For example, if you have a mouse on COM1, avoid moving it to COM2 if at all possible. If you move it, you may need to change the configuration of each of your application programs that use a mouse.

To attach a serial device to your computer, you need a serial port. DOS can access them only if they are physically present on your computer. As explained earlier, a serial port comprises electronic circuitry that contains a UART or equivalent chip that performs the necessary data manipulation for serial data to be sent from the port.

Most PCs, made after about 1985, include two serial ports as standard. These may be built into the computer's main system board or may be in an expansion board. Earlier PCs typically had only one serial port. However, you can purchase inexpensive expansion boards that contain a serial port (or two), or you can purchase an internal modem that incorporates a serial port.

All serial ports that you can plug serial devices into are revealed by a connector in the computer's case. This connector will be either a 25-pin D-type connector (so called because it is D-shaped) or a 9-pin D-type connector. The typical PC standard is for this connector to be male (have pins rather than holes), but some early PCs had female connectors. (The serial port on an internal modem cannot be used to plug in a different serial device.)

If your display adapter is an EGA or earlier, it may also have a 9-pin D-type connector. This must be used to attach your monitor and is not a serial port.

To communicate by conforming with the RS-232 standard, you do not actually need to use all the wires in the 25-pin connector. In fact, all the required connections can be made with the 9-pin connector. IBM used the 9-pin connector in the IBM AT for space considerations (a 9-pin connector is much smaller than a 25-pin connector). Consequently, many PC compatibles also use the 9-pin connector.

Note: The serial ports with a 9-pin connector cannot be used for synchronous communication because this requires more than 9 wires for the signals. Unless you are using a leased line (covered earlier in this chapter), you will be using asynchronous communication and can use the 9-pin connectors without a problem.

You typically assign a serial port's name, address, and interrupt number by altering switches or changing jumpers on the circuit board. Some serial ports can be adjusted by using configuration software instead of physically changing switches.

If you are adding an internal modem, the serial port is incorporated into the expansion board, and you need to assign an available port name, address, and interrupt number.

As with the serial ports, this may be achieved by physically altering switches on the board or by running software. The manufacturer's documentation is essential reading.

An external modem can be plugged into an available serial port in your computer. Whether you add an internal or external modem, you need to write down the serial port name, address, and interrupt number. You need this information for your communications software. If any of your hardware or software does not work after installing a modem, the first question asked by technical support will be "Do you know your serial port name, address, and interrupt number?"

As more and more PCs include extra devices, such as sound boards and network adapters, the port address and interrupts become an important issue. There are a limited number of addresses and an even more limited number of interrupts. Because you only want one device to respond at a time, the assignments can become tricky.

For example, most network adapters are configured to use interrupt 3 when supplied. If you then try to add a second serial device, you may experience conflicts because the second serial device may be configured to also use interrupt 3. These conflicts may not occur when you are not actually using the second serial port but *do* occur when you try to add a modem. In this case, you may need to change the interrupt number assigned to your network adapter or to your second serial port.

Several programs — commercial, shareware, and free — are available that will interrogate your computer hardware and tell you what devices are assigned to which ports. Microsoft Diagnostics (MSD), shipped as part of DOS 6 and later, is particularly convenient, but your computer or modem may be supplied with a utility program. Many Gateway 2000 computers, for example, are supplied with QA Plus, and Qualitas' 386MAX memory management program is supplied with ASQ, a system analysis program.

You can use a serial device on COM1 and another on COM3, and similarly a serial device on COM2 and another on COM4, if you usually only use one serial device at a time.

For example, you may have a mouse on COM1 and a plotter or serial printer on COM3. Because you use the printer only when printing and the mouse to move the cursor around the screen, you can probably make both devices work successfully.

Problems arise when you try to use both devices at once. For example, if your program moves the mouse around while printing to a serial device, you can end up sending incorrect information to the mouse and the serial printer, and neither will work correctly. (This does not occur with parallel printers and mice, because the mouse and printer are configured as different devices with different port addresses.)

This conflict is particularly important for modems and mice. As a general rule, if you are using a mouse on COM1, avoid COM3 for your modem; and if you are using a mouse on COM2, avoid COM4 for your modem. Read the documentation that comes with your modem to help avoid problems.

Choosing a Modem

Like PCs, modems are advertised with lots of buzzwords, and different manufacturers emphasize different features to make a particular modem stand out from the crowd.

Also like PCs, your particular needs will be different from other people's requirements, and selection really is a matter of personal preference. You can buy very inexpensive modems that will probably operate under most conditions but may lack the extra features, technical support, warranty, or company reputation, and not be tolerant of slightly noisy phone lines or work with all other modems.

In contrast, you also can buy expensive top-of-the-line modems from well-known companies with every conceivable feature, excellent technical support, and good warranties. However, you may not need all these features and may be wasting your money.

One important difference between modems and PCs is that you need two modems to communicate, but many PCs never have their compatibility with other PCs tested. Any modem you buy, including fax modems, must be able to establish, maintain, and disconnect with another modem, which may or may not be of the same brand or standard.

Choosing a data modem, fax modem, or combined unit

Your first choice is relatively easy. Decide whether you need or will need a data modem, a fax modem, or a modem that combines data and fax capabilities. (Many new model modems automatically include a fax modem.)

If you know you want to transfer files between computers, call such online services as CompuServe or BBSs, or link your computer to another computer through telephone lines, you need a data modem.

If you want a substitute for a stand-alone fax machine and want to send documents (equivalents of pieces of paper), consider a fax modem. As the concept of the electronic desktop becomes more of a reality, where nothing is handed around on pieces of paper, the desirability of PC fax modems increases. As discussed in a later section, choosing a fax modem requires careful consideration.

Choosing the modem's form factor

After deciding whether you need a data modem or combined data modem and fax modem, you need to assess your current computer system, because this may limit your options. If you are going to use the modem with a desktop or tower computer, you have two main choices: internal or external.

An *internal modem* fits into an expansion slot inside your computer and incorporates a serial port. You need an available slot in which to place it, and you need to assign it an unused serial port name. If you already have four serial ports on your computer, you need to remove one to use an internal modem.

If you have a PS/2 computer and want to use an internal modem, you may need to select one that uses the Micro Channel Architecture (MCA) bus connection rather than the more common Industry Standard Architecture (ISA) or PC bus connector. (Not all PS/2 computers have the MCA bus; some low-end computers, such as the Model 25, use the ISA bus.)

If you want an external modem, you have two general choices. The typical box style is about 6 inches wide, 1½ inches high, and 10 or 11 inches deep, although some companies offer futuristic shapes rather than the boring box. Alternatively, you can choose a small modem, often known as a *pocket modem*, that can be used with a desktop or tower computer or (because of its small shape) taken on the road with a laptop.

If you want to use a modem with a laptop or notebook, you can use an external modem and attach it to a serial port. Some laptops come with built-in modems or modems that are available from the computer manufacturer as upgrade options. Alternatively, if your computer has a PCMCIA type 2 slot, you can purchase a PCMCIA modem that plugs into the expansion slot in your notebook. (These may be products offered by the computer manufacturer or a third-party company.) These modems are also small and lightweight for relatively easy travel.

If you are not limited in your choice, consider the following advantages of external modems:

- Most external modems have LEDs (light-emiting diodes) on the front that give an indication of the current modem status.
- You can use the serial port for other purposes, such as attaching a plotter.
- You can move the modem from one computer to another. This is good if you own multiple computers or decide to upgrade your computer.
- External modems have a separate power supply and do not use PC power. (This is only a factor with older PCs that don't have a power supply sufficient to power the modem along with everything else.)

Consider the following advantages of internal modems:

- Internal modems do not need a data cable connecting the serial port to the modem.
- Internal modems come with a built-in serial port.
- You don't need extra room for the modem on the desk.
- You don't need an extra electrical outlet for the modem's power supply.

Apart from the physical shape of your modem, you need to consider its features. Some modems can transfer data faster than others. The majority of buzzwords, in particular the ones that are not acronyms or abbreviations, used by communications people are related to the communications standards. These standards dictate the speed at which data can be transferred from one computer to another. In some standards, more data is sent at once. In others, the data is compressed before transmission and uncompressed by the receiving modem, which results in more data being sent in a certain amount of time.

Selecting compatibility

For PCs, the first consideration is to buy only a Hayes-compatible or AT-compatible modem. (As explained in Chapter 5, the AT is not the same as IBM's AT but refers to the two characters that are used as the start of modem commands.)

The analogy with PCs is worth reiterating here. The term PC-compatible has come to mean PCs that run and support all software and hardware designed for the IBM PC. In some cases, the PC-compatible standard has been expanded and enhanced by other

manufacturers, and these extras may be desirable features. In other cases, increasingly less significant nowadays, PC compatibility may mean supporting almost all software and hardware designed for the IBM PC. This is where name brand — not necessarily IBM, but a major computer manufacturer — plays an important part in the support you will get from the computer manufacturer and software vendors when you find an incompatibility. They are more willing to support a computer that has brand recognition, because there are more likely to be many customers with the same computer.

The term AT-compatible is comparable. Buying a modem that claims to be AT-compatible is essential for you to use PC-communications software easily and consequently make a connection with another modem easily. However, it does not guarantee full compatibility and/or it may buy you extra features that are not available on all other AT-compatible modems.

For most PC users, there is only one type of telecommunication, and you use an AT-compatible modem to link with other modems. This type of communication is called more formally *asynchronous communication.* Asynchronous (as opposed to synchronous) communication is communication that occurs between two computers without regard to the precise clock timing sequences of the connected computers.

Synchronous communication is used between two computers when the events at each end of the phone line must occur in sync with (or in step with) the devices and computers attached to each other. People who need synchronous communication are usually aware of it, because they either have specialized computer knowledge or are supported by people who do. The most common implementation is linking a terminal to a mainframe computer. You can purchase specialized adapter boards for your PC that perform this type of communication.

However, if you need synchronous communications, you can purchase modems that will do both asynchronous and synchronous communication. You can do synchronous communication on dial-up phone lines and don't need a specialized synchronous communication adapter.

Note: The Hayes AutoSync protocol handles the situation this way. The synchronous application talks to the AutoSync driver, and the AutoSync driver sends special formatting signals to the modem so that the modem can convert the data. The modem has firmware in it that converts the asynchronous data into synchronous and puts it out on the phone line. Between the computer and the modem, the connection is asynchronous, but to the software, it appears to be a synchronous channel, and the data going over the phone line is properly formatted synchronous data. It's a clever way of tricking the hardware to do what the software needs to do.

Modem communications standards

Modems can operate at a variety of speeds and with different options that affect the effective data rate and insensitivity to data errors. A modem's data transfer rate is specified in bits per second (bps). However, this is not the only number you need to consider. When choosing a modem, you want to pick the most advanced standard you can afford.

Remember, however, that you must communicate with a second modem. The fastest and the highest standard that you will be able to communicate to second modem is the *highest common denominator*. For example, if you are only going to communicate with a 1200 bps modem, you do not need the fastest modem on the market.

The history of the modem provides insight into the apparently random numbering of communications standards and the apparent duplication of standards for certain data rates. You simply need to understand that from the time of the very first modem in the 1920s, people have wanted to send more data faster.

As a consequence, when a national or international standard for faster transmission had not been defined but public demand existed, manufacturers created proprietary standards to accommodate their customers. However, in most cases, the manufacturers also supported existing standards as fallback positions. If a modem was not connected to a modem that could support the faster standard, a fallback, and probably slower, standard was agreed on by the modems and used instead.

You don't have to worry about whether the modem you are calling has a particular standard. You set your modem to communicate with all its features, and it negotiates with the other modem automatically to find the best common denominator. However, when you buy a modem, you need to know which features you want, because its top feature will be the best at which you can communicate.

In the U.S., early microcomputer modems conformed with the Bell 103 standard. This transmits data at 300 bps. Although now rarely used, it is still supported by most high-speed modems as the final fallback standard. The next standard in the U.S. that is still used is Bell 212A. This transmits data at 1200 bps.

However, in Europe and the rest of the world, the two equivalent standards that were adopted were specified by the United Nations agency Consultative Committee on International Telephony and Telegraphy (CCITT) and were V.21 for 300 bps transmissions and V.22 for 1200 bps. CCITT is now renamed ITU-T (International Telecommunication Union).

Unless you purchase a very old modem, most modems support both Bell 212A, Bell 103, as well as V.21 and V.22. Bell 212A or V.22 are likely to be the lowest standards your modems will ever use.

For speeds above 1200 bps, U.S. manufacturers tend to adopt the international standards as they become available. However, there are some modems that also incorporate proprietary standards that you can use only when calling a modem of the same brand.

The 2400 bps standard is V.22bis (pronounced *bizz* or *biss* depending on who you talk to), and the 4800 bps and 9600 bps standard is V.32. The 14,400 bps standard is V.32bis.

So far, apart from slightly strange numbers, the bigger the standard number, the faster the modem, and the addition of *bis* to the standard number means a faster modem. As an additional help, most manufacturers use the data speed as part of the product's name to indicate its maximum speed.

Two more important standards are commonly available on newer modems: V.42 and V.42bis. Both of these standards may be available on any 1200 bps or faster modem.

A modem that supports V.42 can do error-checking on your data to help ensure that the data received is the same as the data sent. (Error-checking is explained in more detail in Chapter 6.)

A modem that supports V.42bis can compress and uncompress the data you are transmitting. As a consequence, you send more data between the modems in a given amount of time. Although manufacturers clearly tout the V.42 and V.42bis support, they do not usually make the potentially faster data-throughput speed with data compression part of the product name. The amount of data compression you actually get depends on the data that you are sending.

The ITU-T will agree on a further communications standard called V.FAST in 1994 and will probably name it V.34. This was intended to be the fastest possible standard possible on typical analog phone lines. In fact V.34bis is already being discussed. However, eventually further speed advancements beyond a standard will require specialized telephone lines and other techniques as yet not considered.

However, V.FAST has been very slow in being defined and agreed upon, and many people are clamoring for faster modems now. As a result, several U.S. manufacturers, headed by Multi-Tech Systems Inc., AT&T Paradyne, and AT&T Microelectronics, created a standard known as V.32terbo as an interim solution. This standard has a maximum speed of 19,200 bps. (As with memory capacities, 19,200 bps is usually represented as 19.2 kbps and is often spoken of as nineteen dot two.)

Understanding Modulation

Although you don't need to know the intimate details of the communications standards, a conceptual understanding of *modulation* is valuable. It will help you understand commonly confused terms, such as *speed* and *baud*, and understand why the current analog phone line cannot transmit data infinitely faster. These topics are covered in Chapter 5.

A covered in Chapter 3, the modem is a modulator/demodulator. When sending data, the modem modulates digital data into an analog form, and when receiving data, it recovers the data and changes it into a digital form again.

There are many different ways of modulating data, and each communications standard uses one or several different methods. To understand the basics, consider that steady alternating signal, known as a *carrier signal,* is transmitted at a particular frequency with a particular amplitude. The data signals are detected as changes in the carrier signal. The carrier signal is a *sine wave*, which you may remember from high school mathematics. All modulation schemes use amplitude, frequency, or phase modulation, or a combination of them.

Consider first *amplitude modulation*, which is similar to AM radio. The amplitude, or signal level, of the carrier is changed to indicate the data. A carrier signal with a normal signal level may represent a zero, and a carrier signal with a slightly lower signal level may represent a one. The carrier's signal level goes up and down to represent the data bits. Amplitude modulation on its own is not used in data modems.

In *frequency modulation*, which is like FM radio, the carrier signal's frequency is altered to indicate the data bits. Low-speed modems use frequency modulation.

The third type of modulation is called *phase modulation*. This requires two sine waves with the same frequency to be sent. A sine wave is an oscillating wave; at one point in time, its signal level is a minimum; at another, it is at a maximum. The number of times per second the signal is at a maximum (or at a minimum) is the signal's frequency. Phase modulation delays or advances the second sine wave relative to the first sine wave.

Depending on the communications standards, the modems use one or more of the modulation methods to transmit data. The details of what represents a one and what a zero are part of the communications standard. However, in many of the standards, you are sending more than one bit of data at a time.

Using amplitude modulation again as the example, because it is the easiest to visualize, a normal carrier signal amplitude may represent two zeros in a row. A slightly lower signal may represent a zero followed by a one. A slightly lower signal may represent a one followed by a zero, and a slightly lower signal still may represent two ones in a row. Consequently, each change in the carrier signal represents more than one bit of data. Sending two data bits at a time rather than one means that twice as much data can be sent in a given time.

In a separate effort, a different group of manufacturers, including Rockwell International Corp. and Hayes Microcomputer Products, has created another standard called V.Fast Class or V.FC. Rockwell has developed a modem chip that is available to all modem manufacturers as well as computer manufacturers. Computer manufacturers may include this chip on a system board so that the modem is an integral part of the computer. This standard has a maximum speed of 28,800 bps.

Many manufacturers have announced support for this V.FC standard, but it should be considered an interim standard. If you need to communicate at 28,800 bps now, rather than waiting for V.FAST standard to be finalized, V.FC is of interest. If not, it is worth waiting for the new standard modems.

Table 4-2 summarizes the specification numbers in increasing speed order. Remember that any modem 1200 bps or faster may include V.42 and V.42bis features. Both of these standards are desirable features for your modem to support.

Table 4-2 Summary of Modem Specification Numbers

Speed	Standard
300	103 or V.21
1200	212A or V.22
2400	V.22bis
4800	V.32
9600	V.32
14,400	V.32bis
19,200	V.32terbo
28,800	V.FC or V.FAST (V.34)

Note: V.42 and V.42bis are available on modems 1200 bps or faster. V.42 adds error-detection and V.42bis adds data-compression.

Selecting extras

Besides communications standards and physical shape, you should consider the extras that come with your modem when considering price. The warranty, technical support, or any money-back guarantee offered may be important to you.

If you buy a modem that supports a new or less commonly used standard, examine the upgrade path being offered. The biggest complaints I hear on bulletin boards are from modem buyers who thought they would be able to upgrade their modems inexpensively, but when the time came, the upgrade cost a couple of hundred dollars.

The analogy with PCs applies here again. In most cases, PC upgrades are not as smooth or inexpensive as you may think. A modem is not the same as software, and upgrading costs the modem manufacturer more than the shipping costs.

If you are purchasing a data modem and fax modem combination, take special care to read the product's specifications. Fax modems follow different communications standards than data modems. As a result, you may think that you are purchasing a modem with a particular speed when you are actually reading the fax modem's specification. Consider your data modem needs and your fax modem needs separately, and then look at products that can fill both needs. The following section on choosing a fax modem gives details on what to look for in fax modems.

The other extras that come with modems are the cables and communications software. The cables, typically a data cable for external modems and a phone cable for all modems, are fairly inexpensive to buy for yourself, but you do need to remember to purchase them. Check with the supplier regarding the length of the phone cable in case you need a longer one to reach the phone outlet.

For external modems, check whether a data cable is supplied and the type of connector on the end. You may need to buy a different cable to fit your serial port. Remember that serial ports on the PC may have a 9-pin or 25-pin connector. You can purchase a 25-pin to 9-pin adapter cable or complete replacement cables if necessary. These cables are called a variety of names, including serial cables, RS-232 cables, and DTE to DCE cables.

"Do I need a null modem cable?"

You sometimes hear about a null modem cable, and people get confused about whether this is the cable they need to connect their modem. A *null modem cable* is a cable that allows you to link two pieces of equipment together without using modems. It is also known as a crossover cable, a modem eliminator cable, or a DCE-to-DCE cable, or DTE-to-DTE cable. You don't need one for connecting your modem to a PC.

You also do not want to purchase a specialized plotter cable to use with a modem because, like the DCE-to-DCE cable, the connectors on the ends may be right, but it may not have the correct wires going to the correct pins. This specialized plotter cable may be a DCE-to-DCE cable, which will not work correctly connecting a modem to a PC, or it may be a completely custom cable for a specific plotter configuration.

Understanding Flow Control or Handshaking

The concept of flow control is fairly simple. The flow control settings establish "who talks, when". A ham radio operator, for example, uses the term "over" to signal when it is time for the listener to speak. A committee discussion may use rules so that the person speaking stands up; when the speaker sits down, another person can stand up and speak. This controls the flow of information and prevents two people speaking at once.

The RS-232 specification, used by the serial ports in your computer, supports two forms of flow control, known as *hardware handshaking* and *software handshaking*. As the names vaguely suggest, controlling data flow with software handshaking involves sending a signal within the data stream (comparable with saying "over" in the ham radio example). Controlling data flow with hardware handshaking involves using extra signals separate from the data stream (comparable with the committee discussion example).

In a typical communications connection, you have two areas of flow control. The *local flow control* dictates the type of handshaking used between the serial port and the modem. The *end-to-end flow control* dictates the type of handshaking used between the two modems.

The end-to-end flow control is determined automatically by the error-control features in your modem or by the file-transfer protocol from the communications software you are using. (File transfer is covered in Chapter 6.)

You can alter the local flow control method and choose between hardware and software handshaking. Lower-speed modems may only support software handshaking, in which case you have no choice. However, most higher-speed modems, which include error-correction features, support both hardware and software handshaking. When transmitting data at higher speeds, hardware handshaking is preferable.

Hardware handshaking involves two signal lines called clear to send (CTS) and request to send (RTS). When a serial device (modem or computer, for example) is ready to receive data, it raises the CTS line. When the serial device is ready to send data, it raises the RTS line. When both the CTS and RTS lines are raised, the data is sent. When the CTS or RTS line is lowered, the data transmission between the two serial devices is stopped.

Software handshaking involves characters added to the data stream that signal the beginning and end of the transmission. The XON character (pronounced ex-on) starts the transmission, and the XOFF character stops the transmission. The XON character is the character sent when you press Ctrl+S and XOFF character is the character sent when you press Ctrl+Q.

Occasionally, the XON or XOFF character appears in the middle of a file transmission, due to the particular combination of bits, and the receiving computer gets confused. It sends a message, or your communications program issues a message, depending on which program detected the XON or XOFF in an unexpected location, that indicates an XON or XOFF character was found.

For example, if the data stream starts with an XON character, the receiving program is expecting to see an XOFF to indicate the end of the transmission. If a file contains an XON character, the receiving program may send a message. You can manually send an XON or XOFF character as required, by pressing the relevant keystrokes.

Because it is likely that XON or XOFF may appear occasionally in a file, hardware handshaking is preferable to software handshaking. However, apart from the local control, the handshaking type is determined by the communications protocols and not by the user.

Table 4-3 lists the pin assignments needed to connect a 9-pin serial port connector to a modem. (The names of the signals are also included for reference.) A 25-pin serial port connector needs a straight-through cable where pin 1 at one end of the cable is connected to pin 1 at the other end, and pin 2 is connected to pin 2. You actually need only nine wires connected, but some serial cables have all 25 wires connected at each end of the cable.

Table 4-3 Serial Port Pin Assignments

9-pin Serial Port	25-pin modem connector
1 (CD)	8
2 (RD)	3
3 (SD)	2
4 (DTR)	20
5 (SG)	7
6 (DSR)	6
7 (DTR)	4
8 (CTS)	5
9 (RI)	22

The RS-232 standard works well only with short cables. It is not intended for use with cables much longer than a yard. If you use a longer cable, you are more likely to pick up extraneous noise and consequently data errors. You are better off with a longer phone cable and short serial port cable, although longer phone cables are also susceptible to noise. The serial cable should also be shielded. Many users make an inexpensive cable by using *ribbon cable* (a flat strip of wires frequently used to connect different internal parts within PCs). These will work under ideal conditions, but if you experience data loss, your serial cable may be at fault. Consider investing in a shielded serial cable before blaming noisy phone lines. (You can buy shielded ribbon cable.)

Besides cables, you need communications software to operate your modem. Almost all modems come with some sort of communications software, but the available features in this software vary dramatically, depending on the modem. Choosing communications software is covered later in this chapter.

If you want a data modem and fax modem, you need communications software to operate the data and fax communications software to operate the fax modem. These are two separate communications programs, although a future trend will be to make a

single program perform both tasks. As with the data and fax modem combination, you should still consider your needs for transferring data and faxing separately and then compare them with the software's features.

Choosing a Fax Modem

Fax modems are, on the one hand, obvious add-on products for PCs, but on the other hand, they are not totally suited for the job of faxing and receiving documents. The suitability of fax modems depends on what you want to send out, what you want to receive, and what you want to do with the received fax documents.

Note: Fax is short for *facsimile.*

Data modems are well suited to PCs because they transfer data that is stored on a computer to another computer or peripheral device. Fax documents can be documents or drawings that are stored on a computer, but they are just as likely to be handwritten material, including signed documents or drawings, sales brochures or other informational material.

A brief summary of the advantages of fax machines over fax modems and fax modems over fax machines is worth considering.

The advantages of fax machines over fax modems include the following:

- You can use any fax machine from more or less anywhere. You do not have to carry a computer and fax modem around with you.
- You can send information that is not stored on the computer, such as handwritten notes, sketches, and printed brochures.
- The recipient does not need to own a computer. You can arrange for fax documents to be received at local service centers, hotels, or your customer's site.
- In theory, a fax document is easier to send than a file.
- You can use the fax machine (but not a fax modem) as a low-volume copy machine.

The advantages of fax modems over fax machines include the following:

- You can transfer large files, such as text, programs, or graphic files, complete and without degradation. A fax document is always received as a relatively low-resolution *graphical* representation of the original.
- You can typically transfer the file faster than a document.
- You can do far more with a fax modem than a fax machine. For example, you can do online research. You can send electronic mail intended for many possibly unnamed people to read.

Choosing a fax modem first involves deciding whether you need a fax modem or a stand-alone fax machine. There is no slot for you to insert your paper on a PC fax modem. This may seem obvious to many people, but it is asked about more often than you would think.

Considering sending documents

First consider the material you want to send. If you send only information you have previously printed from your PC, you may make good use of a fax modem. However, if you send handwritten information or signed documents, the fax modem is less applicable because you must find some way of getting the written material into your computer.

The signed-document issue is worth emphasizing. It is actually relatively easy to add a graphic image of your signature onto a fax document in your PC. However, you need to be very comfortable with your PC security before you would want to use that type of mechanism due to the legal implications of anyone having access to your signature.

Sending handwritten or preprinted documents with a fax modem and performing the equivalent of stuffing the document in the fax machine's slot are time-consuming and require extra software and hardware. You need a scanner and scanning software. The scanner translates your paper document into an electronic form and stores it as a file on your computer. You can then use the fax modem and its software to send the document as if you had created it on your computer. The scanning process often takes as long as, if not longer than, sending the document.

Considering receiving documents

To receive a fax document on a fax modem, your computer must be turned on and the fax communications software loaded. (There are ways to install the software so that you can work on your computer while waiting for a fax.)

However, you cannot use the fax modem and the data modem at the same time. Nor can you use a telephone attached to the same phone line. They can share the telephone line, but only one device can use the line at a time.

The fax modem behaves like a fax machine. It can answer the phone, establish a communications link with the sending fax machine, and collect the sent document.

Received fax documents are always graphical representations of the original material. Graphic files are large compared to typical text files.

A single-page document that is 4 KB when stored as ASCII text, might be 1 MB — about 250 times larger — when stored as a graphic file. If you expect to receive 10 to 20 pages of fax documents a week and have a need to keep them all, you can rapidly fill your hard disk.

You can print the graphic file and then delete it from your hard disk if you do not actually need to store the received fax on your PC. Many stand-alone fax machines, particularly the less expensive ones, use photosensitive paper that curls and fades with time. Most PC printers use ordinary paper, so the output from the printer may be better than from the stand-alone fax machine.

However, you need a printer that can print graphic files, and you need to consider the time needed to print. A laser printer, for example, can take several minutes to print a page with a graphic image.

If you need to use the received fax on your computer to make modifications, for example, you probably need to convert the received graphics file into text. This can be done by using an optical character recognition (OCR) program. A few fax communication programs include OCR programs, and many other OCR programs are available. Chapter 11 details the advantages and disadvantages of these programs.

An OCR program looks at the graphic file and tries to interpret the shapes into text characters. Depending on the OCR program, this method is very successful for clear typewritten and printed documents and is useless for handwritten, dirty, or complex documents including a mixture of fonts. For example, it is quicker to retype a document with handwritten marked-up notes and pictures than to try to make the OCR program convert it accurately.

Another potential problem with handling documents received by a PC fax modem is your filing system and organization. The documents are saved with a DOS filename, and you need suitable software that can identify the document easily, or you may spend hours trying to find the fax again.

With printed documents, even if you don't have a filing system, you can skim and dismiss a document within a couple of seconds. On the computer, you will be loading and unloading large graphic files unless you have a good filing system that can identify the document without reading it.

Text documents that you are going to alter rather than just read and throw away are much more appropriate for sending with a data modem than a fax modem. However, fax modems are relatively inexpensive ways of providing yourself with faxing capabilities, especially if you purchase a data modem and fax modem combination. The ability to give people a fax number is becoming an essential part of business today.

If you are buying a data modem anyway, consider getting a combined data modem and fax modem to supplement a stand-alone fax machine. You can use the PC fax modem for outgoing faxes that are documents you created on your PC and the stand-alone fax machine to send pre-printed or handwritten material.

Choosing a fax modem's form factor

As with data modems, you need to consider the physical attributes of the PC fax modem as well as the communications standards. The previous section on choosing a data modem's form factor applies equally to your fax modem.

A PC fax modem will probably be an internal device. However, the data modem and fax modem combinations are commonly available as external units. You need to select an internal fax modem that conforms with the expansion bus of your computer. For example, if you have a PS/2 computer, you may need a fax modem with a Micro Channel Architecture connector, or you may want a PCMCIA fax modem for your laptop.

The fax modem is similar to the data modem in that it uses the serial port to connect to your PC. If you already have four serial ports in your computer, you will have to remove one to make a serial port available for an internal fax modem. An external fax modem needs one of the serial port connectors.

Choosing fax modem compatibility and communications standards

Fax machines (stand-alone and PC fax modems) send documents between each other by using established communication standards. These standards predate PCs and are different from all the standards covered earlier in this chapter for data modems.

However, these standards were established by the same United Nations group, ITU-T (previously CCITT), as for data modems. Fax machine communication standards are known as Groups, and there are standards for Group 1, Group 2, Group 3, and Group 4. Almost all fax machines produced today are Group 3 compliant.

Group 1 and Group 2 fax machines are analog devices and do not include modems. Group 3 faxes are digital devices and scan the document, convert the digital data, and then modulate it for transmission. Upon receipt, they demodulate the data back into digital data for printing. Group 4 fax machines can transmit the fax data over digital telephone lines, such as leased lines or ISDN lines. They are not covered in this book.

"What are fax modem communications standards?"

The group number may be abbreviated in the manufacturer's specification. For example, Group 3 may be referred to as G3. Choose a fax machine or fax modem that is Group 3 compliant. Group 3 modems use ITU-T communications standards that specify how the data is modulated for transmission. Like data modems, the different modulation schemes result in different speeds of data transmission.

The V.21 Channel 2 standard is used for 300 bps fax transmission. V.27ter is used for 4800 bps transmission with 2400 bps transmission as a fallback speed. V.29 is used for 9600 bps transmission with 7200 bps fallback speed. V.17 is used for 14,400 bps fax transmission with 12,000 fallback speed. Table 4-4 lists the standards.

Table 4-4 Fax Modem Standards	
Maximum Speed	**Standard**
300	V.21 Channel 2
4800	V.27ter
9600	V.29
14,400	V.17

As with PC data modems, you should buy the fastest fax modem you can afford. When buying a data modem and fax modem combination, be sure to read carefully and determine which is the data modem's speed and which is the fax modem's. You can, for example, buy a 9600 bps data modem that has a fax modem capable of 14.4 kbps transmission. This will not give you a 14.4 kbps data modem.

PC fax modems, rather than fax machines, also conform with other standards. These standards are the equivalent of AT-compatibility and dictate how the PC will communicate with the fax modem. The standards do not affect how the fax data is transmitted so are not relevant to stand-alone fax machines.

The command set used by the PC to communicate with the fax modem is determined by the fax modem's Class. The Telecommunications Industry Association (TIA, also referred to as EIA/TIA) has established the Class standard for fax modems. However, some PC fax boards, including the most popular, use a proprietary standard.

Class 1 fax modems conform with EIA/TIA-578 and are the most commonly found PC fax modems. Class 2 is a fairly recent new standard that is likely to be more popular than Class 1 because of performance improvements.

The selection of a Class 1, Class 2, or a proprietary fax modem depends on the software you are using. The Class standard for the fax modem affects how the PC talks to the fax modem, not how the data is sent over the phone line. The issue for fax modems is a matter of fax communications software support. A Class 1 or Class 2 fax modem will be supported by almost all fax communications software. If you pick a popular proprietary fax modem, such as Hayes JT Fax or Intel SatisFaxtion, any supplied fax communications software will support it. Do not consider proprietary fax modems that are not specifically supported by the fax communications software you are considering.

A fax modem using proprietary techniques for communicating between the data and fax modem may outperform or be more flexible than Class 1 or Class 2 fax modems. For example, you may be able to send a fax document in the background more efficiently while working on another program. Now that Class 2 fax modems have become available, this is a preferable choice to Class 1 fax modems. (The Class 3 standard is also being discussed.)

Fax modem extras

As with the data modem, you need to consider the extras that may or may not be supplied with your fax modem. This includes such physical items as the phone cable as well as the warranty, technical support, and money-back guarantee. The previous section on modem extras applies to fax modems as well, except that you are less likely to have an external fax modem and consequently will not be concerned about data cables.

The fax communications software is typically a weak point with PC fax modems. The programs work satisfactorily, but you may need other programs to provide file management or conversion utilities. The following section on choosing communications software and Chapters 7 and 11 provide more detail.

Choosing Communications Software

Most modems come with communications software, and most fax modems come with fax communications software. You should consider only modems with associated software. This helps with installation and fault-finding, because the supplier will be unable to argue that they have not tried that particular communications program.

The software supplied with data modems and fax modems varies in quality and features. However, most will probably get you up and running even if they do not include all the extra features you may want when you can do the basics.

You have a wide choice of communications programs to choose from. The most appropriate choice depends on what you want to do. For example, if you want to call BBSs, you will want a different program than if you want to call Prodigy, an online service that requires its own communication software. You probably want a general-purpose communications program and should be prepared to get add-on programs or specially written programs as you communicate with different online services.

For example, a general-purpose communications program will allow you to connect with a friend or business colleague as well as get access to most online services. However, you will dramatically reduce your online charges, which in some cases include a charge for the time you are connected to the service in addition to the telephone connection charges, by using a product that streamlines your actions while connected. If you need a special program to connect with a particular online service, you purchase this program as part of the start-up kit when you first subscribe to the service.

The best communications program to use is the one that includes the features you need and you can get the most help with. It's rather like word processors — you get the best help with your word processor if you talk to other people with the same program.

As a starting point, use the communications software supplied with your modem. If you are unable to make it work and cannot get adequate help from the modem manufacturer, choose a well-known third-party program that supports your modem. For example, QModemPro from Mustang Software and Procomm Plus from Datastorm Technologies are two of the most popular programs used on BBSs. If you are using Microsoft Windows, choose a communications program designed to work with Windows, such as Hayes Smartcom for Windows or DCA's Crosstalk for Windows.

As mentioned in Chapter 3, do not give up on telecommunicating because you cannot understand the communications software supplied with your modem. Chapter 5 introduces the basics of making a connection, and Chapter 6 covers how to transfer files between computers. Your communications software should be able to perform both of these tasks.

Part III explains specific features supplied in communications programs and shows how to consider whether you need a better program than that supplied with your modem. In most cases, if you do more than the very occasional connection, you will want to take advantage of software with more features. If you catch the telecommunications bug, you will have very specific requirements.

Fax communications software is slightly different in scope. Most fax modems are supplied with software. Do not consider a fax modem unless it comes with software. Unlike data modems, where you have a lot of control over the communications parameters, you cannot do many different things with a fax modem. You simply send a document or receive a document.

The fax modem communications software must be able to the basics, such as sending and receiving documents, but needing extra features, such as viewing the received document or converting the file into another form, depends on your application. Rather than expecting your fax communications software to do much more than communicate, you may consider third-party utility programs.

For example, fax software can save your graphic file. To work with the received file, you want a graphic viewing or editing program. You may need to convert the graphic file into another file format before you can edit it in your editing program. You may choose a third-party (or is it sixth-party?) program that can do the file conversion. If you want to convert the file to text, you need an OCR program. OCR programs are sophisticated and will probably not be included in your fax modem.

Chapter 7 shows how to send and receive a fax. The supplied fax communications software should be able to perform this task. Part III explains the extra desirable features, such as file management, that you may want for use with your fax modem. When considering the cost of modems and in particular fax modems, assess the associated software carefully.

Installing Your Equipment and Software

As an external unit, a modem is typically a small rectangular box with a series of LEDs (*light-emitting diodes*) along the front and a series of connectors and an on/off switch on the back. Most modems also include a separate power supply, a small black cube with an incorporated electrical plug, that you plug into the electrical outlet and the rear of the modem. (See Figure 4-1.)

There are typically two or three other connectors on an external modem. The larger connector, known as a DB 25 connector, is the data connector and accepts the cable linking your computer and the modem. You need a cable that can join this connector to a serial port on your computer. Serial ports are covered in detail earlier in this chapter.

The other one, or two, connectors are RJ-11 phone jack connectors. These are the same as the connectors typically found on telephones and telephone outlets. They connect your modem to the phone line and possibly to another telephone extension.

You link one connector, probably labeled *Line* or *Telco*, to your phone line via a normal phone cable. This links your modem to the phone line in the same way that a phone is linked to the phone line. When you issue the appropriate instructions in the communications software, the modem can perform the equivalent of taking the phone off the hook and dialing a phone number.

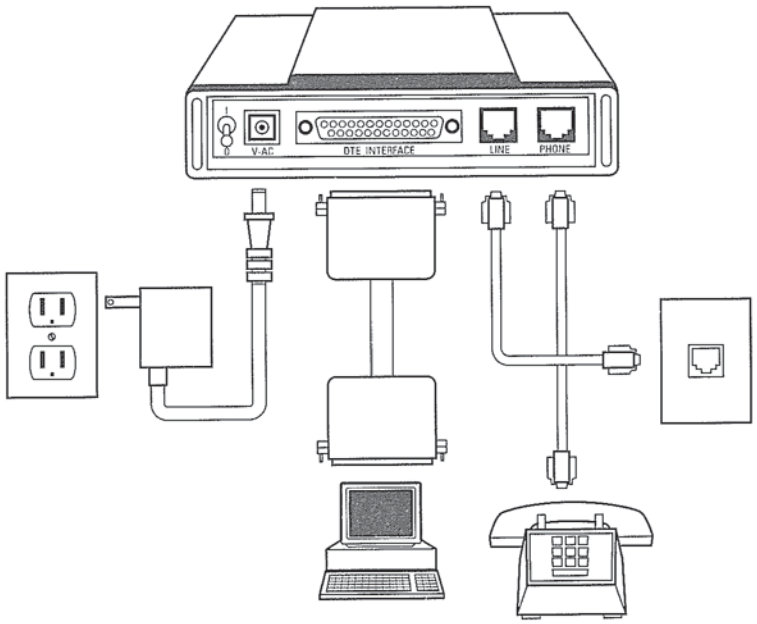


Figure 4-1: The setup of a typical external modem.

The other RJ-11 phone jack, when present, will probably be labeled *Phone* or *Ext.* You can connect a phone via a phone cable to this connector. If there is only one connector on the rear of your modem and it is labeled *Phone*, you should use a phone cable and this connector, the telephone line, and not an extension phone.

Don't be embarrassed if you didn't realize that you need to connect the modem to the phone line. It is very obvious once you know, but an acquaintance of mine spent more than an hour trying to help someone call a BBS with a new modem and software and did not think to ask whether the modem was connected to a phone line.

On some modems, the RJ-11 connectors are not labeled. In this case, they often are linked internally; it doesn't matter which one you use to connect to the phone line and which to an extension phone. However, check your modem documentation to be sure. If the documentation tells you to use a specific one for the phone line and the other for the extension phone, take the time to label the connectors so you avoid confusion later.

If the space around your computer is limited, you should label the cables themselves as you install them. It is surprising how often you have to unplug your external modem to move telephones or computers around.

An internal modem, as its name suggests, is plugged into your PC, laptop, or notebook. It draws its power from the computer and does not include a separate power supply. You can put the modem into any available slot that it will fit into. (See Figure 4-2.)

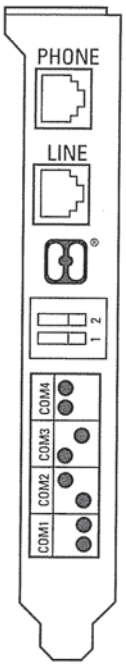


Figure 4-2: A typical internal modem.

Modems have moderate power supply needs, and if you are using a laptop or notebook, you will quickly run down your battery. Wherever possible, plug your laptop or notebook into the main power when using your modem. Because you must be near a telephone anyway to transmit, the addition of external power to the computer is not normally a big problem.

An internal modem has fewer visible connectors than an external modem. The data connector is not required because the computer is linked to the modem via the expansion board or PCMCIA connector.

Like the external modem, the internal modem has one or two, RJ-11 telephone connectors on the rear panel. (The PCMCIA modem also has an RJ-11 connector.) If there is only one connector, you plug a phone cable into the rear panel of the modem and into the telephone outlet in the wall.

If there are two connectors, one probably will be labeled *Line* and the other *Phone* or *Ext.* Use a phone cable to join the connector labeled *Line* to the telephone outlet in the wall, and use a phone cable to join the other connector to a telephone if you prefer.

Tip: Even if you do not want to leave a telephone attached permanently, it is worth adding a telephone while you install your modem and make it work. You can then check that you are actually getting a dial tone and can dial numbers manually from the modem's location.

If there are two connectors and they are not labeled, check the modem documentation for details. As with the external modem, if you must use one for the phone line and one for the extension phone, take the time to make a sticky label and label the connectors so that you avoid confusing the connections.

Take the time when installing your modem to label your phone cables as well as the connectors. This is especially important with internal modems. You will probably need to unplug your modem from the phone line every time you move your computer. Reconnection is much quicker with labeled connectors.

With the hardware installed, you need to install the communications software supplied with your modem. This usually involves running an installation program that copies and uncompresses the files onto your computer. Refer to your modem or communication software documentation for specific details.

After copying the files to your hard disk, you need to configure the software to fit your computer hardware configuration and the modem. This can involve such items as specifying the serial port, its address, and interrupt level, as well as choosing niceties such as screen color.

The actual process depends on the specific modem as well as the specific software. Some communications programs can examine your hardware and determine the modem's location automatically. Others will supply lists for you to choose from, and still others will require to do all the specifying explicitly.

Note: Hayes modems support a feature called AutoStart that enables software to automatically determine the modem capabilities and feature set and configure the software to support them. This feature is supported through the I4 command. With Smartcom

for Windows, for example, the software automatically interrogates the modem to find the speed and feature set and can set up the software to support those feautures.

Make sure that your modem is turned on before starting your communications program. (An internal modem gets its power from the PC and is "turned on" whenever your PC is turned on.) In this way, any initializing or interrogation done by the communications program will occur automatically.

"Why doesn't my communication software start when I type AT and press Enter?"

Although this chapter introduces the concept of modems being Hayes-compatible or AT-compatible and mentioned the term AT commands, this is not the name of your communications software program. AT commands are used within your communications software program to communicate with your modem. In most cases, the software program sends the commands and you do not type them separately.

To start your communications program, you need to type its DOS filename at the DOS command prompt and press Enter. (Although this may seem obvious to experienced DOS users, it can be confusing because an operating mode found within communications programs, known as *terminal mode*, can look deceptively like a DOS command mode.)

The communications program's name will probably be the product name or an abbreviation of the name. For example, you run QModemPro by typing **qmpro** and Smartcom Exec by typing **exec** and pressing Enter.

Your fax software is comparable with the data communications software. Installing involves copying and uncompressing files onto your hard disk or creating working copies. This process is usually automated and may or may not include configuring your fax modem. Be sure to turn your fax modem on before doing the configuration. (As with data modems, if the fax modem is internal, the fax modem is powered on whenever your computer is turned on.)

Your data modem or fax modem requires a communications program to be running as well as the modem being powered on to communicate.

Chapter 5 shows how to communicate with your modem, make your modem communicate with another modem, and create a communications link. Chapter 6 explains file transfer, which you can do after you understand how to make the two modems talk to each other. Chapter 7 shows how to establish a connection and send or receive fax documents at your fax modem.

Summary

This chapter introduced the telephone and RS-232 port essentials. You learned the type of phone line you need and techniques for getting the best service from that line. You also learned the basics on serial ports, their DOS names, port addresses, and interrupt levels.

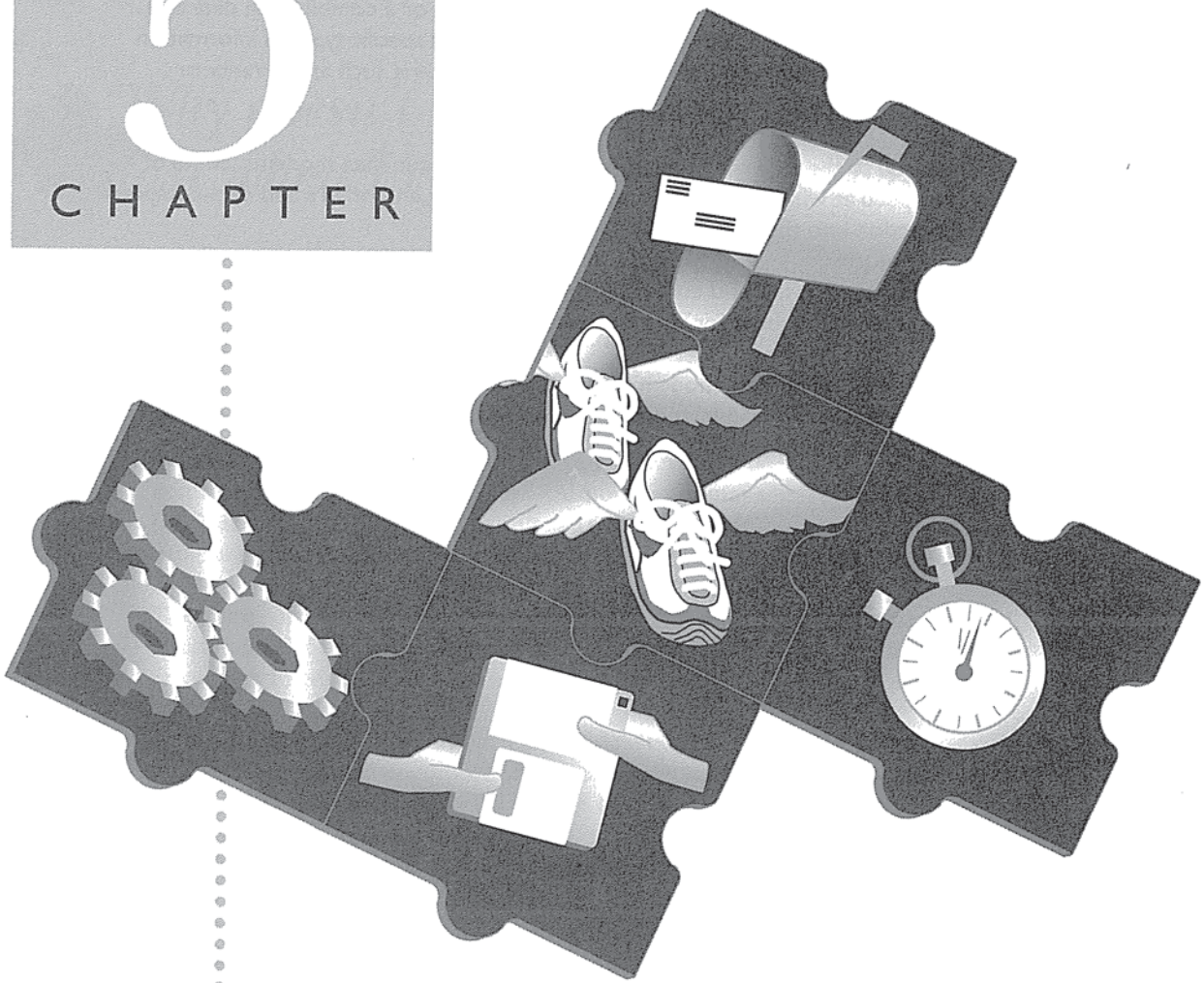
You determined whether you need a data modem, fax modem, or a combination data and fax modem. The selection depends on many factors, including the specific types of information you want to exchange, the computer's form factor, and other criteria such as warranty or technical support.

You were introduced to the ITU communications standards used in data modems. Some of these specify the speed at which your modem can communicate, and others add data-compression or error-detection into the process.

You also were introduced to the ITU communications standards for fax modems. These are not the same as the standards used for data modems.

This chapter covered installing your modem and communications software in preparation for the subsequent chapters. Chapter 5 introduces data modem communication, and Chapter 7 introduces fax modem communication.

CHAPTER 5



Understanding Your Data Modem

This chapter explains how to communicate by using your data modem. You learn the following important steps:

- Preparing to communicate
- Issuing commands to a modem
- Establishing a connection
- Communicating with another computer
- Breaking the connection

This chapter does not explain how to send a fax but focuses on data modems. Chapter 7 is the equivalent chapter for fax modems. Your modem and communications software should be installed before using the techniques in this chapter. Chapter 4 explains modem selection and installation.

Preparing to Communicate

Unless you have used your modem recently and made successful connections, it is well worth checking that you have the following connected:

- A phone line plugged into the rear of the modem and the telephone outlet in the wall
- The data cable for an external modem plugged into the rear of the modem and into the serial port on your computer
- The power supply for an external modem plugged into the rear of the modem and into an electrical outlet

If you find a missing cable or loose connection, be sure to turn off the computer and modem before correcting the problem. When you plug in a connector, you actually make and break the connection multiple times. This can damage the electrical equipment it is attached to. For example, plugging in the power supply to the modem when it is already plugged into the electrical outlet can break your modem. It rarely happens, but there is a chance.

After checking the cables, turn on your modem and start your communications program. An external modem may have an LED to indicate that the power is on and will probably light at least one of the LEDs on its front panel.

An internal modem lacks these LEDs, but you may use a terminate-and-stay-resident (TSR) program on your computer to display the equivalent of the LEDs on your computer screen. A TSR is a program you load into your computer, typically from AUTOEXEC.BAT. It remains in memory and you can run your normal programs with the TSR still resident.

Your internal modem may be supplied with a program of this type or you can obtain one from a computer user group or BBS. These programs remove the disadvantage of not having visible LEDs on an internal modem but may not allow all your other application programs to run successfully. This program is worth considering when you are experiencing problems rather than being a requirement for all situations.

Table 5-1 lists the most common LEDs found on external modem front panels and gives a brief description of their purpose. You will not need to look at these LEDs other than to check that power is on, but they are a valuable tool in troubleshooting. For example, if you do not hear a dial tone when you ask the communications software to dial a number, you can look at the OH LED and see whether the modem performed the equivalent of lifting the phone receiver.

Table 5-1 Modem LEDs

LED	Name	Description when lit
HS	High Speed	Modem is operating at what it considers to be a "high speed."
AA	Auto Answer	Modem answers when phone rings.
CD	Carrier Detect	Modem detects a carrier signal from other modem.
OH	Off Hook	Modem has done equivalent of lifting phone receiver.
RD or RX	Receive Data	Data is being sent from modem to computer.
SD or SX	Send Data	Data is being sent from computer to modem.
TR	Terminal Ready	Computer and modem are linked via serial port.
MR	Modem Ready	Modem is turned on.
DC	Data Compression	Modem is able to compress data.
EC	Error Control	Modem is able to detect errors.

You need to know five items in order to connect with another modem:

1. The phone number
2. Whether you are calling the other modem or the other modem is calling you
3. The *character format*
4. If you are doing the calling, the fastest desired transmission speed
5. The *terminal emulation* both computers will use

Each of these items is detailed in the following sections. Most communications programs include all of these settings within their menus. However, you also can control your modem by sending commands directly. Issuing commands to your modem is covered later in this chapter.

Knowing where you are calling and how

To make a communications link, two modems are connected via a telephone line.

If you are calling another modem, you need to know its phone number. You also want to choose whether you use tone or pulse dialing. As covered in Chapter 4, tone dialing is faster, but you need a phone line that can accept it.

When dialing another modem, you are originating the call. You may need to select this option, called *modem originate*, in your communications program. In most programs, this is the default setting, and if you select a phone number to dial from your program's *dialing directory* (list of phone numbers), this setting is assumed.

If another modem is calling your modem, you need to know the phone number that your modem is attached to. Remember that you cannot use the phone for a voice call at the same time as the modem.

When a modem calls your modem, your modem answers the phone. You need to select the *auto-answer* option or the *modem answer* option in your communications software. This instructs the modem to listen for the phone ringing and then pick up the phone call.

Choosing character format

As explained in Chapter 4, your computer sends serial data to the modem, which is modulated and sent to the other modem for translation back into digital data, where it is accepted by the receiving computer's serial port.

Although the data is sent from the serial port one bit at a time, it is arranged in groups known as *characters*. The character format is comprised of three parts: start bits, data bits, and stop bits. Both computers need to be set to send and receive characters with the same character format.

You define the character format for your connection by choosing the number of *data bits* (7 or 8), the number of *stop bits* (1 or 2), and the *parity* (None, Even, Odd, Mark, or Space).

"What do 7E1 and 8N1 mean?"

The sidebar, "Understanding Character Format," explains parity and gives more detail on how the character is actually formatted when you change these parameters.

In general, if you are calling another PC, such as a BBS or your office personal computer, you will want to choose eight data bits, no parity, and one stop bit. If you are calling a mainframe computer — many commercial online services are run on mainframes — you will probably need to choose seven data bits, even parity, and one stop bit. These are commonly abbreviated to 8N1 and 7E1 respectively.

These values can be changed in most communications programs in a variety of places. The menus are typically named *port settings*, *device settings*, or format settings. In many programs, you can choose a setting that will be the default if you do not specify a different one, and you can specify a different setting for a particular telephone number. (Choosing default settings and customizing settings for particular places you call are covered in Chapter 8.)

For example, you may set 8N1 as the default because you will usually be calling bulletin boards. However, your settings for when you call the online service Genie may be 7E1.

Understanding transmission speed

In addition to specifying the format of the character, the serial port must be set to transmit data to the modem at a particular speed. Surprisingly, the maximum possible speed is not always the most desirable.

With a few exceptions, which are noted in the following sidebar, choose a transmission speed that matches your modem's maximum speed or choose the *automatic* or *maximum* speed option. For example, if you have a 2400 bps modem, choose 2400. If you have a 9600 bps modem, choose 9600.

Many users, experts included, use the term *baud*, or *baud rate*, instead of or as well as *speed* or *bits per second* (bps). You can provoke many arguments trying to understand the correct definition. The technical sidebar "Understanding Baud and Data Rates" addresses these debates.

Understanding Character Format

Your computer sends data to the serial port one byte at a time. The UART and other circuitry rearrange this data into serial data form for transmission from the serial port. The receiving serial port accepts the data one bit at time.

So that the receiving port understands where one byte begins and ends, the bits of data are arranged into characters. The data bits have extra bits known as *framing bits* added on either end of the byte of data. You can control the format of the character by altering the number of data bits, the number of stop bits, and the parity.

The sending and receiving serial ports must be set to use the same character formats to translate the data correctly. Some computers, typically mainframes, can use only seven data bits in a byte, and consequently, the PC, which can use eight data bits in a byte, must alter its serial port settings to conform with the more limiting standard.

Similarly, some serial ports expect one bit to indicate the end of a character being sent on the serial port, and others expect to see two bits to indicate the end of a character.

Using human communication as a comparison, think of needing the same character format at both ends as the difference between local accents. I was in a car accident once in Texas, and the other driver came from New York. I could not understand the slow-talking Texas policeman who left gaps in the middle of words as well as between them. I spoke very precisely with my British accent, pronouncing every syllable, and the policeman had problems understanding me. The New Yorker, who talked extremely fast with no gaps between sentences, let alone words, ended up repeating what the policeman and I said so we could all communicate with each other. You would not have believed that we were all talking English.

The parity bit is an error-checking mechanism that can be added to the data. The parity bit is added so that the receiving serial port has a check that the character sent is the same as the character received. The receiving serial port compares the *parity bit* in the character with the *parity* of the received character. If the two are the same, there is a reasonable assurance that the character actually received is the same as what was sent.

To understand a character's parity, consider a character containing eight data bits, which is a row of eight ones and zeros. The character is considered to have even parity if there is an even number of ones in the row and is considered to have odd parity if there is an odd number of ones in the row.

If you set even parity on your serial port, the serial port will make the parity bit a one or a zero to make the character along with its parity bit have an even number of ones in the row of data. For example, if the eight data bits were 10011001, the parity bit would be a zero to maintain an even number of ones.

You may have odd, even, mark, space, or none parity. Odd parity is similar to even parity, except that the parity bit is made a one to make an odd number of ones in the character. No parity, usually designated as a setting of none, does not add a parity bit.

Mark and space are also no parity, in that they do not represent the parity of the character, but they do add a bit in the parity bit location. Mark adds a one in the parity bit location, and space adds a zero in the parity bit location. They are used most frequently in serial device applications, such as plotting, other than modems.

If you want to avoid the detail, understand that baud (pronounced *bod*) and baud rate are most frequently used incorrectly. As a modem user, talk about your modem's speed in bits per second or bps and you will be correct. I think the confusion has arisen because baud is easier to pronounce than bps.

Your communications software, and other documentation, such as guides to online services, may use the term baud or baud rate to mean transmission speed or data rate. This is where you set the transmission speed. (The modem's actual baud is defined by the communications standard being used and is not directly controlled by the user.)

The speed of the modem you are connecting to is usually not important to the data speed setting. Your modem will try to connect at the fastest speed you specify, and if this is not possible will automatically negotiate a slower speed with the other modem.

The following three sections address the times when you will not want to set the communication software's transmission speed to the maximum value.

Slow modem, fast software

Your communications software may be able to send data at rates that your modem cannot handle. For example, if you have a 2400 bps modem, you don't want to set the port speed to 115200 bps. If you do not pick an appropriate speed, you may experience overrun or dropped characters. (Chapter 4 explains overrun.)

Connect charges

Many online services charge you a connect time charge. You pay an hourly rate for the length of time you are connected to the service. Because you can get more data in a given amount of time if you connect at a faster speed, these services often charge more per hour if you use a faster speed.

For example, you may pay $1 per hour if you connect at 2400 bps or $5 per hour if you connect at 9600 bps. As a beginner, do the basic mathematics to choose the best connection speed. In this example, 2400 bps seems a better choice. You will need to set your communications program to the slower speed, or your modem will try to connect at the faster speed, and you will connect at the higher hourly rate.

As you become more familiar with communications, this is an area well worth revisiting, because understanding the true cost of a connection can save you a lot of money. Even in the preceding example, you may want to use the faster speed for certain operations and the slower speed for others. For your first few connections, the difference of a few cents an hour is probably less important than actually making the

connection and doing the communicating. However, when you are spending ten hours a week online, the few cents add up. Part IV explains how to find out your true online costs and what choices are available.

By the way, if spending ten hours a week online seems a lot to you, consider this. A popular online service changed its rate schedule from a fixed monthly charge to an hourly rate. Many people (home computer users) complained that this would be cost prohibitive for them because they spent in excess of 200 hours a month online to this particular service!

Understanding Baud and Data Rates

When a modem manufacturer called me in 1986 to tout a brand new product that was a 9600 bps proprietary modem with 7.5 baud, I realized I needed to brush up on my engineering degree. I didn't know how to repsond. I knew the terms but bandied them around haphazardly. Baud and data rates are the most abused and confusing terms in modem technology.

First the definitions. *Data rate* is the amount of data that is transmitted in a given amount of time. For example, a data rate of 2400 bits per second means that 2,400 bits of data are sent every second. *Baud* is the signaling rate. For example, 2400 baud means that data is sent 2,400 times per second. (Notice that this definition does not say how much data is sent, only how often data is sent.) Strictly speaking, there is no such thing as *baud rate*. It is like saying "frequency rate" rather than frequency.

Any modem that conforms with the commonly accepted standards, such as V.22bis or V.32, and operates faster than 1200 bps has a baud rate lower than the data rate. To conform with the standards and make best use of the telephone system, more than one bit of data is transferred between the modems at a time.

For example, a 2400 bps modem operates at 1200 baud. Two bits of data are being sent every 1/1200 of a second. The faster modems, such as the 9600 bps modems, have even lower baud rates because even more data is being sent at one time. In my earlier example of 9600 bps at 7.5 baud, 1,280 bits of data were being sent at a time, and they were sent 7.5 times per second.

Many communications programs include a baud setting rather than a bps setting, and many people, including modem manufacturers, refer to their modems as being, for example, 9600 baud modems. In fact, the communications programs are correct and the modem manufacturers are wrong!

Your communications program communicates with your modem through the serial port. The baud setting you choose in the communications program dictates the data speed between the serial port and the modem. Because the data is only sent out of the serial port one bit at a time, the signaling rate is the same as the data rate.

However, the data being sent between the two modems may or may not be sent one bit at a time. The baud is decided by the modems and not the user. In some cases it is chosen based only on the communications standards being used, in others it depends on the standard as well as on the amount of noise on the phone line. When you refer to a modem, you should talk about the data rate and ignore its baud.

Data-compression included

You want to pick a sufficiently fast speed for your communication, which may not be the specified speed of your modem. As explained in Chapter 4, the V.42bis communication standard adds data compression features to your modem. V.42bis is a feature that may be available on all 2400 bps or faster modems.

Other modems may use different data compression methods. These may be proprietary to the particular brand and model of modem or may conform with another fairly common standard called MNP Level 5.

When your modem uses a compression standard, such as V.42bis or MNP Level 5, it compresses the data supplied from the computer and transmits it to the other modem. The receiving modem uncompresses the data and passes it on to the receiving computer. This compression results in more data being passed between the modems in a particular length of time than if the data is sent uncompressed.

In this case, if you set the serial port to the maximum speed of the modem, such as 9600 bps, the modem will not be sending data the whole time because it will compress the data and wait for more data. In effect, the compression will not increase the overall throughput.

However, if you set the serial port speed on both the receiving and the transmitting computer to faster than the modem's speed, the overall throughput can exceed the modem's maximum transmission speed. The sending modem collects a chunk of data and compresses it, then it sends the smaller-sized data to the other end for uncompression or extraction. The data transmission speed (the number of bits per second) remains the same, but the amount of data contained in those bits is more due to the compression.

The amount of compression, which may be none, depends on the type of data being sent. If you have used file compression programs, such as PKZIP, you may be familiar with the concept that certain file types compress more than others. A typical text file, for example, can be compressed by a substantial amount, but a typical program file can be compressed only a small amount.

This compression is possible because of the actual patterns in the data bits and the type of compression techniques used. Text files, for example, include a lot of repetitive characters; words such as "the" or "and" or lots of space characters in a row may occur frequently. A compression technique may take advantage of this to reduce the amount of data sent. Consider the difference in voice communication between ordering "Same again" in a restaurant as opposed to "I'll have a gin and tonic with lots of ice and a lime twist, please."

As a beginner, you don't need to worry about the degree of data compression possible on the material you are sending or receiving. Because the maximum data-throughput with data compression can be eight times your modem speed, you should set your serial port speed higher than your modem speed. (If you do not know about the receiving computer, set the speed higher and be prepared to lower it at a future time if you experience data loss.)

Choosing between compressing data before transmission or using a modem's compression techniques is an intermediate topic that is addressed in Part III. Chapter 9 expands on the possible approaches. Like using the most economical speed if you are being charged based on the time you are on-line, you also want to reduce the time you are on line by using the best compression method.

Understanding terminal emulation

In addition to setting the speed and format of the transmitted information, you need both computers to operate with the same or compatible *terminal emulations.* The parameters discussed previously allow the data to be sent and received at their respective serial ports. The terminal emulation is what the two computers mean by the data that is being sent.

You must remember that you are actually linking two modems together with a communications link. Although each modem may be attached to a computer, the computer types do not have to be the same. Additionally, a second computer does not have to be involved. One modem may be attached to a printer or a device called a terminal.

Terminals consist of a video screen, keyboard, and some control electronics and are typically used to attach to minicomputers or mainframes. They do not contain the processing power of a PC, nor do they usually contain disk drives, but they use the communications link to connect the keyboard to the programs and data on the minicomputer or mainframe. The remote computer in turn controls the terminal. It may send particular data or controlling codes to alter the terminal's operation.

There are a variety of terminal types. Logically, each type is used to connect to each type of computer. Some are very simple, and others are more sophisticated with many more functions available. You can make your PC appear to the other computer as a terminal, hence the name *terminal emulation.* The two computers are able to "talk" to each other because they are using the same terminal emulation.

Using voice communication as a comparison, the terminal emulation is equivalent to establishing the language you are going use, such as French or Spanish. If the two people talking to each other do not use the same language, some information may be heard correctly, but other information may be misinterpreted.

The voice communication analogy can be extended beyond language to gestures and expressions. This is equivalent to the computer sending controlling data and expecting a certain result. In America, looking at the person you are speaking to shows a degree of interest and respect. But in another culture, say Japan, you show respect by keeping your eyes and head lowered. Hand gestures in particular have very different meanings around the world. Even within Europe, a relatively small geographical area, a polite gesture in one country may be lewd in another.

The terminal emulation you choose dictates the effect of your pressing a key on the keyboard and what the receiving computer thinks that key means. For example, when you press an A on your keyboard, the receiving computer considers it as an A. However, if you press F1 on your PC, this may have no meaning to the computer at the other end. Additionally, the remote computer will send data to your computer, and you need to choose the terminal emulation mode that will respond in the expected way to the data or control information sent.

The most basic terminal emulation is known as TTY. This is supported by most computers but has the disadvantage of providing only basic functionality. (TTY was used for teletype machines, an early form of fax machines still in use in many places around the world.) If you do not know what sort of computer you are connecting to, you are unlikely to fail with TTY. Even if the other computer is actually using another terminal emulation, you may still be able to understand most of the characters because TTY is a subset of many more-advanced terminals.

"How do I make BBS menus appear in color?"

If you are connecting with another PC — and most BBSs are run on PCs — you should choose ANSI as the terminal emulation being used by your communications program *and* the BBS. This allows you to see characters in color, bold, or flashing where applicable. Additionally, you will be able to see line characters. These are the characters that make your menus have lines around them instead of funny looking characters.

"When I log on to a BBS, I see funny characters around the menus. Why?"

If there is a mismatch between your terminal emulation selections, your screen will not appear as intended. The BBS probably offers a menu option, and you can adjust the setting in your communications program.

You may have seen similar funny characters by pressing Print Screen and printing a screen with menus on to a laser printer or dot matrix printer.

A recent addition to many BBSs is a graphical user interface called RIP (Remote Image Protocol). This allows you to see graphical menus and to use your mouse online. To take advantage of RIP, you need to use RIP Script or RIP emulation as your terminal emulation. As graphical user interfaces on BBSs become more popular, this will become a preferable alternative to ANSI.

The other terminal emulation types found in many communications programs are very important if you need to connect to other types of computers and are features worth looking out for in your communications software. As a PC user, you will probably be told whether you need to use one of these other emulations. Common terminal emulations include VT 52, VT 100, VT 102, VT 220, and VT 230. These are Digital Equipment Corporation (DEC) terminal emulations. Other popular ones are Wyse 50, IBM 3101, and Heath 19; Prestel and Teletel are widely used in Europe.

When you connect with a system for the first time, you will often be given a choice of terminal emulations. You will probably want to choose the most advanced one your computer can support. If you do not know which to choose, make a note of all of them and try them in turn. For example, choose RIP over ANSI, and choose ANSI over TTY.

The more-advanced terminal emulation may make the system less responsive. For example, the time it takes to redraw the screen with RIP may be much longer than with TTY. If you have a fast modem and fast computer, you may not see any significant difference between the speeds and can enjoy the "pretty screens". However, on slow computers with relatively slow modems, the redrawing can be annoying.

Issuing Commands to a Modem

Almost all communications software includes a menu system that allows you to choose the phone number, character format, transmission speed, terminal emulation, and if necessary an auto-answer feature. You pick the relevant menu and select from a list of options. You are then ready to make a connection.

However, the ease with which you can select these items varies dramatically with the software program. This may be the point where you decide that the communications program provided with your modem is unusable.

As explained in Chapter 3, AT-compatible modems can be controlled from software. (You, or more likely a telephone technician, made adjustments in early modems by opening it up and altering switches inside the cover.) When you make a selection from a menu, the communication software sends commands to the serial port and modem to make the appropriate adjustments.

You can also make these and many other adjustments directly rather than using the communications software. As you become more experienced with communications, you are more likely to need to make the adjustments directly, because your software may not include all the particular modem-controlling commands you need.

For example, you are likely to want to configure your modem in a particular way every time you start your communications software. One method of doing this is to make your software issue a series of commands to the modem, known as an *initialization string*, when you start the communications program. Initialization strings are covered in Chapter 10, because they are an intermediate-level topic. However, controlling the modem directly is an introductory topic.

When you turn on your modem, it powers up in *command mode* and is ready to accept commands that are sent from the computer's serial port. The modem commands (with two exceptions) all begin with the prefix **AT,** and sending commands directly to the modem is commonly referred to as issuing *AT commands*. For example, the command to dial the phone number 555-1111 is ATD555-1111.

"What does AT stand for?"

AT is not an abbreviation but is used by the modem to determine the character format and transmission speed being used by the serial port. The AT allows the modem to self-calibrate. It is expecting to see the two characters AT and can adjust its internal settings so that it will understand the subsequent characters sent as commands from the serial port.

This AT character sequence is not the same as IBM's PC AT computer designation. The AT in an IBM AT computer is an abbreviation for Advanced Technology, because it is a more advanced computer than the IBM PC.

To issue AT commands, you need a method of sending characters from the serial port. For modems, this is done from within your communications program. Your communications program will have a mode where, when you type characters on-screen, they are sent to the serial port. This is most frequently known as *terminal mode* or the *terminal screen* (because you are using the PC as a terminal).

This is the area on your screen where you see data from the computer you are connected to. In a DOS-based communications program, the terminal screen is typically a blank screen with a line of status information at the top or bottom of the screen. In a Windows-based communications program, it is usually a blank window.

"When I type AT at the DOS command prompt, nothing happens. Why?"

Because the terminal screen is mostly blank or has a series of commands with the cursor at the bottom, it can be confused with the DOS prompt screen that may be displayed when you turn on your computer. The two screens may appear similar, but they serve different functions.

The DOS prompt screen, which usually has a drive letter and path, such as `C:\>`, followed by the cursor, is used to issue commands to DOS. The terminal screen is within your communications software program.

To reach a terminal screen, you need to type the name of your communications software program, such as **qmodem** or **exec**, and load your communications software then access the terminal screen. The method for accessing this screen varies, but you may automatically be in this screen when you load the program or may need to press Esc to remove a dialing directory (list of places you can call). Alternatively, your communications program may use the term direct connection to give you access to the terminal screen. Some communications programs do not include a method for you to enter AT commands directly.

When at the terminal screen, the keys you press on your keyboard are sent to the modem. If your modem is in command mode, the keys you press are considered modem commands and the modem tries to respond to them. If your modem is not in

command mode (it is in *online mode*) and is connected to another modem, the keys you press are sent to the modem, passed to the other modem, and in turn passed onto the other computer.

You will need to issue AT commands in two typical situations: configuring and troubleshooting. You may want to change some modem settings, such as how long the modem is to wait after dialing before giving up on making a connection or how many times the phone must ring before your modem should answer it. You may also need to use AT commands to find problems, such as verifying that your modem actually can communicate with the computer or hang up the phone when the remote computer appears to be unresponsive.

There are literally hundreds of AT commands, and no single modem supports them all. Some of these commands are found on all modems, some are found on specific modems, and some are applicable only to fax modems. In most cases, a particular AT command is used in the same way on each modem that supports that command. However, there is no clear-cut definition that says "buy a modem that supports the following AT commands and you won't go wrong." In general, an AT-compatible modem will include support for sufficient AT commands to make communications successful.

Unfortunately, there is no definitive core AT command set. The TIA/EIA 602 standard for data modems, TIA/EIA 578 for Class 1 fax modems, and TIA/EIA 592 standard for Class 2 fax modems are minimum lists of AT commands that a modem or fax modem will include. However, these lists are not enough for most modem use.

This book presents the AT commands in various forms. This chapter includes seven AT commands that all users should memorize. Even if you know about AT commands, you should read the following section because it defines terms like result codes that are used throughout the book.

Appendix A includes a list of the AT commands. Your modem will include many of these commands, but it may include additional ones as well.

Verifying that your modem is there (AT)

The first command all users should know is AT. From the terminal screen, type **AT** and press Enter. As you press each character, it is sent to the modem. However, the modem will not respond until you press Enter to indicate that the command is completed.

Most modems are not case-sensitive, so you can type **AT** or **at**. However, some modems are case-sensitive and only respond **to AT**.

"Can I type at or must AT commands be capital letters?"

Your modem should respond, by displaying on your terminal screen, with `OK` or less commonly `0`. If you do not see the `OK` or `0`, you typed the command incorrectly or your modem is not responding. Check that your modem is turned on and verify that your cables are plugged in. (You may need to exit your communications program and restart it if your modem was not turned on.)

The `OK` and `0` are called *result codes* and are the modem's method of signalling its status as a result of the commands you have sent. Modems support two types of result codes: verbose and short form. As the name suggests, one is brief and the other is more descriptive. See the ATV and ATQ commands later in this section.

If you make an error when typing a command, you cannot use the backspace key to erase it because the modem has already received it. Press Enter and retype the command; remember to press Enter at the end.

You may not see the letters AT on-screen as you type them. You can make them appear by using another AT command (**ATE1**) to make the modem echo the commands back to the screen. However, some communications programs do not operate correctly when you do this, so if you do it for testing, use the **ATE0** command to turn the echo off before resuming normal communication. See the later section on "Communicating" for more information on the local echo command.

Making your modem go off hook (ATH1)

Use **ATH1** to make your modem do the equivalent of picking up the phone receiver. You can use this to make sure that the modem is plugged into a phone line. Type **ATH1** and press Enter. The modem performs the equivalent of taking the phone off the hook. It will return the result code `OK` or `0`, and you should hear a dial tone through your speaker and the OH LED on your modem should light. Type **ATH** or **ATH0** to hang up the phone.

If you type the **AT** correctly, but the modem is unable to understand the rest of the command, your modem will respond with ERROR or 4. You can reissue the command.

If you do not hear a dial tone, your modem may have its speaker volume turned down or off. (Some modems do not include a speaker.) Early-model modems have a knob on the back that you turn to alter the speaker volume. Newer modems use two AT commands to alter the sound from the speaker. **ATM1** turns the speaker on, **ATM0** turns the speaker off. **ATL0** provides the lowest volume and **ATL2** a medium volume.

Making your modem dial (ATD)

The ATD command is used to make the modem dial a phone number. On its own (without the accompanying number), you can use it like ATH1 to make sure the modem is plugged into a phone line. Type **ATD** and press Enter.

To make the modem dial a phone number, you add the phone number to the ATD command. For example, to make it dial directory inquiries, you type **ATD555-1212** and press Enter.

There are many modifiers to this command so that you can make the modem dial in different ways and with different delays. For example, you can add a T to the command to make it dial in tones or add a P to the command to make the modem dial in pulses. You can add a W to make the modem wait until it hears a dial tone before dialing or add a comma (,) to make it pause while dialing.

For example, the command **ATDTW9,555-1212** dials the same number as before but will use tone dialing, will wait to hear a dial tone before dialing the 9, and will pause before dialing the rest of the number. As you may have guessed by now, there are additional AT commands to alter the length of time the modem will pause when you include a comma in the dialing command and additional modifiers that affect the dialing.

You may need any or all of the modifiers depending on how you are configured. For example, many company phone systems require you to dial a 9 to get an outside line, and you may have to pause for a couple of seconds before you get the line.

As a beginner, you do not need to learn all the modifiers, but you should understand that your communications program is issuing these instructions to your modem. It will probably require you to add such items as commas to indicate delays and add the number you would dial to get an outside line.

Getting result codes (ATQ and ATV)

It is much easier to fault find if your modem is supplying result codes. If you are having trouble connecting with a new online service, any one of many things may be going wrong. You may not be dialing the right number, the phone line may not be working, your modem may not be working, the phone number may be busy or ringing with no answer, et cetera, et cetera.

The ATQ command enables and disables the result codes. If you type **ATQ0** and press Enter, result codes are enabled; if you type **ATQ1** and press Enter, result codes are turned off.

The ATV command enables and disables the verbose result codes. If you type **ATV1** and press Enter, you will get verbose result codes that are much more meaningful than the short form. Typical examples include `OK`, `RING`, `BUSY`, `CONNECT 2400`, `CONNECT 9600`, and `COMPRESSION: V.42BIS`.

The short form codes, enabled by typing **ATV** or **ATV0**, return a one- or two-digit number. Most PC users will not use these codes, although obscure communications programs may require their use. These codes are used most frequently when the modem is attached to specialized equipment, such as control circuitry. Rather than the circuitry needing to translate long words, it can translate the number.

Making your modem hang up (ATH)

After the AT command, the most important command you can learn is how to hang up the phone. The AT command to make your modem hang up the phone is ATH or ATH0. Type **ATH** and press Enter or type **ATH0** and press Enter. Your modem should hang up the phone.

This is not as easy as it sounds and is explained in more detail in the section on disconnecting later in this chapter.

Resetting your modem (ATZ)

To reset the modem to a predefined state, use the ATZ command. Type **ATZ** and press Enter.

If no one has reconfigured your modem, this returns you to the factory default settings. However, you should be aware that you can store your own selection of settings in many modems. These settings are known as a *profile*. Profiles are an intermediate topic and are covered in Chapter 9. When you use the ATZ command, the *preferred* profile that you have previously chosen is restored.

Connecting

To make a connection with another modem, you need to set the phone number, character format, transmission speed, and terminal emulation for your connection. This can be done via menus in your communications software or, if necessary, by issuing AT commands directly to your modem.

When each item is chosen, you select the dial command in your communications software to start the connection process. The communications software sends the commands to the modem, which in turn alters its settings where necessary, takes the phone off the hook, and dials the specified number.

The terms *local* and *remote* are frequently used in communications. As with Einstein's theory of relativity, the terms you use depend on your frame of reference. The local computer and the remote computer may vary, depending at which end of the phone connection you are standing. For the following explanation, assume that the local computer is the one you are dialing from (the *originating* modem) and the remote modem and computer (the *answering* modem) are the ones at the other end of the phone line receiving the phone call.

At the other end, the remote, or answering, modem is set to auto-answer and performs the equivalent of picking up the telephone receiver when the phone rings.

(If the phone number is busy, your modem may signal to your communications software that it is busy or it may continue to wait for an answer.) You can make adjustments to many modems (by using AT commands) that alter how long the modem should wait for a connection.

The tones, bleeps, buzzes, and squeals you hear while your modem is making the connection indicate the negotiation process. When the remote modem answers the phone, it sends a tone or series of tones to the local modem that originated the phone call. The particular tone or series of tones specify the particular speed and particular modulation scheme that it is able to connect at.

The local modem either responds or fails to respond to the tones sent from the remote modem. The two modems negotiate the highest possible connection speed that both modems can achieve (or have been set to achieve). Depending on the particular speed and modulation protocol established, the two modems may exchange further information. This information is used, in part, to further tailor the connection by altering the modems' electronics. In this way, the best possible connection, with the least noise and potentially the most error-free transmission characteristics, for the particular phone line connection is made.

On modems that do not support data compression or error detection, when the highest mutually acceptable speed and modulation standard has been agreed to by the modems, the connection is complete, and the modem sends the result code indicating connection at a particular speed to the communications program (unless the result code feature has been disabled).

However, on modems that support data-compression or error-detection, such as V.42, V.42bis, or MNP 5, the connection is not yet complete. The modems do further negotiation to agree on the data compression or error-detection standards. When complete, the modem sends the result code to indicate the connection speed and the compression or error detection to the communications program.

When you are connecting with another modem, look out for this result code. You will not see codes that your modem is not capable of achieving, but you may see standards that are lower than you are expecting.

For example, if you have a 9600 bps modem and connect with a BBS, you may see a result code such as `CONNECT 2400`. This means you are connected at 2400 bps, not 9600 bps. Read the BBS's menus carefully because you may find that if you call an alternate number, you can connect to the BBS at a faster speed, such as 9600 bps. Typically, the commonly advertised phone number for online services is a 2400 bps number.

As another example, you may see a connect result code that says, `CONNECT 2400, COMPRESSION: V.42bis, PROTOCOL: V.42/LAPM` indicating that you have connected at 2400 bps, but the modems are able to use V.42bis compression and V.42 error detection.

After the negotiation, the two modems "keep in touch" with each other by establishing a signal between them, known as a carrier signal. This is the signal that is modulated to transmit the digital data between the modems. Chapter 4 introduces modulation and carrier signals. When a modem stops detecting a carrier signal, the connection is lost. You will have to reestablish the connection by redialing the phone number.

Communicating

After the modems finish negotiating and establish the carrier signal, the two computers, or the computer and the remote serial device, can communicate with each other by using the chosen terminal emulation.

When you press a key on your keyboard, it is received by the remote computer, and if the remote computer sends characters to your computer, you see them on-screen. The computer you call, such as an online service, usually displays a prompt, such as `Last`

Name, or will display a screen of information and a menu. If you call a friend, rather than an online service, you see the modem result codes, and when you press keys on your computer, they will display on your friend's computer screen. When your friend presses keys on the keyboard, you will see them on your screen.

Part IV explains in more detail the type of information you see when you access an online service. For this chapter, you need to understand that the remote computer is running a program that you control when you call it. This program will probably ask for your name and password before providing access to its resources. If you follow the prompts and type carefully, you will quickly reach the menus for the system. If this is the first time you access the system, you may need to register with the system or fill out a questionnaire, but the remote computer sends characters (or graphical images depending on the terminal emulation) that are displayed on your screen.

You respond to the prompts sent from the other computer, and the remote computer follows your instructions. For example, you may ask to read mail or send a message. Depending on what computer you are connected to and what program is running on that computer, you may have to alter a further communications setting. If you do need to alter this setting, make a note of it and set it before the next time you connect with this system.

As explained in the section on issuing AT commands to your modem, when you power up your modem, it is in command mode and your modem tries to interpret any keys you press on the keyboard as commands. When you make the modem dial a phone number, the modem switches from command mode to online mode, and any key you press on the keyboard is passed to the remote modem and onto the remote computer.

"Why do I see repeated characters when I connect to a particular computer?"

In most cases, the remote computer echoes (repeats) the character in the data it sends back to the local computer, so when you press a key, it appears on your screen. However, some computers, depending on how they are configured, do not do this echoing. The remote computer will still accept and process the key press correctly, but you will not see it on the screen. You can correct this by turning on *local echo* and making the local modem echo the characters you press back to your screen.

In the opposite case, where you have local echo turned on and the remote computer also sends back the characters you press, you will see double characters. For example, if you type **hello** you would see HHEELLOO. In this case, turn local echo off.

Local echo is turned on and off either directly from a command in your communications program or by issuing an AT command to your modem. In many communications programs, it is a single key combination to toggle the setting. For example, QModem uses Alt+E and calls it *Duplex toggle;* Smartcom Exec uses the term *Echo typed characters* and you choose yes or no. The AT command to turn echo on is ATE1 and to turn echo off is ATE0.

Unfortunately, this is another area like baud and data speed where terms are misunderstood. Local echo on is sometimes called full duplex, and local echo off is sometimes called half duplex.

Strictly speaking, the terms full duplex and half duplex refer to how the data is transmitted between two serial devices. In half duplex, data is sent in only one direction at a time; in full duplex, data is sent in both directions at once.

As covered in Chapter 6, some file transfer protocols are full duplex and others are half duplex. This is different from local echo being on and off. Local echo on makes the connection between your computer and modem a duplex connection and has nothing to do with the connection between the two modems.

In most cases when you are calling an online service, you want local echo set to off, and the remote system will echo the characters you press. You are most likely to need to turn local echo on when you call a friend's computer and not a BBS. In the calling-a-friend situation, the symptoms of not having local echo on will show as you seeing what your friend types but not what you are typing. Similarly, if your friend can see what you type but not what he is typing, he needs to turn local echo on his computer.

Apart from local echo, communication between the two computers involves you sending characters, which are created by pressing keys on your keyboard, and the remote computer sending characters back to you. Even the graphical interfaces are doing a similar thing — when you pick a menu item this is translated into a series of characters that are accepted by the remote computer. This *character transfer* will allow you to read the information and send information between the two computers.

However, on many occasions, you will send more information than you can type easily at your keyboard, and you will want to send or receive files of information that may or may not consist of characters. *File transfer* is an important part of modem communications and is introduced in detail in Chapter 6.

As a beginner, you should differentiate between *character* and *file transfer* so that you know when you are doing one and when the other. When you press a key, you are doing character transfer, and when you activate a command (run a program) comparable with the DOS copy command, you are doing a file transfer between the two computers.

ASCII, which is an abbreviation for American Standard Code for Information Interchange, is a national and international standard for representing characters that are used on most computers. The internationally approved ASCII character set includes representations for all the alphanumeric keys and punctuation keys and many symbols. There are 128 characters in the set, although not all of the characters are printable. For example, one character represents a carriage return. You can represent 128 characters with seven bits of binary data.

The IBM PC popularized a variation of the standard character set, called the extended character set. This included an additional 128 characters and requires eight bits of binary data for the full set.

"Why do I see gibberish when I connect?"

Sometimes when you are online, you see characters that look peculiar, including for example, smiley faces, squiggles, and other strange shapes. This is a mismatch between your communications program's terminal emulation and the terminal emulation being used on the remote computer.

Your terminal emulation is taking the number of the ASCII character and displaying the character from the extended character set instead of from the basic character set. For example, instead of displaying 2, which is ASCII character number 50, the screen will actually display the extended ASCII character number 178 (128 + 50), which is a shaded rectangle.

Most communications programs have a command that switches the representation so that you see the correct character. This is often called 8-bit toggle or 8-bit on and off, and in QModemPro is toggled by pressing Alt+8. Note that changing this option does not affect characters that are already displayed, but will affect any new characters sent from the remote computer to your screen.

Disconnecting

In theory, disconnecting from the remote modem and computer appears easy. Unfortunately, it is not always simple to make the modem perform the equivalent of hanging up the phone.

When you have made a connection, the modems are transparent to the connection, and you are linked to the remote computer. You expect any data that you send from your computer to the other computer or that you receive from the remote computer to be passed on by the modems and not be changed en route.

However, when you want to hang up the phone, you need to make your modem hang up the phone or the remote computer needs to make the remote modem hang up the phone. You need the modem to stop passing the data it receives onto the other modem and accept an AT command to hang up the phone. When one modem hangs up the phone, the carrier signal is lost and the other modem is aware the connection has been broken.

Unless configured differently by the use of AT commands, Hayes-compatible modems monitor the connection by two methods. They look in the stream of data being sent for a special sequence of characters, referred to as an escape sequence to instruct the modem to switch from on-line mode to command mode. Additionally, they look for the presence of a carrier signal.

When you want to make the modem pay attention to you, rather than pass data onto the other modem, you issue this escape sequence and the modem switches to command mode and awaits AT commands. When you have made any adjustments you need, you can issue a final instruction to make the modem return to online mode if applicable.

"How do I hang up?"

Because the modem uses three different ways to determine whether it can send data, you can use two different ways to break the connection yourself, or you can get the remote computer to break the connection.

If you are connected to an online service, you can use the remote computer program's commands to break the connection. A typical menu will include a command option such as goodbye or logoff. When you enter this command, you will be disconnected

from the service. Wherever possible, particularly if you are paying a connect time charge for the service, use the facilities offered by the remote computer to disconnect. In this way, you can be certain that the remote computer considers you disconnected immediately and does not wait until nothing happens for a certain length of time before considering you disconnected.

If you need to hang up at a time when you do not have a menu option, use your modem to break the connection. For example, if the remote computer locks up or you are unable break out of a file transfer, you can try to make your modem hang up.

The easiest method, and the one you should always use first if you are breaking the connection, is the hang up command. The hang up command in your communications program issues the escape sequence followed by the hang up command. Your modem detects the escape sequence, switches to command modem, then accepts and acts on the hang up command.

Alternatively, you can issue the hang up command directly. You first need to switch the modem to command mode using the escape sequence and then issue the ATH command as described earlier.

The term *escape sequence* does not mean the Esc key on your keyboard. It is a series of keystrokes that you designate as "the keys I want the modem to pay attention to." By default, it is assigned to the plus key, but, surprise, surprise, you can use AT commands to alter the assignment.

To explain the escape sequence, assume the default setting of the plus key. For a true AT-compatible modem, the escape sequence is a period of no data of at least one second (where nothing is typed on the keyboard or is being sent to the modem) +++ followed by another period of no data of at least one second. These no-data periods are known as Guard Time. Hayes Microcomputer Products has a patent on this escape sequence, and other modem manufacturers have been issued a license by Hayes to use it in their modems. This escape sequence is formally called Hayes Improved Escape Sequence with Guard Time.

When the modem sees this escape sequence, it switches to command mode and accepts AT commands. (The escape sequence is one of the two commands that does not use the prefix AT.)

Some modems do not require the Guard Time and will respond to an escape sequence of +++AT followed by a carriage return in most cases without needing a no-data period before or after it. This escape sequence is formally called Time Independent Escape Sequence (TIES).

Understanding an Escape Sequence

Dale Heatherington, inventor of the Hayes Smartmodem along with Dennis Hayes, very carefully chose the escape sequence needed to switch the modem from online mode to command mode. The idea was to create a sequence of characters that would not appear in any data being sent from one modem to the other.

Suppose that the escape sequence had been "cat." The internal electronics in the modem would have been programmed so that when it saw the sequence of characters "c a t," it would switch from online mode to command mode. This would mean if any file transferred with the word "cat" in it — or whenever you typed the word cat when connected — the modem would stop passing data to the other modem and wait for AT commands. The likelihood of the word "cat" occurring in files is high, especially if you consider it as part of other words, such as catalog, advocate, or scathing, and the escape sequence would interrupt far too many transmissions.

On the other hand, you cannot pick an obscure sequence of bits because the user must be able to type it at the keyboard. Remember that the modems were designed to work with any computer, so unusual keys, such as the function keys on a PC or special keys found on terminals, were not suitable.

The addition of Guard Time before and after the escape sequence reduces the chance that a file will switch the modem to command mode. Using the "cat" example again, the need for you to leave a one second no-data period before and after typing the word "cat" probably makes it unlikely that the word catalog or advocate will make the modem switch to command mode. File transfer, which is a continual stream of data, is very unlikely to trigger the modem.

The default escape sequence chosen was a one second no-data period +++ followed by a one second period of no data. On most modems, you can reprogram this sequence to be another character, such as !!!, if you prefer.

The sidebar uses a specific example to explain the escape sequence. However, for a beginner, you need to understand that the modem is designed to respond and switch to command mode when it sees the escape sequence. If you transfer files or data that contain the escape sequence, the modem will assume that the sequence of data bits was intended for it and will interrupt the data transmission by switching to command mode.

The issue is how often the escape sequence actually occurs during data transmission. If you never send a file that has the particular sequence of bits that make up the escape sequence, you will detect no difference between the two types of modems (the one that uses Guard Time and the one that is time-independent).

However, you will not be able to send a file or sequence of characters that make up the escape sequence without the modem switching to command mode and disrupting the transmission. By employing Guard Time, where nothing is transmitted for a second before and after the escape sequence, it is far less likely, as in almost improbable, for the escape sequence to be detected in a normal data stream.

On the other hand, the Guard Time requirement can make it harder to attract the modem's attention. If, for example, you want to sever a connection while a file is being transmitted or streams of characters are scrolling across your screen, the one-second delay without transmission can seem impossible. In this case, you must resort to one of the other methods for breaking the connection.

However, Guard Time may be preferable to you because the transmission will not be broken unintentionally by the data that is being transmitted. Although it is harder to interrupt and get the modem's attention, it is not due to strange — and probably unrecognizable to the user — data content.

The other method for hanging up the phone is to remove power from the modem. Use this as a last resort. An external modem is relatively easy to turn off at its power switch. Exit your communications software, turn your modem back on, and restart your communications software before making another connection.

With an internal modem, you can only remove power by turning off the computer or by pressing the reset button. (This is not the same as pressing Ctrl+Alt+Del.) Wherever possible, exit your communications program before turning off. If you turn your computer off, rather than pressing the reset button, wait until the hard disk drive and cooling fans have stopped before turning it back on again.

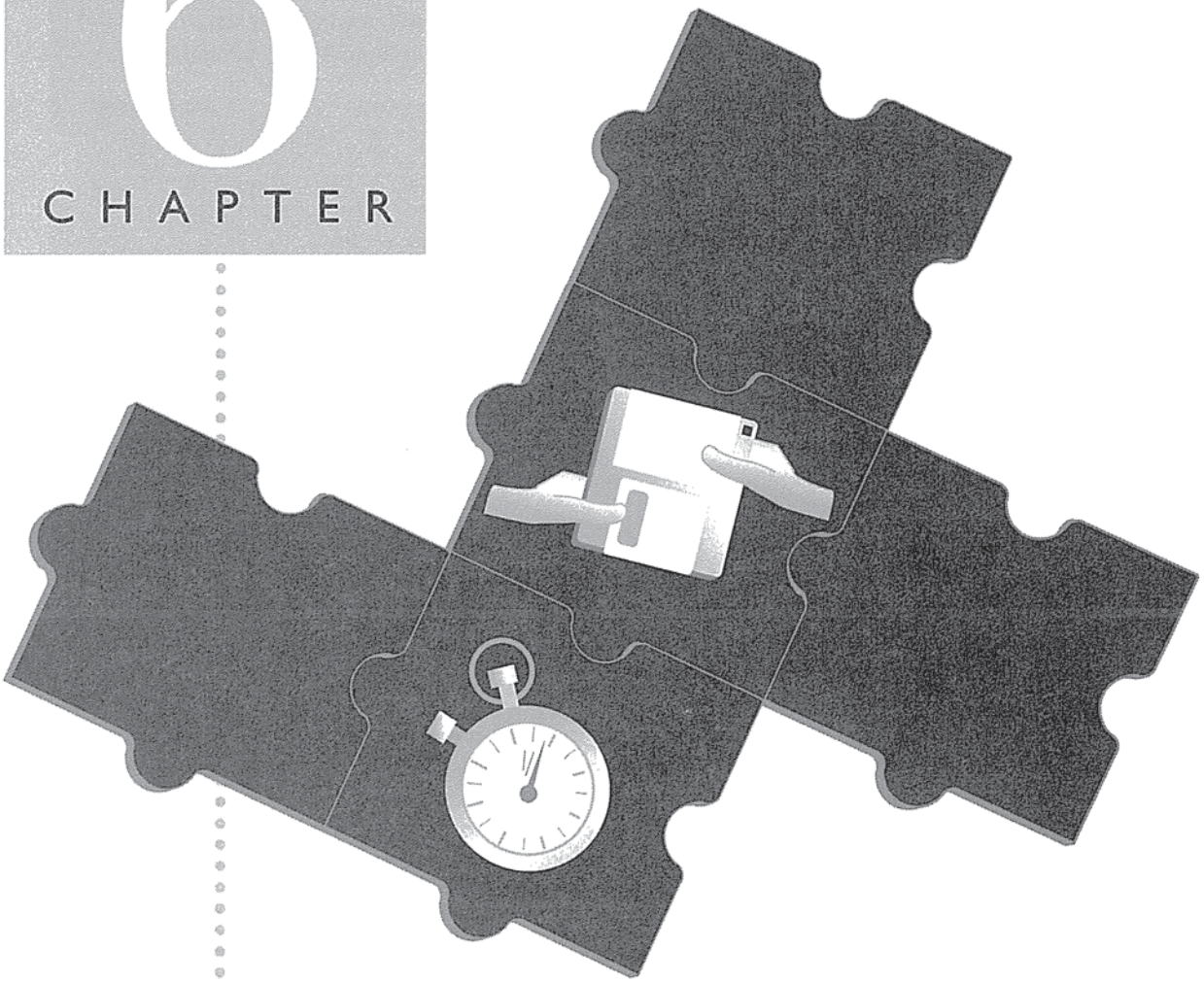
Summary

This chapter introduced the basics of making a connection with your modem. You learned the five items needed before making a connection: phone number, call originator, character format, data speed, and terminal emulation. You also learned that you may need to adjust local echo to be able to see the characters you type.

In most cases, the communications program can issue the commands to the modem, but as you become more experienced, you will need to be able to issue AT commands to your modem directly. This chapter explained that modems can accept commands when in command mode and pass on the data you send from your computer to the other modem when in online mode. You can issue AT commands from the terminal screen, and you were introduced to seven of the most frequently used AT commands in this chapter.

Before reading the next chapter on file transfer, you should understand the three stages of a connection: dialing, connecting, and disconnecting. You should also understand the difference between character transfer, which uses terminal emulation and transfers one character at a time, and file transfer, which uses the equivalent of a DOS copy command to transfer groups of data.

CHAPTER 6



Understanding File Transfer

After users learn to make their modems call other modems and they have access to all these online services, file transfer becomes the primary application. You can copy files from your computer to another, or you can copy files from the other computer to your computer.

This chapter introduces file-transfer protocols. Specifically, you will learn the following techniques:

- Extending your communication to include file transfer
- Understanding the importance of the appropriate file-transfer protocol
- Defining the types of available file-transfer protocols
- Choosing a file-transfer protocol

Extending Your Communication to Include File Transfer

Chapter 5 introduces how to establish a connection with another modem. When you are linked to another computer via two modems, you can type characters on your keyboard and control a program running on the remote computer.

In many cases, in addition to typing messages, you need to transfer files between the two computers. File transfer is comparable to using the DOS copy command: You want to move a file from one computer to another.

However, because different computers are involved, and you probably won't have the same communication software running on both computers, and you may not even have the same operating system running on both computers, you need to use commands that will operate on both computers.

These commands are actually small programs that you run on both computers. You have a choice of programs that you can run. These programs support a particular standard, and the standards are known as *file-transfer protocols.*

In human communication, the analogy for file-transfer protocols is mailing a letter. In most countries, the general principle is to write the address on an envelope, put a stamp on it, and mail the letter in a mailbox. The postal service handles the routing and delivery.

In reality, the precise rules vary from country to country. In the U.S., for example, mail boxes are blue; in England, post boxes are red. In Germany, you typically place the town below the name and the street below that. In the U.S., the street address comes below the name and the town on the subsequent line. However, you can still send mail from one country to another without worrying about the details related to the receiving countries.

"What are uploading and downloading?"

In modem communication, the terms *uploading* and *downloading* refer to sending and receiving a file respectively. Like the remote and local computers, if you are standing at the other end of the connection, the uploading and downloading terminology is reversed.

The general procedure is to prepare the communication program at one end of the connection to send or receive the desired file and then start the program at the opposite end. For example, if you are calling an online service and want to download a file to your computer, you issue the command on the remote computer to start the download. Then you issue the command to your communication software to accept the downloaded file.

In most cases, when you control the program at both ends of the connection, you start the process at the remote end and then instruct your program to receive or send the file. If you are calling a friend, one of you should prepare to accept the file, and the other can instruct the communication program to send it. The typical procedure is to start the download command (the Page Down key in many communications software programs), and then start the upload command (the Page Up key in many communications software programs).

Remember that you are uploading the file from the sending computer. The receiving computer, which is accepting the file, needs to use the download command to receive the file. In the case of an online service, both ends use the terms downloading or uploading, because the program is only being controlled from one end. In the case of you and a friend, you are both controlling your own ends of the connection.

Although you probably will not use exactly the same file-transfer *program* at both ends of the connection, both computers must agree on the file-transfer *protocol*. There are a variety of protocols in use; some are variations on each other; others have specialized applications.

Although the technical detail may seem cumbersome to understand, you should realize that the faster the protocol you use, the faster the file is transferred. Different file-transfer protocols operate at different speeds and have different levels of error-checking.

On the other hand, it does not matter how fast the file is transferred if it is received inaccurately. Inaccuracies occur during file transfer because of noise on the telephone lines. You do not get 10, 20, or 30 minutes of completely noise-free phone lines. When you are talking, you can easily repeat a sentence or stop talking when the noise occurs; but in telecommunications, you need a mechanism to verify that what was sent is what is received.

Understanding Error-Correction

Apart from ASCII, the file-transfer protocols, such as XMODEM and ZMODEM, are *error-detecting* protocols, although they are often referred to as *error-correction* protocols. As a modem user, you should understand the difference between error-detection and error-correction, because you may need to be the judge of how long you are prepared to let a poor transfer last.

An error-detection protocol can identify when an error has occurred during transmission. It then signals the equivalent of "That can't be right" and resends a portion of the file.

An error-correction protocol, which is not employed by the file-transfer protocols, is actually able to *correct* an error rather than *detect* it. This is much more difficult than simply detecting the error.

Suppose that I send you an equation, and I tell you that it will be three numbers added together, such as 8 + 6 + 3 = 17. If you actually receive 8 - 6 + 3 = 17, you know there is an error and can correct the minus sign to a plus sign. On the other hand, if you actually receive 7 + 6 + 3 = 17, you know there is an error, but you cannot tell which number is incorrect.

The different error-detection methods, such as checksum, 16-bit CRC, and 32-bit CRC, offer different levels of error-detection. You can think of them as providing different levels of detail about the transmitted block. The additional detail gives the receiving computer more of a chance of detecting an error.

File Compression (Zipped Files)

File transfer takes time. Anything you can do to reduce this time saves you in telephone and connect-time charges. Most files stored by online services ready for you to download are stored in compressed form. This makes the file much smaller and consequently reduces the file transfer time.

The most commonly used file compression program is PKZIP, a shareware program from PKWare. This program can compress a file or many files into a single file. The compressed file is typically referred to as being *zipped* and has a file extension of .ZIP. Other file compression programs include ARC, from SEA, and LHA, a public domain program created by Haruyasu Yoshizaki (familiarly known as Yoshi).

However, you cannot run these files in their compressed form. After you download them, you need to uncompress them before use. Consequently, one of the first files you should obtain is the set of zipping and unzipping programs. This file, which has a name that includes its version number, such as PKZ204G.EXE, which means Version 2.04 of PKZIP, is available on most online services as well as from most user groups. It is a special type of zipped file, known as a *self-extracting* file, which can uncompress itself to give you the various utility programs and documentation.

If you find an online service using a different file compression program, you need to obtain the equivalent program. In general, you can find the appropriate file compression program for a particular online service in the files listing for that service.

"What do I do with this zipped file?"

PKZIP, which zips the files, and its partner, PKUNZIP, which unzips the files, are a vital part of your PC utility programs. If you intend to do any telecommunicating, you will use PKZIP on a regular basis and should register your shareware copy.

The PKZIP utility is used to group files together and enables you to download a single file to obtain all the elements you need. Suppose that you download a popular game with the filename MICEMN.ZIP. This single zipped file contains all the files you need to play, configure, and register the game, including the documentation.

You look for the game in a files list on the online service and download it to your computer. You then exit your communications program and issue the command **PKUNZIP MICEMN.ZIP**, at the DOS prompt. PKZIP uncompresses and separates all the files, and you have the game file, MICEMEN.EXE, the documentation, MICEMEN.DOC, and other supporting files.

Many online services accept files that you upload. However, many require you to zip the files before you upload them to save disk space and online time. Zipping files involves compressing them, which is the reverse of unzipping them. This process is easy when compressing only one file but requires a little care when compressing multiple files into a single zipped file. However, the documentation supplied with PKZIP provides good instructions.

Determining when to use PKZIP and when to take advantage of your modem's compression is an intermediate topic covered in Chapter 9.

File-Transfer Protocols

The following sections introduce the popular file-transfer protocols. Although you do not need to remember the technical details, you should understand the basics and

make sensible selections. Additionally, this section can be used as a reference when you have trouble matching a file-transfer protocol available in your communications program with a file-transfer protocol available on an online service.

ASCII

The most basic file-transfer protocol is *ASCII*, which is used to transfer *text* files between computers. As Chapter 5 explains, ASCII is used on most computers throughout the world, and IBM PCs and compatibles have popularized the extended character set, which provides even more characters. The ASCII file-transfer protocol is *not* used to transfer programs. (Hayes communication software, including Smartcom for Windows LE, uses the term *autotype*.)

The ASCII file-transfer protocol sends each character in the file in turn as if you had typed it on the keyboard. It does not include error-checking features nor any compression features.

Understanding File Compression

File compression is extremely important when doing file transfer because it can dramatically reduce the size of your files and consequently reduce the file-transfer time. For example, I regularly capture screen shots for inclusion in magazine articles, and a file-compression program can compress the file to about 4 percent of its original size.

There are many ways to compress files, and a good file-compression program will use the best method for each particular file. Some compression algorithms are quite complex, but an understanding of simple algorithms can help you understand why some files cannot be compressed and others can be compressed substantially.

If a file contains truly random data, it cannot be compressed. However, most files, including program files, contain repetitive patterns. A file-compression program looks for these repetitions and, instead of sending the full pattern each time, sends an abbreviated version.

One approach is to eliminate the repetition in a series of bits. For example, a typical bit-mapped screen (without color) is a series of black dots and a series of white dots. Instead of sending the code for "white, white, white, white, black, black, black," the compression program may say "four whites, three blacks."

Another, much more involved approach is to use substitution and send the decoding instructions for the substitution. For example, if the text of this chapter was considered, the words *communications* and *file-transfer protocol* would occur very frequently. If I said "substitute Fred for communications and Jane for file-transfer protocol," I could save a lot of space.

This substitution approach is complicated because the substitutions are not the same for each file. For example, if the file was a bird-watching guidebook, the expression *file-transfer protocol* would not occur. However, *Fred* could be used as a substitute for *binoculars* or *winter habitat*.

XMODEM

XMODEM is the next most common file-transfer protocol found on PCs after ASCII. XMODEM was invented by Ward Christensen, the designer and programmer of the first BBS (see sidebar in Chapter 2). It is known as a half-duplex, error-correcting protocol. In fact, it is an error-detecting protocol and not an error-correcting protocol. *Half-duplex* means that data flows in only one direction at a time.

The file is divided into blocks that are 128 bytes in length, and a checksum is added to each block. The *checksum*, which is, as its name suggests, a number that verifies the data within a block, serves a similar purpose to a parity bit. See the sidebar "Understanding Error-Correction" for more detail.

The transmitting computer sends the first block of data along with its checksum. The receiving computer calculates the checksum for the block and compares its compared value with the checksum value sent with the block.

If the two checksums are the same, the receiving computer sends a character, called an *ACK* (for acknowledge), to the originating computer. The originating computer then sends the next block of data and its checksum.

If the checksums do not correspond, the receiving computer sends a *NACK* character (for no acknowledge) to the originating computer. The originating computer resends the block of data and its checksum.

"Why is the file bigger after I download it?"

The process of sending a block, verifying the checksums, and sending the next block continues. The last block of data in the file contains a special code called an *EOT* (end of text) code that logically indicates the end of the file. When the receiving computer gets the EOT code, it combines all the blocks received into a single file and completes the file transfer.

XMODEM blocks are all 128 bytes in length. If your file is not an even multiple of 128 bytes, XMODEM adds bits to the last block to make it a complete block. These are not removed by the receiving computer. Consequently, you may see a slightly larger

number of bytes in the file on the receiving computer than on the sending computer. This does not affect the file because the end-of-file indicator within the last block is unchanged, but the padding on the end is stored on the disk.

XMODEM variations

Several variations on the basic XMODEM protocol are used in the PC environment. *XMODEM/CRC* uses a *cyclic redundancy check* (CRC) instead of a simple checksum. The CRC is a two-byte code that provides better error control than the one-byte checksum. Its value is derived from a more complex algorithm, and as a result, any transmission errors are more likely to be detected.

In many cases, XMODEM/CRC has replaced the original XMODEM. Consequently, you may run into poor nomenclature and find an online service referring to XMODEM/CRC as XMODEM. If you choose XMODEM in your communications program and XMODEM on the online service, but have trouble with a file transfer and none of the blocks are accepted, the nomenclature may be the problem. Try setting XMODEM/CRC in your communications program, keep XMODEM on the online service, and try again.

Although XMODEM, with its 128-byte blocks, is reliable, it is not particularly fast. Additionally, the transfer protocol does not provide a method of keeping the time and date stamp for the file. For example, if you download a new version of your favorite game by using XMODEM, the date and time that appears with the filename is the date and time of your downloading and not the file-creation date.

XMODEM-1K is another variation of XMODEM. It is similar, except that the block size is 1,024 bytes (1 KB) instead of 128 bytes. The larger block size gives a faster transfer rate because the receiving computer sends fewer ACK or NACK signals. Terminology is mixed with XMODEM-1K, too. Some online services use XMODEM-1K but call it *YMODEM*. YMODEM is explained in the following section. You may need to choose XMODEM-1K in your communications software to use the YMODEM protocol on the online service.

YMODEM

YMODEM is an extension of XMODEM-1K but includes two important but optional extra features: time and date stamp transfer and batch file transfer. YMODEM uses a 1 KB block size but also transfers such file information as the file date and time stamp.

You can also specify more than one file at a time to be transferred. This is known as a *batch-file-transfer protocol*. YMODEM is used on many online services but is often not implemented with the batch-file-transfer protocol feature and is sometimes not implemented with the time and date stamp transfer.

ZMODEM

ZMODEM has become the most popular file-transfer protocol in recent years. It is faster than the XMODEM variations and is usually implemented with the batch-file-transfer protocol and is more tolerant of errors.

In a ZMODEM transfer, the blocks are sent continuously to the receiving computer. Each block includes a 16-bit or a 32-bit CRC. The 32-bit CRC picks up more errors than the 16-bit CRC, in the same way that the 16-bit CRC detects more errors than a checksum. Consequently, you can be more confident that your data-transmission errors (due typically to noise on the telephone line) will be detected.

The sending computer does not expect to receive an ACK signal until the file is completely sent. If the receiving computer detects an error, however, it sends a NACK signal immediately and specifies which block contained an error. On receipt of the NACK signal, the sending computer aborts and restarts the file transmission, beginning with the incorrectly received block. ZMODEM also supports batch-file transfer; you specify a series of files and start uploading or downloading. ZMODEM automatically handles each file in turn.

ZMODEM is faster than the other file-transfer protocols because even less response is necessary from the receiving computer. If there are no detected errors, the receiving computer acknowledges at the end of a file transfer. However, its recovery from errors is particularly desirable. If you are downloading (or uploading) a file and start to see a lot of errors, you can assume that the telephone line has become noisy. You can hang up the phone, redial, and restart the file transmission. ZMODEM will not retransmit the whole file but will start from where it left off.

Suppose that you are downloading a 1 MB file and have received half of it when lots of errors start appearing. With the other file-transfer protocols, such as XMODEM, it's a toss-up whether it's quicker to let the file transfer continue with the error detection and retransmission or hang up and start all over again. With ZMODEM, when you try to download the file again, only the unreceived part is sent.

Other protocols

Several other file-transfer protocols are used in PC communications, and new ones are developed as new technology is available or programmers become more creative. Some are used for specialized purposes, and others are enhancements that can squeeze that extra ounce of performance for you.

When selecting communications software, look for a program that supports all the file-transfer protocols you need and that has the capability to add additional file protocols through add-on programs. In this way, you can expand your communication program's features without buying a whole new program.

For example, HSLINK is a relatively new file-transfer protocol that is not yet supported in every communications program but may be very popular in the online services you choose to frequent. You can buy an add-on program that allows you to use HSLINK with a communications program that supports external file transfer protocols. DSZ and GSZ, enhanced implementations of ZMODEM, are other examples of external protocols you may need.

The communications programs that support external file-transfer protocols allow you to select them from the upload or download menu as if they were part of the main program. In fact, the communications program passes control to the external file-transfer program and waits until the file-transfer program passes control back again to continue.

The following sections introduce two specialized but important file-transfer protocols: CompuServe B and Kermit. You may never run into these, but if you need them, nothing else will suffice.

CompuServe B

As its name suggests CompuServe B, along with its close relatives CompuServe B+ and Quick B, are proprietary file-transfer protocols used by the popular online service CompuServe. They are optimized for use with the particular software used by CompuServe. If you do not subscribe to CompuServe, you won't need it. On the other hand, if you do subscribe to CompuServe, it's the file-transfer protocol of choice.

Kermit

Kermit is another specialized file-transfer protocol. It is actually very flexible but has not become popular in the PC-to-PC communications world. It's the only commonly available file-transfer protocol for transferring 7-bit files.

As described in earlier chapters, you are not necessarily connected to another PC when you use your modem. The remote computer may be a minicomputer or mainframe. Most mainframes use 7-bit characters and not the more familiar (to PC users) 8-bit characters. You need to use Kermit to transfer files to a system that uses 7-bit files.

Kermit transfers can include blocks with varying sizes. This feature is useful if the receiving computer can accept only particular block sizes, but it is also particularly useful on noisy line conditions. When a line is clear, the blocks can be larger and received accurately, but when the line is noisy, smaller blocks are sent, which reduces the number of retransmissions of data necessary to receive the file accurately.

Kermit also allows you to specify wild-card transfers, (similar to the DOS wild-card feature) and employs some file-compression techniques. However, all this flexibility can make successful Kermit transfers difficult because of the number of options that must match, and consequently, it has not gained extensive popularity in the PC communications world, although it is used extensively in academic circles.

HSLINK

HSLINK is a bidirectional file-transfer protocol that is used with full-duplex modems, and both computers can send data at the same time. For example, you can use HSLINK with 9600 bps modems that support V.32, V.32bis, and V.FC communications standards. (As other communications standards bocome able to connect at full-duplex, they will also be able to use HSLINK.) HSLINK takes advantage of this feature and allows you to upload and download files at the same time.

This bidirectional protocol has tremendous performance advantages because you can transfer your files more rapidly. However, because it only works on high-speed modems that are operating in full-duplex, it is not very popular yet.

GSZ and DSZ

The ZMODEM file-transfer protocol was developed and put into the public domain by Chuck Fosberg. He also created two popular ZMODEM implementations called DSZ and GSZ. These shareware products are examples of add-on products that can supplement your communication software. Other add-ons for different purposes are covered in Chapter 10.

GSZ (an implementation of ZMODEM that uses graphical characters) and DSZ (an earlier implementation) include ZMODEM compression and MobyTurbo (TM) accelerator features. You may see 2-5 percent improvements in the time required to download a file when you use this program. Although this may seem small, every minute you save online is saved connect time and, where applicable, phone call

expenses; these minutes leave you with more time for other online activities when you call a time-limited system that may, for example, allow only 90 minutes connect time each day.

Selecting a File-Transfer Protocol

With the exception of ASCII transfer, which is used for a particular purpose, you can choose any file-transfer protocol that your communications software and the remote computer's software support. However, there is a significant difference in performance between the protocols, and faster is probably better.

When to use ASCII

You are most likely to use the ASCII file protocol for uploading messages you have previously typed. For example, if you are looking for information on a new type of motor car oil you heard about, you may want to leave the same message on several online services. Typing the message before going online will save typing the same thing several times. Alternatively, you may want to ask detailed questions about a problem you are having with configuring your word processor, and writing the message before going online will ensure accurate typing.

Which protocol to choose

If you are transferring a file of data, rather than a text file, you cannot use ASCII and must choose another protocol. As a general rule, choose ZMODEM. Remember that the chosen protocol must be supported by the receiving *and* sending computers. If ZMODEM is not supported by your computer or the receiving computer, choose XMODEM. Although it's not a particularly faster protocol, it is widely supported.

The following are exceptions to this rule:

- When transferring 7-bit files, choose Kermit.
- When transferring from CompuServe, choose CompuServe B or one of its variants.
- If you have a modem that supports V.32 and are connected to a modem that supports V.32 and HSLINK is supported, HSLINK is preferable.
- If you have a modem operating in full-duplex mode at 9600 bps or higher and HSLINK is supported, HSLINK is preferable.

Summary

This chapter introduced file transfer, the other application for communication by modem besides sending and receiving messages. In particular, you learned that most files are stored in compressed form. You need a file compression and uncompression utility, such as PKZIP, to use the compressed files that you obtain.

You also learned that you upload a file from your computer to another computer and download a file from a remote computer to your computer. You must choose the same file-transfer protocol at both ends for the transfer to occur. In general, you should choose ZMODEM or XMODEM, except in specialized situations. You choose ASCII to upload or download text.

Chapter 7 introduces fax modem use. Part III extends your modem knowledge so that you can streamline your modem use and make the best use of your communications software.