? Even if a remote modem will successfully negotiate with yours for a very high connection speed, is that the speed you really want to use? Commercial online services charge a great deal more for 9600 bps than for 2400 bps, and your wallet may demand that you throttle back. How do you account for all those possibilities when so many different high-speed modems speak so many different dialects of Hayes?

For me it would be easiest to tell you still again to read your modem manual until you thoroughly understand your modem's entire command set, but to insist on that is like insisting that you push a bunch of bamboo slivers underneath your fingernails. So let's look at what your other options are, starting with the simplest.

1. **Spend some money and buy the latest version of MicroPhone II.** MicroPhone II comes from a firm called Software Ventures, and the people there spend a lot of time playing around with, and probably (God help them) even reading the manuals of, a great many modems. When they feel they've caught a modem's essence—mas-
tered the settings it likes best and the Hayes dialect it understands—they write *modem drivers*, scripts that serve as go-betweens. What the driver goes between are you and your modem, which is to say that you tell MicroPhone II what speed and handshaking settings you want and which modem driver to use, and from that point onward MicroPhone II uses the driver to talk to your modem in its native dialect. (It's analogous to the way your other Macintosh applications use printer drivers—or Chooser extensions in System 7-ese—to talk to your printer in its native dialect.)

Figure 14.1 shows the current set of MicroPhone II drivers as of this writing, but be aware that the elves at Software Ventures continually refine their drivers and also create new ones. Additions and revisions to the current set are usually posted in the communications libraries of CompuServe, GEnie, and America Online, as well as on Software Ventures' own BBS (phone # 510-849-1912).

![Figure 14.1. Modem drivers from MicroPhone II, version 4.0. You select the one you want to use in MP II's Communications Settings window.](image)
Currently there are 18 MicroPhone II drivers in all, but some support more than one modem. Here's a list of the supported modems:

**Standard Driver**
- Apple Personal Modem 1200
- Avatex 1200HC
- Hayes SmartModem 1200

**Standard (autobaud) Driver**
- Anchor Automation 2400E
- Applied Engineering DataLink 2400
- Avatex 2400
- Bizcomp Intellimodem 2400
- Everex 2400
- GVC Technologies SM24
- Hayes SmartModem 2400
- Intel 2400EX
- La Cie 2400 Mac External
- Mirror Technologies 2400
- Practical Peripherals 2400SA
- Shiva NetModem 2400
- SupraModem 2400
- Zoom 2400

**MNP Standard Driver**
- Abaton Fax 96/24
- Cardinal 2400 MNP
- Applied Engineering DataLink 2400 MNP
- Intel 2400EX MNP
- Mirror Technologies 2400 MNP
- Practical Peripherals 2400SA MNP
- Many generic 2400 bps V.42/V.42bis modems

**No MNP Driver**
The same modems as those listed under the MNP Standard Driver when you don't want to try for an MNP connection
Telecommunications: The Macintosh Modem Book

- **V.32 Standard Driver**
  - UDS FasTalk V.32
  - Mirror Technologies 9600 V.32
  - Some generic V.32bis modems

- **Hayes V-series/Ultra Driver**
  - Hayes Optima
  - Hayes V-series
  - Hayes Ultra 96/14.4

- **Prometheus V.32 Driver**
  - Lightspeed 9624E
  - Prometheus ProModem 9600
  - Prometheus Ultima

- **Telebit T2500 Driver**
  - Telebit TrailBlazer
  - Telebit TrailBlazer Plus
  - Telebit T1000
  - Telebit T2000
  - Telebit T2500

- **Telebit T1600 Driver**
  - Telebit T1600
  - Telebit T3000
  - Telebit QBlazer

- **USR Courier Driver**
  - USR Courier HST
  - USR Courier Dual Standard
  - USR Sportster V.32

It's reasonably safe to say that other modems will be covered soon.

2. **Buy a modem that has useful factory settings.** Many high-speed modems are able to save one or more factory configurations in non-volatile memory (memory that isn't erased every time you turn your modem off). Both the new Zoom VFX V.32bis and the SupraFaxModem V.32bis support this feature. When you power up either modem, it defaults to the factory profile. And if you issue an AT&F command to the Zoom or an AT&F1 to the Supra, it causes the full factory profile to be restored.
Typically, these profiles configure your modem in a way that allows you to use all of its best attributes in the “normal” fashion. Sending an AT&F to the Zoom VFX V.32bis, for instance, sets it up to use hardware (RTS/CTS) flow control and, when it connects to another modem, to negotiate for the highest mutually acceptable carrier speed (14,400 bps with fallback to 12,000, then 9600, etc.), the best mutually acceptable error control scheme (LAP-M with fallback to MNP-4, then to no error control at all) and the best mutually acceptable in-modem data compression scheme (V.42bis with fallback to MNP-5, then to no compression).

However, initiating all those negotiations can sometimes make for problems. Not every modem will be able to handle them, and it’s not uncommon for the remote modem to disconnect in the middle of negotiations in the same way that Soviet diplomats used to walk out in the middle of U.N. sessions. Even if the negotiations appear to have come to a successful conclusion, you may find that the remote modem sends you a long string of garbage characters.

When those things happen, the best idea is often to experiment by cutting back on all the many connection possibilities that your modem presents to the remote modem.

Here are some rough guidelines for initialization strings that set some limits for the handshaking that occurs when two modems first connect. They’re based on the command set for the Rockwell chips that control the Zoom and Supra, and your own modem may want things put a little differently. (So you’ll have to read your modem manual after all.) But the sense of what the strings accomplish ought to, I hope, be plain enough.

Start with carrier speed. In many modems it’s controlled by the contents of an S-register, commonly S37. We’ve already noted that the values stored in S-registers govern various aspects of a modem’s operation. We’ve also noted that you set an S-register to a given value with the command ATSn=x, where n is the number of the S-register and x is the value you want to put into the register.

With many high-speed modems, a zero in S37 lets the modem negotiate for the highest mutually acceptable carrier speed. But on the next page are some other possible values for S37.
S37=12  Try to connect at 7200 bps (V.32bis only)
S37=11  Try to connect at 14,400 bps (V.32bis only)
S37=10  Try to connect at 12,000 bps (V.32bis only)
S37=9   Try to connect at 9600 bps
S37=8   Try to connect at 4800 bps
S37=6   Try to connect at 2400 bps
S37=5   Try to connect at 1200 bps
S37=3, 2 or 1 Try to connect at 300 bps

To set S37 to 9, you can issue an ATS37=9 command all by itself or put S37=9 into an initialization string.

Many modems use the Nx command in conjunction with S-register number 37. N1, usually the default for high-speed modems, allows negotiations to take place at any speed of which both connected modems are capable. NO (that's a zero) permits handshaking to occur only if the carrier speed is the speed prescribed by S37. Still another modem command that affects negotiation is the &Qx command. Fairly commonly, &Q5 tells the modem to negotiate mutually acceptable error-control and/or compression protocols, while &Q6 tells it not to use such protocols.

Now let's put all that together. When you first connect with another modem, your modem may ask the remote modem for too much speed or for an error checking or compression scheme it's never heard of. In your experimentation, you'll want to put the remote modem at its ease. So instead of using all your high-powered factory settings, you can pull back on your own modem with strings like these:

**AT&Q6N1^M**  Negotiate connection speed (N1) but not compression/error checking (&Q6).

**AT&Q6S37=11N0^M**  Connect at 14,400 bps only (S37=11N0). No error control and/or compression protocols (&Q6). This applies only to V.32bis modems, because V.32 doesn't handle 14,400 bps.

**AT&Q5S37=10N0^M**  Connect at 12,000 bps only (S37=10N0). Negotiate error-control and/or compression protocols (&Q5). This ap-
plies only to V.32bis modems, because V.32 doesn't handle 12,000 bps. It does apply to the Zoom Turbo, which straddles the _terra incognita_ between V.32 and V.32bis.

**AT&Q6S37=10N0^M** Connect at 12,000 bps only (S37=10N0). Don't negotiate error-control and/or compression protocols (&Q6). This applies only to V.32bis modems and the oddball Zoom Turbo, because V.32 doesn't handle 12,000 bps.

**AT&Q5S37=9N0^M** Connect at 9600 bps only (S37=9N0). Negotiate error-control and/or compression protocols (&Q5).

**AT&Q6S37=9N0^M** Connect at 9600 bps only (S37=9N0). Don't negotiate error-control and/or compression protocols (&Q6).

**AT&Q5S37=6N0^M** Connect at 2400 bps only (S37=6N0). Negotiate error-control and/or compression protocols (&Q5). _You might use this string if you're dialing an online service such as GEnie and want to force a less-expensive 2400 bps connection that still provides for error checking._

**AT&Q6S37=6N0^M** Connect at 2400 bps only (S37=6N0). Don't negotiate error-control and/or compression protocols (&Q6).

**AT&Q5S37=5N0^M** Connect at 1200 bps only (S37=5N0). Negotiate error-control and/or compression protocols (&Q5).

**AT&Q6S37=5N0^M** Connect at 1200 bps only (S37=5N0). Don't negotiate error-control and/or compression protocols (&Q6).
AT&Q6S37=1N0&M Connect at 300 bps only (S37=5N0).
Don't negotiate error-control and/or compression protocols (&Q6).

I can't guarantee these strings will work for your particular modem, but if you catch the sense of what they're trying to accomplish, you may be able to cobble together some equivalent strings of your own. You will, I'm afraid, need to consult your manual.

3. **Try the strings below, paying special attention to your modem's flow control commands.** I've assembled some "starter" strings for a number of different modems, and below I'll list the ones in which I feel most confident. All are designed to let the modem negotiate for its highest speed and for the best error-control and compression scheme of which it's capable. With each string I've also pointed out how hardware handshaking can be turned off in case you don't have a hardware handshake cable.

- **Most "standard" MNP and V.32 modems, most 2400 bps V.42/V.42bis modems, and even a few V.32bis modems.** Included in this category are such modems as the Practical Peripherals PM2400SA with MNP-5 and/or V.42bis and the Intel 9600EX and 96/14.4. Try:

  AT E1 Q0 V1 X4 &C1 &D0 \G0 \J0 \N3 \Q3 \V1 S0=0

  To turn off hardware flow control, change the \Q3 to \Q0. To use XON/XOFF flow control, change the \Q3 to \Q1.

- **Multimodem V.32 and Farallon V.32.** Try:

  AT E1 Q0 V1 X4 &C1 &D0 $BA0 &E1 &E4 &E10 $R1 S0=0

  To turn off hardware flow control, change the &E4 to &E3. For other forms of flow control, check your manual for other values that can follow &E.

- **Hayes V Series and Optima modems, Practical Peripherals 9600SA, and 9600/14/4 modems and the Zoom V.42.** Try:

  AT E1 Q0 V1 X4 &C1 &D0 &K3 &Q5 S0=0 S36=7
To turn off hardware flow control, change the &K3 to &K0. To use XON/XOFF flow control, change the &K3 to &K4.

**Prometheus V.32 and V.32bis modems.** Try:

```
AT E1 Q0 V1 X4 &C1 &D0 *S1 *E1 *F3 *Q2
```

To turn off hardware flow control, change the *F3 to *F2. For other forms of flow control, check your manual for other values that can follow *F.

**USR Courier modems.** Try:

```
AT B0 E1 Q0 V1 X6 &A1 &B1 &H1 &I0 &M4 &N0 &R2 &S0 S0=0
```

To turn off hardware flow control, change the &H1 to &H0 and the &R1 to &R0. For other forms of flow control, check your manual for other values that can follow &H and &R.

**Telebit TrailBlazer, TrailBlazer Plus, T1000, T2000 and T2500 modems.** Try:

```
AT E1 Q0 V1 X3 S0=0 S50=254 S51=255 S58=2 S66=1 S68=255 S95=2 S111=0
```

To turn off hardware flow control, change the S58=2 to S58=0. For other forms of flow control, check your manual for other values that can be assigned to S58.

**Telebit QBlazer, T1600, T3000.** Try:

```
AT E1 Q0 V1 X12 &C1 &D0 S0=0 S58=2 S68=255 S111=255 S180=2 S181=1
```

To turn off hardware flow control, change the S58=2 to S58=0. For other forms of flow control, check your manual for other values that can be assigned to S58.

**Supra V.32/V.32bis Modems.** Try:

```
AT &F1 &K3
```

To turn off hardware flow control, change the &K3 to &K0. To use XON/XOFF flow control, change the &K3 to &K4.
Zoom V.32/V.32bis Modems. Try:

```
AT &F &K3
```

To turn off hardware flow control, change the &K3 to &K0. To use XON/XOFF flow control, change the &K3 to &K4.

For all high-speed modems, when you’re using either RTS/CTS or XON/XOFF flow control make sure to set your serial port speed according to the recommendations in the section on speed that appears earlier in this chapter. To reprise what I’ve already said about CompuServe Navigator, it differs from general-purpose terminal programs in that it doesn’t support hardware handshaking, so when and if you’re using it set your serial port speed to two or three times your connection speed and initialize your modem for XON/XOFF flow control.

4. **Seek viable initialization strings wherever you can find them.**

   Recommending initialization strings is asking for trouble. It’s fairly safe to pass along a string you’ve tested thoroughly to someone who uses the same terminal program that you use yourself and whose modem is identical to yours down to the version of the firmware in the ROMs (two physically identical modems can contain different firmware). But vouching for initialization strings that don’t meet all of those conditions is a little like vouching for tips on who’s going to win the fifth race at Pimlico. Like commercial software, my tips come with no implied warranty.

   Keeping that in mind, some places to which you can turn for initialization strings include: 1) your modem’s manufacturer, 2) the Macintosh communications message bases on CompuServe, GEnie, or America Online, 3) the FidoNet MACNETCOM echo on any Mac BBS that carries it, 4) the local message base of any good Mac BBS, and 5) your modem manual. With any or all of them—even with this book—a touch of luck will help.
Modem Connect Responses

When your modem succeeds in making a connection, it normally tells you so by sending a message to your computer. The message shows up in your terminal window in a form called a verbal response. Non-high speed modems typically respond with "CONNECT 2400" when they've made a 2400 bps connection. If an MNP modem has been able to negotiate a reliable link (a connection governed by MNP-4 error-control) it might say "CONNECT 2400/REL." A V.42 modem might put it slightly differently by saying "CONNECT 2400/ARQ." If you take to writing scripts for your terminal program, you may need to examine those responses to determine what kind of connection you've arrived at. Your terminal program may examine them even if you don't, and when it examines them it may want them in a particular format.

ZTerm is a good example. When you use its auto-dialing feature, it waits for a connect response by looking for the string CONNECT. With many high-speed modems, however, it's possible to send an initialization string which invokes a response set that uses the word CARRIER instead of CONNECT, and if you do that while using ZTerm, ZTerm won't know that you've connected even after you've indeed connected. And when that happens it may try to redial the number, despite the fact that you're already online with the service you've just called.

Most MNP-only modems and some older V.32/V.32bis modems don't give you all that much control over verbal responses. If they connect at 2400 bps without error-control, they'll say only "CONNECT 2400." If they connect with error-control, they will or won't add a /REL or /ARQ after the 2400 depending on whether you've initialized your modem with a \V1 or \V0 respectively (\V1 is the normal default). As for the 2400, such modems always report the carrier speed and not the serial port speed when they respond, so ZTerm users needn't touch a thing.

Many newer high-speed modems, however, can respond in several different ways when they connect. For one thing, they may let you choose whether the connection speed they report is the carrier (modem-to-modem) speed or the serial port (local modem-to-local computer) speed. For another thing, they may let you choose between a one-line response such as "CONNECT 2400/ARQ," or a detailed response like:

CARRIER 14400
Among the latter variety of modems, a great many use S-register 95 to control which responses they send. In such a modem, if you want full result messages with the CONNECT response reporting modem-to-computer speed, you should include $95=44$ in your initialization string. If you want the same thing except that the CONNECT statement reports *carrier* speed, include $95=45$ in your string. If all you want is a simple one-line report with CONNECT followed by carrier speed followed by /ARQ if some form of error-control has been achieved, try $95=3$.

And yes, there is an element of alchemy in brewing up initialization strings for high-speed modems. And yes again, you’ll get it right eventually, even if you have to spring for MicroPhone II (provided, of course, that one of its drivers happens to work with your modem).

May good fortune be with you in all your high-speed ventures.
Voice Calls, Faxes, and Networks

Voice Calls

If there’s a modem eating up valuable footprint space on your desk, you may as well use it even when you’re not transferring files or making or receiving data calls. In a pinch all Mac peripherals can serve as paperweights or boat anchors, but you can also make voice calls only with your modem. In fact, in conjunction with your Mac and some decent software, your modem can be a better voice call dialer than any piece of specialized voice call hardware I know of. There are, to be sure, all kinds of elaborate telephones that store the numbers you use frequently and dial them at the touch of a button, but your Mac can do that too, and the number of numbers it can store is limited only by the size of your hard disk. What’s more, on the Mac it’s a breeze to edit name/address/phone number entries, and you can print envelopes or address lists whenever the impulse overtakes you.

All manner of heavy-duty business software such as More and Panorama support voice dialing. (Panorama conceals its dialing feature by not including it on any of its menus, but if you keep your phone numbers in a Panorama database you can dial them from a macro by using Panorama's DialModem...
macro command; see your manual for the command format.) Most versions of HyperCard are distributed with an address stack that lets you dial voice calls, and several more sophisticated phone number stacks are available as shareware on BBSs and online services. Two popular ones are called PhoneLog and Phone Book.

Of course, when you're not using System 7.0 or MultiFinder, voice call dialing isn't much of a convenience if you have to quit your application and then open More or HyperCard or Panorama just to make a single call. But there's a shareware Desk Accessory/F-Key combination called Dialer that lets you dial a number from any application simply by selecting the phone number and typing the F-Key. (You invoke an F-Key by holding down Command + Shift and typing a designated digit between 1 and 9; the System software module that ejects a floppy when you type Command + Shift + 1 is an F-Key.)

Even if you work in System 7.0 or MultiFinder all the time, I'd be reluctant to encourage you to use a full-size application such as HyperCard or Panorama for voice call dialing. Opening a program of that magnitude just to call your uncle Louie uses too much time and memory. What you really want for voice dialing is a simple Desk Accessory that opens fast no matter what else you happen to be doing with your computer and then lets you access and dial your phone numbers equally fast. Over the years a shareware DA called Address Book has tried to be such an entity, but through all its upgrades and modifications I've never been able to become a devotee. By trying to be too many things to too many users, Address Book is bigger and slower than it ought to be, and when it sends a phone number to your modem it doesn't append a semicolon to the dialing string. You'll remember from Chapter 6 on the Hayes language that adding a semicolon to a dialing string tells your modem to return to the command mode after dialing and not keep looking for a carrier. Without the semicolon your modem hangs up if it fails to detect a carrier after a few seconds, and when you dial a voice call with Address Book and don't pick up the phone as soon as you connect, that's just what happens. Address Book lets you stretch the delay before hanging up to the time it takes a phone to issue nine full rings, but that's not enough to listen to a weather or stock market report on your modem speaker.

Thus far, the shareware voice dialer I've come to prefer over all the others is a Desk Accessory called PhoneBook (just one word and therefore not to be
confused with the HyperCard stack called Phone Book). Because PhoneBook is so compact, it doesn’t use much disk space or memory, and beyond that it loads fast, it’s compatible with System 7.0, and its interface makes it a pleasure to use as an everyday voice call dialer. Here’s the window it gives you when you open it:

To get a weather report, you’d click once on Weather and type Command-D (for dial). That’s it. Nothing else to do. If you aren’t in the mood to use your mouse you can type a w and then an e. That would highlight Weather too, just as typing w and i would highlight Wilson, Fred. If you don’t feel like using your keyboard, you can select Dial from PhoneBook’s one lone menu. Whichever method you use, PhoneBook puts up the window shown on the next page when it starts to dial.

Figure 15.1. PhoneBook’s main window. Actually, it opens to the A’s, but I’ve already clicked on W.
Click OK when your party has answered and you’ve picked up the phone.

OK

Until you hit OK it keeps your modem online with the party you’re calling, so you can listen to long recorded messages through your modem speaker without having to touch your phone at all.

PhoneBook’s address books are fully searchable, creating new entries is fast and easy, and editing existing ones is easy too. You simply double-click on the entry you want to edit, or you highlight it and press Enter. That brings up this window:

You’ll also notice that it’s also from the edit window that you dial an entry’s alternate number. And if you surround area codes with parentheses, you can configure PhoneBook to add a long distance prefix when it dials outside your local area, but to omit the long distance prefix for 800 numbers and omit the area code altogether if it happens to be the same as yours. (Phone companies
generally don’t let you make local calls using your own area code.) PhoneBook also supports multiple phone books and prints envelopes, and it’s bundled with a utility that imports and exports data to and from your PhoneBook files and also prints your phone books with or without addresses. All told, PhoneBook deserves to be classified as Macintosh shareware that’s genuinely useful. I recommend that you download it for a test run and pay the shareware fee if you decide to use it. Good shareware keeps commercial software developers on their toes, and it’s important that the tradition be sustained.

Fax Modems

To the best of my knowledge, Apple’s own AppleFax modem was the first such device for the Mac. When it first appeared it was something of a high-tech novelty item. Quite a few were sold, and many are still in service. It’s true that the AppleFaxes are fax-only modems, but if an owner wants one to share his modem port with a data modem it’s simple enough to hook up an AB switch that allows switching back and forth between both units (in Chapter 5 there’s a BTW on setting up AB switches). It’s also true that Apple’s software for the AppleFax was downright execrable, but an alternative soon came along in the form of BackFax, and BackFax is only semi-execrable. Unlike the Apple software, it lets fax transmissions take place in the background so you can work on something else while they’re in progress, but all fax programs have to convert your outgoing documents into fax format before sending them and your incoming faxes from fax format before you can view or print them, and in BackFax both conversions are painfully slow. Moreover, every respectable Mac program has the basic decency to put up some sort of progress indicator when performing a time-consuming task, but BackFax is unforgivably miserly with progress indicators. Regardless of whether its performing a conversion or sending or receiving a fax, you never know if you’re going to have to wait a minute or an hour.

When other fax modems began appearing, they were almost always dual purpose units: data plus fax. That obviated the need for AB switches, but most modems in the early wave also came with BackFax, and debate persisted in the Macintosh community about whether fax modems were worth the bother. They’re certainly less expensive than dedicated fax machines, and even if they’re sometimes cumbersome to use, the documents they send look
virtually perfect. Lines of type are always flawlessly horizontal, and the absence of smudges and fly specks puts them in a whole other class than their dedicated fax machine-sent counterparts. Of course, their perfection stems from the fact that they aren't optically scanned. They're imaged directly from computer documents in very much the same way that the Mac images documents when it prints them. On the other hand, you can't just pick up any old piece of paper and fax it out. Fax modems can send faxes only from existing Macintosh documents. If the image you want to send exists only on paper, you first have to find a way to get it into your computer. You can scan it in if you have a scanner, but when you do that you introduce all the smudges, fly specks, and other inexactitudes that are part and parcel of optical scanning. And even if you do have a scanner, scanning an image into your Mac and then faxing it from a scanner document is significantly more troublesome than sending out the image with a fax machine.

Given those considerations, are fax modems worth the bother, or are they doomed to remain high-tech novelty items? For you, I'll refrain from levying a verdict; as for me, I think they're marvelous. Almost every time I need to send a fax, it's something I've produced either with a word processor or a graphics program, and on the rare occasion that I want to fax out something like a newspaper clipping, I turn to the outdated but adequate Apple Scanner that dwells in my office on semi-permanent loan. Those received faxes that I take the trouble to print at all (any but the smudgiest received faxes can be read directly on your screen) come out looking crisper on my printer than on most dedicated fax machines, and my private quest for the paperless office is well served. About the only thing that mitigates what would otherwise be my unreserved enthusiasm is the current state of fax software. Some of the newer fax programs aren't as awful as fax programs used to be, but even the best of them can be a little — how shall we say? — hairy.

Let's take a look at how fax programs work. The one I'll use as an example is Quick Link II Fax, which comes bundled with Zoom fax modems as well as any number of generics and can also be bought from discount mail order suppliers for as little as $10. Versions of the Quick Link fax software earlier than 2.0 were sources of considerable frustration among Macintosh faxers, but my own recent experience has been that Smith Micro Software can now be said to have ironed out the lion's share of the kinks.
When you send a fax from a Macintosh document, the governing metaphor is that you're printing the document to the recipient's fax machine. You therefore go to the Chooser and choose not a printer driver but the fax driver that came with your fax software. Here's how you'd choose QL Fax Print, the Chooser extension that comes with Quick Link II Fax:

![Figure 15.3. Selecting a fax driver in the Chooser. To choose it, all you do is click on it once.](image)

Every time you change your Mac's current printer driver, the Chooser leaves you with a warning when you close its window:

![You have changed your current printer. Please choose "Page Setup..." in all of the open applications.](image)
So when you choose a fax driver you probably ought to do what the Chooser recommends. You may or may not have noticed that different printer drivers give you different Page Setup windows. The following is QL Fax Print's:

![Page Setup window](image)

Figure 15.4. The Page Setup window put up by QL Fax Print, which, as you can see, is pretty standard.

Once you've chosen Page Setup, you send your document by "printing" it. In other words, you choose Print from the File menu of whatever application you happen to be using and then deal with the options window that's put up by the fax driver:

![Options window](image)

Figure 15.5. QL Fax Print lets you fax your document immediately (Send Fax), save it in fax format so you can send it later from the Quick Link II Fax application (Save Fax), or both. That's more or less normal for fax software.
After you've asserted your options, you promptly get another window:

![send fax information window]

Figure 15.6. In this window you can schedule and direct your fax from scratch, or you can click on Phone List and use a phone number you've already entered in the list. You create the list from within the Quick Link II Fax application.

Now you schedule your fax and indicate where it's supposed to go. You can also decide whether you want to include a cover page and, if you do, to whose attention the fax will be sent. Clicking on the Send button is what causes the software to grab your modem, dial the number of the destination fax machine, and then start transmitting the fax. Actual transmission takes place in the background, so you needn't wait until the fax is sent before going back to work.
Fax drivers typically offer you some other options, and the options are typically set from within the fax application that works with the driver. In the Quick Link II Fax application you set your sending options in this window:

![Send Fax Setup]

...and your receiving options in this window:

![Receive Fax Setup]

It's also from within the fax application or a supplementary Disk Accessory that you create your phone lists, view and print received faxes, manually receive faxes (when you're already on the phone with someone and they ask you to turn on your fax machine), and convert fax documents from fax format to formats more compatible with graphics applications.

You should be aware that faxes normally come in two resolutions: high and low. In the fax world, low is generally normal. High resolution is 200 dots per inch (dpi); low is 200 dpi horizontally and 100 dpi vertically. That means a high-resolution fax image contains twice as many dots as a low-resolution
image, which therefore means it takes around twice as long to send and twice as much disk space to store. On the other hand, it will look considerably sharper when it’s printed, although it probably won’t look twice as sharp.

You should also be aware that some fax drivers—QL Fax Print is one of them—are always looking for data at the serial port when fax receiving is turned on. That can cause a major conflict when you’re using another terminal program. Actually it’s much the same kind of conflict that arises when you have two terminal programs open at once, and it usually manifests itself as random loss of characters by your terminal program. To avoid problems, keep fax receiving turned off when you’re using your terminal program.

BTW, Quick Link II Fax also has its own version of a bug that plagued Microsoft Word 4.0 users. When you’re using TrueType, either with System 7.0 or with the TrueType INIT for System 6.x, Quick Link II Fax sometimes displaces hyphens an inch or so to the left, especially when the hyphens come at the end of a line. Bothersome? Yes, indeed. But the history of fax software has been so dreadful that any fax program that worked perfectly probably wouldn’t feel like the real thing.

Networks

Modems can be used on networks. Once that was only true for Shiva’s NetModems, which have AppleTalk and even Ethernet interfaces. NetModems may be expensive, but they’re solid products. You plug one into your network just as you’d plug a LaserWriter into your network, and then every computer on the network can use the NetModem just as every computer on the network can print to the LaserWriter.

Recently, however, several less expensive software alternatives have surfaced, and one I’ve come to like is called PortShare. Like QL Fax Print, PortShare is a Chooser extension. Look at Figure 15.3 again. If you choose PortShare instead of QL Fax Print, the “QL II Fax Print is Selected” message in the upper-right-hand corner of the window changes to “Select a remote port:” and the box just underneath the message lists all the modem ports on the network to which you’ve assigned names. How do you assign names to modem ports? PortShare puts an Options button in the Chooser window, and when you click on it you get PortShare’s options window, as shown in Figure 15.7 on the next page.
What you can do within the window is ask PortShare to "trick" the Macintosh System into letting you use the modem ports of other Macs on the network or letting those other Macs use yours. Let's say that I have a no-frills modem attached to my modem port. It's adequate for occasional use, but it's not very fast and it can't send faxes. I've used the window in Figure 15.7 to name my modem port, and as I use my modem only very occasionally and only to send small files, I normally keep PortShare set to Use Local Port. However, two other Macs on the network are also equipped with modems. Jack's modem transmits data at only 2400 bps, but it has fax capability. Jill's modem can hit 14,400 bps, but it can't handle faxes. Jill has used the window shown in Figure 15.7 to name her modem port Jill's Modem. Jack has likewise named his modem port Jack's Modem. So now, if I want to send a fax, I bring up the Chooser window on my screen, I look for Jack's Modem in the box under Select a remote port:, highlight Jack's modem, then, in the options window, I click on Use Remote Port. From that point forward, PortShare sees to it that my computer behaves as though the modem port of my computer were actually the modem port of Jack's computer. Whatever terminal or fax software I use will communicate through Jack's modem and not through mine. Any data I
send or receive would actually come in or go out via my printer port (through which all network activity is always routed), but PortShare would see to it that my software doesn't know the difference.

Naturally, if I had a gigantic file that I wanted to modem out posthaste, I'd try using Jill's V.32bis modem rather than my own, and because my network can still move data a lot faster than any modem—even if it's using nothing more than plain-Jane LocalTalk—I might even be able to attain 14,400 bps (provided the network isn't too busy).

PortShare is a particularly nice example of some new telecommunications software that's turning up more and more frequently now that Macintosh owners have begun using modems more and more frequently. Back in the introduction to this book I opined that the marriage of Macs and modems was a natural from the beginning. A good marriage sheds a salutary glow on everyone who comes into contact with it. Eventually the headlong march of new technology will surely trample Macs and modems as we know them now, but 'til death do them part you may as well sit back—hunch forward over your keyboard is actually more like it—and enjoy the glow.
Macintosh Bulletin Board Services Around the Country

The following is a list, sorted by area code, of some Mac BBSs you can try calling. I say "try" because BBSs often come and go very fast. A sysop loses interest, his equipment breaks down, he runs out of time or money—and the BBS evaporates. So when you dial a BBS with your modem, there's a chance you'll get an operator announcing that the number's been disconnected or someone telling you the BBS doesn't live there anymore. Still, some BBSs have proven to be solid and tenacious. A few have hung on long enough to become bona fide Macintosh institutions. If you call around, you'll soon discover which ones those are.

The list has been culled from many different sources, and available information about each BBS varies greatly from one to the next. Host refers to the host program the sysop uses to operate the BBS. At one time Red Ryder Host was easily the most popular host program for Macintosh-based BBSs. Then its author, Scott Watson, brought out a sequel, which he called Second Sight. Second Sight retains many of Red Ryder Host's features, and, if it's known that a BBS hosts with one or the other, the host is listed as "2nd Sight/RRH." Boards hosted with Hermes have a very different feel than Second Sight/Red
Ryder boards. When you call around, you'll see how the different host programs operate.

As for Speed/Protocols, when a BBS's highest available connect speed is known, that's what's listed. Lower speeds will almost always be supported too. Where you see “HST” or “Telebit,” you can try connecting with one of the U.S. Robotics or Telebit proprietary protocols. Some companies chose not to wait around for V.32 or V.32bis and instead created high-speed protocols of their own. PEP is Telebit's; HST is USR's. If you know your modem can handle one of them, there's no reason not to try it out.

Where it's known that a BBS is a FidoNet node, its Fido node address is often given. As a general rule, BBSs that participate in FidoNet are more substantial than the others. They also give you access to any number of national and worldwide message bases. If you want to sell your old computer, FidoNet will get your ad to people all over the country. AlterNet is a smaller national network, but it works too.

Where a four-digit alternate number is given for a BBS, use the same area code and prefix as the main number. Most of the rest should be pretty much self-evident. In the end, if a BBS is near where you live and interests you, the best thing to do is call it with your modem (the phone numbers in the list are all data numbers, not voice) and try it out.

201-293-7778  Shadow Spawn BBS. FidoNet 1:269/203.0
201-335-1797  The Insane Asylum. Speed/protocols: 2400
201-348-0576  Power Strip Mac BBS. Host: Hermes; speed/protocols: 2400
201-387-9232  Rock Pile. Host: 2nd Sight/RRH; speed/protocols: HST; Opus 1:107/554; 7:520/554
201-398-1133  Microcosm BBS. Speed/protocols: HST; FidoNet 1:269/202
201-403-1990  Video Dome BBS
201-460-1673  The Phantom of the Mac. Host: Hermes; 300 megs
201-503-0929  Videopolis
201-543-6950  Mac Atlantic BBS. Speed/protocols: 2400; FidoNet 1:269/125
201-622-5928  Essex Computers BBS. Host: 2nd Sight/RRH; speed/protocols: HST
201-666-2013  Macintosh BBS. Host: 2nd Sight/RRH; speed/protocols: HST, V.32; V.42bis; FidoNet 1:269/327
<table>
<thead>
<tr>
<th>Phone Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>201-741-5208</td>
<td>Realm of Insanity. Speed/protocols: 2400</td>
</tr>
<tr>
<td>201-767-6337</td>
<td>Electronic Pen BBS. Host: 2nd Sight/RRH; speed/protocols: 2400</td>
</tr>
<tr>
<td>201-805-9819</td>
<td>Jungle. Host: Hermes; speed/protocols: 2400; Mac, DOS, Apple II; Home Brewers' Forum; anti-viral tactics section</td>
</tr>
<tr>
<td>201-832-6294</td>
<td>Silent Tower. Host: Hermes; speed/protocols: 2400</td>
</tr>
<tr>
<td>201-838-0113</td>
<td>Finishing Technology Hotline. Host: 2nd Sight/RRH; speed/protocols: 2400; PCB special interest</td>
</tr>
<tr>
<td>201-857-8880</td>
<td>RockBoard</td>
</tr>
<tr>
<td>201-863-5422</td>
<td>Da Cave AE. Host: Hermes; over 1.8 gigs online</td>
</tr>
<tr>
<td>202-547-0435</td>
<td>Mac OnLine. Host: Hermes; speed/protocols: 2400; MacONLINE! magazine</td>
</tr>
<tr>
<td>203-226-6694</td>
<td>The Empire BBS. Host: Hermes; speed/protocols: 2400</td>
</tr>
<tr>
<td>203-287-9120</td>
<td>Anxiety Closet</td>
</tr>
<tr>
<td>203-294-1206</td>
<td>Mac's Last Stand. Speed/protocols: 2400</td>
</tr>
<tr>
<td>203-371-7073</td>
<td>Sixth Sense BBS. Host: Hermes; speed/protocols: 2400; Mac, IBM, ]</td>
</tr>
<tr>
<td>203-429-3789</td>
<td>Algorithmica. Host: Hermes; speed/protocols: 9600, V.32; MNP-5; 173 megs online</td>
</tr>
<tr>
<td>203-481-3663</td>
<td>Castle Anthrax</td>
</tr>
<tr>
<td>203-481-3747</td>
<td>Atomic Cafe</td>
</tr>
<tr>
<td>203-742-5699</td>
<td>Stellar Sabre BBS. Host: 2nd Sight/RRH; speed/protocols: 2400; donation requested; Sysop: Daniel Delamatta; Coventry, CT; 24 hrs.</td>
</tr>
<tr>
<td>203-761-1469</td>
<td>Pretend BBS. Host: Hermes; speed/protocols: 2400</td>
</tr>
<tr>
<td>203-762-8481</td>
<td>Wilton Woods OPUS. Speed/protocols: HST; Opus; 1:141/250</td>
</tr>
<tr>
<td>203-763-3485</td>
<td>Earth Network. Host: 2nd Sight/RRH; speed/protocols: HST; FidoNet 1:142/500</td>
</tr>
<tr>
<td>203-790-6612</td>
<td>Web. Speed/protocols: 2400; FidoNet 1:141/735.0</td>
</tr>
<tr>
<td>203-854-9716</td>
<td>T.S.C. Speed/protocols: 2400; Opus; 1:141/245</td>
</tr>
<tr>
<td>203-863-8866</td>
<td>Greenwich High School Online. Host: Hermes; speed/protocols: 2400</td>
</tr>
<tr>
<td>204-269-4343</td>
<td>Headboard BBS. Speed/protocols: 2400</td>
</tr>
<tr>
<td>205-262-3735</td>
<td>Speed of Light. Speed/protocols: 2400; FidoNet 1:375/15</td>
</tr>
<tr>
<td>205-826-9205</td>
<td>The Lyceum</td>
</tr>
<tr>
<td>Phone Number</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>206-232-7426</td>
<td>Lerxstwood Mall. Host: Hermes; speed/protocols: HST; alternate number -7556</td>
</tr>
<tr>
<td>206-282-3065</td>
<td>Bumbershoot BBS. Host: 2nd Sight/RRH; speed/protocols: 9600; FidoNet 1:343/43</td>
</tr>
<tr>
<td>206-323-7578</td>
<td>Abraxas Information. Host: Hermes; speed/protocols: 2400; 10 p.m.-8 a.m. PST</td>
</tr>
<tr>
<td>206-432-0657</td>
<td>Mac Street</td>
</tr>
<tr>
<td>206-452-2012</td>
<td>Port Angeles/Evergreen Micro Network. Host: 2nd Sight/RRH; speed/protocols: 9600; FidoNet 1:354/1</td>
</tr>
<tr>
<td>206-475-0402</td>
<td>Beyond BBS. Host: Hermes; speed/protocols: 2400</td>
</tr>
<tr>
<td>206-562-1223</td>
<td>Uneasy Alliance. Host: Hermes; speed/protocols: 2400</td>
</tr>
<tr>
<td>206-637-2398</td>
<td>SeaSoftNet. Speed/protocols: 2400; FidoNet 1:343/8.9</td>
</tr>
<tr>
<td>206-641-5756</td>
<td>Pac/Mac. Host: Hermes; speed/protocols: 9600; T2500 node; alternate number -6767</td>
</tr>
<tr>
<td>206-643-4826</td>
<td>Pac/Mac. Host: Hermes; speed/protocols: 2400; 2nd node</td>
</tr>
<tr>
<td>206-725-6629</td>
<td>Sea/Mac. Host: Mansion; speed/protocols: HST; FidoNet 343/31</td>
</tr>
<tr>
<td>206-726-1484</td>
<td>MACHine BBS. Host: Hermes; speed/protocols: 2400; 2 nodes</td>
</tr>
<tr>
<td>206-767-7681</td>
<td>MacWASH BBS</td>
</tr>
<tr>
<td>206-863-1526</td>
<td>Bunny Line. Host: Hermes; speed/protocols: 2400; #2 Internodal Chat; 24 hrs.</td>
</tr>
<tr>
<td>206-863-8662</td>
<td>Data Control. Host: Hermes; speed/protocols: 2400; FidoNet 1:138/13</td>
</tr>
<tr>
<td>206-883-1383</td>
<td>Crystal Cavern. Speed/protocols: 2400; FidoNet 1:343/105</td>
</tr>
<tr>
<td>206-889-9802</td>
<td>Mac Exchange. Speed/protocols: HST; FidoNet 343/49</td>
</tr>
<tr>
<td>206-964-2683</td>
<td>The Cubby Hole</td>
</tr>
<tr>
<td>206-964-4149</td>
<td>Basic Training BBS. Host: Hermes; speed/protocols: 2400; alternate number -1913</td>
</tr>
<tr>
<td>207-799-9880</td>
<td>Ice Palace</td>
</tr>
<tr>
<td>208-455-3312</td>
<td>JJHS-BBS. Host: 2nd Sight/RRH; speed/protocols: HST; Jefferson Jr. High</td>
</tr>
<tr>
<td>209-432-9778</td>
<td>Fresno Connection. Host: Hermes; speed/protocols: 2400</td>
</tr>
<tr>
<td>209-575-1411</td>
<td>Ellipsis BBS. Host: Hermes; speed/protocols: 2400; FidoNet 1:208/107</td>
</tr>
<tr>
<td>Phone Number</td>
<td>Service Name and Details</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>212-439-6126</td>
<td>Frontal Lobe. Speed/protocols: HST; FidoNet 1:107/747</td>
</tr>
<tr>
<td>212-597-9083</td>
<td>Metro Area MUG. Host: 2nd Sight/RRH; speed/protocols: HST; FidoNet 1:107/705; AlterNet 7:520/705</td>
</tr>
<tr>
<td>212-645-9484</td>
<td>NYMUG BBS. Speed/protocols: HST</td>
</tr>
<tr>
<td>212-663-8707</td>
<td>Odyssey BBS. Host: Hermes</td>
</tr>
<tr>
<td>212-697-3713</td>
<td>Sappho's Exchange. Host: 2nd Sight/RRH; speed/protocols: 2400</td>
</tr>
<tr>
<td>212-923-3049</td>
<td>Theatre Arts. New York, NY 19200</td>
</tr>
<tr>
<td>212-927-6919</td>
<td>Super 68 BBS. Speed/protocols: 2400; FidoNet 1:107/108</td>
</tr>
<tr>
<td>212-982-4444</td>
<td>Jeff's BBS. Speed/protocols: 9600; 8 p.m.-2 a.m. only</td>
</tr>
<tr>
<td>213-214-0153</td>
<td>MACrow #2</td>
</tr>
<tr>
<td>213-214-3604</td>
<td>MACrow #1</td>
</tr>
<tr>
<td>213-254-4133</td>
<td>Glassell Park BBS. Alternate number -4852</td>
</tr>
<tr>
<td>213-273-1158</td>
<td>Ye Olde Pawne Shoppe</td>
</tr>
<tr>
<td>213-275-6975</td>
<td>Olympus II. Host: Hermes; speed/protocols: 9600, V.32</td>
</tr>
<tr>
<td>213-372-4800</td>
<td>Manhattan Transfer. Host: 2nd Sight/RRH; speed/protocols: HST; FidoNet 1:102/135</td>
</tr>
<tr>
<td>213-376-2150</td>
<td>Kirk's BBS. Host: 2nd Sight/RRH; speed/protocols: 2400; 7 a.m.-midnight PDT; FidoNet 1:102/132</td>
</tr>
<tr>
<td>213-458-6157</td>
<td>MacMaster's</td>
</tr>
<tr>
<td>213-472-0643</td>
<td>The MousePIT. Host: Hermes; speed/protocols: 9600; 250 megs of Mac and IBM files</td>
</tr>
<tr>
<td>213-552-3278</td>
<td>Smash Palace Mac</td>
</tr>
<tr>
<td>213-732-4387</td>
<td>Menlo</td>
</tr>
<tr>
<td>213-791-7060</td>
<td>Big-Mac Attack BBS. Host: Hermes; speed/protocols: 2400</td>
</tr>
<tr>
<td>213-820-4320</td>
<td>Digital Dungeon. MacCitadel BBS</td>
</tr>
<tr>
<td>213-874-0809</td>
<td>CP BBS</td>
</tr>
<tr>
<td>213-936-6923</td>
<td>Nibbler's Node. Host: Hermes; speed/protocols: 9600</td>
</tr>
<tr>
<td>214-360-0550</td>
<td>Southwestern Macintosh. Host: Hermes; speed/protocols: 19,200</td>
</tr>
<tr>
<td>214-363-0428</td>
<td>Southwestern Macintosh. Host: Hermes; speed/protocols: 19,200; Telebit T-2500 node; alternate number -0517</td>
</tr>
<tr>
<td>Number</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>214-556-2982</td>
<td>Southern Crossroads</td>
</tr>
<tr>
<td>214-739-0645</td>
<td>MACRO Mouse</td>
</tr>
<tr>
<td>214-867-5809</td>
<td>Wolverine's Lair</td>
</tr>
<tr>
<td>215-293-9703</td>
<td>Crystal Citadel</td>
</tr>
<tr>
<td>215-364-3324</td>
<td>SATALINK</td>
</tr>
<tr>
<td>215-368-1162</td>
<td>Naughty Bits</td>
</tr>
<tr>
<td>215-387-8095</td>
<td>PennMUG BBS. Host: Hermes</td>
</tr>
<tr>
<td>215-426-5765</td>
<td>Free Lino BBS. Host: Hermes</td>
</tr>
<tr>
<td>215-446-7670</td>
<td>Bob's Mac. Host: 2nd Sight/RRH</td>
</tr>
<tr>
<td>215-464-9068</td>
<td>Narnia BBS</td>
</tr>
<tr>
<td>215-635-0244</td>
<td>PAMMUG</td>
</tr>
<tr>
<td>215-638-7391</td>
<td>Outer Limits BBS. Speed/protocols: 2400; FidoNet 1:273/217</td>
</tr>
<tr>
<td>215-745-9774</td>
<td>Turbo-386. Speed/protocols: 9600; FidoNet 1:273/906.1</td>
</tr>
<tr>
<td>215-895-2573</td>
<td>Dragon Keep. Host: 2nd Sight/RRH</td>
</tr>
<tr>
<td>216-587-3435</td>
<td>Hal's BBS. Online games</td>
</tr>
<tr>
<td>216-752-4921</td>
<td>Monstrous MAC BBS. Host: Hermes</td>
</tr>
<tr>
<td>216-943-2788</td>
<td>Frayed Ends of Sanity II. Host: Hermes</td>
</tr>
<tr>
<td>217-337-0764</td>
<td>Avalon BBS. Speed/protocols: 2400; Sysop: Santi Zorzopulos</td>
</tr>
<tr>
<td>217-384-3128</td>
<td>Avalon (CUMUG). Host: 2nd Sight/RRH; speed/protocols: 2400; FidoNet 1:233/14</td>
</tr>
<tr>
<td>219-282-1054</td>
<td>MCN 1. Speed/protocols: 2400; multi-line; FidoNet 1:227/2</td>
</tr>
<tr>
<td>301-325-1027</td>
<td>Pizza's Place BBS. Speed/protocols: 14,400</td>
</tr>
<tr>
<td>301-337-0780</td>
<td>The Mosquito's Den. Host: Hermes</td>
</tr>
<tr>
<td>301-384-5847</td>
<td>Dog's Dwelling. Host: Hermes</td>
</tr>
<tr>
<td>301-467-7119</td>
<td>The Bayview MUG BBS. Speed/protocols: 9600, V.32; V.42; V.42bis; Sysop: Mark Fleming</td>
</tr>
<tr>
<td>301-471-4263</td>
<td>Eastern Mac Exclusive. MacCitadel</td>
</tr>
</tbody>
</table>
CRABBS. Host: 2nd Sight/RRH; speed/protocols: 2400; FidoNet 1:261/1053; Annapolis Apple Slice UG

John's BBS. FidoNet 50:5301/1083; 1:261/1083

Capt Peg's. Host: Hermes; speed/protocols: 2400

MacCity BBS. Host: Hermes; speed/protocols: 2400

MacSachel Printing, Inc. Host: Hermes; speed/protocols: 2400; Imagesetting Service Bureau

Ravenous Abyss

Ravenous Abyss

Overlord's Party Castle. Lots of messages

BlackDog BBS. Host: Hermes; speed/protocols: 2400; Mac, Amiga, IBM

Mouse Event. Speed/protocols: 9600

Striker. Speed/protocols: 2400; IBM, Mac

Nut House. Host: Hermes; speed/protocols: 9600, HST

Divinity II. Speed/protocols: 9600

FireStation BBS. Speed/protocols: 2400; FidoNet 1:261/1044

The Clone

Galactic Express

HARDWARE BBS. Host: Hermes; speed/protocols: 2400; 10:30 p.m.-7:00 a.m.

Boulder Mac Maniacs. Speed/protocols: 2400; FidoNet 1:104/49

Smoky Net. Host: Hermes; speed/protocols: 2400; a school BBS with BattleTech

Fort Mac. Host: 2nd Sight/RRH; speed/protocols: 9600 (HST); Sysop: Greg Shaw; Aurora, CO; 306/17; FidoNet 24 hrs.

Mile High Mac BBS. Host: Hermes; speed/protocols: 9600 (HST); V.42

The Mosquito Coast

Sunshine Online Service. Host: 2nd Sight/RRH; speed/protocols: 2400; donation requested; Sysop: Louis Oaken; Miami, Florida; FidoNet 1:135/92; 24 hrs.

Gameport BBS. Game section

MACabre BBS. Speed/protocols: 2400
<table>
<thead>
<tr>
<th>Phone Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>305-748-7993</td>
<td>NatMAC BBS. Host: Hermes; speed/protocols: 2400</td>
</tr>
<tr>
<td>305-753-4605</td>
<td>MACATTACK! BBS. Host: Hermes; speed/protocols: 2400</td>
</tr>
<tr>
<td>305-763-7743</td>
<td>Fluid Power. Speed/protocols: 2400; FidoNet 1:369/4</td>
</tr>
<tr>
<td>305-891-1062</td>
<td>South Beach BBS. Host: 2nd Sight/RRH; speed/protocols: 19,200; Sysop: Craig Moser; South Miami Beach, FL; 24 hrs.</td>
</tr>
<tr>
<td>309-692-0786</td>
<td>PCGS/CIG BBS. Host: Hermes; speed/protocols: 1200</td>
</tr>
<tr>
<td>309-755-8274</td>
<td>MouseCapades. Host: 2nd Sight/RRH; speed/protocols: 2400</td>
</tr>
<tr>
<td>312-583-7679</td>
<td>Convolutions. Speed/protocols: 2400</td>
</tr>
<tr>
<td>312-769-2020</td>
<td>20/20 TBBS MultiLine. Speed/protocols: 2400; FidoNet 1:115/769</td>
</tr>
<tr>
<td>312-923-1932</td>
<td>Data-Mania. Speed/protocols: 2400; Mac and Atari</td>
</tr>
<tr>
<td>312-943-3498</td>
<td>Galapagos. Host: Hermes; speed/protocols: 9600</td>
</tr>
<tr>
<td>313-291-5571</td>
<td>Gateway Online</td>
</tr>
<tr>
<td>313-557-0759</td>
<td>The MacGroup-Detroit BBS. Speed/protocols: 9600, V.32; Sysop: Terry White</td>
</tr>
<tr>
<td>313-572-9536</td>
<td>MaxMac</td>
</tr>
<tr>
<td>313-695-6744</td>
<td>The Association. Speed/protocols: 14,400</td>
</tr>
<tr>
<td>314-349-5344</td>
<td>Cheswick's RBBS. Speed/protocols: 2400; FidoNet 1:100/375</td>
</tr>
<tr>
<td>314-846-8982</td>
<td>Mac Paradise. MacMaster; alternate number -8929</td>
</tr>
<tr>
<td>314-997-6912</td>
<td>Show Me More Stacks BBS. Speed/protocols: 2400; Opus 1:100/255</td>
</tr>
<tr>
<td>315-342-6960</td>
<td>Toad Harbor. Host: Hermes; speed/protocols: 19,200</td>
</tr>
<tr>
<td>315-685-3367</td>
<td>Bird's Nest. Host: 2nd Sight/RRH; speed/protocols: 2400; FidoNet 260/337; EE</td>
</tr>
<tr>
<td>315-695-4436</td>
<td>Galaxia. Speed/protocols: 2400; TSAUG MUG; Mac section; FidoNet 1:260/328</td>
</tr>
<tr>
<td>316-793-8819</td>
<td>X-tronic Connection. Alternate number -9008, -792-5925</td>
</tr>
<tr>
<td>317-290-9070</td>
<td>Falx Cerebri QBBS. Speed/protocols: 2400; FidoNet 1:231/80</td>
</tr>
<tr>
<td>317-849-4007</td>
<td>IndyServe QBBS. Speed/protocols: HST; FidoNet 1:231/250</td>
</tr>
<tr>
<td>318-424-0375</td>
<td>MAClan/Shreepport</td>
</tr>
<tr>
<td>319-365-4775</td>
<td>Mouse College. Host: Hermes; speed/protocols: 2400</td>
</tr>
<tr>
<td>358-084-4970</td>
<td>Bomb Shelter. Host: Hermes; speed/protocols: 9600</td>
</tr>
</tbody>
</table>
Appendix A  Macintosh Bulletin Board Services
Around the Country  

21st Century BBS
MacNet Omaha. Host: Hermes; speed/protocols: 2400; OMUG support; Mac echos; files
Wind Dragon. Speed/protocols: 2400; FidoNet 1:285/637.0
TKG BBS. Speed/protocols: 2400; SS
SunValley/MicroLink. Speed/protocols: HST; FidoNet 1:129/34; MicroLink 8:7500/16.1
Mac_Xen_Link. Host: Hermes; speed/protocols: 9600
Calgary’s Resource. Speed/protocols: 9600; ST, IBM, Mac areas; FidoNet 8:7500/11
MacLORE BBS. Official Apple Corp. Node 2; alternate number -2465
Calgary Online. Host: Hermes; speed/protocols: 2400
Nimbus BBS. Host: Hermes; speed/protocols: 2400
Mouse BBS. Login w/ ID=guest, password=guest
TimeWarp Tavern II. Host: 2nd Sight/RRH; speed/protocols: 2400
Rockin’ Man
USS Republic. Speed/protocols: HST; FidoNet 1:133/1371
Shoe’s CPU. Speed/protocols: HST; FidoNet 1:3621/425.0
Tatau! Host: Hermes; speed/protocols: 2400; Atlanta, GA; FidoNet 133/111; 11 p.m.-8 a.m.
BIKENET. Speed/protocols: 2400; Opus 1:346/5
Pyrotechnic’s
California Dreamin’
Cornucopia TBBS. Speed/protocols: 2400; FidoNet 1:363/18
Abacus Information Center. Opus 1:363/7
Storage House BBS
Mac Outpost. Speed/protocols: 2400; alternate number -0208
Skeletons/Closet
The Realm of Wonder. Host: Hermes; speed/protocols: 2400; Mac hardware/software tech support
MacDaze. Host: 2nd Sight/RRH; speed/protocols: HST; FidoNet 1:143/33; Eggnet 99:9403/33; messages only; no files; formerly Phoenix 2 BBS
<table>
<thead>
<tr>
<th>Phone Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>408-281-4895</td>
<td>South Bay Soaring Society BBS. Host: Hermes; speed/protocols: 2400</td>
</tr>
<tr>
<td>408-736-2607</td>
<td>Sphynx BBS. Speed/protocols: 2400; FidoNet 1:143/205</td>
</tr>
<tr>
<td>408-738-1119</td>
<td>Sunnyvale Fido Connection. Speed/protocols: 2400; messages only; no files; FidoNet 1:143/207.1</td>
</tr>
<tr>
<td>408-738-5791</td>
<td>Mousehole. Speed/protocols: 2400; Galacticomm BBS; MacTUTOR! magazine code disk downloads available for fee</td>
</tr>
<tr>
<td>408-866-4933</td>
<td>MacScience BBS. Host: 2nd Sight/RRH; speed/protocols: 9600 (HST); Sysop: Ray Terry N6PHJ; San Jose, CA; FidoNet 1:143/36</td>
</tr>
<tr>
<td>408-866-4933</td>
<td>MacScience! BBS. Host: 2nd Sight/RRH; speed/protocols: HST; FidoNet 1:143/36</td>
</tr>
<tr>
<td>409-297-5466</td>
<td>The treeHOUSE. Host: Hermes; speed/protocols: 9600; alternate number -5609</td>
</tr>
<tr>
<td>412-268-8974</td>
<td>Mac@Night. Speed/protocols: 9600; FidoNet 1:129/107</td>
</tr>
<tr>
<td>412-344-8504</td>
<td>Pittsburgh Macs. Speed/protocols: 9600; 7 lines; FidoNet 8:7001/1; Network Echogate 1:129/34</td>
</tr>
<tr>
<td>412-696-5500</td>
<td>NetP Classifieds On-Line. Speed/protocols: 2400; HyperBBS; free classified ads</td>
</tr>
<tr>
<td>412-765-0532</td>
<td>Pgh Computer Connection. Speed/protocols: 2400; FidoNet 1:129/29.0</td>
</tr>
<tr>
<td>412-928-8452</td>
<td>Endeavor Starflight. FidoNet 1:129/120</td>
</tr>
<tr>
<td>413-545-4453</td>
<td>SpaceMet. FidoNet 1:321/110</td>
</tr>
<tr>
<td>413-562-1870</td>
<td>Baudville. Host: 2nd Sight/RRH; speed/protocols: HST, 9600; FidoNet 1:321/304</td>
</tr>
<tr>
<td>413-746-3202</td>
<td>Mac Only BBS. FidoNet 1:321/307</td>
</tr>
<tr>
<td>414-541-9426</td>
<td>Forecast Office. Speed/protocols: HST, 9600; FidoNet 1:154/970</td>
</tr>
<tr>
<td>415-284-5343</td>
<td>The Strangeland. Host: Hermes; speed/protocols: HST; 320 megs</td>
</tr>
<tr>
<td>415-285-6862</td>
<td>Jasmine Support BBS</td>
</tr>
<tr>
<td>415-342-0511</td>
<td>Draco Redux Apple Info System. Host: Hermes; speed/protocols: 2400; Mac and [GS; alternate number -3661</td>
</tr>
<tr>
<td>415-349-7322</td>
<td>Mac F-X BBS. Host: Hermes; speed/protocols: 2400</td>
</tr>
<tr>
<td>415-365-4194</td>
<td>Nwonknu HQ. Speed/protocols: HST; 8:914/701.0</td>
</tr>
<tr>
<td>Phone Number</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>415-387-5117</td>
<td>Leviathan RBBS. Speed/protocols: 2400; 1:10/8; 8:914/204</td>
</tr>
<tr>
<td>415-426-0470</td>
<td>Records Department TBBS. Speed/protocols: 2400; 1:161/42</td>
</tr>
<tr>
<td>415-462-3347</td>
<td>Leading Technology. Speed/protocols: 2400; 1:161/91</td>
</tr>
<tr>
<td>415-484-4412</td>
<td>MacCircles. Host: 2nd Sight/RRH; speed/protocols: 2400; 1:161/555</td>
</tr>
<tr>
<td>415-567-0217</td>
<td>Twelfth Night (or, what you will). Host: Hermes; speed/protocols: HST; 1:125/17; 8:914/214; alternate number -5094</td>
</tr>
<tr>
<td>415-574-4544</td>
<td>Dimension X. Host: Hermes; speed/protocols: 2400</td>
</tr>
<tr>
<td>415-672-1944</td>
<td>Elite's HideAway</td>
</tr>
<tr>
<td>415-681-9594</td>
<td>Macademe/Emma. Speed/protocols: HST, V.32; 1:125/222.0</td>
</tr>
<tr>
<td>415-751-8396</td>
<td>MacWARP BBS. Host: Hermes; speed/protocols: HST</td>
</tr>
<tr>
<td>415-769-8874</td>
<td>TOPS Support BBS. Speed/protocols: 2400</td>
</tr>
<tr>
<td>415-775-2384</td>
<td>Bay. Speed/protocols: 9600, V.32; FWB support (HD Util)</td>
</tr>
<tr>
<td>415-776-3021</td>
<td>Holy Grail</td>
</tr>
<tr>
<td>415-795-8862</td>
<td>MACINFO BBS. Host: 2nd Sight/RRH; speed/protocols: HST; 1:204/555</td>
</tr>
<tr>
<td>415-795-8862</td>
<td>MacInfo BBS. Host: 2nd Sight/RRH; speed/protocols: 9600 (HST); donations accepted; Sysop: Norman Goodger; Newark, CA; 204/555; FidoNet; 24 hrs.</td>
</tr>
<tr>
<td>415-849-3539</td>
<td>Psychic Link BBS. Host: Hermes; speed/protocols: 2400</td>
</tr>
<tr>
<td>415-851-4143</td>
<td>Snood's Board</td>
</tr>
<tr>
<td>415-854-2559</td>
<td>Heebner's Hotel. Sysop: Jim Heebner</td>
</tr>
<tr>
<td>415-898-1696</td>
<td>MicroLINK. Host: 2nd Sight/RRH; speed/protocols: 2400</td>
</tr>
<tr>
<td>415-932-8293</td>
<td>Thought Plane. Host: NovaLink</td>
</tr>
<tr>
<td>415-948-1349</td>
<td>OneNet BBS</td>
</tr>
<tr>
<td>415-948-1349</td>
<td>The Forum. Speed/protocols: 2400; private; Sysop: Scott Convers</td>
</tr>
<tr>
<td>415-961-3708</td>
<td>Kremlin</td>
</tr>
<tr>
<td>415-965-1525</td>
<td>The Sound Mind BBS. Host: 2nd Sight/RRH; speed/protocols: 9600, v.32; donations groveled for; Sysop: John R. MacWilliamson; Mountain View, CA; 24 hrs.</td>
</tr>
<tr>
<td>415-966-1667</td>
<td>Useless Slime. Mac Exchange; alternate numbers -1668, -1669</td>
</tr>
<tr>
<td>416-288-1767</td>
<td>Magic BBS. Toronto, Canada</td>
</tr>
<tr>
<td>Area Code</td>
<td>BBS Name</td>
</tr>
<tr>
<td>----------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>416</td>
<td>Daily Planet BBS. Annual fee</td>
</tr>
<tr>
<td>416</td>
<td>NYNET BBS. Toronto, Canada</td>
</tr>
<tr>
<td>416</td>
<td>SoftArc Online. Toronto, Canada</td>
</tr>
<tr>
<td>416</td>
<td>LOGIC Information Systems. Host: 2nd Sight/RRH; speed/protocols: 2400; $45/yr.</td>
</tr>
<tr>
<td>416</td>
<td>MacMECCA. Host: Hermes; speed/protocols: 9600</td>
</tr>
<tr>
<td>417</td>
<td>TriStar BBS. Speed/protocols: 9600; Telebit; various computer SIGs</td>
</tr>
<tr>
<td>418</td>
<td>Bab-O-Manie. Speed/protocols: 9600; FidoNet 1:240/507</td>
</tr>
<tr>
<td>419</td>
<td>POST Office BBS. Host: Hermes; speed/protocols: 2400</td>
</tr>
<tr>
<td>419</td>
<td>Zephyr BBS/SASS. Speed/protocols: 2400; FidoNet 1:234/4</td>
</tr>
<tr>
<td>44</td>
<td>Macintosh UK BBS. Speed/protocols: HST; Kent, England</td>
</tr>
<tr>
<td>502</td>
<td>Magic Castle. Speed/protocols: 2400; FidoNet 1:11/17</td>
</tr>
<tr>
<td>502</td>
<td>Club Mac of Kentuckiana. Speed/protocols: HST; TF; Bill Pittman</td>
</tr>
<tr>
<td>503</td>
<td>LinguaBase. Host: Hermes</td>
</tr>
<tr>
<td>503</td>
<td>The Infinite Loop BBS. Host: Hermes; speed/protocols: 2400; mostly private</td>
</tr>
<tr>
<td>503</td>
<td>Greyland-2. Host: Hermes; speed/protocols: 9600</td>
</tr>
<tr>
<td>503</td>
<td>Greyland. Host: Hermes; speed/protocols: 9600; Node 1. Mac support board; V.42bis.</td>
</tr>
<tr>
<td>503</td>
<td>LateNight BBS. FidoNet 1:152/208</td>
</tr>
<tr>
<td>503</td>
<td>Event Horizons. GIFs</td>
</tr>
<tr>
<td>504</td>
<td>BizMac</td>
</tr>
<tr>
<td>505</td>
<td>Call BBS. Host: 2nd Sight/RRH; speed/protocols: 9600; FidoNet 1:301/4</td>
</tr>
<tr>
<td>507</td>
<td>Caesar's Palace. Host: 2nd Sight/RRH; speed/protocols: 9600</td>
</tr>
<tr>
<td>508</td>
<td>Cul-De-Sac BG. Speed/protocols: 2400; FidoNet 1:322/360</td>
</tr>
<tr>
<td>508</td>
<td>SoundStage BBS. Host: NovaLink</td>
</tr>
<tr>
<td>508</td>
<td>Greyhawk BBS. Speed/protocols: 9600; Mac, IBM</td>
</tr>
<tr>
<td>508</td>
<td>Think Tank. Speed/protocols: 9600</td>
</tr>
<tr>
<td>508</td>
<td>Graveyard BBS. Speed/protocols: 2400; 24 hrs.</td>
</tr>
<tr>
<td>508</td>
<td>Main Street U.S.A. Host: 2nd Sight/RRH; speed/protocols: 2400; FidoNet 1:322/575; 3 p.m.-7:30 a.m. M-F; 24 hrs. SS</td>
</tr>
<tr>
<td>Phone Number</td>
<td>Service Description</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>509-924-5364</td>
<td>MACS BBS. Host: Hermes; speed/protocols: 2400</td>
</tr>
<tr>
<td>509-966-8555</td>
<td>Acey BBS. Speed/protocols: 2400; FidoNet 1:347/12.0</td>
</tr>
<tr>
<td>510-275-9759</td>
<td>Dear Theophilus BBS. Host: 2nd Sight/RRH; speed/protocols: 2400; Sysop: The Rev. Mark A. Spaulding; Danville, CA, SF Bay Area Via PC Pursuit; FidoNet 161/703; 8 a.m.-1 a.m.</td>
</tr>
<tr>
<td>510-250-2279</td>
<td>'02 Register BBS. Host: Hermes; speed/protocols: 2400</td>
</tr>
<tr>
<td>512-288-0914</td>
<td>Outlandos d'Amour. Host: Hermes; speed/protocols: 2400</td>
</tr>
<tr>
<td>512-322-0265</td>
<td>Zen Wedgie</td>
</tr>
<tr>
<td>512-331-2967</td>
<td>Chatsubo</td>
</tr>
<tr>
<td>512-343-1612</td>
<td>Bull Creek BBS. Host: 2nd Sight/RRH; speed/protocols: HST; FidoNet 1:382/54.11; AppleTalk/MacServe network; $15/yr.</td>
</tr>
<tr>
<td>512-345-9469</td>
<td>ForeignDesign-EliteTexan BBS. Host: Hermes; speed/protocols: HST, 9600</td>
</tr>
<tr>
<td>512-366-0556</td>
<td>M.S.I. After Hours. FidoNet 1:387/304</td>
</tr>
<tr>
<td>512-392-4366</td>
<td>Impact Crater BBS. Host: Hermes; speed/protocols: 2400</td>
</tr>
<tr>
<td>512-462-1425</td>
<td>Spanky's SportsNet. Host: Hermes; speed/protocols: 2400; Austin, TX; sports, media</td>
</tr>
<tr>
<td>512-472-6220</td>
<td>Necropolis of Dreams. Host: Hermes; speed/protocols: 2400; 2nd Node; alternate number -6905</td>
</tr>
<tr>
<td>512-474-1512</td>
<td>Hellhole. Host: Hermes; speed/protocols: 9600</td>
</tr>
<tr>
<td>512-482-9183</td>
<td>Akbar Jeff's BBS Hut. Host: Hermes; speed/protocols: 2400; FidoNet 1:384/64</td>
</tr>
<tr>
<td>512-641-2063</td>
<td>Lean, Mean, 32 Bit Clean. Bruce Tomlin's BBS</td>
</tr>
<tr>
<td>512-658-3212</td>
<td>MAC Exchange. Speed/protocols: 2400; Sysop: Greg Lewis; FREE! alternate number -6224</td>
</tr>
<tr>
<td>512-891-3733</td>
<td>Motorola Freeware BBS. Speed/protocols: 2400; Motorola support</td>
</tr>
<tr>
<td>512-993-4291</td>
<td>MacWildness. Host: Hermes; speed/protocols: 9600; online games; echos; files area</td>
</tr>
<tr>
<td>512-993-7595</td>
<td>Stomp'n Brew. Host: Hermes; speed/protocols: 2400</td>
</tr>
<tr>
<td>513-253-2476</td>
<td>Currents. Host: Hermes; speed/protocols: 2400; FidoNet 1:110/430; MacList 6:6026/1</td>
</tr>
</tbody>
</table>
Mac Mania BBS. Host: Hermes; speed/protocols: 9600, V.32
Grand Finale. Host: Hermes; speed/protocols: 2400
Queen City Mac BBS. Host: Hermes; speed/protocols: 2400
Mac a Mac. Host: 2nd Sight/RRH; speed/protocols: 2400; Primarily French msg base; FidoNet 1:167/113; $25/yr. for downloads
ACS Lab -> ELITE. Host: Hermes; speed/protocols: HST, V.32; V.32bis
MAC-LINK. Host: 2nd Sight/RRH; speed/protocols: HST; FidoNet 1:167/182; $60/yr.; new number
PComm. Speed/protocols: 2400; FidoNet 1:167/1.0
Zoo System. Speed/protocols: 9600, V.32; password: PLEASE; FidoNet 1:290/2.1
Enchanted Mansion BBS. Host: Mansion; speed/protocols: HST; FidoNet 1:290/628
Cyber-Net. Host: Hermes; speed/protocols: 2400
Mac's Delight. Host: Hermes; speed/protocols: 9600, V.32; Sysop: Bob Taub
Starship Pegasus BBS
MECCA/ENVIRONUS. Host: 2nd Sight/RRH; speed/protocols: 2400; AlterNet 7:526/305; FidoNet 267/10
MasEnP BBS. Conversation-oriented Mac BBS
MacHaven BBS. Host: 2nd Sight/RRH; speed/protocols: 9600 (HST); V.42bis; $10 for extended access time; Sysop: Ray Leninger; Jackson, Mississippi; FidoNet 3632/6; 24 hrs.
Stacks R Us/WCTV. Speed/protocols: 2400; HyperCard stacks
Carpe Diem! Speed/protocols: 9600; FidoNet 1:114/50
BioSPHERE. Host: NovaLink; 10 p.m.-8 a.m.
Arizona Macintosh User's Group BBS. Host: 2nd Sight/RRH; Speed/protocols: 2400; FidoNet 1:114/56; AMUG members only $30 annual fee
Tucson Apple Core (TAC). Host: 2nd Sight/RRH; speed/protocols: HST; FidoNet 300/15; $20/yr.
Mactivities. Host: 2nd Sight/RRH; speed/protocols: HST; fee for downloading
Phoenix Red Ryder Host #1. Host: 2nd Sight/RRH; speed/protocols: 9600
<table>
<thead>
<tr>
<th>Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>602-881-8720</td>
<td>First DIBS. Host: 2nd Sight/RRH; speed/protocols: 2400; FidoNet 1:300/7</td>
</tr>
<tr>
<td>602-941-3747</td>
<td>EyeNet &quot;HS&quot;. Speed/protocols: 9600; Opus 1:114/14; a.k.a. Mac Shack. Stack Shack, Mac Wizards</td>
</tr>
<tr>
<td>602-947-0587</td>
<td>4th Wave/AMUG II. Host: 2nd Sight/RRH; speed/protocols: 2400; FidoNet 1:114/53; licensed NewsBytes source</td>
</tr>
<tr>
<td>602-996-0078</td>
<td>Tiger's Den. Host: 2nd Sight/RRH; speed/protocols: HST; FidoNet 1:114/27; $25/yr. for DL access</td>
</tr>
<tr>
<td>603-228-0705</td>
<td>Recovery BBS. Speed/protocols: HST; FidoNet 1:132/131.0</td>
</tr>
<tr>
<td>603-352-3001</td>
<td>No Name BBS</td>
</tr>
<tr>
<td>603-431-6441</td>
<td>SeaMac NH</td>
</tr>
<tr>
<td>604-465-0017</td>
<td>B.C. Macintosh BBS. Host: 2nd Sight/RRH; speed/protocols: 2400</td>
</tr>
<tr>
<td>604-574-1199</td>
<td>Generic BBS</td>
</tr>
<tr>
<td>604-943-1612</td>
<td>Sunshine BBS. Host: 2nd Sight/RRH; speed/protocols: 2400</td>
</tr>
<tr>
<td>606-291-6502</td>
<td>Blue Chip Mac</td>
</tr>
<tr>
<td>606-572-5375</td>
<td>MacCincinnati BBS. Speed/protocols: 2400; Covington, KY</td>
</tr>
<tr>
<td>607-256-0200</td>
<td>The Lowlands. Mac software; V.32</td>
</tr>
<tr>
<td>607-256-0824</td>
<td>C.A.D.E.F. Speed/protocols: HST; TF; V.32bis; Sysop: Scott Mandell</td>
</tr>
<tr>
<td>607-256-3937</td>
<td>Mega-City One. V.42; V.42bis; 24 hrs.</td>
</tr>
<tr>
<td>607-257-4238</td>
<td>Backdoor BBS. Speed/protocols: 1200; 24 hrs.</td>
</tr>
<tr>
<td>607-257-5822</td>
<td>Memory Alpha. Speed/protocols: 2400; 24 hrs.; GIFs</td>
</tr>
<tr>
<td>607-257-6134</td>
<td>Incoherent Light. Speed/protocols: 2400; Midnight-10 p.m.; except Sun. 10 a.m.-5 p.m.</td>
</tr>
<tr>
<td>607-257-7421</td>
<td>Not Just Another BBS. Speed/protocols: 2400; 24 hrs.</td>
</tr>
<tr>
<td>607-257-7820</td>
<td>All Broccoli Aside. Speed/protocols: 2400; 24 hrs.</td>
</tr>
<tr>
<td>607-272-0601</td>
<td>The Night BBS. Speed/protocols: 2400; alternate number -6453 before 9 p.m.</td>
</tr>
<tr>
<td>607-272-1371</td>
<td>The Forum BBS. Speed/protocols: 2400; 24 hrs.</td>
</tr>
<tr>
<td>607-272-3399</td>
<td>The Police Box. Speed/protocols: 2400; 10 p.m.-6 a.m.</td>
</tr>
<tr>
<td>607-272-4060</td>
<td>Total Perspective Vortex. Speed/protocols: 2400; 24 hrs.</td>
</tr>
<tr>
<td>607-272-7002</td>
<td>Rainbow Bridge BBS. Speed/protocols: 19,200; 24 hrs.</td>
</tr>
<tr>
<td>Phone</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>607-277-7979</td>
<td>Alchemy International. Speed/protocols: 14,400; 24 hrs.</td>
</tr>
<tr>
<td>607-529-3487</td>
<td>Nightlife. Speed/protocols: 19,200; 24 hrs.</td>
</tr>
<tr>
<td>607-844-4475</td>
<td>Tompkins/Cortland BBS. Speed/protocols: 2400; 24 hrs.</td>
</tr>
<tr>
<td>607-844-9891</td>
<td>The GRAFT Magazine BBS. Speed/protocols: 1200</td>
</tr>
<tr>
<td>608-221-3841</td>
<td>Mac BBS. Host: 2nd Sight/RRH; speed/protocols: 2400; $15/yr. for download access</td>
</tr>
<tr>
<td>608-244-0852</td>
<td>Buyer's Review. Opus; reviews SIG; 1:121/6</td>
</tr>
<tr>
<td>608-256-1177</td>
<td>Station 699. Host: Hermes</td>
</tr>
<tr>
<td>608-836-9473</td>
<td>Madisound. Audio BBS</td>
</tr>
<tr>
<td>609-985-4750</td>
<td>Armoury II. Host: Hermes; speed/protocols: 9600</td>
</tr>
<tr>
<td>612-290-9777</td>
<td>MacExchange II. Host: Hermes; speed/protocols: 2400</td>
</tr>
<tr>
<td>612-338-8844</td>
<td>System BBS. Speed/protocols: 2400; HW/SW sales</td>
</tr>
<tr>
<td>612-377-2197</td>
<td>Railway Post Office. Host: Hermes; Speed/protocols: 2400</td>
</tr>
<tr>
<td>612-420-5850</td>
<td>Tower Exchange</td>
</tr>
<tr>
<td>612-420-7811</td>
<td>Tower Exchange BBS. Host: Hermes; speed/protocols: 9600</td>
</tr>
<tr>
<td>612-426-6687</td>
<td>Electronic Albatross</td>
</tr>
<tr>
<td>612-535-3196</td>
<td>Real American BBS. Host: Hermes; speed/protocols: 14,400</td>
</tr>
<tr>
<td>612-546-1624</td>
<td>Creative Solutions. Host: Hermes; speed/protocols: 2400</td>
</tr>
<tr>
<td>612-557-8925</td>
<td>Glacier BBS. Host: Hermes; speed/protocols: 2400</td>
</tr>
<tr>
<td>612-571-7774</td>
<td>ExchangeNET. Speed/protocols: 2400; FidoNet 1:282/65</td>
</tr>
<tr>
<td>612-572-8370</td>
<td>Syndicate. Host: Hermes; speed/protocols: 2400</td>
</tr>
<tr>
<td>612-636-4285</td>
<td>Desert Oasis. Host: Hermes; speed/protocols: HST</td>
</tr>
<tr>
<td>612-636-7580</td>
<td>DTP Exchange. Speed/protocols: 9600; FidoNet 1:282/61; DTP only; over 300 laser fonts online</td>
</tr>
<tr>
<td>612-642-4629</td>
<td>Conus BBS. Host: 2nd Sight/RRH; speed/protocols: 9600; FidoNet 1:282/24; 6 p.m.-8 a.m. CST M-F; 24 hrs.</td>
</tr>
<tr>
<td>612-778-1222</td>
<td>Caverns of Depth. Host: Hermes; speed/protocols: 2400</td>
</tr>
<tr>
<td>612-825-0595</td>
<td>WALK in the SHADOWs of the NIGHT. Host: Hermes; speed/protocols: 19,200; 1.5 gb storage; alternate number -0596</td>
</tr>
<tr>
<td>612-888-3712</td>
<td>Bloomington BBS. FidoNet 1:282/22.1</td>
</tr>
</tbody>
</table>
### Appendix A  Macintosh Bulletin Board Services

**Around the Country**

<table>
<thead>
<tr>
<th>Number</th>
<th>Service</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>612-937-9464</strong></td>
<td>Mac Shaak</td>
</tr>
<tr>
<td><strong>613-233-1474</strong></td>
<td>JUNGLE_electric. Host: NovaLink; speed/protocols: HST; 3 nodes; design DTP support</td>
</tr>
<tr>
<td><strong>613-562-2624</strong></td>
<td>The Great MacHouse. Speed/protocols: 9600, V.32; Victoria, Australia</td>
</tr>
<tr>
<td><strong>613-729-2763</strong></td>
<td>MacOttawa. Host: 2nd Sight/RRH; speed/protocols: 2400; 10-day trial; $45/yr.; FidoNet 1:163/124</td>
</tr>
<tr>
<td><strong>613-739-1030</strong></td>
<td>EntrNet-Q. Speed/protocols: 2400; FidoNet 1:163/224</td>
</tr>
<tr>
<td><strong>614-385-3870</strong></td>
<td>Beta Traders. Speed/protocols: 9600</td>
</tr>
<tr>
<td><strong>615-383-0727</strong></td>
<td>Nashville Exchange. Speed/protocols: 2400; FidoNet 116/19; 18 lines</td>
</tr>
<tr>
<td><strong>615-434-2551</strong></td>
<td>INFO*LINK. Speed/protocols: 2400; Mac, DOS, others; 3 lines; alternate numbers -2995, -2875</td>
</tr>
<tr>
<td><strong>616-364-1958</strong></td>
<td>MOB Scene. Speed/protocols: 2400; 24 hrs.</td>
</tr>
<tr>
<td><strong>617-227-7986</strong></td>
<td>BCS Info Center TBBS. FidoNet 1:101/121</td>
</tr>
<tr>
<td><strong>617-266-6370</strong></td>
<td>WOLF'S DEN. Host: Hermes; speed/protocols: 2400</td>
</tr>
<tr>
<td><strong>617-324-7310</strong></td>
<td>OnRecord BBS. Host: NovaLink; speed/protocols: 2400; MDI/music; professional and amateur musicians; multi-line; music SW sales; $30/annual fee; FAX service free to members</td>
</tr>
<tr>
<td><strong>617-367-2427</strong></td>
<td>Nova Central. Host: NovaLink</td>
</tr>
<tr>
<td><strong>617-472-8612</strong></td>
<td>PhotoTalk. Speed/protocols: 2400; Opus; 1:101/206</td>
</tr>
<tr>
<td><strong>617-494-0565</strong></td>
<td>4th Dimension BBS. Host: Mansion; speed/protocols: 2400; FidoNet 1:101/450; Mac, IBM, Apple, Tandy</td>
</tr>
<tr>
<td><strong>617-522-9827</strong></td>
<td>Dicaeopolis. Host: Hermes; speed/protocols: 2400; Boston</td>
</tr>
<tr>
<td><strong>617-592-5772</strong></td>
<td>NPI III. Speed/protocols: 2400; Opus; 1:101/193</td>
</tr>
<tr>
<td><strong>617-593-7228</strong></td>
<td>REFLECTIONS BBS. Serving the Boston area</td>
</tr>
<tr>
<td><strong>617-599-0691</strong></td>
<td>Reflections. Host: Hermes; speed/protocols: 2400</td>
</tr>
<tr>
<td><strong>617-625-6747</strong></td>
<td>BCS Mac BBS. FidoNet 101/485</td>
</tr>
<tr>
<td><strong>617-786-9788</strong></td>
<td>BCS Telecomm. Speed/protocols: 9600; Opus; 1:101/122</td>
</tr>
<tr>
<td><strong>617-859-9478</strong></td>
<td>Crystal Palace. Host: NovaLink; speed/protocols: 2400</td>
</tr>
<tr>
<td><strong>617-965-7816</strong></td>
<td>Prism. Host: 2nd Sight/RRH</td>
</tr>
<tr>
<td><strong>618-394-0065</strong></td>
<td>Emerald Keep. Speed/protocols: 9600; Ami Fido; FidoNet 1:288/601</td>
</tr>
<tr>
<td><strong>618-549-1129</strong></td>
<td>Mac Underground. Alternate numbers -6918, -2005</td>
</tr>
<tr>
<td>Phone Number</td>
<td>Service Details</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>619-230-2530</td>
<td>MacHackers. Shadowfax</td>
</tr>
<tr>
<td>619-272-2059</td>
<td>MacUG SafeHouse. Speed/protocols: 2400; FidoNet 1:202/710</td>
</tr>
<tr>
<td>619-371-4776</td>
<td>Kittyhawk. Speed/protocols: 9600; Mac, DOS, and music sections</td>
</tr>
<tr>
<td>619-571-9010</td>
<td>N. San Diego Apple UG. Speed/protocols: 2400; 24 hrs.</td>
</tr>
<tr>
<td>619-697-8709</td>
<td>The FunHouse BBS. Host: 2nd Sight/RRH; speed/protocols: 2400; $10/yr. for access to CD-ROM; Sysop: Daniel Kennedy; La Mesa, CA; 24 hrs.</td>
</tr>
<tr>
<td>619-697-8714</td>
<td>Fun House. Host: 2nd Sight/RRH</td>
</tr>
<tr>
<td>619-758-1700</td>
<td>MacNERDS. Host: Hermes; speed/protocols: 2400; trust deed sales info; alternate number -1105</td>
</tr>
<tr>
<td>702-258-0660</td>
<td>We the People BBS. 1:209/765</td>
</tr>
<tr>
<td>702-293-4655</td>
<td>Mac+ BBS. Host: Hermes; speed/protocols: 2400</td>
</tr>
<tr>
<td>703-241-5492</td>
<td>MacProgrammers. Speed/protocols: 2400; Falls Church, VA</td>
</tr>
<tr>
<td>703-338-6025</td>
<td>ASTEC Support BBS. Speed/protocols: 9600, V.32; Sysop: Rod Paine</td>
</tr>
<tr>
<td>703-351-9173</td>
<td>Radioactive Paradise. Host: Hermes; speed/protocols: 2400</td>
</tr>
<tr>
<td>703-352-1502</td>
<td>Brewer's Barn BBS. Speed/protocols: 9600; FidoNet 1:109/30</td>
</tr>
<tr>
<td>703-524-5520</td>
<td>The Crystal Cave. Invite-only BBS in VA</td>
</tr>
<tr>
<td>703-524-7312</td>
<td>Mount Olympus. Host: Hermes; speed/protocols: 2400</td>
</tr>
<tr>
<td>703-533-3938</td>
<td>BBS THALIA. Host: Hermes; speed/protocols: 2400; show business/actors/agents</td>
</tr>
<tr>
<td>703-631-8772</td>
<td>Bull Board. Speed/protocols: 9600; IBM, Atari, TI, Mac, OS/2, Unix; FidoNet</td>
</tr>
<tr>
<td>703-658-0086</td>
<td>MainLine Mac. Host: Hermes; speed/protocols: HST; alternate number -0087</td>
</tr>
<tr>
<td>703-709-9381</td>
<td>Mormac BBS. Host: Hermes; speed/protocols: 2400; FidoNet 1:109/348</td>
</tr>
<tr>
<td>703-720-1624</td>
<td>End of the Line. FidoNet 1:274/16</td>
</tr>
<tr>
<td>703-825-7533</td>
<td>Culpeper Connection. Host: NovaLink; speed/protocols: 2400</td>
</tr>
<tr>
<td>703-860-1427</td>
<td>Macintosh Network. Speed/protocols: 2400; FidoNet 1:109/328.715; MacVirus echo origin</td>
</tr>
<tr>
<td>703-893-3632</td>
<td>Cluster BBS. Opus 1:109/131; 5 p.m.-7 a.m.</td>
</tr>
<tr>
<td>704-563-6233</td>
<td>Charlotte Apple Comp Clb</td>
</tr>
<tr>
<td>Area Code</td>
<td>Area Code</td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>707</td>
<td>258-2960</td>
</tr>
<tr>
<td>707</td>
<td>485-0987</td>
</tr>
<tr>
<td>707</td>
<td>545-0746</td>
</tr>
<tr>
<td>707</td>
<td>747-0306</td>
</tr>
<tr>
<td>707</td>
<td>795-1721</td>
</tr>
<tr>
<td>707</td>
<td>874-1135</td>
</tr>
<tr>
<td>708</td>
<td>260-9660</td>
</tr>
<tr>
<td>708</td>
<td>291-6660</td>
</tr>
<tr>
<td>708</td>
<td>293-1886</td>
</tr>
<tr>
<td>708</td>
<td>295-6926</td>
</tr>
<tr>
<td>708</td>
<td>352-9282</td>
</tr>
<tr>
<td>708</td>
<td>395-4914</td>
</tr>
<tr>
<td>708</td>
<td>596-3648</td>
</tr>
<tr>
<td>708</td>
<td>654-2064</td>
</tr>
<tr>
<td>708</td>
<td>657-1113</td>
</tr>
<tr>
<td>708</td>
<td>668-8287</td>
</tr>
<tr>
<td>708</td>
<td>741-1995</td>
</tr>
<tr>
<td>708</td>
<td>759-9214</td>
</tr>
<tr>
<td>708</td>
<td>948-7008</td>
</tr>
<tr>
<td>713</td>
<td>356-6004</td>
</tr>
<tr>
<td>713</td>
<td>471-5142</td>
</tr>
<tr>
<td>713</td>
<td>664-9873</td>
</tr>
<tr>
<td>713</td>
<td>799-9016</td>
</tr>
<tr>
<td>714</td>
<td>259-4390</td>
</tr>
<tr>
<td>714</td>
<td>593-6144</td>
</tr>
</tbody>
</table>
MacElite
Unknown Elite #3
Unknown Elite #4
Unknown Elite #2
Orange County Mac Data Exchange. Host: Hermes; speed/protocols: 9600
Spider Island Software BBS. Demo of Mac-like telecom interface: Telefinder
Unknown Elite #1
Nymphotic Zitron. Alternate number -1971
Beyond This World. Sysop: Marshall Blake
SIGnet.Canada. Speed/protocols: 2400; FidoNet 1:103/328
MacVille
Grandpa's. Host: 2nd Sight/RRH; speed/protocols: 2400
Mac's Last Stand. Host: 2nd Sight/RRH; speed/protocols: 2400; FidoNet 1:260/211
Mac's Last Stand Node 1. Host: 2nd Sight/RRH; speed/protocols: 2400; donation for more online time; Sysop: James VanGeyten; Rochester, NY; FidoNet 260/256; 24 hrs.
Mac's Last Stand Node 2. Host: 2nd Sight/RRH; speed/protocols: 9600; donation for more online time; Sysop: James VanGeyten; Rochester, NY; FidoNet 260/256; 24 hrs.
AARDVARK Burrow. Host: 2nd Sight/RRH; speed/protocols: 2400
Rochester Hamfest BBS. Speed/protocols: 2400
Forest Hills BBS. Host: 2nd Sight/RRH; speed/protocols: 9600; Queens, NY
Wall. Host: 2nd Sight/RRH; speed/protocols: HST; FidoNet 1:107/604
Laserboard. Speed/protocols: HST
Black Box. Speed/protocols: 9600; FidoNet 107/615; MNP-5; 3 lines
The Movie BBS
NY Online. Speed/protocols: 2400; Brooklyn, NY
Not Even Odd for Your Mac. Forest Hills, NY
719-598-8413 Eagles' Nest. Speed/protocols: 2400; Opus 1:128/18; School Dist. #20
719-637-1458 Scorpion. Host: 2nd Sight/RRH; speed/protocols: HST, 9600; FidoNet 1:128/46
801-374-5438 Mainly Macintosh. Host: Hermes; speed/protocols: 2400
801-634-3655 Mac Plus BBS. Host: 2nd Sight/RRH; speed/protocols: 2400; 3 p.m.-8 a.m. MST; FidoNet 1:15/10
802-388-9899 Green Mountain Mac. Speed/protocols: 9600; FidoNet 1:325/201
802-482-2110 CVU BBS. Speed/protocols: 9600; multi-line; TBBS; FidoNet 325/107
802-496-9330 The Veil of Illusion
803-548-0900 Fort Mill BBS. Host: 2nd Sight/RRH; speed/protocols: 9600; FidoNet 376/24
803-576-5710 MacMoore. Host: Hermes; speed/protocols: 9600
804-423-1338 C.F.I. BBS. FidoNet 1:275/328
804-486-7322 Sand Castle. Speed/protocols: 2400; FidoNet 1:275/32; FBBS; a.k.a. RoundTable BBS
804-784-3771 Ye Ole' World BBS
804-978-1076 SubZero II. Host: Hermes; speed/protocols: 9600, V.32
805-687-1001 Fred's Place. Speed/protocols: 2400
805-687-5414 Fred's Place BBS. All Mac files
805-967-0194 UnLimited BBS. Host: Hermes; speed/protocols: 2400; elite fairly new; pretty good
805-986-1277 Farside BBS. Host: Hermes; speed/protocols: 9600; Mac, II, DOS, NUP: Adobe
806-795-6751 Homeboy BBS. Host: Hermes; speed/protocols: 2400; Mac, IBM, Amiga
808-456-8498 MacBBS. Speed/protocols: 2400; FidoNet 1:345/21
808-486-6673 MacBBS. Host: 2nd Sight/RRH; speed/protocols: 2400; EE
813-321-0734 Mercury OPUS. FidoNet 1:3603/20.0
813-371-3600 Grapevine BBS. Speed/protocols: 9600; FidoNet 1:137/201
813-542-5482 ENTREvous II. FidoNet 1:371/1302
813-733-3666 Crash 'n' Burn BBS. Host: Hermes; speed/protocols: 2400
814-337-2021  Magical Mystery Tour BBS. Host: 2nd Sight/RRH; speed/protocols: HST; FidoNet 1:237/503; EE
815-332-3014  Castle Glen Finnain. Host: Hermes; speed/protocols: 2400
817-346-9552  Six Macs Over Texas. Speed/protocols: 2400; FidoNet 1:130/13
817-383-3268  DAMUG BBS. Host: Hermes; speed/protocols: 2400
817-794-5641  WendellNet BBS. Host: Hermes; speed/protocols: 9600; Univ. of Texas
817-924-2922  Obligatory Hendrix Perm. Host: Hermes; speed/protocols: 2400
817-927-2873  Mad Island. Speed/protocols: V.32bis
818-240-5799  The Pit BBS. Host: Hermes; speed/protocols: 2400; any computer; large transfer section
818-355-7872  GreyCastle
818-704-1365  CVMUG BBS. Host: Hermes; speed/protocols: 9600; FidoNet 1:102/804
818-792-1661  Realm of the Darkness. Host: Hermes; speed/protocols: 2400
818-794-4943  Red Dwarf Node 2. Host: Hermes; speed/protocols: 2400; alternate number -5843
818-951-4445  Mac Valhalla. Speed/protocols: 9600; FidoNet 1:102/942
818-965-6241  The Drawing Board
818-966-3630  Mountain Retreat. Sysop: Tom Slick
901-396-7300  USA-Net. Speed/protocols: HST; FidoNet 1:123/25
901-754-9823  NiteMare BBS. Speed/protocols: HST; 1:123/13
901-795-3453  Memphis Online
904-488-9344  Rickards High School BBS. FidoNet 1:3605/40
907-333-4090  Apple Diggins (line 1). Host: 2nd Sight/RRH; speed/protocols: 2400; alternate number -338-4373
907-345-3277  The Druid's Keep. Host: Hermes; AK
907-452-1460  Northermost Node. FidoNet 1:17/38; Fairbanks, AK
907-488-9327  Rice Paddy in North Pole. Speed/protocols: 2400; 5 p.m.-9 a.m. PST M-Sat.; 24 hrs. Sun.
908-388-1676  NJMUG BBS. Host: 2nd Sight/RRH; speed/protocols: 9600 (HST); Sysop: Mike Bielen; Clark, NJ; 107/947; FidoNet; 24 hrs.
Appendix A Macintosh Bulletin Board Services Around the Country

908-469-3450 Dragon's Cave BBS. Host: 2nd Sight/RRH; speed/protocols: 9600 (HST); Sysop: Ralph Merritt; Caldwell, NJ; 269/102; FidoNet 7:520/802; AlterNet; 24 hrs.

908-506-0686 Zeppelin BBS. Speed/protocols: HST

908-666-2013 NJ Mac BBS. Speed/protocols: HST

908-988-0706 Castle Tabby. Host: Mansion; speed/protocols: HST; AlterNet 1:107/412; FidoNet 7:520/412; formerly Mouse's Cottage

912-764-7701 Mac BBS. Host: Hermes; speed/protocols: 2400

913-649-2484 MacVault. Collectors

913-832-2246 The ReaperBahn. A human sexuality board

913-841-2752 Lawrence News Center. Multi-line; FidoNet 1:280/102

913-841-3059 Battleship Armageddon. Speed/protocols: 9600; FidoNet 1:280/101

913-841-9446 The MacRocosm BBS. Host: 2nd Sight/RRH; speed/protocols: 2400; Sysop: Rob Dewhirst; Lawrence, KS; 24 hrs.

914-565-6696 Info-Center BBS. Host: 2nd Sight/RRH; speed/protocols: 9600, HST

914-682-0404 Dead Deckers Society. Host: Hermes; speed/protocols: 2400

914-961-7032 DataShack BBS

914-967-8162 MacHell

915-590-9798 Health Professions BBS. Speed/protocols: HST; FidoNet 1:381/61

916-365-5600 UNICOM BBS. Host: Hermes

916-446-0926 MacNexus. Speed/protocols: 9600; TF; v.42bis; hardware handshake; Bill Davies; Guest/Guest; 8 p.m. - 8 a.m.

916-649-1720 DMUG

918-492-0087 MegaByte Image Center. Speed/protocols: HST; alternate number -298-1901; GIFs

918-747-0250 ComputerCenter Multiuser Online Svcs. Speed/protocols: 9600; multi-line; FidoNet


919-779-6674 Micro Message Service. Speed/protocols: 2400; FidoNet 151/102
The United States Government runs a number of bulletin board systems that you can access with your modem.

The Economic Bulletin Board

If the government can be said to have a flagship BBS, it's probably EBB, the Economic Bulletin Board, which is operated by the U.S. Department of Commerce's Office of Business Analysis. You can inquire about it by making a voice call to (202) 377-1986, or you can dial it up directly with your modem at (202) 377-3870. It's essentially a subscription-based system that describes itself as "a one-stop source of current economic information." It contains "press releases and statistical information from the Bureau of Economic Analysis, the Bureau of the Census, the Federal Reserve Board, the Bureau of Labor Statistics, the Department of Treasury, and several other federal government agencies." It lists over a thousand files, and if you happen to need economic data, the subscription is cheap at the price.
To explore EBB you can log on as a guest (at the “User ID?” prompt just type <guest>). Guests are allotted 20 minutes per call. As a guest, you won’t be able to go to EBB’s file section, but you’ll be allowed to list or download some information files and some sample data files:

# ECONOMIC BULLETIN BOARD INFORMATION

1  EBB Introduction
2  EBB Registration Form
3  EBB Quickstart Manual
4  EBB Technical Help Documentation
5  EBB Modem Numbers Information
9  1992 Release Dates Calendar
10  Files Being Updated This Week
12  List of all EBB Files

SAMPLE ECONOMIC BULLETIN BOARD FILES

20  Federal Reserve Bank of New York Quotations
21  State and Local Government Series
22  Daily Treasury Statement
23  Daily Trade Opportunities
24  Treasury Yield Curve Points
25  International Marketing Insight Reports
26  U.S. AID Procurement Awards Information System

Type Selection or L for list, P to set protocol, <CR> to exit:

For dialing information, I'll quote one of EBB's own information files:

The EBB runs two separate speed services depending on what speed modem you are using. Both services contain identical information and are only separated due to modem speed.

300/1200/2400 bps parameters are: No parity; 8 bit data length; 1 stop bit; the fastest transmission speed your modem supports. Dial 202-377-3870. The EBB supports 32 concurrent normal-speed users.

9600 bps parameters are: No parity; 8 bit data length; 1 stop bit. The 9600 bps service uses U.S. Robotics Dual Standard HST/V.32 modems. Dial 202-377-2584. The EBB high-speed service supports six concurrent users.

Both EBB services are available 24 hours/day, 7 days/week.
EBB's annual subscription fee for 300, 1200, or 2400 bps users is $35, which includes $12 of connect time. For 9600 bps users, the annual subscription fee is $100, which does not include any connect time. Non-9600 bps connect time is billed quarterly at 20 cents per minute 8 a.m.-noon, EST; 15 cents per minute noon-6 p.m., EST; and 5 cents per minute 6 p.m.-8 a.m., EST and 24 hours on weekends. 9600 bps connect time is billed quarterly at 50 cents per minute, 24 hours a day, 7 days a week.

Other U.S. Government BBSs

EBB also posts a list of other U.S. Government BBSs as a service to its subscribers. It always prefaces the list with the assertion that it "cannot guarantee the information." And well it can't. Over the years government BBSs as a group have proven to be almost as evanescent as amateur BBSs run by teenage sysops. Just as unpredictable are their access privileges. A board that's open to the public one day may be closed the next. One reason for the unpredictability is the state of the Federal budget in any given month, and I needn't remind you that the government's ability to regulate its budget isn't so different from that of a teenager with normally low impulse control.

Still, if you're willing to spend some money on telephone time and you have a taste for arcane facts, you may want to try out some of the numbers on the next three pages. You'll probably find that some of the boards no longer exist and that some others no longer permit public access. As for settings, a good first approximation is 2400-8-N-1-full duplex. The bulletin boards in the table are sorted by name within state.
<table>
<thead>
<tr>
<th>State</th>
<th>BBS Name</th>
<th>Sponsoring Agency</th>
<th>Data Phone Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wash. D.C.</td>
<td>ADAIC BBS (ADA Programming)</td>
<td>ADA Information Center</td>
<td>202-694-0215 Line 2: 301-459-3865</td>
</tr>
<tr>
<td>Wash. D.C.</td>
<td>FDIC BBS</td>
<td>Federal Deposit Insurance Corp.</td>
<td>202-371-9578</td>
</tr>
<tr>
<td>Wash. D.C.</td>
<td>Federal Highway Administration</td>
<td>U.S. Dept. of Transportation</td>
<td>202-366-3764</td>
</tr>
<tr>
<td>Wash. D.C.</td>
<td>GAO Information Technical Center</td>
<td>General Accounting Office</td>
<td>202-275-1050</td>
</tr>
<tr>
<td>Wash. D.C.</td>
<td>GSA-IRSC</td>
<td>General Services Administration</td>
<td>202-535-7661</td>
</tr>
<tr>
<td>Wash. D.C.</td>
<td>Megawatts One</td>
<td>U.S. Dept. of Energy</td>
<td>301-353-5059</td>
</tr>
<tr>
<td>Location</td>
<td>Service/Program</td>
<td>Description</td>
<td>Phone Number</td>
</tr>
<tr>
<td>----------</td>
<td>----------------</td>
<td>-------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Wash. D.C.</td>
<td>NADAP BBS</td>
<td>U.S. Navy</td>
<td>202-693-3831</td>
</tr>
<tr>
<td>Wash. D.C.</td>
<td>NASA HQ Info Technology Center</td>
<td>NASA</td>
<td>202-453-9008</td>
</tr>
<tr>
<td>Wash. D.C.</td>
<td>Naval Observatory (1200/E/7/1)</td>
<td>U.S. Naval Observatory</td>
<td>202-653-1079</td>
</tr>
<tr>
<td>Wash. D.C.</td>
<td>NAVDAC BBS</td>
<td>Naval Data Automation Command</td>
<td>202-433-2118</td>
</tr>
<tr>
<td>Wash. D.C.</td>
<td>Science Resource Studies BBS</td>
<td>National Science Foundation</td>
<td>202-634-1764</td>
</tr>
<tr>
<td>Maryland</td>
<td>ALF - Agricultural Library Forum</td>
<td>National Agricultural Library</td>
<td>301-344-8510</td>
</tr>
<tr>
<td>Maryland</td>
<td>Census Bureau BBS</td>
<td>U.S. Bureau of Census</td>
<td>301-763-4576</td>
</tr>
<tr>
<td>Maryland</td>
<td>Census Bureau Personnel BBS</td>
<td>U.S. Bureau of Census</td>
<td>301-763-4574</td>
</tr>
<tr>
<td>Maryland</td>
<td>FCC Public Access Link</td>
<td>Federal Communications Commission</td>
<td>301-725-1072</td>
</tr>
<tr>
<td>Maryland</td>
<td>Lipid Nutritional Library</td>
<td>U.S. Dept. of Agriculture</td>
<td>301-344-1277</td>
</tr>
<tr>
<td>Maryland</td>
<td>MSG - RBBS</td>
<td>David Taylor Naval Research Center</td>
<td>301-227-1042 Line 2: 301-227-3428</td>
</tr>
<tr>
<td>Maryland</td>
<td>NIST/Data Management Info Exchange</td>
<td>National Institute for Standards &amp; Technology</td>
<td>301-948-2048</td>
</tr>
<tr>
<td>Maryland</td>
<td>NIST/Microcomputer Elect. Info Exchange</td>
<td>National Institute for Standards &amp; Technology</td>
<td>301-948-5717 Line 2: 301-948-5718</td>
</tr>
</tbody>
</table>
(table continued)

<table>
<thead>
<tr>
<th>State</th>
<th>Service</th>
<th>Agency/Department</th>
<th>Phone Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maryland</td>
<td>NOAA BBS</td>
<td>National Oceanic &amp; Atmospheric Administration</td>
<td>301-770-0069</td>
</tr>
<tr>
<td>Maryland</td>
<td>State Data Ctr./Business - Ind. Data Ctr.</td>
<td>U.S. Bureau of the Census</td>
<td>301-763-1568</td>
</tr>
<tr>
<td>Virginia</td>
<td>Computer Communications Network (CCN)</td>
<td>U.S. Dept. of the Navy</td>
<td>703-602-3693</td>
</tr>
<tr>
<td>Virginia</td>
<td>DASC-ZSA</td>
<td>Defense Logistics Agency</td>
<td>703-274-5863</td>
</tr>
<tr>
<td>Virginia</td>
<td>Fort Myer O Club BBS</td>
<td>Fort Myer Officers Club</td>
<td>703-524-4159</td>
</tr>
<tr>
<td>Virginia</td>
<td>Geological Survey BBS</td>
<td>U.S. Geological Survey</td>
<td>703-648-4168</td>
</tr>
<tr>
<td>Virginia</td>
<td>National Biological Impact Assessment Program</td>
<td>U.S. Dept. of Agriculture</td>
<td>703-231-3858</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Line 2:</td>
<td>800-624-2723</td>
</tr>
</tbody>
</table>
Obtaining Hardware Handshake Cables

Ever since the appearance several years ago of MNP modems, the subject of Macintosh modem cables has been a source of considerable confusion and frustration. A good part of the blame lies with Apple Computer. Modems attempt to communicate with computers through the internationally standard RS-232 serial connection. That's what virtually all external modems are made to do, and it's also what most non-Macintosh computers are prepared for them to do.

On the Macintosh, however, RS-232 is implemented only partially. If you have a non-MNP, non-V.42 modem whose top speed is 2400 bps or slower, none of this need concern you, because you won't derive any particular benefit from using hardware handshaking. If you need flow control at all (in practice that won't be very often) XON/XOFF will serve your purposes perfectly well, and any old Macintosh modem cable that works at all should do the job.
Hardware handshaking will also be of only academic interest if you have an older Mac with a DB-9 serial port (see Figure B.1 below), as you can’t use hardware handshaking regardless of what cable you have.

![Figure B.1](image)

Why are older Macs left out in the cold? Well, as we’ve noted elsewhere, another name for hardware handshake flow control is RTS/CTS flow control. RTS (request to send) and CTS (clear to send) are two of the lines that make up an RS-232 serial connection, and in hardware handshaking your Mac and your modem use both lines to tell each other when to start and stop sending data. Table B.1 is from Apple’s “Macintosh Technical Notes #10” entitled “Pinouts”:

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description/Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Ground</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>+5V</td>
<td>See <em>Inside Macintosh</em> for power limits</td>
</tr>
<tr>
<td>3</td>
<td>Ground</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>TxD+</td>
<td>Transmit data line</td>
</tr>
<tr>
<td>5</td>
<td>TxD-</td>
<td>Transmit data line</td>
</tr>
<tr>
<td>6</td>
<td>+12V</td>
<td>See Macintosh hardware chapter for power limits</td>
</tr>
<tr>
<td>7</td>
<td>HSK</td>
<td>HandShake: CTS or TRxC; depends on Zilog 8530 mode</td>
</tr>
<tr>
<td>8</td>
<td>RxD+</td>
<td>Receive data line; ground this line to emulate RS-232</td>
</tr>
<tr>
<td>9</td>
<td>RxD-</td>
<td>Receive data line</td>
</tr>
</tbody>
</table>

As you can see, the DB-9 serial port has only one handshake line. Apple envisions it supporting the CTS line of a modem, but that leaves no support for the RTS line, which, as you’d expect, makes RTS/CTS flow control impossible. Still, don’t take that to mean you can’t use a high-speed modem with an older Mac. See Chapter 14 on high-speed modems for advice on how
to get by without a hardware handshake cable regardless of what kind of modem you use.

If you have a Mac Plus or newer computer and a modem that does support MNP or V.42 or one whose top speed exceeds 2400 bps, then I'd recommend getting a true Macintosh hardware-handshake cable. Such an animal may have come with your modem when you bought it, but it also might not have. Quite a few high-speed modems, hardware error-checking modems, and hardware data-compression modems have been sold with cables that don't support hardware handshaking. That's happening less and less frequently as both vendors and the modem-buying public become more and more sensitive to the problem, but some non-hardware handshake cables are still being sold with modems that would be best off using hardware handshaking.

If you're not sure if your cable supports hardware handshaking, keep reading. You'll see how hardware handshake cables are typically wired, and you can always use a continuity checker to see what kind of cable you have. If you know your cable doesn't support hardware handshaking and you'd like one that does, you can make a cable yourself or buy one ready-made. In the latter event, if your regular source for Macintosh products is reputable and knowledgeable, you may prefer to try them first. Here are some alternative sources for ready-mades:

<table>
<thead>
<tr>
<th>Source</th>
<th>Address</th>
<th>Phone</th>
<th>Fax</th>
</tr>
</thead>
<tbody>
<tr>
<td>CompUnite Cables</td>
<td>P.O. Box 10949</td>
<td>1-800-835-5514</td>
<td></td>
</tr>
<tr>
<td>Celestin Company</td>
<td>Oakland, CA 94610-0949</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Computer Friends, Inc.</td>
<td>14250 N.W. Science Park Drive</td>
<td>(800) 547-3303</td>
<td></td>
</tr>
<tr>
<td>MacConnection</td>
<td>14 Mill Street</td>
<td>1-800-800-4444</td>
<td>603-446-7791</td>
</tr>
<tr>
<td>Maya Computer</td>
<td>Waitsfield, VT 05673</td>
<td>1-800-541-2318</td>
<td>802-496-8110</td>
</tr>
</tbody>
</table>

Some Cable Anatomy

As formally defined, an RS-232 serial connection consists of a number of lines with specific names and purposes. Each line name comes with a standard abbreviation. There's the transmit data (TXD) line, the receive data (RXD) line, the data terminal ready (DTR) line, and so forth. All the lines fall into two broad categories: data lines and signal lines. The data lines carry data
in serial fashion; i.e., one bit after another. The signal lines are typically two-state (i.e., on/off) indicators through which the modem and computer can tell each other what state they're in ("I'm ready," "I'm not ready," "I'm on," "I'm off," etc.). A specified voltage on a signal line signifies "on" or "true," while another voltage signifies "off" or "false." Thus, if a computer is raring to function as a data terminal, it might turn DTR on to let the modem know that the data terminal to which it's connected is indeed ready.

The purpose of any Macintosh modem cable is to link the serial port of a modem to one of the serial ports on the computer. I say "one of" because there are, of course, two Macintosh serial ports, the modem port and the printer port. Either one can handle a modem, but using the modem port is generally a better idea. If both ports happen to be busy at the same time, it's the modem port that gets priority when the Mac checks for data-transfer activity. And if the other port is allowed to hog CPU time, it's conceivable you might lose some data connection that you've already made.

The modem end of a modem cable looks more or less like Figure B.2 below.

Figure B.2. The DB-25 serial port of nearly all external general purpose modems. Pin numbers apply when looking head-on at a female connector.
And this is how your modem's pins are likely to be wired:

*Table B.2. Modem Pinouts*

<table>
<thead>
<tr>
<th>Signal</th>
<th>Abbreviation</th>
<th>DB-25 Pin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame ground</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Transmit data</td>
<td>TXD</td>
<td>2</td>
</tr>
<tr>
<td>Receive data</td>
<td>RXD</td>
<td>3</td>
</tr>
<tr>
<td>Request to send</td>
<td>RTS</td>
<td>4</td>
</tr>
<tr>
<td>Clear to send</td>
<td>CTS</td>
<td>5</td>
</tr>
<tr>
<td>Data set ready</td>
<td>DSR</td>
<td>6</td>
</tr>
<tr>
<td>Signal ground</td>
<td>GND</td>
<td>7</td>
</tr>
<tr>
<td>Data carrier detect</td>
<td>DCD</td>
<td>8</td>
</tr>
<tr>
<td>Data terminal ready</td>
<td>DTR</td>
<td>20</td>
</tr>
</tbody>
</table>

If you read your modem manual, you'll probably also discover a high-speed line (probably pin 12) and a ring indicator line (probably pin 21), as well as two or three lines that are optional, proprietary, special, reserved, or whatever. As regards the basic Macintosh modem cable, those other lines aren't immediately relevant.

At the Mac end of a Macintosh modem cable, we have either a DB-9 or a mini DIN-8 connector. If it's a DB-9, we put Table B.1 together with Table 8.2 and come up with:

*Table B.3. Typical "Older Mac"-to-Modem Cable*

<table>
<thead>
<tr>
<th>DB-9 connector on Mac</th>
<th>DB-25 connector on Modem</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Signal</strong></td>
<td><strong>Pin</strong></td>
</tr>
<tr>
<td>Ground</td>
<td>GND</td>
</tr>
<tr>
<td>Transmit data</td>
<td>TXD-</td>
</tr>
<tr>
<td>Data terminal ready (+12V)</td>
<td>DTR</td>
</tr>
<tr>
<td>Handshake</td>
<td>HSK</td>
</tr>
<tr>
<td>Tied to pin 3 of DB-9</td>
<td>GND</td>
</tr>
<tr>
<td>Receive data</td>
<td>RXD-</td>
</tr>
<tr>
<td>Frame ground</td>
<td></td>
</tr>
</tbody>
</table>
Occasionally you'll see some variations on the configuration in Table 8.3. Some manufacturers haven't bothered to connect the frame grounds (i.e., the chassis-to-chassis ground wire) running between pin 1 of each connector. Others don't implement the DCD line running between pin 7 of the DB-9 and pin 8 of the DB-25. Even when they don't, the cable—quite miraculously—works just about all the time.

Macs from the Plus on up have mini DIN-8 serial ports, so the computer end of their modem cables will look like Figure B.3 below.

![Figure B.3. The mini DIN-8. Pin numbers apply when looking head-on at a female connector.](image)

The connector's pinouts are as follows:

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>HSKo</td>
<td>Output handshake</td>
</tr>
<tr>
<td>2</td>
<td>HSKi</td>
<td>Input handshake</td>
</tr>
<tr>
<td>3</td>
<td>TxD−</td>
<td>Transmit data line</td>
</tr>
<tr>
<td>4</td>
<td>Ground</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>RxD−</td>
<td>Receive data line</td>
</tr>
<tr>
<td>6</td>
<td>TxD+</td>
<td>Transmit data line</td>
</tr>
<tr>
<td>7</td>
<td>Not connected</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>RxD+</td>
<td>Receive data line</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Grounded to emulate RS-232)</td>
</tr>
</tbody>
</table>
Now, back before hardware handshaking was an issue, a modem-to-mini DIN-8 cable for the Mac would likely be wired more or less like this:

**Table B.5. Typical Non-Hardware Handshake Modem Cable**

<table>
<thead>
<tr>
<th>Mini DIN-8 connector on Mac</th>
<th>DB-25 connector on modem</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Signal</strong></td>
<td><strong>Pin</strong></td>
</tr>
<tr>
<td>Output handshake</td>
<td>HSKo</td>
</tr>
<tr>
<td>Input handshake</td>
<td>HSKi</td>
</tr>
<tr>
<td>Transmit data</td>
<td>TXD-</td>
</tr>
<tr>
<td>Ground</td>
<td>GND</td>
</tr>
<tr>
<td>Receive data</td>
<td>RXD-</td>
</tr>
</tbody>
</table>

Cables of that ilk are sometimes called *standard* Macintosh modem cables. The standard turns out to vary a bit from maker to maker, but with the TXD- and RXD-lines hooked up properly and the grounds properly grounded it will very likely do the job.

**Hardware Handshake Cables**

Enter RTS/CTS flow control. It arrived in force with the first wave of 2400 bps MNP-5 modems. Until then, XON/XOFF (software) flow control was all that Mac telecommunicators needed. An XON character (AQ) meant "Gimme data" and an XOFF character (AS) meant "Hold your horses!" If you were wise, you didn’t use XON/XOFF flow control during XModem file transfers, because there was always the chance that some eight-bit string of binary data might happen to be the ASCII code for AS or AQ.

But the new MNP-5 modems came with several new capabilities. They could service the RTS (request to send) and CTS (clear to send) lines in the RS-232 serial connection for purposes of flow control, and they could also compress and decompress the data they transmitted. Your modem might take data from the phone line at 2400 bps, but if the data were compressed very tightly
the modem could, after decompressing the data, send it on to your computer a lot faster—theoretically at 9600 bps. At times that could be faster than your computer was able to handle it, so the modem could also be instructed to monitor the RTS line continuously. If your terminal program was hardware-handshake savvy, it could vary the voltage level on the RTS line in such a fashion as to tell your modem either that your computer wants more data or to hold off sending because the computer is busy processing the data it already has.

That's if you are receiving data. When you are sending data, your modem talks to your computer through the CTS line. One voltage level tells the computer it is all right to send more data; a second voltage level tells it to wait until the modem handles the data already sitting in its buffer.

To implement this newer form of handshaking, Hayes Microcomputer suggested that Macintosh cables for its V-series modems be wired as follows:

Table 8.6. Hayes V-series Modem Cable

<table>
<thead>
<tr>
<th>Mini DIN-8 connector on Mac</th>
<th>DB-25 connector on modem</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Signal</strong></td>
<td><strong>Pin</strong></td>
</tr>
<tr>
<td>Output handshake</td>
<td>HSKo</td>
</tr>
<tr>
<td>Input handshake</td>
<td>HSKi</td>
</tr>
<tr>
<td>Transmit data</td>
<td>TXD-</td>
</tr>
<tr>
<td>Ground</td>
<td>GND</td>
</tr>
<tr>
<td>Receive data</td>
<td>RXD-</td>
</tr>
<tr>
<td>Transmit data</td>
<td>TXD+</td>
</tr>
<tr>
<td>Receive data</td>
<td>RXD+</td>
</tr>
</tbody>
</table>
And soon afterward there came into being a consensus hardware hand­shake cable whose wiring has become more or less standard. This is the way it's configured:

Table B.7. The "Standard" Hardware-Handshake Modem Cable

<table>
<thead>
<tr>
<th>Mini DIN-8 connector on Mac</th>
<th>DB-25 connector on modem</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Signal</strong></td>
<td><strong>Pin</strong></td>
</tr>
<tr>
<td>Output handshake</td>
<td>HSKo</td>
</tr>
<tr>
<td>Input handshake</td>
<td>HSKI</td>
</tr>
<tr>
<td>Transmit data</td>
<td>TXD-</td>
</tr>
<tr>
<td>Ground</td>
<td>GND</td>
</tr>
<tr>
<td>Receive data</td>
<td>RXD-</td>
</tr>
<tr>
<td>Transmit data</td>
<td>TXD+</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If you compare the wiring schemes in Tables B.6 and B.7, you'll see they're largely similar. The Hayes cable has pins 4 and 8 on the Mac side going separately to pins 1 and 7 tied together on the modem side, but that yields much the same result as you get from the “standard” cable by tying pins 4 and 8 together on the Mac side and sending just one wire to pin 7 on the modem side. The fact that frame ground and signal ground aren't tied together on the modem side of the standard cable ought not to affect performance.

What about the fact that the “standard” cable has the RTS and DTR lines tied together on the modem side while the Hayes cable doesn't? Well, in most cases that shouldn't affect performance either. Essentially, the DTR (data terminal ready) signal tells the modem whether the terminal, in this case your Mac, is on or off. That's so the modem can take some sort of action—such as hang up—when you turn off your computer. But most modems nowadays default to ignoring DTR anyway. So check with your manual and see if your
modem does indeed default to ignoring DTR. If it does, you're in the clear with either cable. If not, find out which command it uses to ignore DTR (the command sometimes varies from modem to modem) and make sure that the command is part of your initialization string whenever you're using hardware handshaking.

BTW, if you do get a hardware-handshake cable bundled with your non-Hayes modem, it will probably be wired like the "standard" cable in Table B.7. And if you buy a generic hardware-handshake cable, it will probably be wired the same way.

Making Your Own Hardware Handshake Cable

Anyone who's handy with a soldering iron and has access to a decent electronic parts store can make a hardware-handshake cable pretty easily. You'll need a standard DB-25 male connector at one end and a male mini DIN-8 connector at the other. (Make sure it's a mini DIN-8. There's also a DIN-8 that isn't mini. It's often used to connect microphones, as well as for other audio applications, and it's too big to fit the Macintosh serial ports.)

As for wire, you can use shielded audio cable. The cable will use only five lines, but six-conductor shielded cable is far easier to come by. Ground the shield to the frame ground on the modem side (probably pin 1, but check your modem manual just to be safe). Then connect five of the color-coded wires as per Table B.7 and solder the connections. (A tip: the innards of some DB-25 connectors make them much easier to solder to than others; use ease of soldering as a criterion for buying DB-25s.)

If you can't obtain any of the parts locally, here are some mail order alternatives:

Cables to Go
26 West Nottingham
Suite 200
Dayton, OH 45405
1-800-826-7904

Digi-Key Inc.
701 Brooks Avenue South
PO Box 677
Thief River Falls, MN 56701
1-800-DIGI-KEY
Fax: 218-681-3380

Electro Products Inc.
1609 South Central Ave.
Suite 2
Kent, WA 98032
1-800-423-0646
This appendix contains a list of modems widely available in stores and by mail order.

Each unit is shown with its top speed and the protocols it supports. You can assume that V.42bis modems also support MNP-5 unless otherwise indicated. (The reverse isn't true.) Most of the modems in the list are external; those that go inside a Mac are duly noted. Street prices are always volatile, so the ones shown should only be used as guidelines. You'll note that two modems that offer pretty much the same set of features can differ quite a lot in price. As a general rule, high-end modems like those from Hayes, Telebit and U.S. Robotics are better-made and also offer better user support. But low-end modems actually do work, and if you run into trouble and can't get the manufacturer's tech support people on the phone, you can always turn to the bulletin boards for help.
**Generic Modems**

- **MacZone MacTurbo modems, MacWarehouse PowerUser Modems, Mac Products Magic Modems, etc.** Top speed: 2400-9600 bps; Protocols: from MNP-5 on up. They're all basically generic, but they work. Many come with terminal software (usually MacKnowledge or MicroPhone 1.6) and, if they support faxing, with fax software. Some come with cables, others require that you buy your cable separately. The general rule is to buy them by price.

**Global Village**

- **TelePort.** Top speed: 2400 bps; Protocols: MNP-5. Uses ADB port instead of serial port, therefore requires Mac SE or newer. Street price: $139

- **TelePort/Fax 9600.** Top speed: 2400 bps; Protocols: MNP-5. Essentially the same as TelePort, but also sends fax (*but not data*) at 9600 bps. Street price: $185

- **TelePort/FullFax.** Top speed: 2400 bps; Protocols: MNP-5. Essentially the same as TelePort/Fax 9600 but also receives fax at 4800 bps. Street price: $219


**Hayes**

- **Optima 24.** Top speed: 2400 bps; Protocols: V.42bis. Street price: $155

- **Optima 96.** Top speed: 9600 bps; Protocols: V.32, V.42bis. Generic. Street price: $389

- **V-series SmartModem 9600 MAC.** Top speed: 9600 bps; Protocols: V.32 (half-duplex only), Hayes proprietary V-series protocol. Discontinued but available. Street price: $610
- **Ultra 96.** Top speed: 9600 bps; Protocols: V.32, V.42bis, Hayes Express 96 modulation protocol. Many special features for special connections. Street price: $669

- **Ultra 144.** Top speed: 14,400 bps; Protocols: V.32bis, V.42bis, Hayes Express 96 modulation protocol. Expensive. Street price: $799

**Intel**

- **14.4EX.** Top speed: 14,400 bps; Protocols: V.32bis, V.42bis. Essentially generic. Street price: $449

- **9600EX.** Top speed: 9600 bps; Protocols: V.32, V.42bis. Essentially generic. Street price: $399

**Mass Microsystems**

- **Fax Modem.** Top speed: 2400 bps; fax 9600 bps. Portable external. Needs no battery because it uses power from Mac’s serial port. Appropriate for PowerBooks. Street price: $185

**Practical Peripherals**

- **2400SA V.42bis.** Top speed: 2400 bps; Protocols: V.42bis, MNP-5. Essentially generic. Street price: $125

- **PM9600SA.** Top speed: 9600 bps; Protocols: V.32, V.42bis. Compatible with Hayes Ultra 96, but doesn’t support every last one of its commands. Street price: $289

- **SmartPack 9600SA.** Same as above, but comes with Hayes Smartcom II and cable. Street price: $389

**Prometheus**

- **ProModem 2400 Plus.** Top speed: 2400 bps; Protocols: V.42bis, MNP-5. External. Comes with MacKnowledge communications software and cable. Street price: $225

- **ProModem 2400SE.** Top speed: 2400 bps. An internal modem for the Mac SE or SE/30. Doesn’t use the expansion
slot. Can be upgraded to ProModem 2400SE Plus. Comes with MacKnowledge communications software. Street price: $225

- **ProModem 2400SE Plus.** Top speed: 2400 bps; Protocols: MNP-5. An internal modem for the Mac SE or SE/30. Doesn’t use the expansion slot. Comes with MacKnowledge communications software. Street price: $245

- **ProModem 2400 MII Plus.** Top speed: 2400 bps; Protocols: MNP-5. An internal (NuBus) modem for the Mac II family. Comes with MacKnowledge communications software. Street price: $245

- **ProModem Home Office.** Top speed: 2400 bps; Protocols: V.42bis. Send/receive Group III fax at 9600 bps. Voice-mail capability. Street price: $259

- **ProModem 24/96 Mini Fax.** Top speed: 2400 bps. Send/receive fax at 9600 bps. Comes with MacKnowledge, MaxFax software and cable. Street price: $219

- **TravelModem.** Top speed: 2400 bps. For the Mac Portable; send-only fax at 9600 bps. Street price: $225

- **ProModem 9600-EC.** Top speed: 9600 bps; Protocols: V.32, V.42bis. Includes MacKnowledge communications software and cable. Street price: $489

- **ProModem Ultima Home Office.** Top speed: 14,400 bps; Protocols: V.32bis, V.42bis. Send/receive Group 3 fax at 9600bps. Voice-mail capability. Street price: $489

- **ProModem 9600 Plus.** Top speed: 9600 bps; Protocols: V.32, V.42bis. Send/receive Group III fax at 9600. Street price: $645

- **PowerModem.** Top speed: 2400 bps. Fax send at 9600 bps, receive at 4800 bps. An internal modem for Mac Power-Books. Street price: $205
Shiva

- **NetModem V2400.** Top speed: 2400 bps. Has LocalTalk interface for shared use on a network. Allows dialing into network from a remote computer. Street price: $339

- **NetModem V32.** Top speed: 9600 bps; Protocols: V.32. Has LocalTalk interface for shared use on a network. Street price: $1,029

- **NetModem/E.** Top speed: 9600 bps; Protocols: V.32. Has Ethernet as well as LocalTalk and interface. Street price: $1,479

Supra

- **SupraFaxModem Plus.** Top speed: 2400 bps; Protocols: V.42bis. Send/receive fax at 9600 bps. Comes with communications and fax software. Street price: $225

- **SupraFaxModem V.32bis.** Top speed: 14,400 bps; Protocols: V.32bis, V.42bis. Send/receive fax at 9600 bps. Comes with MicroPhone 1.6 and cable. Good buy. Street price: $379

Telebit

- **T3000.** Top speed: 14,400 bps; Protocols: V.32bis, V.42bis, PEP (Telebit's proprietary protocol) optional. The high end of Telebit's line. Street price: $949

- **T2500.** Top speed: 19,200 bps; Protocols: V.32, V.42bis, PEP. Street price: $949


- **T1000.** Top speed: 9600 bps; Protocols: Telebit's proprietary PEP protocol. Fewer features than T2500. Street price: $699

- **T1600.** Top speed: 9600 bps; Protocols: V.32, V.42bis. Security features. Street price: $699

- **QBlazer.** Top speed: 9600 bps; Protocols: V.32, V.42bis. Portable, battery-operated. Street price: $699
U.S. Robotics

- **Courier HST Dual Standard.** Top speed: 14,400 bps; Protocols: HST, V.32bis, V.42bis. Supports HST and V.32bis. A fine piece of equipment. Street price: $775

- **Courier V.32bis.** Top speed: 14,400 bps; Protocols: V.32bis, V.42bis. Doesn't support USR's HST protocol. Street price: $525

- **WorldPort 9600 V.32.** Top speed: 9600 bps; Protocols: V.32, MNP-5. Does not support V.42 or V.42bis, as most other V.32 modems do. Portable, pocket-sized, battery-run. Street price: $475

- **Sportster 9600 V.32.** Top speed: 9600 bps; Protocols: V.32, V.42bis. Street price: $625

- **Sportster 2400.** Top speed: 2400 bps. Essentially generic, but of high quality. Street price: $149

Zoom Telephonics

- **V.32 Turbo.** Top speed: 12,000 bps; Protocols: V.32, V.42bis. Has a 12,000 bps turbo mode that lets it connect to a V.32bis modem at 12,000 bps. Street price: $339

- **VFXV32bisMAC.** Top speed: 14,400 bps; Protocols: V.32bis, V.42bis. Can send/receive Group 3 fax, Classes 1 and 2. Comes with Mac cable and fax software. Good buy. Street price: $319

- **VFXV32MAC.** Top speed: 9600 bps; Protocols: V.32, V.42bis. Can send/receive Group 3 fax, Classes 1 and 2. Comes with Mac cable and fax software. Street price: $279

- **FXVMAC.** Top speed: 2400 bps; Protocols: V.42bis, MNP-5. Can send/receive Group 3 fax, Classes 1 and 2 at 9600 bps. Comes with Mac cable and fax software. Street price: $149

- **AFXMAC.** Top speed: 2400 bps; Protocols: V.42bis, MNP-5. Can send/receive Group 3 fax, Classes 1 and 2; receive fax is at 4800 bps. Comes with Mac cable and fax software. Street price: $99
Being notoriously nomenclature-rich, the field of Macintosh telecommunications demands a glossary. To help you with cross-referencing, phrases within a definition that appear in boldface also have glossary entries of their own.

**AB switch** A device that allows you to connect either of two **cables** to a given port. You might use an AB switch to let your **modem** and a printer share your **modem port**.

**ACK** An "acknowledge" or acknowledgment. In **file-transfer protocols**, a signal from the receiving terminal to the sending terminal acknowledging that a block of data has been received. Acknowledgments come in two types: positive and negative. A positive ACK means the block of data was received without errors; a negative acknowledgment (or **NAK**) means it arrived with errors. On receiving a negative acknowledgment, the sending terminal usually tries to resend the **block** of data.

**ADB** **Apple Desktop Bus**, the circuit that supports the mouse, keyboard and other small peripherals for all Macs of more recent vintage than the Plus.

**ADB keyboards** When Macs began using the Apple Desktop Bus, Mac keyboards changed. Most relevant to telecommunications is the fact that they acquired a **Control** key.
America Online  A commercial online service whose interface is more Mac-like than any of the others. When you connect with America Online you can't use an ordinary terminal program; you have to use their proprietary software.

ANSI  The American National Standards Institute, which sets standards for many areas of the computer industry.

ANSI graphics  A set of standards by which data terminals send and receive graphic images. Most commercial terminal programs support some or all of the ANSI graphics standard. ZTerm, a shareware terminal program, supports a satisfactory subset.

Answerback  A message sent to a remote terminal in response to a specific enquiry upon connecting. The correct response means it's all right to proceed with the connection.

AOL  Abbreviation for America Online.

Apple Modem Tool  A communications tool from Apple Computer that provides a software interface to Apple's Communications Toolbox.

AppleFax modem  An early fax-only modem made by Apple Computer.

ARC  A data compression format for IBM-compatible computers. StuffIt Deluxe and ArcPop.ARC are two Mac programs that can decompress ARC files.

Archive  A Macintosh file that contains one or more other files in compressed form. Files created by the StuffIt family of Macintosh compression programs are commonly called archives.

Archiving  The Mac terminal programs Red Ryder and White Knight use this term one way; everyone else uses it another way. In its more general usage, archiving refers to the process wherein files are compressed and saved in archives. But when White Knight or Red Ryder save the contents of your terminal window, they "archive" your screen to a text file.

Archiving utilities  Programs that perform file compression.

ArcPop.ARC  A Macintosh program that can create or decompress ARC archives.

ASCII  Pronounced ASS-key, it's the American Standard Code for Information Interchange. It defines a set of 128 characters with decimal values ranging from zero to 127. The Mac supports 256 characters, but only the first 128 of them are specified by the ASCII standard.

ASCII Chart  A valuable shareware Desk Accessory that displays all 256 of the Mac's characters along with their keystrokes and decimal and hexadecimal codes.

Asynchronous transmission  A means of transmitting data over phone lines in which transmitted characters need not leave or arrive at any particular time intervals. Each character is framed by a start bit and stop bit so the receiving terminal can tell where one character ends and the next one begins.

AT  The first two characters of most commands in the Hayes language, the standard command set with which computers talk to modems.

ATZ  The Hayes command for resetting a modem to its default settings.
Auto-answer  A feature of most modems in which the modem answers the phone automatically when a call comes in. When auto-answer is disabled, you have to tell your modem to answer the phone by issuing an ATA command. Most modems default to a state in which auto-answering is disabled.

Auto-dialing  A feature of many terminal programs that automates the process of dialing up any of a number of services.

Autotyping  In Smartcom II, the term for sending text.

Background  A program is said to be operating in the background when none of its windows is the frontmost window on your screen. A terminal program that operates in the background lets you work within another program during long file transfers. Virtually all modern Mac terminal programs can function in the background.

Baud, baud rate  The number of times per second that the carrier signal on a given communications channel changes state. Back when transfer speeds of 300 bits per second were the edge of the telecommunications envelope, each such change of state corresponded to the sending of a single bit of data, so “baud rate” and “bits per second” meant essentially the same thing. Nowadays modems typically send more than one bit per state change. A 2400 bit-per-second modem, for instance, sends four bits per state change, so when it’s sending at 2400 bps it’s actually sending at 600 baud. The general public has responded to the confusion by using “baud rate” and “bits per second” interchangeably.

BBS  A computer bulletin board system.

Bell 212A  The standard modulation protocol in the U.S.A. for transmitting data at 1200 bps.

Binary file  All computer files are binary (in that they consist only of ones and zeroes), but in everyday usage a binary file is one whose contents are not all alphanumeric characters. This distinguishes them from text files.

Binhex  Sometimes written “BinHex,” a file format in which any text or binary file is expressed as a string of ASCII characters. Converting a Mac file into Binhex format makes it compatible for transmission to, and storage in, non-Macintosh computers. BinHex is also the name of a Mac program that converts files to and from Binhex format. Various members of theStuffIt family of programs can also perform these conversions.

Bits per second  The number of data bits transmitted during each second along a communications channel. Often abbreviated as "bps." Also see Raw speed.

Block  A string of characters or other binary data transmitted as a single unit from one terminal to another. Also called a packet.

bps  See Bits per second.

Break  A signal sent from one system to another whose purpose is usually to interrupt a given process. The control character ^C (Control + c) often serves the same purpose.
BTW Among telecommunicators, a common abbreviation for "by the way."

Buffer An area of the internal memory of either a computer or a modem in which data is stored temporarily.

Buffered keyboard A technique in which a line of data entered on a keyboard is stored in a buffer and not sent to the modem until the typist issues a carriage return.

Bulletin board system Called simply a BBS, it's a computer-based system for exchanging messages and files over telephone lines. A number of modem manufacturers and public agencies operate bulletin board systems, but the majority are run by amateurs.

BUSY The message normally sent by your modem when it tries to make a data call and encounters a busy signal.

Byte A group of eight binary bits that together represent a single character. Since eight binary bits can take any value between 0 and 255, a byte can represent any one of 256 different characters.

Cable Normally several wires or lines running inside a single coat of insulation.

Capture file A text file in which incoming and outgoing text is saved to disk during an online session.

Carrier, carrier signal The audio frequency signal that modems use to transmit data. A modem imposes changes of state on the carrier signal that correspond to the bits of data it's transmitting. When it does so, it's said to be modulating the carrier. When a modem interprets the state changes of an incoming carrier in order to extract digital data from the carrier, it's said to be demodulating the carrier.

CCITT The International Telegraph and Telephone Consultative Committee, a U.N. agency that lays down international standards for data transmission.

Checksum A technique for verifying the integrity of transmitted data. It's less reliable than a newer technique called CRC.

Chooser Apple System software for the Mac, accessed via the Apple menu, that lets you choose an output device for your computer. Normally the output device is a printer, but it can also be a fax modem.

CIS Abbreviation for CompuServe Information Services.

Clear to send Abbreviated CTS, it's one of the active lines in your modem cable if (and that's a big if) your cable and your Mac both support it. Hardware handshake cables coupled with Macs of Plus-or-later vintage support the CTS line, which is essential to hardware flow control. See Flow control and Request to send.

Command mode One of two modes your modem can assume. In this mode it carries out the Hayes commands you send it. Your modem's other mode is the Online mode. Also see Escape sequence.
Communications tools  System extensions that give you access to the Communications Toolbox. In turn, other applications such as terminal programs give you access to communications tools, so communications tools can be seen as intermediaries between applications and the Communications Toolbox. Communications tools need not be written by Apple Computer, although some are.

Communications Toolbox  Apple System software built into System 7.0 and installable in later versions of System 6.x. A collection of software routines to facilitate communication through the Macintosh modem port and printer port.

Compact Pro  A popular Macintosh program for compressing and decompressing files. It used to be called Compactor.

Compactor  See Compact Pro.

Compression  A technique in which files are made smaller by encoding redundant data. Compression can be performed by software or accomplished directly within your modem. The most popular Macintosh compression utilities are Stuffit and Compact Pro. The most widely used in-modem compression schemes are MNP-5 and V42.bis. Also see Data compression protocols.

CompuServe  The largest commercial online service.

CompuServe B, B+ and Quick-B  Proprietary file-transfer protocols used by CompuServe. B+ is the newest of the three, and it offers advantages similar to those of ZModem.

CompuServe Navigator  A special-purpose commercial terminal program designed for use with CompuServe.

CONNECT  The message normally sent by your modem when it first establishes a connection with the remote modem.

Control characters  Special characters that two connected systems can use to signal each another. See Table 3.1 in Chapter 3 for information on what control characters do.

Control key  A key on ADB keyboards used for sending control characters. If your keyboard doesn't have a Control key, many terminal programs let you "fake" it with either the Command or Option keys.

Copernicus  A Macintosh program for relaying FidoNet messages.

CPU  Central processing unit, the chip that runs your computer.

CRC  Cyclic redundancy checking, a means of verifying the integrity of transmitted data. Also the algorithm used for such verification, in which a number is computed for a given block of data and then transmitted together with the data. When the block is received, its CRC is recomputed. If the recomputed CRC is identical to the received CRC, the data is considered to have been transmitted intact. If not, a transmission error is deemed to have occurred. Cyclic redundancy checking has pretty much superseded the older checksum method of verification.

Creator  Means one thing in the Bible, another on the Mac. All Mac files have a four-character creator code that identifies the application that created them.
CTS  See Clear to send.

D

D200  A data terminal manufactured by Data General Corp. It can be emulated on the Mac by the terminal programs VersaTerm and VersaTerm-PRO.

DART  A utility from Apple to convert bit-for-bit images of floppy disks into compressed files for storage on hard disks. From those files, DART can then reconstruct the floppies.

Data bits  In telecommunications, the number of bits used to represent each character transmitted. Terminal programs normally use either seven or eight data bits per character.

Data compression protocols  Sets of rules that govern data compression performed within modems. The ones most widely used nowadays are V.42bis and MNP-5.

Data rate  See Baud rate and Bits per second.

DB-9 connector  On all Macs of vintage prior to the Plus, the nine-pin connector that constitutes both serial ports (modem port and printer port).

DB-25 connector  On nearly all modems, the 25-pin connector to which you attach your modem cable.

DDExpand  A freeware Macintosh utility program for decompressing files compressed by DiskDoubler.

DEC  Short for Digital Equipment Corp., manufacturer of the VT52, VT100, VT102, and VT220 data terminals.

Default  The initial state of a program or modem before you change its settings.

DG   Short for Data General Corp., manufacturer of the D200 data terminal.

Dialing string  In Hayes language, a dialing command followed by the numbers you want to dial. "ATDT 555-1212" is an example. Sending that string to your modem would cause it to dial 555-1212.

Direct hardware connection  A serial connection between two computers without use of intervening modems. Normally you run what’s called a null modem cable between the serial ports of each computer. It happens that the Mac’s standard ImageWriter I cable is wired so that it can function as a null modem cable.

DiskCopy  A utility from Apple to convert bit-for-bit images of floppy disks into files for storage on hard disks. Like DART, DiskCopy can reconstruct the floppies from those files. Unlike DART, DiskCopy doesn’t compress the disk images.

DiskDoubler  A commercial file-compression program for the Macintosh.

Docs  Short for documentation, usually of a computer program.

DOS  The most widely used operating system on IBM-compatible computers.
**DOS Mounter** Macintosh software that lets you mount DOS-formatted 3-1/2" floppy disks on your Mac. Normally you need a SuperDrive, which the Plus and early SEs lack. But there are external floppy drives for Pluses and SEs that can accept DOS floppies.

**Downloading** Transferring a file from a remote system to your own computer.

**Duplex** See Full duplex, Half duplex, and Local echo.

---

**E-mail** See Electronic mail.

**EAASY SABRE** An online service run by American Airlines. It offers worldwide flight information, online airline reservations, and assorted travel packages. Generally it's accessed through gateways from other online services such as GEnie, CompuServe, and America Online.

**EBB** The Economic Bulletin Board operated by the U.S. Department of Commerce's Office of Business Analysis. Access is available to anyone for a small fee.

**Echo** In the world of BBSs, a networked message base, usually one that's carried on FidoNet, devoted to a given topic.

**Echoing** The process in which a remote terminal echoes each of the characters it receives back to the sending terminal. This is the normal full-duplex arrangement for most connections, and when it's in force the characters you see in your terminal window—including the ones you've typed yourself—have all come from the remote terminal. During half-duplex connections the remote terminal doesn't echo back your typing, so in order for your typing to appear in your terminal window you have to rely on your own terminal to echo your keystrokes. When the local terminal echoes your typing, the arrangement is called local echo.

**Echomail** Messages relayed among participating FidoNet systems. Echomail pertains to public conferences devoted to a given subject, while Netmail generally goes person-to-person.

**Electronic mail** Messages you send and receive using telecommunications techniques.

**ERROR** The message normally sent by your modem when it's in command mode but can't understand some command you've issued.

**Error-checking protocols, error-control protocols** Nearly all of the software file-transfer protocols provide for some form of error checking. YModem-g is the skimpiest in that it checks for errors but isn't capable of asking for a retransmission when it discovers them. For error checking that takes place within your modem, V.42 and MNP-4 are the dominant protocols.

**Error-checking, error control, error-correcting** In telecommunications, the process in which transmitted data is verified and retransmitted, if it was received with errors. Error checking can be performed by your software or inside your modem or both. The procedures that govern error checking are known as protocols. Nearly all software-based file-transfer protocols provide for some kind of error checking, and
modems that support V.42 or MNP-4 can perform error checking without any help from your software.

**Escape sequence** The characters +++ typed in quick succession to your modem. When your modem is in its **online mode**, the escape sequence causes it to switch over to **command mode**.

**Even parity** A method of **error checking** that's not **protocol**-dependent. A character is a string of ones and zeroes. During transmission of data the **parity bit** of each character is set so that the total number of ones is even. That is, if there's already an even number of ones in the character, the parity bit is set to zero. If there's an odd number of ones in the character, the parity bit is set to one. See also **odd parity**.

**Express 96** A proprietary **modulation protocol** used by **Hayes V-Series modems**.

---

**FastComm** A **shareware terminal program** for the Mac.

**Fax modem** A **modem** capable of fax transmission. Some fax modems are send-only devices, but most can send and receive. Fax modems typically send faxes at 9600 bps, but many can receive faxes only at 4800 bps. Fax modems let you "print" a Mac document to a remote fax machine. Received faxes can be printed or viewed on your screen.

**FidoNet** An amateur network that uses ordinary phone lines to link thousands of personal computers around the world. **Echomail** and **Netmail** messages are relayed from system to system until they arrive at their intended destinations.

**File capture** The process in which everything that appears in your terminal window is captured to a **text file** on disk.

**File transfer** The transmission of computer files from one **terminal** to another. Text files are sometimes transmitted without use of any **error-checking protocol**, but **binary files** are normally sent using a protocol like **XModem** or **ZModem**.

**File transfer protocol** The ensemble of **handshake** signals, data formats, and **error-checking** algorithms that two connected **terminals** use when a file is transferred from one to the other. In general, the term refers to software-based **protocols** like **XModem**, **YModem**, **ZModem**, **Kermit**, etc. See **Error checking** and **Error-checking** protocols.

**Firmware** The computer code that's stored in **ROMs**, read-only memory units.

**Flow control** A technique in which a **modem** or computer that's receiving data can signal the sender to halt transmission when it can't handle any more incoming data. When the receiver is ready for more data, it can then signal the sender to resume sending. Flow control in both directions between your own modem and computer is called **local** flow control, and it can be achieved via hardware or software. **Software flow control** uses the **XON** (green light) and **XOFF** (red light) characters as signals. **Hardware flow control**, also called **hardware handshaking** or **RTS/CTS** flow control, uses the **RTS** and **CTS** lines in your **modem cable** to accomplish the same thing.
Full duplex A connection in which data can be transmitted in both directions at the same time. Also see Echoing.

Gateway A means by which one online service gives you access to another online service.

GEnie A popular commercial online service.

Half duplex A connection in which data can be transmitted only in one direction at a given time. Also see Echoing and Local echo.

Handshake, handshaking Signals exchanged between two connected terminals or modems or between a modem and a computer. Modems often do a lot of handshaking when they first connect in order to negotiate the protocols that will govern the connection. Also see Hardware handshake.

Hardware error-checking Error checking that takes place within your modem. See Error checking and Error-checking protocols.

Hardware handshake See Flow control.

Hardware handshake cables Modem cables that facilitate hardware handshaking. Such cables support the RTS and CTS serial lines between your modem and your Mac, but your Mac has to be a Plus or newer to use one. See Flow control.

Hayes compatibility The ability of non-Hayes modems to execute Hayes commands.

Hayes Hayes Microcomputer manufactures modems. The Hayes language, a set of modem commands, was first used in Hayes modems but is now in virtually universal use with all modems.

Hayes command A Hayes language command issued by a computer (or other terminal) to a modem.

Hayes language Also known as the Hayes command set, an ensemble of commands through which you control your modem from your terminal.

Hayes Modem Tool A communications tool from Hayes Microcomputer. It's one of the best such tools for use with modems.

Hermes A host program for the Macintosh.

Host, host program, host mode A system that's set up to answer and respond to data calls when it's unattended is called a host. Bulletin board systems are all hosts. When systems function in this fashion, the host computer is under the control of a host program. Several sophisticated Macintosh terminal programs such as White Knight and MicroPhone II can allow your Mac to be a host. When White Knight does this, it's
said to be in **host mode**. Two Macintosh programs specifically designed for hosting are **Second Sight** and **Hermes**.

**Hot menus** A feature offered by many **host programs** in which your responses to **prompts** don’t have to be followed by carriage returns.

**HST** Short for **High Speed Technology**, a proprietary **modulation protocol** used by U.S. Robotics’ Courier modems.

---

**Initialization string** A string of commands sent to your **modem** before making a connection. The commands enable or disable whatever features of your modem are best suited to the connection.

**ISDN** **Integrated Services Digital Network**, a service offered by telephone companies in which digital data can be transferred at very high speeds without need of any modems. Probably the wave of the future.

---

**K**

**Kermit** A software-based **file-transfer protocol** originally created for use by large mainframe computers. Nowadays it’s giving way to protocols of more recent origin.

**Keyboard buffers** See **Buffered keyboard**.

**Keyboard mapping and remapping** A software-based technique offered by some Macintosh **terminal programs** in which keystrokes are made to produce effects other than the ones they normally produce. A typical example is the remapping of the Option key so that it acts as a **Control key**, which can be useful if your keyboard doesn’t have a Control key and you want to send **control characters**. Keyboards can also be mapped to send a Delete instead of a common Backspace. When your terminal program is emulating one of the **DEC VT** terminals, your numeric keypad is often remapped to support the emulation. Consult your program’s manual for available remappings.

---

**L**

**LAN** See **Local area network**.

**LAP-M** **Link Access Procedures for Modems**, the preferred **error-checking protocol** during a **V.42** connection. If the remote modem doesn’t support LAP-M, V.42 attempts to fall back to **MNP-4**.

**Line noise** Spurious signals that interfere with a telephone connection. They can originate along the lines themselves or within connected modems. Most **modems** contain some form of noise-suppression circuitry. It’s line noise that usually accounts for data transmission errors.
Local area network  A network of two or more computers, printers, modems, etc. in close proximity. In the Macintosh world nowadays, the networked devices are usually connected to one another with ordinary telephone wire. The protocol that governs the movement of data from one device to another is usually either LocalTalk or Ethernet.

Local echo  An arrangement in which your own computer echoes your typing to your terminal window during a half duplex connection. See Echoing.

Log-off  The procedure in which you disconnect from a remote service.

Log-on  The procedure in which you establish a connection with a remote service.

LZH  A data compression format for IBM-compatible computers.

MacBinary  A file format that's particularly well-suited to the transfer of binary Macintosh files in that it allows for the preservation of such Mac-specific attributes as type, creator, resource fork, etc. Most Mac terminal programs default to converting binary files to MacBinary before sending them. When files are being received, most terminal programs recognize the MacBinary format and convert incoming files back to normal Macintosh format on the fly. MacBinary is the successor to BinHex, an earlier format to which binary Mac files were typically converted before sending. MacBinary has significantly less overhead than BinHex.

Macintosh PC Exchange  Software from Apple Computer that makes it relatively easy to mount and read DOS floppies on a MAC. Also see DOS Mounter.

MacKnowledge  A commercial terminal program often bundled with modems.

Macros  Simple automated procedures for your computer. They're sequences of actions normally triggered by a keystroke or the click of a button. Many Macintosh terminal programs have macro capability.

MacUnZip  A shareware Macintosh program for decompressing files in ZIP format.

Mark parity  A non-protocol error-checking technique in which each character's parity bit is set equal to one before transmission. Nowadays it's rarely used. See Even parity for additional information.

McSink  A shareware text processing Desk Accessory that's especially useful for manipulating text received from remote services and for preparing text for sending to remote services.

Memory partition  An area of your Mac's internal memory that's reserved for use by a given application under MultiFinder or System 7.0.

Message base  An area of a bulletin board service or online service devoted to exchanging messages, often on a particular topic.

MicroPhone, MicroPhone II  Popular commercial terminal programs for the Mac. MicroPhone II is one of the most powerful such programs around.
Mini DIN-8 On all Macs from the Plus on up, the connector that constitutes both serial ports (modem port and printer port). Not to be confused with the plain DIN-8, which is larger.

MNP Microcom Networked Protocol. Actually a collection of hardware-based protocols for in-modem error checking and data compression. The protocols are separately called MNP-1 (or MNP class 1 or MNP level 1), MNP-2, etc. For example, MNP-3 governs the removal of start and stop bits from each character before sending. The modem at the receiving end then restores the start and stop bits before passing characters along to the computer, and the process results in a gain of about eight percent in the transmission rate. MNP-4 regulates error-checking and MNP-5 governs compression.

Modem A device for sending and receiving digital data along ordinary telephone lines. Through a process called modulation, modems convert digital data into audio signals by effecting changes in a base audio tone called a carrier. The inverse of that process is demodulation, the process by which receiving modems extract digital data from an incoming carrier. Modem is an acronym for modulator-demodulator.

Modem cable The cable with which you connect your modem to your computer. See also Hardware handshake cable.

Modem driver Software that serves as an intermediary between an application, usually a terminal program, and a particular modem. MicroPhone II makes extensive use of modem drivers.

Modem port One of your Mac's serial ports, the other being the printer port. Your modem can be connected to either one, but because of the way your Mac services its serial ports, the modem port is preferred.

Modulation See Modem, carrier, and Baud rate.

Modulation protocol A set of rules that governs how a modem performs modulation and demodulation. Some common modulation protocols are V.32, V.32bis, PEP, HST, and Express-96.

MUG A Macintosh User Group. The Berkeley Macintosh User Group is known as BMUG, for example.

MultiFinder A feature of Apple's System 6.x which permits several programs, each in its own memory partition, to run at the same time on one Mac. System 7.0 does the same thing as a matter of course.

MultiXfer A simple but interesting shareware terminal program for the Mac.

NAK A negative acknowledgment, usually a signal from one system to another indicating that a given block of data has been received with errors. See ACK.

Netmail Messages relayed among participating FidoNet systems. See Echomail.
NO CARRIER The message normally sent by your modem when it fails to detect a carrier signal or when it loses a carrier signal. Your modem loses a carrier signal when a service to which you've been connected hangs up or otherwise disconnects.

Node In FidoNet, a participating system that relays messages to and from other systems.

Noise See Line noise.

Odd parity Like even parity, except the parity bit is set so that the number of ones in each transmitted character is odd.

Online mode One of two modes your modem can assume. In this mode your modem expects to receive data from a remote service, not commands from your own terminal. See Command mode and Escape sequence.

Online service A commercial system that offers a variety of services to telecommunicators who log on with their modems. The most popular ones among Macintosh users are CompuServe, GEnie, and America Online.

Overhead Extra data and transmission time introduced by a given protocol or by conversion to a special file format.

Pacing See Text pacing.

Packet See Block.

PackIt A Macintosh file-compression program that has fallen into disuse. Files in PackIt format can be decompressed with some other compression programs like StuffIt.

Parity bit The bit in each character that's used for parity checking.

Parity checking A simple form of error control. See Even Parity, Odd Parity, Mark Parity, and Space Parity.

Partition See Memory partition.

PEP Packetized Ensemble Protocol, a proprietary modulation protocol used in some Telebit modems.

PhoneBook A useful shareware Desk Accessory for making voice calls with your modem and maintaining address books.

Pinouts Information about which pins do what for a given connector.

Plain text See Text file.

PortShare Useful commercial software in the form of a Chooser extension that allows several Macs on a local area network to share a modem or other serial device.
Printer port One of your Mac's serial ports, the other being the modem port. It's through the printer port that you connect to a LocalTalk LAN.

Prodigy A commercial online service. It's not especially popular with Macintosh telecommunications.

Prompt A message from a remote service asking you to type in a reply.

Protocol A set of rules that define the way two connected systems perform software-based error checking during file transfers or the way two connected modems perform modulation, error checking, or file compression. See also File-transfer protocol and Modulation protocol.

QuickKeys A popular and powerful commercial macro program for the Macintosh.

Raw speed The number of bits per second that a modem can send or receive unaided by any data compression.

Receive folder The folder on your hard disk to which your terminal program normally directs all incoming files.

Red Ryder Once the most powerful shareware terminal program for the Mac. When it went commercial its name was changed to White Knight.

Red Ryder Host Once the most popular host program with Macintosh bulletin board systems. Its successor is called Second Sight.

Reliable connection A connection between two modems that's overseen by a hardware error-checking protocol like MNP-4 or V.42.

Request to send Abbreviated RTS, it's one of the active lines in your modem cable if (and that's a big if) your cable and your Mac both support it. Hardware handshake cables coupled with Macs of Plus-or-greater vintage support the RTS line, which is essential to hardware flow control. See Flow control and Clear to send.

Result codes Responses from your modem, such as OK, BUSY, or NO CARRIER. Result codes can be verbal or numeric. See your modem manual for how to get the kind of result codes you want.

RJ-11 The designation for ordinary modular telephone connectors. Modems typically connect to your phone line through RJ-11 connectors.

ROMs Read-only memory units. Being non-volatile, their contents can't easily be erased or changed. In modems, ROMs are used to store internal settings and the code that performs various protocol procedures. Also see Firmware.

RTS See Request to send.
RTS/CTS  Another name for hardware-based local flow control. See Request to send, Clear to send, and Flow Control.

S-

S-registers  Storage areas within a modem. The values to which the S-registers are set control various aspects of the modem's performance. Different modems contain different numbers of S-registers. In Hayes language, you set an S-register with the command ATSn=v, where n is the number of the S-register and v is the value to which you want to set it. See your modem manual for more on what its S-registers do.

Screen buffer  The area in your computer's memory in which your terminal program stores text that has appeared in your terminal window. When you scroll your terminal window backwards, your terminal program retrieves data from its screen buffer in order to know how to draw your screen.

Script  A series of instructions to a terminal program that allows the automation of frequently used procedures.

Scroll-back buffer  See Screen buffer.

Sealink  A variant of the XModem file-transfer protocol designed for use with special-purpose phone systems like packet-switching networks and satellite relays. Few terminal programs support it, but it's enough like XModem so that you might be able to get away with a Sealink transfer by telling your terminal program that it's an XModem transfer.

Second Sight  The successor to Red Ryder Host.

Self-extracting archive  A file that contains one or more other files in compressed format and also the computer code needed to decompress them. Because you can run the decompression code directly, SEAs, as they're sometimes called, are actually applications.

Serial ports  See Modem port and Printer port.

Service  Any remote system to which you make a data connection. It can be an online service, a BBS, or your friend's or company's computer.

Session  The time between log-on and log-off with a given service.

Shareware  A method of distributing software that lets you try out a product before paying for it. Shareware is posted in the file libraries of commercial online systems and amateur BBSs. Its quality is variable, but the price is generally right.

Smartcom EZ  A truncated version of Smartcom II.

Smartcom II  A powerful, if sometimes idiosyncratic, general-purpose terminal program from Hayes Microcomputer.

Software handshaking  Another term for software flow control. See Flow control.
Space parity  A non-protocol method of error control in which the parity bit for each character is always set to zero, no matter what the value of the other bits in the character. Nowadays it's rarely used.

Standard Files dialog box  The dialog box that normally appears on your Mac's screen when you ask an application to open or save a file. Within the SF dialog box you can name and direct the file you're saving or pick the file you want to open. Applications can add buttons and other text to the SF dialog box, so each application's version may look a little different.

Start bit  The bit that denotes the beginning of a character in a stream of digital data. Together with the stop bit, it's used to frame each character during asynchronous transmissions.

Stop bit  The bit that denotes the end of a character in a stream of digital data. Together with the start bit, it's used to frame each character during asynchronous transmissions.

String  A contiguous succession of characters.

Strip 8th bit  While eight bits are used to represent each Macintosh character, data is commonly (but not always) sent using just seven bits per character. This tells your terminal program to strip the high order (leftmost) bit of each character when sending and receiving.

Strip high bit  Same as Strip 8th bit.

Stufflt  A family of powerful Macintosh compression programs that includes Stufflt Deluxe (commercial), Stufflt Classic (shareware), and Stufflt Expander (freeware). The latter can decompress files in archives created by either of the other two.

Synchronous transmission  A means of transmitting data over phone lines in which transmitted characters have to leave and arrive at constant time intervals. Also see Asynchronous transmission.

Sysop  Short for system operator; a person who operates a bulletin board system.

Tabby  A program that enables Macintosh systems to relay FidoNet messages and otherwise participate in FidoNet functions.

TeachText  Apple's own no-frills text processor. It comes bundled with all recent versions of the System.

Tektronix 4014 graphics terminal  A special-purpose telecommunications terminal. It can be emulated on the Mac using VersaTerm and VersaTerm PRO.

Tektronix 4105  A special purpose graphics terminal. It can be emulated on the Mac using VersaTerm PRO.

Telebit  A major manufacturer of high-end high-speed modems.

Term-Plus  A shareware terminal program for the Macintosh.
**Terminal** Hardware that allows you to telecommunicate with another system. A teletypewriter is a terminal, as is a DEC VT52 or a Macintosh when it’s running **terminal emulation** software (i.e., a terminal program). Modern terminals typically include a keyboard and a screen. If a terminal uses a **modem**, some people regard the modem as part of the terminal, other people don’t. Also the name of a **shareware** terminal program for the Macintosh.

**Terminal emulation** The process by which a terminal program enables a computer to emulate another terminal. Most Macintosh terminal programs default to emulating either TTY (Teletype) or VT100 terminals. **VersaTerm** is notable among Mac terminal programs for the number of different terminals it can emulate and the accuracy of its emulations.

**Terminal program** A program that makes your computer behave like a data terminal. General-purpose telecom programs for the Mac are terminal programs.

**Terminal window** The main window that a terminal program displays on your Macintosh screen.

**Termulator** A shareware terminal program for the Macintosh.

**TermWorks** A shareware terminal program for the Macintosh.

**TEXT** When it’s all in uppercase, TEXT refers to the file type that the Macintosh assigns to text files.

**Text file** A file that contains only text. In text files there are no formatting information or other data embedded in the text. Each group of eight bits meaningfully corresponds to a character in the Mac’s set of 256 text characters.

**Text pacing** The process by which a terminal program regulates the speed at which characters are sent to the remote terminal to make sure the characters don’t arrive too rapidly for the remote system to handle them. Text pacing is independent of the **data rate** to which your modem or serial port is set.

**Tick** On the Mac, one sixtieth of a second.

**Timeout** In file transfers, the allowable elapsed time between sending a packet of data and receiving an acknowledgment—either an **ACK** or a **NAK**—from the remote system. Usually, if the interval exceeds the timeout value, the system waiting for the acknowledgment terminates the transmission (although it doesn’t disconnect).

**Transfer rate** The rate, usually measured in bits per second, at which data moves from one connected system to another. This is not the same as raw speed, in that data compression can significantly enhance the transfer rate between two systems.

**TTY** Abbreviation for Teletype, one of the most widely used data terminals. Emulating TTY terminals is a relatively simple task for most Mac terminal programs, and it’s the emulation most Mac telecommunicators use most often. It’s also the emulation most often used by BBSs and **online services** such as GENie.

**Type** A four-character code associated with every Macintosh file that identifies its file type. Text files are of type TEXT, applications of type APPL, and so forth. Also see **Creator**.
U

U.S. Robotics A major manufacturer of high-speed modems, especially the Courier line of modems.

UATerm A shareware terminal program for the Macintosh.

UnZip A shareware Macintosh program for decompressing files in ZIP format.

Uploading Transferring a file from your own computer to a remote system.

V

V-series A series of modems manufactured by Hayes Microcomputer. Early high speed V-series modems use only Hayes's Express 96 modulation protocol, but later ones use CCITT protocols as well.

V.22bis The CCITT modulation protocol for 2400 bps.

V.32 The CCITT modulation protocol for 4800 bps through 9600 bps. A V.32 modem must support 9600 bps.

V.32bis The CCITT modulation protocol for modems operating at 4800 bps through 14,400 bps. It includes V.32. A V.32bis modem must support 14,400 bps.

V.42 The CCITT error-checking protocol for modems. It supports two error checking schemes, LAP-M and MNP-4. LAP-M is preferred.

V.42bis The CCITT data compression protocol for modems. It can only be invoked when LAP-M error checking is also in effect.

VersaTerm, VersaTerm-PRO Two very powerful commercial terminal programs for the Mac. They're not for scripting, but they include solid support for the Communications Toolbox and are capable of a number of sophisticated terminal emulations.

VT100 A data terminal manufactured by Digital Equipment Corporation. Quite a few Macintosh terminal programs can emulate the VT100, although some do it better than others.

VT220 A data terminal manufactured by Digital Equipment Corporation. It's not as widely emulated as the VT52 and VT100, but both VersaTerms do a VT220 emulation.

VT52 A data terminal manufactured by Digital Equipment Corporation. White Knight does a good emulation of this terminal.

W

White Knight A popular and powerful terminal program for the Mac. One reason for its popularity is that many Mac telecommunicators began using its shareware predecessor, Red Ryder. It has extraordinary scripting capability.
**Glossary**

**Word wrapping**  Breaking up lines of text in a Macintosh window so that the breaks fall between, and not in the middle of, words.

**X**

**XModem**  Once the standard **file-transfer protocol** for personal computers. The original version provided only for **checksum error checking** and 128-byte packet lengths, but it was improved to use optional **CRC error checking** (in which case it's sometimes called XModem CRC) and 1024-byte packets (in which case it's sometimes called XModem 1K).

**XOFF**  By itself, the ^S (Control-s) character. In **XON/XOFF flow control**, it serves as a "red light" to halt transmission of data.

**XON**  By itself, the ^Q (Control-Q) character. In **XON/XOFF flow control**, it serves as a "green light" to resume transmission of data.

**XON/XOFF flow control**  Another name for software-based flow control. See **flow control**.

**Y**

**YModem**  A **file-transfer protocol** based on XModem. Like XModem 1K, it uses 1024-byte blocks. YModem allows for multiple files to be sent as a batch during a single transmission.

**YModem-g**  A variant of **YModem**. YModem-g is a "streaming" protocol in that there are no **handshaking** pauses between the transfer of individual blocks. The resulting savings in **overhead** make it notably faster than ordinary YModem, but reduced handshaking makes it impossible for erroneous blocks to be retransmitted. If an error is detected in a block, the entire transfer is terminated. YModem-g is therefore really suitable only for connections governed by **hardware error checking**.

**Z**

**ZIP**  A data **compression** format for IBM-compatible computers. **StuffIt Deluxe**, **MacUnZip**, and **UnZip** can all decompress ZIP files on the Mac.

**ZModem**  A very efficient software-based file-transfer protocol of more recent origin than **XModem** and **YModem**. It's a "streaming" protocol like YModem-g, but it allows for retransmission of blocks with errors and accommodates the transfer of files in batches. The protocol of choice for everyday use.

**ZTerm**  A **shareware terminal program** for the Macintosh. It was the first such program to offer ZModem file transfers and has remained extremely popular with Mac telecommunicators.
A
AB switch, 69
ADB, 69
  keyboards, 43, 46, 70
  modems, 69
Add/Strip, 182
Address Book DA, 288
Adobe Type Manager, 41
AlterNet, 302
America Online, 191-2, 215
American National Standards Institute (ANSI), 237
American Standard Code for Information Interchange, See ASCII
Ampère, André, 10
ANSI graphics, 237
Answer phone modem command, 95
Answerback, 101
Apple Desktop Bus, See ADB
Apple File Exchange, 177, 184
Apple Modem Tool, 42, 251
Apple Personal Modem, 18
AppleFax modem, 5, 291
AppleTalk, 24, 297
ARC, 184
Archive, archiving, 54, 143-7
ArcMac, 184
ArcPop.ARC, 184
Arizona Macintosh Users Group (AMUG), 128, 192
Arrow keys, 43
ASCII characters, 16, 43, 61, 136
  file transfer, 134
ASCII Chart DA, 62
  ASLTalk DA, 129
Asynchronous
  telecommunications, 75, 15
  telegraphy, 12
AT command, 82-3
Attention command, See AT
Auto wraparound, 100-101
Auto-answering, 132
Auto-dialing, 103, 234, 239
Autotype, 135
B
BackDown DA, 129
BackFax, 291
Background, 32, 147, 170-71
  file transfers, 170
Backspace key, 45-6, 68, 70, 100, 237
Batch file transfers, 163, 167
Baud rate, 12
Baudot, Émile, 12-13
BBS, 2, 4, 23, 35, 60-61, 73, 76, 115, 127, 133, 191-2
  file libraries, 129
  in a Box, 128
  list (Mac), 301
  list (U.S. Govt.), 325
  operators, See Sysops
  validation survey, 109
Bell character, 45
Bell, Alexander Graham, 13
Berkeley Macintosh Users Group (BMUG), 192
Beta testing, 153
Binary
  files, 22, 136-7, 192
  numbers, 137
  system, 11
BinHex, 139-40, 146
  DA, 140
Bits per second, 12
Block (See also Packet), 157
BMUG, 192
Break signal, 48-49
BTW, 61
Buffer, 25
  archiving, 54
Bulletin board systems, See BBS
Byte, 13
C
Cable
  hardware handshake, 23, 27, 167, 273, 331, 333
  hardware handshake, making your own, 340
hardware handshake, wiring, 338-9
modem, 23, 25-7
modem, DB-9, wiring, 335
non-hardware handshake, wiring, 337
Call waiting, 104
disabling, 104
Capture file, 53-6, 259
Captured text, appending, 55
Capturing text, 52, 127
Carrier, 14
speed, 271, 279
telegraphy, 14
Case sensitivity, 112
CCITT protocols, 20, 22, 266-71
Checksum, 162
Chooser, 293-4
extensions, 276
CIS, See CompuServe
ClarisWorks, 30
Clear screen character, 45, 101
Clear to send, See CTS
Comma in dialing string, 85
Command mode, 79
Communications settings, 72, 76
Communications Toolbox, 42, 243-5
Communications tools, 42, 244, 250
Compact Pro, 23, 109, 138, 140, 144-147, 182
Compressor, See Compact Pro
Compression, 21-23, 73, 143, 146, 269,
272, 279
formats, 143-5
optimizers, 146
protocols in modems, 269
CompuServe, 58, 73, 76, 150, 167, 168, 191,
192, 194, 195
B, B+ and Quick-B protocols, 154, 167
file libraries, 196
Macintosh System Forum, 150
membership kit, 202
Navigator, 274
pricing, 197
vendor support forums, 194
Connection tools, 245
Conrad, Joseph, 23
Control characters, 43-44, 102, 111
Control key, 43, 46, 63, 100
in MicroPhone II, 47
Cooke, William, 10
Copernicus, 131, 222
Courier font, 40, 99
Courier HST modem, 265
Crash recovery, 166
CRC, 162, 170, 268
error, 164
Creation date, 139
Creator code, 53, 136-9, 146
CTS, 26, 273, 337
Cyclic redundancy checking, See CRC
D200, See DG D200
DART, 145-6
Data bits, 74
Data fork, 138, 177
Data rate, 72
Data terminal ready, See DTR
DB-25 connector, 24-5, 340
DB-9 connector, 24-5
serial port, 332
DCA/RFT interchange-format, 186
DDExpand, 145-6
DEC, 16, 37
terminals, 69
Decimal equivalent of any character, 13
Decompression, 22, 145-7
in background, 147
Delay between characters, lines, 65
Delete key, 45-6, 68, 70, 237
Demodulate, 14
DeskWriter, 103
Destructive backspace, 70
DG (Data General) D200, 37
Dialer DA/F-Key, 288
Digital data, 10
Digital Equipment Corp., See DEC
Diplexing, 11
DiskCopy, 145-6
DiskDoubler, 145
DiskTop DA, 138
Docs, 2, 59, 176
DOS, 182-4
file suffixes, 185
files, 139, 182
files, decompressing, 182
files, opening, 184
-based conversion utilities, 185
DOS Mounter, 184, 187
Downloaded files destination, 56
Downloading files, 3, 30, 56, 129, 134, 146-7, 150, 160
Dragon's Cave BBS, 103
Driver, See Modem drivers
dTR, 333
Duplexing, 11, 76

E

E-mail, 126, 192
networked, 126
EASY SABRE, 192
EBB, 326
Echo duplex, 77
Echoes, 4, 114, 117
Echoing text, 65, 76
Echomail, 222
Economic Bulletin Board, 325
Edison, Thomas, 11
Effective speed, 18
Efficiency of the file transfer, 158, 166
Ellipsis, 62
Emulation, See Terminal emulation
Error-checking, 19-21, 160
file transfer protocols, 64, 160
protocols for modems, 268
Escape key, 47-8
remapping, 47
Escape sequence, 95
Ethernet, 297
Even parity, 15
Excel 3.0, See MicroSoft Excel
Express 96 protocol, 266
Extended characters, 61-3, 99, 237
Extractor, 144, 153

F

F-Key, 288
Factory settings, 275, 278
Farallon V.32, 282
FastComm, 31
Fax, 291
cover page, 295
driver, 294-6
modems, 291
programs, 292
resolution, 296
FidoNet, 4, 105, 126, 192, 222, 302

File

captures, See Capture file
compression, See Compression
header, 139
name, 139
suffixes, 183
transfer protocol, 19-20, 32, 160
transfer status window, 157
transfers, 23, 128, 134-5, 139, 149, 163
Finder information, 139
Firmware, 267, 284
Flow control, 25-6, 57, 93, 99, 273
characters, 167
Font, 41
for terminal window, 40
size, 99
monospaced, 40
Ford, Ford Maddox, 23
Forks, 138-9, 140
FreeSoft, 3
Freeware, 34
Full duplex, 58, 65, 76, 99, 132, 234

G

Garbage characters, 102, 177
Gateways, 191-2
Gauss, Carl, 10
GEnie, 58, 73, 76, 160, 163, 167, 168, 191-2, 203, 242
file libraries, 212
Macintosh RoundTable, 207
membership info, 214
pricing, 203
RoundTables, 204
stock quotes, 206
GetInfo, 138
Global Village modems, 342

H

Half duplex, 58, 76-7, 82, 132, 234
Handshaking, 21, 32, 93, 163, 279, see
also Hardware handshaking
Hardware handshaking, 26, 57, 73, 99, 332
Hardware-based
compression, 21-3, 93, 143
error checking, 20, 93, 164
Hayes
BBS, 264
commands, 88
commands for modem speakers, 89
compatibility, 92-3
language, 79, 89, 93, 96
Modem Tool, 245, 247-51
Optima modems, 283, 342
SmartModem 9600, 270
Ultra modem 144, 342
Ultra 96 modem, 342
V Series modems, 283
V-series 9600 modem, 266
V-series SmartModem 9600 MAC, 342
Hermes, 125, 171, 301
High level language, 254
High-speed modems, 265, 270
Host programs, 2, 104, 134, 301
Hot keys, 107, 112
HST protocol, 93, 265, 270
HyperCard, 288
HyperTalk, 254

ImageWriter, 103
Initialization strings, 60, 93-5, 242, 274-80, 284
Integrated Services Digital Network, See ISDN
Intel
14.4EX modem, 343
9600EX modem, 282, 343
International Telegraph Alphabet No. 2, 13
International Telegraph and Telephone Consultative Committee, 20, also see CCITT protocols
Interrupt character, 45
ISDN, 60-61

Kermit, 164-5
Keyboard, 43, 46
buffer, 70-71
remapping, 46
settings, 43, 67
Khmer Rouge, 274

LAN (local area network), 5
LAP-M, 268, 269, 279
LapLink Mac, 185
LaserWriter, 103
Leading, 41
Line prompt, waiting for, 66
Linefeed, 44-5, 67, 102, 178
add and remove, 180-82
Link access procedures for modems, See LAP-M
Local area network, 5
Local echo, 76-7, also see Half duplex
Log off, 131
Log on (to service), 49, 60, 104, 191
Log-on procedures, 101
Long form result codes, 91
Low level language, 254
Lowercase modem commands, 83
LZH, 183

Mac 128K, 24
512K, 24
512KE, 24
Classic, 272
Plus, 24, 272
SE, 272
MacBinary, 140-42, 175, 182, 183
file header, 140
MacBinary II, 142
Macintosh PC Exchange, 184
MacKnowledge, 30
MacLink, MacLink Plus, 177, 187-90
Macro, 225
programs, 62
sets, 228
MacroMaker, 231
MacUser, 217
MacWorld, 217
MacWrite, 146, 176
MacWrite II, 176
MacWrite to Word application, 176
Mass Microsystems Fax Modem, 343
McSink DA, 136, 138, 141, 181
Memory partition, 51-2
Message bases, 109, 113, 115
networked, 115
Microcom, 19, 21-2
MicroPhone II, 30, 35-6, 40, 56, 64-5, 80-81, 99, 135, 156, 243, 276
Communications Settings window, 240
modem drivers, 241, 275-7
scripts, 264
Terminal Settings window, 100
auto-dialing in, 239
MicroSoft
Excel 3.0, 187-8
Word, 146, 176-7, 186-7
Works, 30
Microtech 9600-bps modems, 270
Mini DIN-8 connector, 24-5, 69, 119
MiniWriter DA, 136
MNP, 19, 21, 158, 241, 267, 271
Class 5, See MNP-5 modems, standard, 282
MNP-4, 19-21, 57, 164, 268-71, 279
MNP-5, 20-23, 57, 269-70, 272, 279
MNP-9, 270
Modem
answer phone with, 132
commands, 95
connect responses, 285
default settings, 84
dialing phone with, 83
drivers, 241, 276
error checking, 268-9
generic, 342
high-speed, 265
internal/external, 17-18
negotiating speed, protocols, 72-3, 279
off hook, on hook, 87-8
pinouts, 335
port, 24, 32, 42
protocols, 267
result codes, 91-2, 94
specifications, 341
speed, 19
speed settings, 271
Modification dates, 139
Modulation, 14
protocols, 22, 265-6, 268-9
Monaco font, 40, 99
Morse
code, 11, 13
telegraphy, 11
Samuel F. B., 11
Mountimage, 145
MultiFinder, 32, 51-2, 147, 168, 170, 288
Multimodem V.32, 282
Multiplexing, 11
MultiXfer, 31, 34, 232

N
National Mail Hour, 222
Netmail, 4, 222, 223
Networks, 297
New Jersey Macintosh Users Group, See NJMUG
NJMUG, 113, 192
echo, 117
No extended characters, 63, 99
No-protocol, 135
Node, 120, 222
address, 105, 222
Noise, 15, 159, 163
Non-ADB keyboard, 100
Non-MacBinary file transfers, 142
Non-Macintosh files, 142
Null duplex, 77


Online mode, 79
Online service, 23, 61, 67, 191, 275
Origin line, 119, 223

P
Packet, 157, 160, 162-3, 166
Packlt, 144, 146
PageMaker, 187
Panorama, dialing phone with, 287-8
Parity, 15
   bits, 15, 74-5
   odd, 15
Partition, See Memory partition
PC-oriented BBSs, 175
PCs, sending files to, 141
PEP protocol, 93, 266
Person-to-person connections, 132
Phone Book stack, 288-9
PhoneBook DA, 288-90
PhoneLog, 288
Pinouts, 332
   Macintosh Mini DIN-8 serial connectors, 336
   Macintosh DB-9 serial connectors, 332
modem, 335
Plain text, See Text file
Point, 222
PortShare, 297-9
Positive ACK, 162
PowerBook, 272
Practical Peripherals
   2400SA V.42bis, 343
   9600SA, and 9600/14/4 modems, 283
   PM9600SA, 343
   SmartPack 9600SA, 343
Printer port, 24, 42
Printing online sessions, 102
Prodigy, 6
Product support, 193
Prometheus
   ProModem 24/96 Mini Fax, 344
   ProModem 2400 Mill Plus, 344
   ProModem 2400 Plus, 343
   ProModem 2400SE, 343
   ProModem 2400SE Plus, 344
   ProModem 9600 Plus, 344
   ProModem 9600-EC, 344
   ProModem Home Office, 344
   ProModem Ultima Home Office, 344
   TravelModem, 344
   V.32 and V.32bis modems, 283
Prompt, 105, 112, 257
Protocols, See File transfer protocols,
   Modem protocols, Modulation
   protocols
PSI PowerModem, 344
Pulse dialing, 84

Q
QL Fax Print, 294
Quadruplexing, 11
Quick Link II Fax, 292-5, 297
QuicKeys, 62, 231-3

R
Radiotelegraphy, 11
Receive folder, 56
Red Ryder, 3-4, 35
   Host, 76, 99, 301
Reliable connection/link, 21, 269
ResEdit, 137
Resource fork, 138, 141
Result codes, See Modem result codes
Retransmissions, 160
Return Key, 67
RJ-11 phone cords, 27
ROMs, 60, 267, 284
RS-232 serial connection, 273, 331-3
RTS (request to send), 26, 273, 337
RTS/CTS flow control, 57, 273, 279, 332, 337
Rubout, 46, 68-70, 237
RXD (receive data), 333
S
S-register, 60, 85, 95-6, 279
Scanner, 292
Screen
   buffer, 49-53
   clear, 101
   scrolling, 41
   updating, 41
Scripting, 35, 253
Scroll Buffer, 49, see also Screen buffer
Sealink, 165
Search for text, 50
Second Sight, 76, 98-9, 104-5, 301
   BBSs, 98
Self-extracting archive, 144-6
   DOS, 184
Semicolon in dialing string, 86
Sending
   files person-to-person, 173
   text files, 134-5
Serial port, 23-4, 42, 334
   settings, 72
   speed, 73, 271
Serial Tool, 251
Settings, 39, 58-9, 99, 236-7
   saving, 57
   files, 242
Shareware, 1-2, 30-31, 54, 144, 146, 291
Shiva NetModems, 297, 345
   NetModem V2400, 345
   NetModem v32, 345
   NetModem/E, 345
Smartcom EZ, 30
Smartcom II, 30, 35, 80-82, 134, 156
   Autopilot language, 264
   communications tools, 245
Software
   error checking, 164
handshaking, 26
Software-based
  compression, 22-3, 143
  flow control, 25-6
  protocol, 20, 274
Speaker commands for modems, 89
Special characters, See Extended characters
Standard Files dialog box, 55, 58, 134-6, 155, 187
Start bit, 16
STF fax software, 266
Stop bit, 16, 75
Streaming protocol, 164, 166
String, 256
Stringing modem commands together, 92
Strip hi bit/8th bit, 63, 237
Structured command language, 263
StuffIt, 23, 54, 109, 145, 182
  Classic, 138, 140, 144, 146
  Deluxe, 144, 146
  Expander, 144-5
Suffix of file name, 139-40, 143-4
Supra Modems, 199, 210, 266, 284, 345
  SupraFaxModem, 266
  SupraFaxModem Plus, 345
  SupraFaxModem V.32bis, 278, 345
Synchronous telecommunications, 12, 15, 75
Sysops, 2
System 7.0, 32, 42, 51-2, 147, 168, 170, 288

T
Tabby, 222
TeachText, 102, 177
Tear line, 223
Tektronix
  4014 emulation, 37
  4105 emulation, 37
Telebit, 93
  QBlazer, 283, 345
  T1000, 283, 345
  T1600, 283, 345
  T2000, 283
  T2500, 283, 345
  T3000, 283, 345
  TrailBlazer, 265, 283
  TrailBlazer Plus, 283, 345
Telegraphy, 10-13
  time division system of, 12
Telenet, 166
TelePort modem, 342
TelePort/Fax, 342
Teletype, 10, 237
  emulation, 99
  terminals, 10, 15
Tempo, 231
Term-Plus, 31
Terminal
  application, 29
  personalizing, 58
  program settings, 61
  programs, commercial, 35
  window, 41, 49, 53-4, 71, 99-100, 233
Terminal emulation, 16, 37, 41
  programs, 10, 25, 29-42, 46-55, 63-4, 101-102, 136, 140, 147, 157
Terminator, 31-2, 34, 55
TermWorks, 31
Text file, 20, 136, 140-41, 177
  transfer, 133-4
  "universal" format, 178
Text pacing, 64-5, 67
Tick, 66, 256
Timeouts, 168
Tone/pulse dialing, 84
Transmission errors, 159-60
TTY, 16, 99, 237
TXD (transmit data), 333
Type code for Mac files, 53, 136-9, 146

U
U.S. Robotics, 93, 265, 346
  Courier HST Dual Standard, 346
  Courier modems, 283
  Courier V.32bis, 346
  Sportster 2400, 346
  Sportster 9600 V.32, 346
  WorldPort 9600 V.32, 346
UATerm, 31
UNIX, 237
Upload, 2, 134, 171
Upload/download ratio, 109

V
V.32, 22, 266-270
  modems, 270, 282
V.32bis, 22, 266, 268-9
  modems, 266, 272, 282
V.42, 19-20, 57, 164, 267-70
V.42/V.42bis modems, 269, 282
V.42bis, 22-3, 57, 158, 267, 269-70, 272
Vail, Alfred, 11
Validation survey, See BBS
Variables, 257-8
VersaTerm, 35, 37, 244
VersaTerm-PRO, 35, 37, 42-3, 57, 77, 244
Kermit options, 165
Voice call dialing, 132, 287-8
VT100
keypad, 237
terminal, 16, 37, 237
VT102, 16
VT220, 37
VT52, 16

W
Wheatstone, Charles, 10
White Knight, 3, 35-7, 40, 49, 73, 99, 102, 155, 159, 264
capture files, 56
file transfer options, 142
key mapping, 47-8
macros, 229
procedure files, 264
sample script, 263
screen buffer, 51
serial port settings, 72
status bars, 36-7
saving settings, 58
Word 4.0/5.0, See MicroSoft Word
Wrap text, 101, 182

X
XModem, 33, 154-7, 162, 165, 269
1K, 162-3
XOFF, XON characters, 25, 45, 65, 167, 273, 337
XON/XOFF flow control, 25-6, 57, 65, 67, 74, 99, 135, 167, 273-4, 284, 331, 337
Xpass-MC, 60
XYWrite III, 189

Y
YModem, 20-21, 33, 163
YModem-g, 21, 33, 164

Z
ZIP, 184
Zmac, 217
ZModem, 32-4, 73-4, 154, 164, 166-7, 269, 274
ZModem-90, 166
Zoom modems, 266, 346
AFXMAC, 346
fax modems, 292
FXVMAC, 346
V.32 Turbo, 346
V.32/V.32bis Modems, 284
V.42, 283
VFX V.32bis, 266, 278, 279
VFXV32bisMAC, 346
VFXV32MAC, 346
ZTerm, 31, 34, 63, 93-4, 102, 155, 159-60, 171, 234, 236, 238
macros, 226
sample script, 261
Settings menu, 94
scripting, 255, 256
Telecommunication—the transmission of data over wires—is the most vital part of computer technology today. Whether you're sending chapters of a novel, scanning a photograph, or receiving a fax, these person-to-person tasks can all be done through phone lines and your Macintosh.

This book, by the publishers of the highly-touted teach yourself... series, is designed to introduce all levels of Macintosh users to basic telecommunications, as well as help more advanced users tap into the most effective ways to telecommunicate.

Steven Taylor takes the sometimes overwhelming subject of telecommunications and all its complex terminology and makes it logical and easy to understand. He also focuses on what modems and software are right for your needs, and what online services are available and the benefits they provide.

Stephen Taylor is a computer consultant, writer, and teacher. As a senior systems analyst with the Corporation for Economic and Industrial Research, he developed a number of large computer-based systems, among them a simulation of U.S. domestic air traffic for the FAA. He subsequently worked as an independent consultant to several major corporations and now operates a Macintosh-oriented consulting firm on Long Island.

Mr. Taylor taught courses on computing and film history at New York's Cooper Union. He also taught screenwriting at the School of Visual Arts and conducted in-house corporate seminars on a variety of computer topics.