

FRANKLIN ACE 1000 User Reference Manual



Prepared by the Documentation Department,
Franklin Computer Corporation

©1982 by Franklin Computer Corp.
7030 Colonial Highway
Pennsauken, NJ 08109
609/488-1700

This manual is copyrighted. All rights are reserved. This document may not, in whole or part, be copied, photocopied, reproduced, translated or reduced to any electronic medium or machine readable form without prior consent, in writing, from Franklin Computer Corporation.

The cartoon character of Ben Franklin appearing throughout this publication constitutes a trademark of Franklin Computer Corporation.

Franklin ACE is a trademark of Franklin Computer Corporation.

The following registered trademarks belonging to the indicated parties are referred to in this manual:

APPLE	Apple Computer Inc.
APPLE II	Apple Computer Inc.
CP/M	Digital Research, Inc.
Space Invaders	Taito America Corp.
PAC-MAN	Bally Midway Mfg. Co.

WARNING: This equipment complies with the requirements for a Class A computing device in FCC Rules Part 15 Subpart J. Operation of this device in a residential area may cause harmful interference requiring the user to take whatever steps may be necessary to correct the interference.

Table Of Contents

Uncertainties, Mysteries, and Doubts	1-1
They Won't Bite — But You Can Bite Yourself	1-3
The ON/OFF Switch Is Never Out Of Reach	1-3
Getting A Feel For Computerdom	1-5
The World Can Live Without Bits and Bytes	1-6
Dealing With The Knowledgeable	1-7
You're Not A Programmer — It's A Dirty Job	1-9
What Can The Computer Do, Anyway?	1-10
What It Can And Can't Do	1-11
What You Have To Remember About Memory:	
Size Counts	1-14
A Good Time To Be Interested In A Personal Computer	1-16
Compatibility — The Good, The Bad, And The Ugly	1-18
Be Choosy About The Company You Keep	1-20
The Dealer	1-21
True Grit: The Adventure Of Mail Order	1-22
Clubs	1-23
There's More To It Than Just The Computer- Additional Equipment	1-24
TV's And Monitors	1-25
Printers — For Paper Copies Of What You Do	1-26
Disk Drives	1-29
Video Cards — For Lines Longer Than 40 Columns	1-31
Communications Devices — For Talking To Another Computer	1-35
Computers In A Computer — If One Is Good, Two Are Better	1-38
Programs	1-40
Paddles And Joysticks	1-43

Before You Try To Set Up A Personal Computer	1-45
It Isn't All That Difficult	1-46

**What You Have To Know: Seven Commandments
(Developing The Uncommon Sense Of Computerdom)** 2-1

I. Electrical Power -- Don't Change A Bulb With The Power On	2-1
II. Diskettes Are Not Dry Cleanable	2-1
III. Reset Switches Are Your Worst Enemy	2-4
IV. An Elephant Never Forgets, But You Didn't Buy An Elephant	2-6
V. You Can't Fix It But You Can Make It Worse	2-10
VI. Promises: Take Each With A Grain Of Salt	2-11
VII. Be Patient -- It's Not A Perfect World	2-13

A good title for this section might be . . .

The Territorial Imperatives Of The Trumpeter Swan	3-1
It's Out Of The Box! Now What?	3-2
Getting It Ready	3-3
Getting The Lid On And Off	3-3
Attaching The Cables	3-4
The Video Cable	3-4
The Disk Drives	3-5
The Power Cord	3-7
Turning It On -- What It Should Do	3-8
Why It Might Not Be Working	3-9

The Communications Connection	3-11
Say Hello To Your Keyboard	3-11
Lower Case And Alpha Lock	3-11
Numeric Pad	3-13
Special Keys	3-13
Keycode Differences	3-14
Just Where Is The Reset Switch?	3-15

The DOS Diskette	4-1
Giving The DOS Diskette A Boot Up	4-1
Now What?	4-3
How To Use The Programs On The DOS Diskette	4-4
How To Use The COPY Program	4-5
How To Use DOS 3.3	4-11
CATALOG (Displaying The Table Of Contents Of A Diskette)	4-13
RUN (Running A BASIC Program)	4-16
BRUN (Running A Binary Program)	4-17
LOCK (Protecting A File)	4-18
UNLOCK (Removing The Protection From A File)	4-19
RENAME (Renaming A File)	4-20
DELETE (Deleting A File)	4-20
INIT (Initializing A Diskette For Use)	4-21
A Few Generalizations About File Names	4-25
How To Use FUD	4-26
Diskette Management Commands	4-28
Selection F (Format A Diskette)	4-28
Selection M (Make A Master Diskette)	4-32
File Management Commands	4-36
Selection S (Show The Files On A Diskette)	4-39
Selection D (Delete Files)	4-41
Selection L (Lock Files)	4-42
Selection U (Unlock Files)	4-42
Selection V (Verify Files)	4-42
Selection C (Copy Files)	4-43
FUD Error Messages	4-44
oops, diskette is write-protected	4-44
oops, wrong volume ID	4-45
oops, file is locked	4-45
oops, no more room on diskette	4-45
oops, can't find that file	4-46
oops, bad diskette or none in drive	4-46

How To Use Boot13 4-46

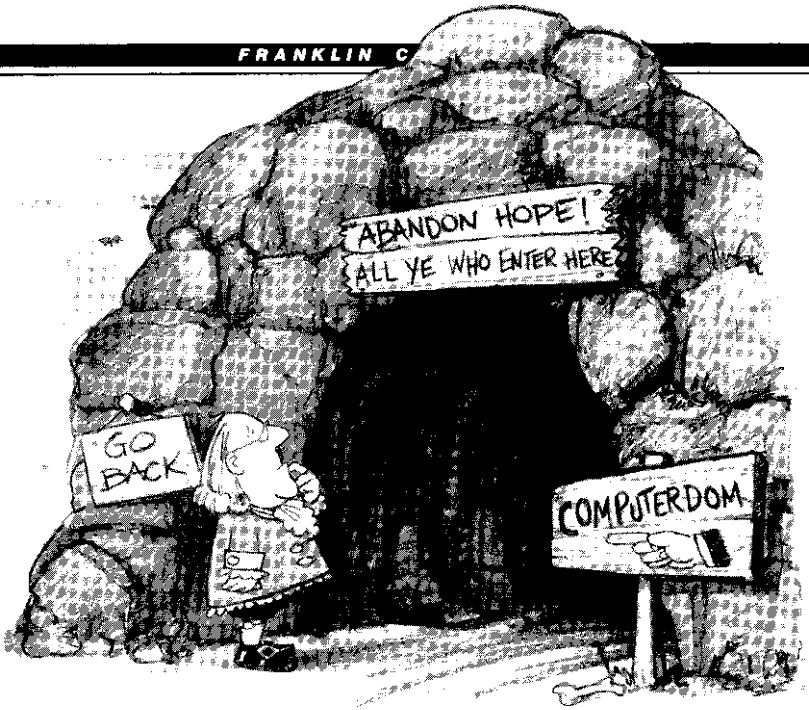
A Synopsis Of The Programs On The DOS Diskette 4-48

APPENDICES 5-1

Glossary 6-1

Index 7-1

DOS And FUD Command Formats 8-1



Uncertainties, Mysteries, and Doubts

A lot of people find computers overwhelming, even threatening. That's sad because, in reality, they're neither. If you're curious and interested but still a little bit wary, that's only natural. Until recently, doing anything with computers required special skills, even special languages, and those who knew what it was all about weren't very interested in initiating everyone else.

Personal computers like the Franklin ACE changed all of that. True, no computer is as easy to use as a toaster or a calculator, but what you need to know to operate a personal

computer is explainable and comprehensible. What's more, it's quick and almost painless to learn. This booklet will give you a start.

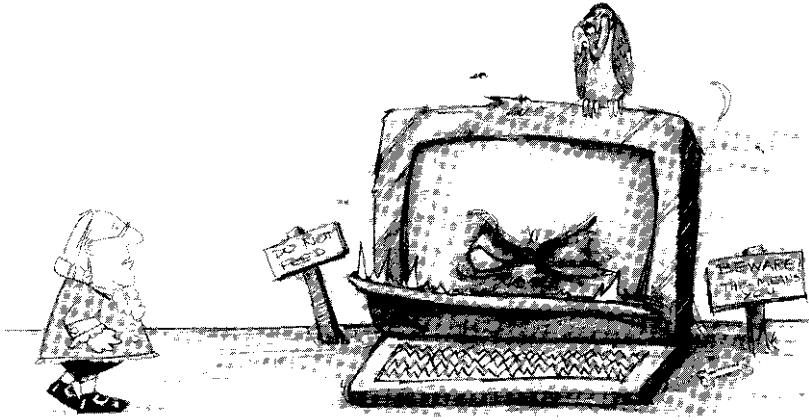
If you want to learn how to program, how computer graphics work, what disk transfer rates are, or how to master the dozens of other exotic sounding skills you hear about in computerdom, there are hundreds of other books on the market to help you learn all those things.

This one won't.

What you will find in this booklet and throughout all Franklin manuals is plenty of practical information, suggestions, and answers to questions that you might otherwise have to learn the hard way. You'll find advice about computer products -- what they're for, how they work, what their quirks are, and so forth. In a field that's growing and changing as fast as computer science, you'll need this sort of information in order to make sound purchasing decisions.

Someone may already have begun to persuade you that a Franklin ACE will make your life complete. It may; it may not. People use computers for all kinds of reasons. Some people buy them simply to keep their kids quiet. There are, however, many other practical applications for a computer, ranging from financial forecasting to Space Invaders™, from word processing to setting up a blind date.

If you're having second thoughts about your curiosity, your interest, or your purchase, dispel those doubts. With a little practice you can put a computer to work and enjoy the process.



They Won't Bite — But You Can Bite Yourself

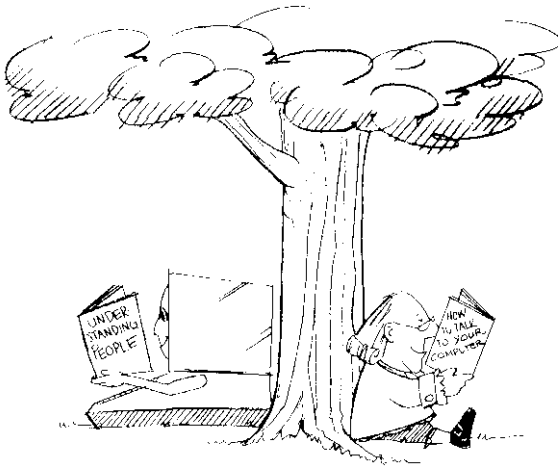
Even though computer types are always talking about bits and bytes (pronounced “bites”), most people aren't afraid that they're going to get bitten. But many worry that the machine will do something they won't understand or that it will suddenly start to spit out all kinds of symbols and nonsense. Computer phobia is the fear of feeling incompetent and helpless in the face of high technology. With personal computers, however, there's nothing to fear.

The On/Off Switch Is Never Out of Reach

One of the joys of using a personal computer is the feeling of complete control that you get by having the ON/OFF switch so close at hand. You're in charge. Nothing is going to happen

unless you make it happen. And if the machine gets you confused, all you have to do is turn it off and start over. Your computer will do whatever you ask, whenever you ask.

True, computers do have a reputation for being fussy about the way you phrase your questions and commands. Some people even think they're evasive, somewhat like a witness in court who manages to avoid providing information by sidestepping the intent or spirit of a question. The fact is, you do have to develop something of a rapport with the way the computer understands what you tell it. That comes with a little practice. If you ever find yourself frustrated, just remember that ON/OFF switch.



As in all of life, awesome power is not without some responsibilities. Since you're in charge, if something goes wrong, then you're probably the one who's to blame. Exercise some discretion when using the machine, and look at instructions and directions as a way of developing a common sense about computers, as well as a way to assemble the components of your machine and get it running.

Getting a Feel For Computerdom

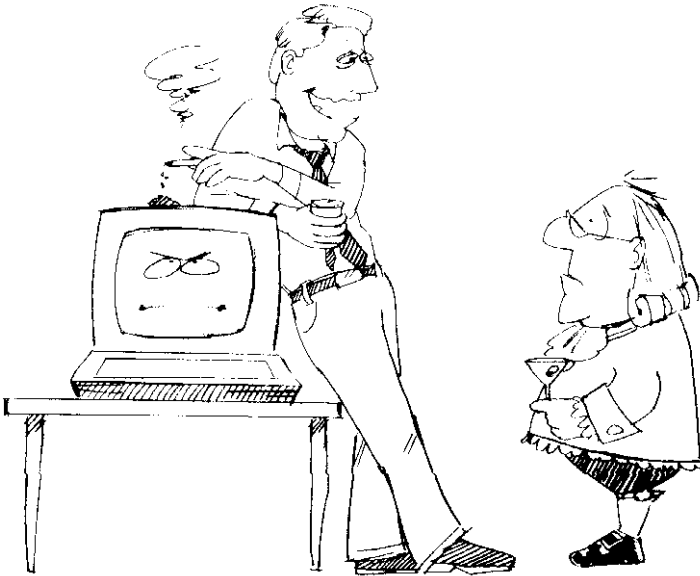
When you're first starting out, it's hard to get a feel for computerdom. Making a computer work for you takes some knowledge and skill, but it also takes a proper attitude or disposition. You can acquire the knowledge and skill readily, since those are both teachable and learnable. The attitude is a little more tricky.

It's a matter of staying loose while at the same time remaining methodical, analytical, and meticulous. That sounds impossible, but it isn't. Don't worry about making mistakes. Using a personal computer is a hands-on, learn-by-doing process that is somewhat less than perfectly straightforward. As long as you're analytical, you'll learn more from your errors than anything else.

Once you have your computer set up, you won't be able to hurt it with anything you may do at the keyboard. You can lose programs or data, but as long as you follow the correct procedures, there's little chance of your doing irreparable harm to the machine.

Making mistakes at the keyboard and mishandling the equipment are not the same thing, so be warned. You can do damage if you're not careful. Probably the most sensible

approach is to treat a computer, any attached equipment, and any diskettes as if they were an expensive stereo system and records or tapes. What you wouldn't do to your stereo and your record albums, you shouldn't do to an ACE or diskettes.



The World Can Live Without Bits and Bytes

You won't find out what bits and bytes are here. If you really want to know, look in a dictionary of computer terms. That may sound a little brusque, but the point is that you don't have to know these or hundreds of other specialized terms to use an ACE or any other personal computer effectively.

“Bits” and “bytes” are examples of the terminology that computer experts use to communicate with one another. You

can use most programs popular today without knowing or caring about bits, bytes, enable pins, annunciators, nibble, or the rest of the other jargon that keeps much of the computer industry detached from the rest of the world.

Dealing with the Knowledgeable

Probably few people in the history of the world have done a better job of making themselves incomprehensible than computer scientists. It's too bad, really, because the technology is finding its way into every aspect of life. Yet it seems that virtually all the experts' explanations of how computers work do little more than make computer processes absolutely opaque. There are reasons why this happens, but those aren't particularly relevant at this point.

Don't let them confuse you! And more importantly, don't be intimidated if you don't understand what a computer person tells you. Everything that you need to know can be explained in plain English. If someone begins to lose you, it's not your fault. Often those who are technically competent are among the least articulate members of society.

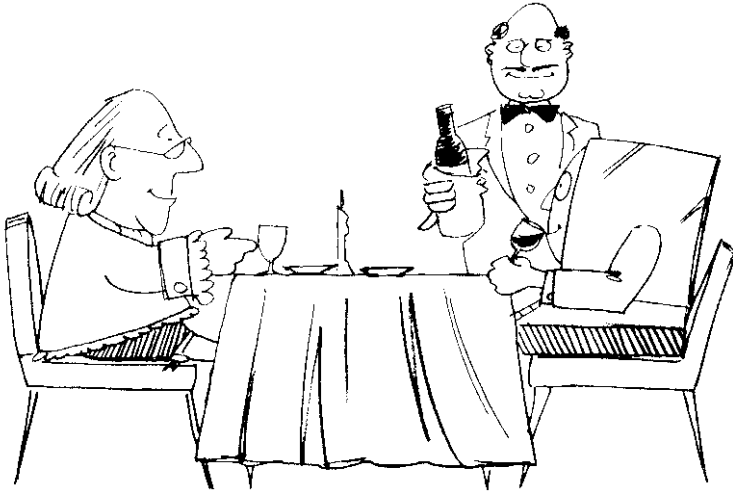
There are computer people who can explain things clearly. You'll know who they are by their patience and their pace. They'll explain things slowly and simply without seeming condescending. They'll illuminate the powers and workings of the machine. They won't, as a rule, take themselves too seriously and they'll refrain from inundating you with the quasi-English jargon and syntax of computerdom. Such people are few and far between. When you find one, stick like glue.

You're Not a Programmer — It's a Dirty Job

Programming is something else you don't have to know anything about in order to use a personal computer. You could probably spend the rest of your life doing nothing more than using the programs that have already been written. Some computers, as a matter of fact, were designed with users, not programmers, in mind. The Franklin ACE, for example, is a user's delight, but a programmer's nightmare.

That's fine, because you don't want to be a programmer. Not yet, at least. Instead of trying to write a program as one of the first things you do, spare yourself the headaches and use what already exists.

You'll need to learn how to use others' programs before you try to create your own. As you get accustomed to other people's programs, you'll probably come up with ideas for simple tasks that you'd like to write programs for yourself. Then, if you want to get down and dirty, you'll have some basis from which to learn to program. At that point, you'll find dozens of books on the market to teach you how to do just that.



What Can The Computer Do, Anyway?

A lot. With a “spread sheet” program, you could easily manage budgets for, let’s say, a real estate office, a little league team, a suburban high school, or the average household. With the same computer, you might go into the office or come home from work (depending on whether you’re coming or going) and read the financial pages of six newspapers from around the country.

You might write a letter or a report and be reluctant to let anyone read it until the spelling was checked. The computer could do that for you, too. When you were finished, you could play chess or find out about the weather or get a list of recommendations for wines to serve with Terrine Maison.

Then the computer could help you plan a banquet. Or a diet.

Or compound interest. The list goes on and on. Whether you want to find out the price of wheat in Kansas or gold in London, a personal computer can help you do it instantly.

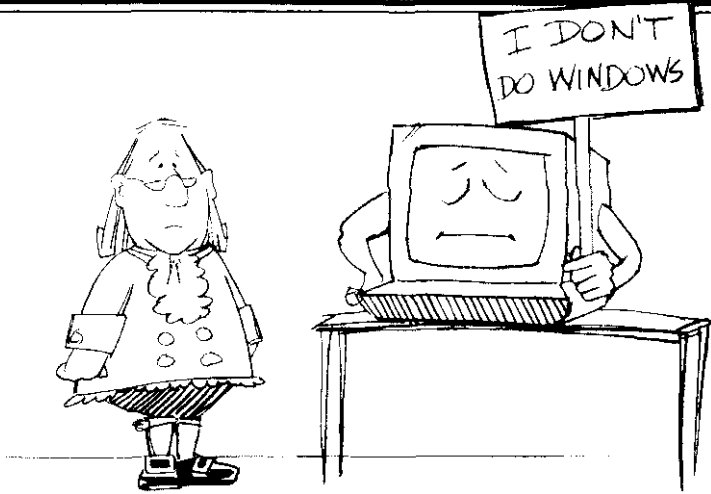
In general, if you can imagine it, someone, somewhere has already figured out a computer program that will do it for you. Since the Franklin ACE is compatible with one of the most widely used personal computers in the world today, there's a vast library of programs available to use on the machine.

A good rule of thumb to keep in mind is that you shouldn't buy a computer unless you know of at least two things that you can use it for *before* you buy. Using the packing box as a planter for your petunias shouldn't be one of the two uses for your new acquisition. But even if it is, you're sure to find more.

Virtually everything you can do with paper and pencil, you can do with a computer in a fraction of the time, with a fraction of the effort. The machines are durable, relatively inexpensive, and thoroughly adaptable to the needs and preferences of small businesses and home users.

What It Can and Can't Do

Lest you be misled, a personal computer is NOT designed to do all your work for you. The computer's role is more that of a helper; it will take care of the tiresome jobs that bore you to death. The computer's strengths lie in its ability to perform miserably dull tasks endlessly and accurately, leaving you free to think about solutions to the real problems at hand.



Since a computer won't think for you, expect no miracles. If you don't already know how to do something, then there's absolutely no way that you'll pull it off just because you're using a computer.

If you can't write a novel with a pencil or a typewriter, then you'd better hire a ghost writer, preferably one who knows something about word processing, before you buy a computer. The computer can take much of the mechanical drudgery out of any task, but it can't think or organize all by itself. Not yet, anyway.

Besides what it can't do, there are certain things you wouldn't want it to do even though it could. Just as you wouldn't take out your calculator to add five plus three, you shouldn't use your computer to write down a shopping list of five items. The pad and pencil will be faster every time. You'd look funny carrying an ACE to the supermarket with you, anyway.

Bear in mind, too, that personal computers can do only one job at a time. Suppose that you have computerized your



appointment book. Further suppose that you're using a spread sheet program to organize your bookkeeping. The phone rings. The IRS wants to make an appointment with you for an audit. You're in trouble, not just because of the audit, but because your appointment book program isn't available. What to do? Miss the appointment or destroy the records that could save you? Either way, you're lost. Always keep the availability factor in mind when deciding whether or not to computerize.

Personal computers also have certain physical and functional limitations. The ACE, for example, will choke very quickly if you try to feed it the payroll information for a Fortune 500 company. It just doesn't have the capacity for such a task. The system for storing information is too small, and the computer itself can't hold enough information at any one given time to perform the necessary calculations efficiently.

Imagine a filing cabinet with four drawers. You could probably store all of your family data in that cabinet with room to spare. It could hold all of your canceled checks, your medical records, your kids' old report cards, insurance

policies, and the like. But if you were filing the customer information for your local electric company, you'd be lucky if all of the information on customers whose names begin with "A" would fit in one cabinet.

So too with a personal computer like the ACE, except that instead of a filing cabinet, you have a floppy diskette, a sort of electronic filing cabinet. It stores information that you can bring into the computer to review, update, and manipulate. Theoretically, you could store an electric company's customer files on a floppy disk system. Practically, it would be a disaster. You'd need several thousand floppy diskettes (file cabinets) to hold all the information. And because a personal computer's memory size is also limited, you could only do a few calculations at a time. It's like using a shovel where you need a bulldozer.

Generally, if your data is of a personal or small business nature, you should be able to use a personal computer effectively.

What You Have to Remember About Memory: Size Counts

If you make the mistake of talking to a salesperson or a computer expert about memory size, you might find yourself descending into a hole from which you have no hope of returning. The only thing you really have to know is that 64 is larger than 48 and that 48 is larger than 32. The K and the reference to RAM that invariably accompany these numbers are right at the edge of the precipice you want desperately to avoid. (The fatal question is, "What are they?")

In order to use some programs you need at least 32, and for others you need at least 48. Others take only 5. The "at least" is very important. It usually means "barely." Programs that take 32 usually work better on 48, and those that take 48 often work better on 64.

Better in what sense? Let's say you're using a data base management system that takes at least 48. You're working up a report that covers information gathered and entered over several months, and you want the computer to compile the data for the entire period. It may not be able to. Perhaps it will be able to do it for one month and then another month and then a third month. You'll have to put the three sections together yourself. With larger memory capacity, the computer could probably do the whole job for you.

Speed is another factor. The machine can pull data from memory much faster than it can retrieve the same data from a diskette in a disk drive. If you were using a word processing program, you'd be able to go through the text, editing and moving paragraphs from one place to another much more quickly if your computer had 64 instead of the 48 that the program manufacturers recommend as a minimum. With a long text, a computer with a smaller memory would be constantly moving text to and from the diskette in the disk drive. It gets cumbersome.

There's also the future to consider. Programmers are constantly writing programs that are bigger and better than those that came before.

A Good Time To Be Interested In A Personal Computer

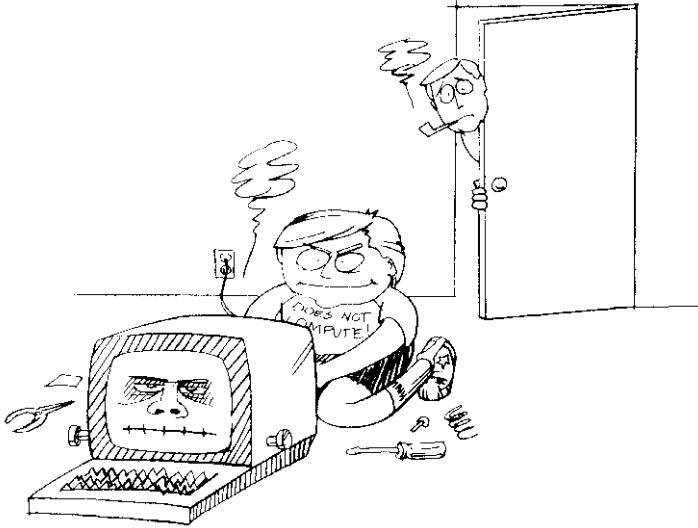
Within the last decade, computer systems that might have filled a small room with board after board of transistors, tubes, and integrated circuits have been reduced in size to where they fit easily into a desktop case. Systems that once cost tens of hundreds of thousands of dollars are now affordable for the home or small business. The sheer speed of the advances in computer technology makes all other advances in the history of technology look like molasses in January.

In the early seventies, Hewlett-Packard and Texas Instruments came out with the first programmable calculators. Suddenly it was possible for the average person to carry an instrument that would do simple arithmetic, chart biorhythms, and provide the calculations needed to navigate by stars.

Some people became so enamored of these devices that they lost sleep playing with them. A few were so overwhelmed that they quit their jobs to learn about computers and programming. Some went to school unabashedly. Others hid in garages, basements, or dusty attics. Eventually, most emerged to take high paying jobs in a booming industry desperate for people with technical expertise.

Not long after that, build-it-yourself computer kits became available. Otherwise sensible people bought soldering guns and began to attach electronic components to plastic-like boards riddled with tiny holes.

Exactly what they were doing was a mystery to all, even



them. These kits seldom had anything approximating complete instructions. Untold hours were invested with minimal success. There's no telling how many dining room tables were marred by scorch marks from soldering irons left glowing by sleeping fanatics whose own fervor had already burned out for that evening.

It's all different now. The machines come assembled and, for the most part, with adequate instructions. They're relatively inexpensive, reliable, easy to use, and remarkably versatile. There's also enough competition in the marketplace to keep quality high and prices stable.

It's definitely the right time to buy. Being overcome by fanatical fervor is still a possibility, but burn marks in the finish of fine furniture are no longer a likely result of the affliction.

Compatibility — The Good, The Bad, And The Ugly

Today there's some controversy in personal computerdom over what constitutes compatibility. You'll probably hear something about it when shopping for a computer. Usually compatibility means something different to the manufacturer than it does to the computer buyer.

To the buyer, it means that if something — a program or some attached device — works on one computer and another is compatible with the first, then that program or device should also work on the second computer. Very simple. The buyer thinks compatible means "the same as." But the manufacturer means that the program or device was *designed* to be 100% compatible.

Did the manufacturer succeed in his design? Well, almost everything works. Does the manufacturer know of anything that doesn't work? Well, there are one or two programs that don't quite work. Watch out! You're about to confront Murphy's Law of Compatibility: If the manufacturer claims it's compatible, the purchaser assumes all responsibility for determining whether or not the manufacturer is right.

Franklin, for example, doesn't claim that the ACE 1000 line is 100% compatible with comparable APPLE® computers. Because the ACE has features not found in the APPLE, it's different. Perfect compatibility is lost.

A good example is the keyboard. It produces upper and lower case characters, a definite advantage to the user over the

APPLE, which has only upper case capability. This means that ACE owners confront a problem now and then if they use a few of the programs specifically designed for the APPLE.

Take solace in the fact that the advantages provided by the ACE's unique design more than make up for any problem you might encounter. Whenever possible, reliable companies warn you about compatibility problems and try to offer solutions.

Be Choosy About The Company You Keep

When you buy a personal computer, you're getting more than just a machine. You're really establishing a relationship with a company, so it's important to choose carefully. Every computer company has a somewhat different attitude toward its customers.

Unfortunately, that attitude is seldom readily apparent before purchase and sometimes only too painfully clear afterwards. In an industry where the oldest company may have been manufacturing machines only a few months longer than the youngest, relative longevity isn't necessarily a reliable indicator of quality.

Probably the best way to judge a company is to determine whether you, the customer, are of primary importance or if you're merely incidental to the marketing scheme.

To see where you fit in, check the instructional manuals provided with the computer. Are they cute but short on practical information? Are they written in computerese by technicians totally blind to the realities of the English language? Does the company acknowledge the limitations of the technology or were the manuals written by some ad agency bent on creating an image of corporate perfection? Some of the largest, most

powerful companies in the marketplace take their customers for granted. It shows in the documentation they provide.

Does the company have service centers available or are repairs strictly the customer's problem? Some companies can't be bothered. They don't even provide warranties. Give the main office a call to see if there's a customer service department.

The Dealer

Another important relationship you'll establish is with your dealer, probably your most valuable ally if you want to get the most out of an ACE or any other personal computer. What kind of help do you get when you walk in the door? Do you get good, clear, objective advice or sales hype? You'll probably need a lot of help making initial and future purchases, so find a dealer that you feel comfortable with and can trust. If, when you walk into the store, the salespeople seem more interested in the computers than they are in you, look for another dealer.

Many dealers offer evening courses on how to use personal computers. Sit in on a session and see if the instructor is a good communicator, sensitive to the needs of the class. These classes are usually available for a minimal charge, but the quality of the teaching varies. A good dealer will be concerned that customers learn how to get the most out of their computers.

Often, dealers provide repair service for machines. Check to see if other customers have been satisfied and see if the company provides training for dealers' technicians. This is the case with Franklin, but not so with every company. Look for evidence of teamwork between company and dealer.



True Grit: The Adventure Of Mail Order

If you decide to buy a computer from a catalog, best of luck. You can find some tremendous bargains, but you'd better know what you're doing before you even consider it. The principle of caveat emptor applies here more than anywhere else.

What are you going to do if the computer you ordered comes disassembled, all the parts in plastic bags, the electronic components stuck in chunks of styrofoam? You have no dealer to rely on for help. Assuming you get it all together, what will you do if it doesn't work? One of the components may be bad. How are you going to know which one? Maybe you can send it back to the company, but you'll have to anticipate a long wait before you get your new computer back.

It's awfully hard to know exactly what you're ordering unless

you're very familiar with personal computers. Although mail order prices can be tempting, the safest way to buy a computer is through a company authorized dealer. This is your best guarantee of getting the kind of support you'll probably want.

Clubs

A good source of information and bargains in computers, accessories, and programs is the users' club. Generally, these clubs are groups of people using the same or similar computers. They're interested in sharing information and programs, as well as helping one another solve problems that arise when using new technologies.

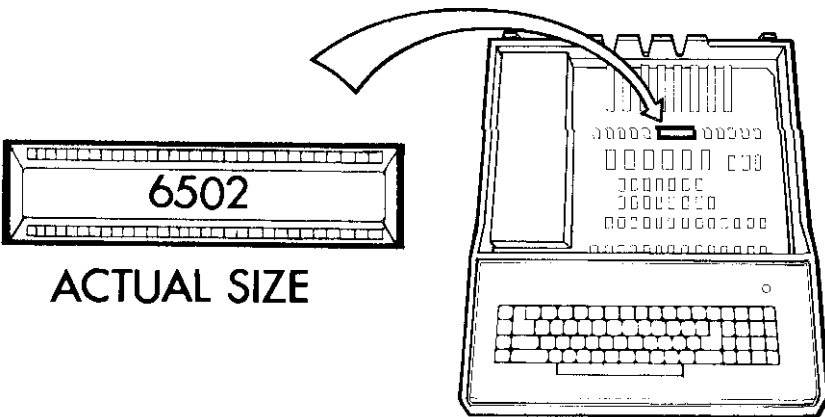
Often the clubs have a general meeting once a month with a speaker or special presentation. Between general meetings, groups of people with special interests frequently get together to help one another or just discuss what they're doing. The atmosphere is usually convivial and people have a chance to meet others who share their interest.

The companies and dealers also find that these clubs provide a good medium of communication. Dealers announce special prices to club members and companies let the groups know about new products. It's a good way to keep in touch.

There's More To It Than Just The Computer- Additional Equipment

If you thought the computer was expensive...

Inside an ACE and similar computers is a little electronic device called a microprocessor. How small is it? The one that runs the ACE is only about as big as the **SHIFT** key on the keyboard. But it's the heart of the system. Without a microprocessor, you wouldn't have a computer.



Everything else in the case is there to help the microprocessor do its job. You have the power supply to convert and regulate the electrical current going to the computer and its support devices. You have some devices called memories that hold information in the computer while you use it. You have some slots at the back that allow you to plug in devices that expand the capabilities of your machine. You have the keyboard. The microprocessor can't do anything by itself. It needs all the elements of the system to do its job.

It also needs other components so that you can exploit its capabilities.

A computer by itself is akin to a car engine with no body or chassis. Its incredible capacity for gorging and disgorging data is useless unless the user has help in handling all this information from other devices known as “peripherals.” They aren’t really peripheral, though. They’re essential. They’re on the periphery, the outside boundary, of the main activity of the computer itself, but they’re much more than mere accessories. The computer is virtually useless without them.

TVs And Monitors

The TV screen is to an ACE or any personal computer what a piece of paper is to a typewriter. Everything you type on your keyboard is also displayed on the screen. Also, everything that the computer wants to tell you — the results of your calculations, the contents of a diskette, the score of the Dodgers’ game — is displayed on the screen. A monitor or TV is absolutely essential. Without one of these two types of display screens, you’ll never be able to have two-way communication with your computer.

A monitor is a trimmed-down, souped-up TV. It doesn’t have a channel selector and antenna on it, but it does have a much sharper and clearer picture of the words that the computer is sending out than a normal TV screen does.

Should you use a TV or a monitor? That depends on what you’re primarily using your computer for. If you intend to spend most of your time performing useful work on the machine, then you should use a monitor. If you’re interested in color displays and those programs (generally games) that make the most extensive and intensive use of color, then you should probably

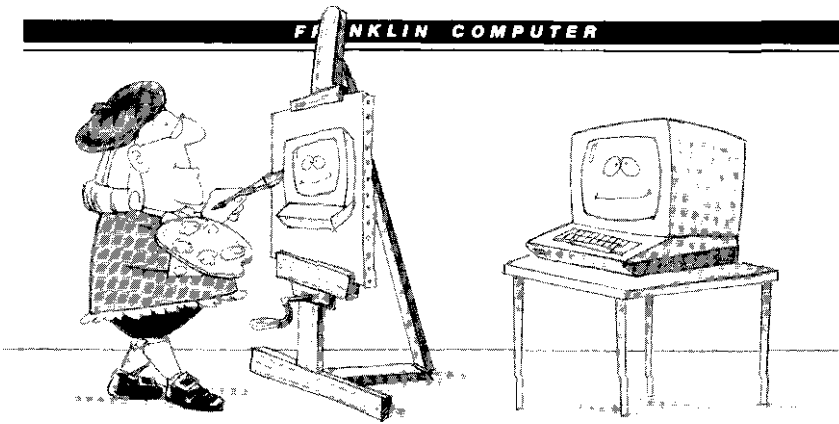
use a color TV. You can still play games if you use a monitor but they just won't be in color.

If you do decide to use a TV, then you'll need another device, an RF modulator. The name sounds exotic, but all it does is let you connect the TV to an ACE or similar computer. Typically, it's a little box about as big as a pack of cigarettes with a cable similar to regular TV antenna wire coming out of one end. This wire attaches to the antenna connections on your TV. Another cable comes out of the other end and plugs into the video connector on the back of your computer. Tune your TV to the channel suggested by the modulator manufacturer and away you go.

Printers — For Paper Copies Of What You Do

A printer is simply a typewriter without keys. After all, why would you need another set of keys when you've already got a perfectly good one on your computer? With a printer, you can transfer the results of your computerized activity onto a piece of paper.

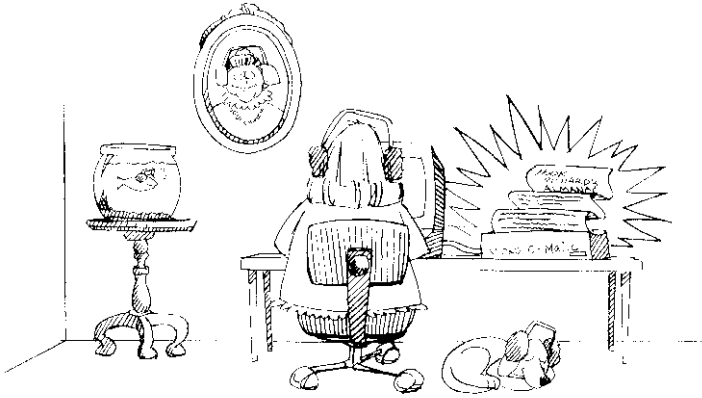
Why would you need a copy of your work on paper when you can easily see it on the screen? Well, screens are terrific for browsing through files, looking at the contents of a floppy diskette, or performing simple calculations. The computer can display data on a screen much faster (and more quietly) than it can print it. If, however, you want a permanent record, the screen hardly suffices. How would you give a copy of your work to someone? Send a snapshot of your screen? You'll want a paper copy of your results, and for that you need a printer.



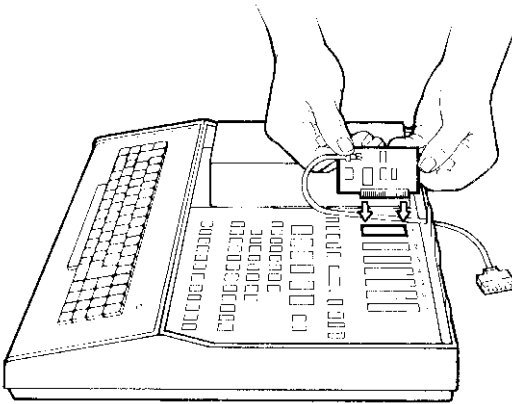
Printers also help to solve a problem sometimes called “window vision.” When you’re browsing through a multi-page document by viewing it on a screen, your overall view is limited by the number of lines of text displayed on the screen, normally 24. Window vision can be aggravating when you’re constantly scrolling through a document to cross reference information that appears in the beginning with information that appears elsewhere in the document. With a printed copy you can spread out the pages on your desk for easy access to all the required information.

Consider buying a printer. But which one? There are hundreds available. The first thing to look for is the printer’s compatibility with the programs you want to use. Another consideration is whether or not you’ll need typewriter quality copies. Some can give it to you; others can’t. There are also the matters of noise and speed. Be sure you listen before you buy. If you get one that’s loud, you may have to build a separate room for it. Also check the manual. No printer will do you any good if you can’t figure out how to use it.

If you do buy a printer, you’ll have to get several other items before you can use it. You’ll need a printer interface card, a cable, and, of course, paper and ribbons.

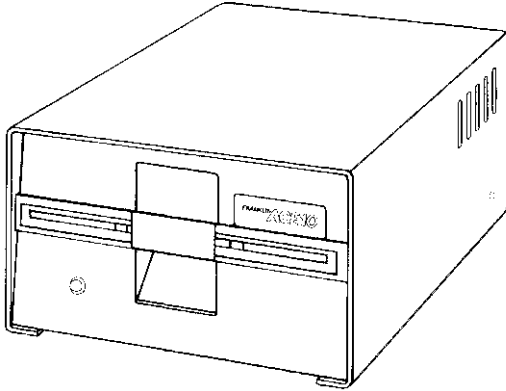


What's a printer interface card? Remember the RF modulator? Well, a printer interface card permits a similar arrangement. Your printer cable connects easily with your printer, but there's no place on the computer to plug in the other end. The printer interface card solves this problem. It's nothing more than a plastic board about the size of a 3 x 5 index card.



Although you should follow the directions when putting it in, installation is only a little more tricky than plugging in a toaster. Once the card is plugged in, you'll have a connector where you can plug in the free end of your cable. That's all there is to installation.

Before you try to operate it, you'll have to check the printer manual to see what you should do next.



Disk Drives

Typing things twice is fine if you're learning how to type, but otherwise it's a waste of time. You might be wondering why you'd have to type things twice if your computer has a memory. Well, it does, but the memory in the computer itself is very forgetful. When the switch goes off, it forgets everything. Poof! If you want to retain what you've typed in, you've got to transfer it out of the computer's memory and store it onto floppy diskettes, your files. Memory, you see, is very different from storage.

All you have to do is instruct the computer to empty its memory onto the diskette before you turn off the power. The program you're using should provide you with a procedure for doing this. Then, when you need the information again, you can instruct the computer to read what's on the diskette back into its own memory. Voila! Whatever you typed in before is now back

in the computer so that you don't have to enter it again in order to work with it.

Disk drives are the electronic transcribers and filing clerks. They record and correctly file information on your diskettes. The electronic filing system remains invisible on the diskette, but don't worry because you can't see it. Your computer and disk drives can find and retrieve your files more quickly than you can find a file folder in a filing cabinet drawer.

There's no limit to the number of diskettes you can use, but you can only have one diskette in the drive at a time.

How many files can you store on a single diskette? Well, how many file folders can you put into a file drawer? It depends on how big the file drawer is, what kind it is, and how big the file folders are. The same thing is true with diskettes. Just as there's a limit to the number of file folders you can stuff into a file drawer, there is also a limit to how many files you can store on a single diskette.

Using a computer is almost impossible with no disk drives and still difficult with only one. In fact, some models of the ACE come with two disk drives built right into their cases. Two drives are more than twice as good as one. Copying data from one diskette to another for backup purposes, a crucial operation, is much faster and easier with two disk drives than with one. Some personal computers let you use a tape recorder for data storage, but models like the ACE don't, simply because the cassette tape method is a very, very slow and cumbersome system. It offers very little flexibility and can't be used at all in many applications.

There are many drives on the market. By far the most widely used are the drives commonly referred to as 5-inch floppy disk drives. There are smaller drives (not too many) and bigger drives

(a good many), but the 5-inch gives you the most for the money.

The bigger drives are physically larger and they store more data. The 8-inch floppy (standard) disk drive is actually about twice as big as the 5-inch in terms of size, data capacity, and price.

A 5-inch Winchester “hard disk” stores about 50 times more data than a 5-inch floppy and you can store and retrieve the data at a much faster rate. As you might expect, the price for this kind of performance is high, especially when you consider that in most systems the Winchester doesn’t completely replace floppies. You still need a storage device (either floppies or a special high-speed tape device) to “backup” the data on the Winchester. It’s also impossible to remove a Winchester disk from the drive itself, so whatever happens to the drive also happens to your data. Backup becomes twice as important.

Don’t get too excited about 8-inch floppies and Winchesters. They present the typical first-time computer user with a lot of problems, only some of which have been mentioned here. Some are solvable, none cheaply. If the machine you buy doesn’t come with disk drives already installed, get a floppy disk system, learn how to use it, and then you’ll be ready to consider the next step up.

Video Cards — For Lines Longer Than 40 Columns

Computers like the ACE can easily display text on your TV or on a monitor screen, but the quantity of the image is a function of your screen and your eyesight. The standard personal computer can put up to 24 lines of text horizontally on the

screen. Each line may hold up to 40 characters. This works out to a maximum of 960 characters on the screen at any one time. Sounds like a lot, doesn't it? It's not. The majority of computer applications are easier using a line anywhere from 60 to 80 columns wide.

Most programs handle this problem by providing you with a "scrolling" or "paging" capability.

Imagine a long document. Further imagine the document broken up into "pages," each 24 lines long. You choose the page (set of 24 lines) that you want to look at at any particular time. Usually, you don't have to worry about specific page numbers, because most programs let you move through the document a page at a time in either of two directions. You can ask to see the previous page or the next page. This process is called paging.

Scrolling is similar, except that you move through your document one line at a time instead of one page at a time. For example, if what you see on the screen represents the 10th through the 33rd lines of a document and you scroll forward, you will then see lines 11 through 34. If you scroll backward, you will see lines 9 through 32.

You can scroll horizontally, too. Suppose you were working on a document 80 columns wide. That may be a little hard to imagine, but it's quite possible. Since the computer can display only 40 characters on a line, you wouldn't be able to see the entire line at any one time. You'd have to scroll the screen right and left to view data outside the range of the original 40 columns. Unfortunately, this horizontal scrolling isn't as helpful as vertical scrolling. Your data might seem disjointed because you'd see half of the first line, half of the second line, etc., the whole way down the screen. If data on any particular line had to

be interpreted in the same context as data on the preceding line, then what you'd see could be very confusing.

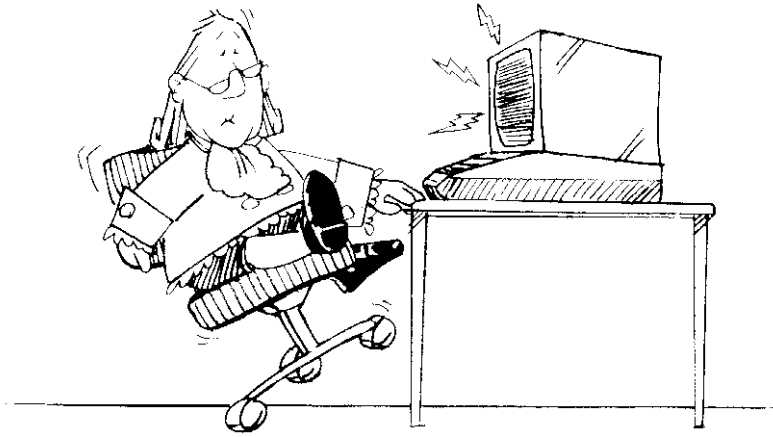
One solution is to allow each line of text to “wrap” around to the next line so that anything beyond the 40th column is displayed on the line below. Much of the time, this can be very effective. In fact, if you never knew that it was possible to display 80 characters on a line, you might be perfectly satisfied with only 40 columns. However, now you do know. Once you try the larger display, you'll never want to go back to 40.

Some machines come equipped with 80 column capability. With others you have to buy an 80 column video card, throw away your TV, and buy a monitor. You probably expected having to buy the video card, but the news about the TV might come as an unwelcome surprise. Unfortunately, you'd never be able to read the display if you tried to cram 80 characters on one line of a TV screen. Because the monitor can display a clearer and sharper picture than the TV, the characters will be easy to read — as long as you get a good quality video card.

A video card looks much like a printer interface card, except that it's a little bigger. It's installed much the same way as the printer interface card.

There are several such video cards available today, each with various advantages and disadvantages. Probably the two most important things to look for in selecting a video card are the quality of the display it puts on the monitor screen and the number of programs that will work properly with it.

The quality of the display is a matter of how clearly the card displays 80 character lines on the monitor. This isn't easy. Test any card you might consider buying. Pay particular attention to any symptoms of oncoming headaches, eyestrain, or seasickness



as you read your screen full of 80 character lines. This is Video Card Test 1. If none of these symptoms develops, and you can easily read the text displayed using the card (make sure you try lower case characters, too), then the card passes Test 1.

Test 2 is called, "Will it work with the programs I'm going to want to use?" You see, the video card makers all do things a little bit differently, while the program writers are busy doing things even more differently. This creates some finger pointing and, more importantly, many incompatibilities. To avoid such problems, get your dealer's advice and try the card.

These are the **ONLY** two qualities that you should look for in buying an 80 column card. Don't let yourself be confused by the endless variety of miscellaneous features that the various card manufacturers try to impress you with (unless, of course, you really care if your video card can whistle "Dixie" in which case, that's the card for you.) In particular, don't pay extra for a lower case feature if your machine already has one. All ACE computers do.

Communication Devices — For Talking to Another Computer

The purpose of communication devices isn't to establish a social life for your new computer. Communicating with another computer is actually one of the more important things you can do with an ACE or other personal computer. In fact, telecommunications is the main reason why some people buy personal computers.

You can hook your computer up to your standard telephone, dial the number of your local computer center, and start sending and receiving information. This is an amazingly powerful setup. Just by calling, you have access to a large, very powerful computer well beyond the financial means of any individual.

Since many people can be hooked up to it at the same time, the costs of purchasing and maintaining the big machine are distributed among many users. To the occasional user like you, this translates to cheap. In fact, in some cases, it translates to free, except for the cost of the phone call.



Once you're connected, you can instruct the central computer to do things for you just the way you do on an ACE. A wide variety of services is available to lonely people out there who like to call up computers. Typically, a dealer sells you the service and gives you a phone number to call, an instruction manual, and some identification numbers so that the big computer will talk to you. It won't, unless its humans know where to send the bill.

As usual, there's a catch, a common one in computerdom. You settle up with your dealer, run home all excited to call up big brother for a chat, try to plug the phone into your computer and say, "Hold on here. Where do I plug the phone in?" Well, unless your dealer has sold you a telephone interface card, you can't. Just like the printer, the telephone needs an interface card.

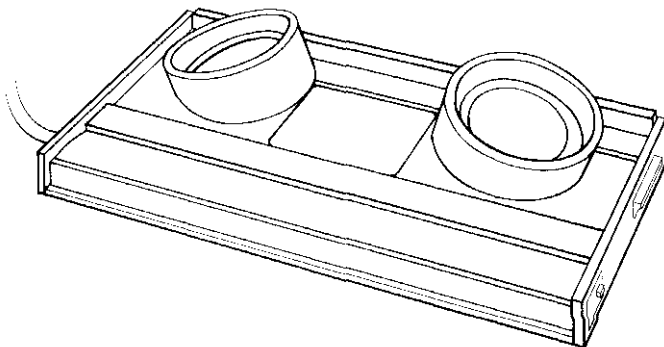
Computer people being the way they are, though, the telephone interface card isn't called a telephone interface card. You have to learn a buzzword — modem — if you want to buy one. Whisper the word to your dealer, and he'll know what you're talking about.

You plug the modem card into one of the slots in the back of your computer (now you begin to see why there are so many slots back there), connect a cable to the card, run it out the back of the ACE, and . . . now what?

How you connect the modem to your phone depends on which type of card you buy. The first type of modem is called a "direct connect" because it attaches directly to your phone line. Where? Into the little outlet jack on your wall where your phone plugs in. The little jack for wall phones is hidden behind the phone.

If you don't feel like unplugging your phone and plugging in your modem every time you get the urge to call the central computer, contact your local phone company and have them install an extra jack for you. It's not too expensive. The only problem is the hassle you generally get if you tell the phone company that you want to hook your computer up to their precious lines. This unpleasant encounter can be avoided. If you don't tell them why you want the extra jack, no one else is likely to.

The second type of modem uses what's called an "acoustic" connection to the phone. Instead of directly wiring the modem card into the phone line, you establish the connection by letting the modem use your telephone handset to "talk" and "listen" to the big computer. This takes a little more than just an extra wall jack.



The acoustic coupler is a box with two foam rubber rings on it and a cable coming out. The handset of your phone plugs into the rubber rings, and the cable plugs into your modem. With that done, you're all set.

Which type of modem should you buy? The acoustic method is usually cheaper. It is also less reliable, and it can't provide you with as many features. If you can afford it, buy the direct connect type. It's easier to use and more versatile as well.

A word of warning: Don't buy a modem unless you already have a disk drive. Modems come with diskettes containing programs that tell your computer how to communicate through the device. Without a disk drive, your computer won't be able to get the instructions off the diskette, and your modem card will be worthless.

Computers In A Computer — If One Is Good, Two Are Better

The ACE is a very powerful machine. With it you can take advantage of a huge library of programs. You can buy almost any conceivable device to attach to it.

Some people, however, are never satisfied with what's on the market. They want more, and sometimes they find a way to get it. Well, a certain group of malcontents found a way to get what they wanted and every owner of an ACE-compatible computer is better off because of it.

Think of computers as countries, each with its own language. Computer A understands Swahili, computer B speaks French, and so on. This means that computer A can't understand the same set of instructions as computer B. In this Tower of Babel, the sets of instructions are called programs, and the computer that understands "Bonjour" doesn't understand "Uhuru."

Further suppose that someone named Joe Coder has come up with a fantastic idea for a program. It's so terrific that the world is going to beat down his door trying to buy it. Joe Coder is going to get rich!

Well, not just yet. First Joe has to develop the set of instructions (the program) that will make the computer do

what he wants it to do. That's not easy, and that's why people will pay Joe handsomely for his efforts. Unfortunately, Joe has a problem. Since computer A will not understand a program written for computer B, which computer should he write his program for?

Well, Joe isn't stupid. He develops the program for the computer that he thinks will give him the largest market. Fine for Joe but not so good for you, unless you share Joe's preference in computers.

In the personal computer field, there are two major types of computers (countries, if you will). There's the type known as "**DOS 3.3**" that you find in the ACE and ACE-compatible machines like the APPLE, and the type known as "**CP/M®.**" **CP/M** systems work in many kinds of computers, all of which understand the same language. Programs written for one **CP/M** system usually work on any other **CP/M** system. However, as you'd probably expect, programs written for **DOS** will not work on a **CP/M** system and vice versa. This is too bad, because there are terrific programs written for each system that half the personal computer owners can't use.

Thanks to those who are never satisfied, a solution is available. All you have to do is plug a **CP/M** card (optional on some models of the ACE, standard on others) into one of the slots at the rear of the machine. With the card in place, the computer can use either type of program.

This feature amounts to putting another computer inside your computer, a significant achievement. The designers even made it very easy for you. You don't have to know which one of those little devils in your computer is running the program! All you do is insert a diskette, start the computer up, and everything necessary to allow you to use that program is done automatically.



Programs

Equipment alone -- no matter how sophisticated -- is not all you need to operate your computer. Computers need programs.

Programs are easy to come by, but few people can really explain what it is that they have even after they get them. Nobody can tell you exactly what programs are, although everyone seems to be buying them, selling them, writing them, swearing at them, or pirating them these days.

The standard definition is that a program is a set of instructions to a computer. That's correct as far as it goes, but what does it really tell you? Does it help you to understand

“PAC-MAN™,” the “computer errors” you get on your bank statement, and the workings of NORAD buried in Cheyenne Mountain in Colorado? Programs are involved in all of these computer processes, but people who write them probably couldn’t agree on a definition more specific than the one just given.

Why can’t people come up with a good definition of “program”? They’re so difficult to explain primarily because you can’t see, hear, touch, or smell them. Since programs are for the machine’s “eyes” only, you’ll never see the programs themselves, only their effects on the machine. They’re somewhat akin to a “life force” that lets a machine be productive when the electricity is turned on. Perhaps yet another reason is that a definition isn’t really necessary. You really don’t have to know what they are in order to use them.

The last point is probably the most important, and yet it’s also the one that causes the most profound suspicions and rankst uneasiness. People generally like to see a definition to know what they’re dealing with. But if you think about it, you probably use things every day that you don’t fully understand or feel compelled to define. What do you really know (or care) about the telephone signal, without which you couldn’t reach out and touch someone? How about those electronic signals that come out of thin air to make your television set full of *The Dukes of Hazzard* or *Hill Street Blues* or *Live at the Met* with Itzhak Perlman?

You don’t really need to know what a television signal is in order to like or dislike a program, to turn it on or turn it off. So too with computer programs. The fact that they’re also electronic signals is about as necessary for you to know as is the fact that David Hartman is nothing but reconfigured electronic impulses when you watch him over coffee in the morning.

Admittedly, when you use a computer program you're more involved in some of the intricacies of how it works than you are in the operation of the television when you sit down to watch a program. You don't really watch computer programs. You use them. You play with them. And, eventually, you develop a sense of what they are.

No matter how they define the term, most computer people agree that the best way to understand programs is to use them. Learn what they are inductively, empirically, not through a textbook definition or a muddled series of analogies that do as much to confuse as they do to clarify.

Just dive in. Start with simple, utilitarian programs that you can master quickly. Then decide what you want to do with your computer, buy some programs, and go for it.

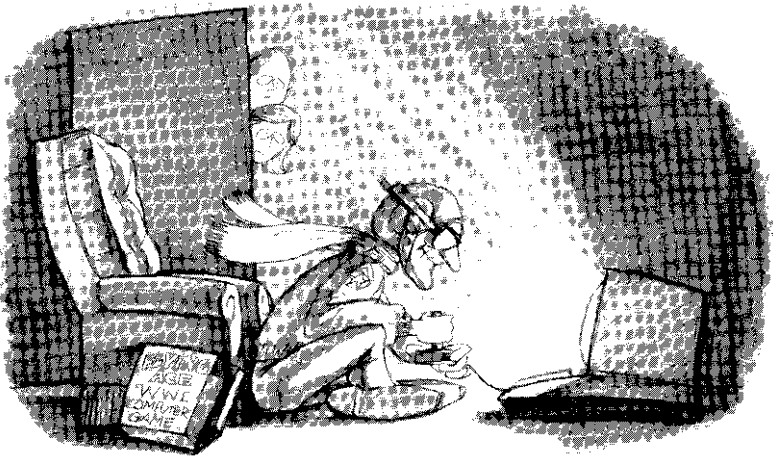
Be warned, however. Programs are expensive and they vary greatly in quality. Before you buy one, make sure it has some practical application to justify its cost, and be sure you're comfortable with the documentation that comes with it. If the manual or instruction booklet is written in computer gibberish, the program may be more trouble than it's worth. Look for lucid and complete explanations.

Be careful not to confuse impressive activity with valuable service. Programs and add-on devices are proliferating beyond imagination these days. Some of them are good, some of them merely cute, and some of them downright ridiculous. (A clock program, you say? One that will make my video screen look like a chunk off of my kitchen wall? Great! With second hands yet?)

On closer analysis, however, the sole purpose of many of these wonders in programming is to separate you from your money.

Don't fall into the trap of buying a program simply to have something running. There are many worthwhile products available, so take your time, look around, try things out, and then buy what you can really use.

Be forewarned that somewhere, sometime, someplace, some enterprising young man who seems to know ten times what you do about computers is going to try to convince you that his program will make a jug of cider jump off the table and turn ducks' eggs into solid gold. Look this man straight in the eye and ask him for names of people who are successfully using his program. **DO NOT**, under any circumstances, bet him that he can't do it. There's no telling what someone might be able to make a computer do.



Paddles and Joysticks

If you want to play video games in the privacy of your own home (or office), or if you're sick of sending the kids to school with plenty of lunch money, only to find them dragging home

from the arcade with empty stomachs and aching wrists, maybe you should consider buying some game programs and a set of paddles or joysticks. These gizmos let you beep or zap or pursue or escape, depending on what video game you happen to be playing.

Paddles, usually sold in pairs, vary somewhat in size and shape. They all have dials, one to move something vertically on the screen, the other to move something horizontally. What kind of something? That depends on what game you're playing. It might be a helicopter or a spaceship, a tank or some creature. By turning both dials at the same time, you can control your direction all over the screen, even make your something move in circles. To zap an oncoming invader, you have buttons on the sides of paddles. Often they make the machine go beep.

A joystick provides the same kinds of directional control, except that instead of two dials you have one stick on one little unit. Your movements on the screen correspond directly and immediately to the direction you push the stick. Just as with the paddles, you have a button that lets you beep and go zap. Neither device would leave you defenseless.

Joysticks and paddles connect to your computer at a tiny socket marked I/O. The exact location of this socket varies from computer to computer, but in most cases it's the only place inside the machine where the 16 pin plugs for the joysticks or paddles look like they might plug in. The I/O socket comes empty, waiting for you to plug in one of those devices or the other.

By the way, the games themselves come on diskettes. You'll find everything you need at your computer store to turn your computer into an arcade.

One last word of advice: Protect the plugs on the joysticks and paddles by sticking them in styrofoam when you're not using them. The little pins at the end have a way of getting mangled otherwise.

Before You Try To Set Up A Personal Computer

There are a few items you should bring home with you once you decide to get a computer. Do you have at least one floppy disk drive? The ACE and most other personal computers aren't much good without one. Did you get some floppy diskettes? You should have a box to get yourself started. How about a printer? If you get one, make sure you have the cable for it. And don't forget paper.

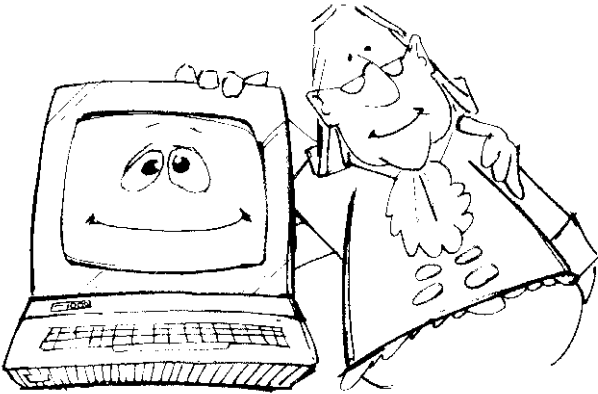
Here is a short checklist of other things to consider:

1. Did you get an RF modulator? If you plan to use a TV as a screen, you'll need one.
2. Are there enough AC sockets where you're going to set up your system? You'll need at least two sockets, one for the ACE and one for the TV or monitor. You'll need another if you also have a printer.
3. Did you buy a game program that you can't wait to fire up? Then pick up a set of paddles or joysticks.

It Isn't All That Difficult

After reading through this booklet, you have all the general information you need to set up and use a personal computer with complete confidence. You may run into snags every now and again. Everyone does. If you stay loose, follow directions, and experiment a little, you'll soon resolve any difficulties. Remember that you can count on the dealer, the club, and the company for help if you need it. You're not alone, so don't let yourself feel isolated.

You're ready now to get set up and go. Franklin hopes this introduction has been helpful and that you have the best of luck with your new computer.



What You Have To Know: Seven Commandments (Developing The Uncommon Sense of Computerdom)

To use an ACE properly, there are a few general rules that you should know and obey. Observing Franklin's Seven Commandments of Computerdom can save you from a lot of grief, not the least of which is the noxious horror of silicon on fire.

I. Electrical Power — Don't Change A Bulb With The Power On

The ACE is an electrical device, so never tinker with anything inside it when the power is on. Never. Never. Never. It's dangerous for you, and you can seriously damage your ACE.

Never plug in (or take out) one of the interface cards while the power is on. You'll ruin something almost every time. Unfortunately, it's easy to forget this rule, especially when you



want to try a new card or do something with your computer that requires a different set of cards.

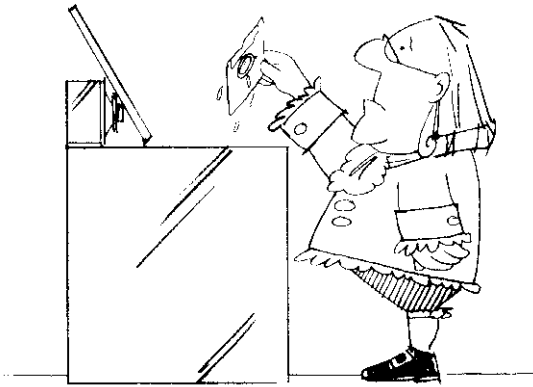
Train yourself to be good. Make a habit of turning off the power and pulling out the plug in the back of the unit before you make any changes inside. Then, every time you touch a card, check to make sure that the power cord is disconnected. Always. Force yourself.

Someday you're going to forget. Not even computerdom is free from sin. When you do, immediately turn off the power. Look for obvious signs of damage such as smoke or glowing circuits. Whether you detect any damage or not, consider your system suspect until you or your dealer can verify that everything is OK. In particular, don't take chances with any of your important programs until you're sure that the system is functioning properly. In other words, don't use any diskettes that you can't afford to destroy.

II. *Diskettes Are Not Dry Cleanable*

Diskettes are very delicate objects. Treat them gently, tenderly. Imagine damaging a diskette holding several hundred dollars' worth of programs or several days' worth of work. Unless you had a backup copy, you'd have nothing to show for your investment or your time. Here are a few rules to follow so that you won't mangle your diskettes.

1. Let nothing touch any of the exposed areas on the diskette.
2. Don't bend, fold, staple, or otherwise mutilate your diskettes. Avoid paper clipping anything to them. Don't even write on the label of the diskette unless you use a felt-tipped pen and a featherlight touch.
3. Never put diskettes in the oven or the refrigerator. More seriously, don't store them anywhere you would feel uncomfortable in a light sweater.



4. Keep diskettes away from magnetic fields. Some devices that generate these fields are TVs, some display monitors, telephones, and airport metal detectors. For this reason, don't lay your diskettes on, near, or under your display screen or your telephone.
5. Keep diskettes clean and dust-free by storing them in a case whenever they're not in use. Never wipe them off to clean them. Refrain from smoking around your diskettes or drives. The smoke contains dust particles.

III. Reset Switches Are Your Worst Enemy

Like most personal computers, the ACE has a dangerous little button called a reset switch. On older models (the ACE 100 line), this button is at the upper right-hand corner of the keyboard. On the newer ACEs (the ACE 1000 line), **RESET** is hidden from view but still easily accessible. Don't go looking for it yet, though. If you don't know where it is, you'll never have any problems with it.

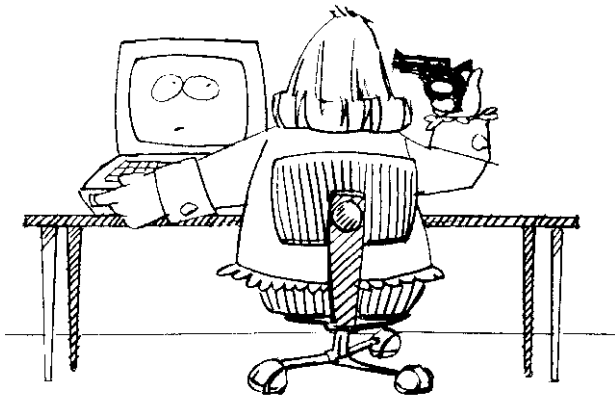
What's **RESET** for if you're not supposed to push it? Well, there are times when a program can become so confused that it will start ignoring you. (The program seems to go into never-never land.) Why does this happen? That depends on the program, the computer, the diskette, the phases of the moon, the return of the swallows to Capistrano, all kinds of wacko things. It happens.

This is where **RESET** comes in. **RESET** forces the errant

program to pay attention to you. Regardless of whether it's gone off on its own while playing Tank Trek, figuring amortization schedules, or calculating the distance of the nearest black hole in space, it stops its wayward behavior and answers the **RESET**. This is where you can get into trouble.

If the program is sending any data to the disk drive for storage when you hit **RESET**, it will stop. This can wreak havoc, wiping everything off your diskette.

Even if you avoid this particular problem by leaving **RESET** alone when the disk drive's light is on, **RESET** can still cause trouble. For example, if you've typed information into your ACE, but you haven't sent it out to a diskette for storage yet, hitting **RESET** may destroy all the information you have in the machine. The effect that this has on a typical user is not pretty.



Here's a good rule regarding **RESET**. Don't ever touch it. If you just can't resist the temptation, hit it only when you're *completely* finished with whatever program you're using. Never use it while the computer is sending something to or receiving something from a disk drive. If you're unsure

whether or not some of your data is on the diskette, try everything else you can think of to bring the program back under control before you hit **RESET**.

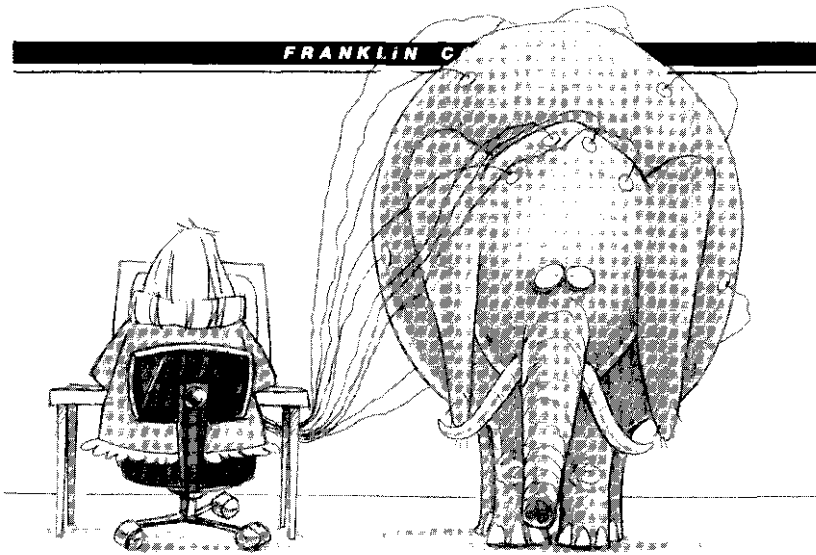
These same rules apply to the ON/OFF switch in the back of the ACE. Turning the ACE off while you're using a program or while the computer is sending or receiving a program or data from the diskette is worse than hitting **RESET**. Be even more strict about following the **RESET** rules when it comes to turning the unit off.

IV. An Elephant Never Forgets, But You Didn't Buy An Elephant

No matter how hard you try, you'll never make a diskette last forever. They wear out. Whenever your disk drive's red light is on, a hard ceramic surface is rubbing on the diskette as it turns in the drive. Sooner or later, ceramic prevails over plastic and the diskette wears thin in spots. In fact, the program or data will actually rub off.

Fortunately, this usually takes a long time. How long depends primarily on how you treat the diskette. It also depends on how often you use it, whether or not your drive is properly adjusted, and whether or not you keep your fingers off the plastic surface beneath the diskette jacket. Never entrust your life, your love, your data, or your time to anything as fickle as a diskette. Not unless you back it up.

Backup, in computerdom, means making copies, especially of diskettes. Once you've made a backup copy, you can rest easy. The ACE **DOS** diskette has a program that copies diskettes.



When the original diskette goes bad, you'll have nothing to worry about. Make a copy of your backup and keep right on going. As long as everything is working properly, copies of copies of copies, etc., will all be identical to the original.

The second greatest joy in personal computerdom is to reach nonchalantly for your backup diskette when your original gives up the ghost. The first greatest joy is when the backup works. In numbers there is safety and reason to act blasé.

Backup. Backup. Backup. To fail to do so is folly. A blank diskette and the time it takes to make a copy are a bargain if you compare them to the cost of replacing the data and programs on a diskette that gets damaged. Backup. Backup. Backup.

From what you know about the impermanence of diskettes and the importance of making backup copies, you may find this hard to believe, but it's true. There are programs out there designed so that it's almost impossible to make copies of them.

You can't do it unless you happen to be a computer genius with plenty of free time on your hands. With some, you can try 'til doomsday and you'll never make a successful copy. You're more in a position of being used by the program than you are of being a program user.

Program manufacturers tend to regard all customers as potential thieves. They know you're going to try to backup their program, despite their best efforts to stop you, and that sooner or later you're going to succeed. Once you do, conceivably, you could start to hand out copies of the program like candy to everyone you meet, thereby depriving the manufacturers of deserved revenues.



Many companies ask you to sign a licensing agreement that specifically prohibits you from such dissemination of the program. These licensing agreements typically stop just short of requiring you to sign away your life, your house, and your first-born child. From the manufacturer's viewpoint, though, it's all perfectly reasonable, a simple matter of self-preservation. Typically, manufacturers won't even talk to you about any problems with programs unless they have your release on file. It's tough to be judged guilty until proven innocent, but the manufacturers have some legitimate

concerns, too.

Legitimate concerns or not, some of their programs are downright diabolical! You think you've made a perfectly good copy and then just after you've typed in your 10,000th character (or some other magic number), the program checks itself to see if it's a copy. This is one program function that the programmer has tested and retested so that it works perfectly every time. When the program discovers that it's a copy, your program, along with all your data, vanishes.

Fortunately, the ACE can't explode under program control. Some programmers will stop at nothing to punish you for making a copy of their program!

There is a way to fight back, though. You might not win the war, but you can launch a strong counterattack. What you need is a weapon, a program called a nibble copier. It's a cute name, but its purpose in life is noble. It's designed to copy uncopyable programs.

The problem is that a nibble copier can't take on everything the enemy throws against it. Some programs are vulnerable to it; others aren't. There are a number of nibble programs available, each capable of copying a different group of programs. By buying two or three nibblers, you'll have a pretty good arsenal to use against these villains. You still won't be able to copy every single program that you encounter, but you'll be successful a good percentage of the time.

Unfortunately, nibble copiers are expensive and take time and skill to use effectively, but there isn't any real alternative, short of refusing to buy uncopyable programs. A boycott isn't practical, however, since nearly every program of any value

has some sort of copy protection built into it. Until consumer protests become loud enough to force manufacturers to stop this practice, nibble copiers are your only hope for good program backup.

V. You Can't Fix It But You Can Make It Worse

It's easy to pop the lid off the ACE and get at the inner workings of the machine. This "pop-top" design, like anything else, has advantages and disadvantages.

The principal advantage is that you can add a wide variety of peripheral devices very quickly and easily. The principal disadvantage is that you also have easy access to all the other internal parts of the machine that you really should leave alone unless you know exactly what you're doing.

Be strong. Resist the temptation to touch things that don't pertain to connecting peripheral devices. A computer isn't like an electric lamp or a vacuum cleaner. You won't find any loose wires that need reconnecting or any fuses to replace. In fact, you can't service any of the components, so don't even consider it, no matter how handy you may be working on a stereo. It takes special equipment to test and repair the ACE. Your dealer is the one best equipped to do computer maintenance, so leave any repair work to the experts at your dealer's shop or a Franklin service center.



VI. Promises: Take Each With A Grain of Salt

One of the first things you discover when you decide to buy additional equipment or programs for your ACE is that there are a lot of people out there anxious to separate you from your money. There are a staggering number of accessories available. Anything you want you can find. But purported capabilities are often different from what the program or peripheral device will actually do. You've got to try before you buy.

Some people will try to sell you anything, whether it works or not, so be prepared for hype. Marketing departments and advertising agencies know that sizzle sells units. Slick, professional-looking packaging accompanies almost every product on the market today. Don't be deceived. Leather-like binding and gold letters do not a good program make.

In general, avoid buying when performance is based on a

promise, and be leery of products if their usefulness depends on developments expected "any day now." In this industry, availability promises are, at best, hopes and, at worst, outright lies. If something is guaranteed to appear in 30 days, go back in three months.

A reputable dealer will be happy to demonstrate equipment or programs that you're interested in. Just ask.

If you want to remain friends with your dealer, though, try to limit extensive evaluations to higher priced products. Don't expect hours of explanation and demonstration for every \$15 game program. However, if you're buying a \$300 word processing program, you have every right to insist on a complete trial run.

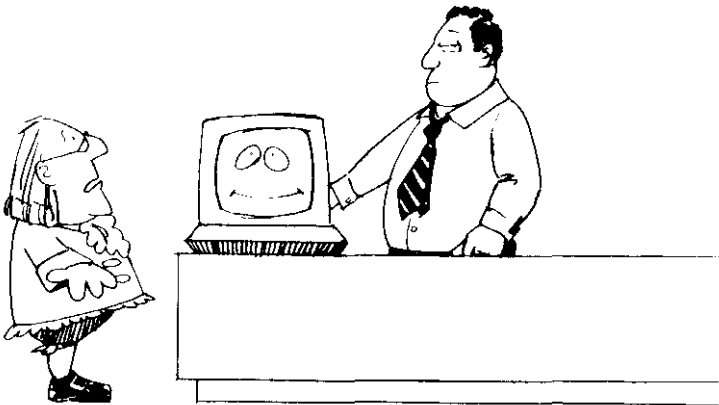
No dealer stocks every product that might pique your interest; it would take half a city warehouse district just to store them. Fortunately, there are evaluation methods other than a trial run at your dealer's showroom. Search out independent reviews of the product published in the personal computer magazines. Some of these publications are helpful for both product evaluations and general information on what's happening in the industry. Consider subscribing to several. It pays to keep up with developments in the personal computer field.

Your local computer club is another alternative. These are excellent sources of information about the quality of accessories. If several club members decide to buy the same item at the same time, they can often get a discount by placing an order together. Other times, companies simply offer special deals to club members. If the products are no good, you'll hear boos and catcalls when the programs or devices are mentioned. You might want to take what you hear as a form of counsel.

Although most advice should be taken with a grain of salt, in general, users know best.

VII. Be Patient – It's Not A Perfect World

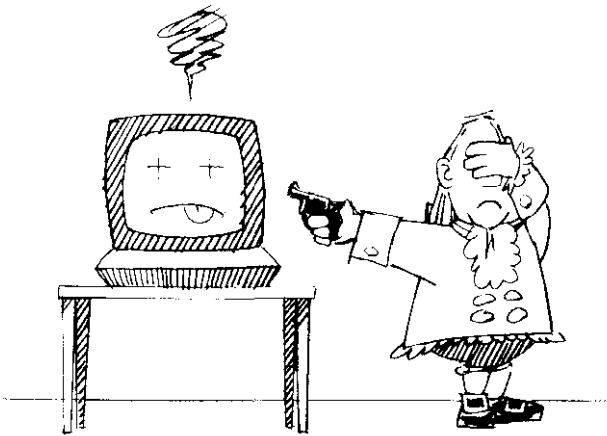
As a computer user you should be aware of the “I am not crazy” syndrome. Typically, a person takes a machine into the dealer for repair of a problem, only to find that the machine works fine. The repairperson then looks quizzically at the customer, who sputters, “I am not crazy. It really didn't work before!” This happens so often in the personal computer industry that most old hands know that the quickest way to fix a problem is to try to show it to someone else.



This is stretching the facts a bit. Most of the time, if the computer doesn't work at home, it won't work at the repair shop either. But the inexplicable does happen.

Occasionally a program won't work the first time you try it but will work perfectly the next day. You may try to print something and nothing happens. Try it again and, sure enough, it prints. These kinds of occurrences are rare, but something similar probably will happen to you sometime.

Problems of this sort aren't cause for concern unless they occur consistently with certain programs or peripherals. Persevere. Try whatever is giving you the problem several times. If it doesn't go away by itself, read the manual and try again. Fiddle with it. There may be an error in the manual. Try commands similar to the ones in the documentation. If you have another program that works perfectly, try running it to make sure the problem is the new program and not your machine. Look at the manual and try again. If that doesn't work, call on your dealer, an experienced friend, or the manufacturer.



A good title for this section might be . . .

The Territorial Imperatives Of The Trumpeter Swan



This title may seem a little out of place in a computer manual, but there's a reason for the incongruity. Unless you were fascinated by birds, you probably wouldn't turn to this section of the manual first. If it had been entitled *Let's Get Started*, the cliché generally used to describe sections of this sort, you might have begun reading here, ignoring all the introductory information.

Assuming that you've read what precedes this, you probably realize that, contrary to the popular maxim, what you don't know can hurt you. To get the most out of your ACE, you have to have the right information and the right attitude. You could probably assemble the components of the computer itself without either, but there probably would be no end to the frustration you'd endure trying to put together the larger picture of personal computerdom.

If you've turned here out of curiosity, turn back. Establish your own territorial imperatives by starting at the beginning and getting a feel for the terrain.

It's Out Of The Box! Now What?

First, make sure you've got everything that you're supposed to have. Here's a list:

- * the ACE 1000
- * this manual
- * an AC power cord (black, about six feet long)
- * a video cable (thin, approximately three feet long, with a male plug on each end)
- * a warranty card

And, if you bought a disk drive (or drives):

- * the disk drive (or drives)
- * the disk controller electronics card
- * the disk cable (a broad ribbon about two feet long)
- * a **DOS** diskette

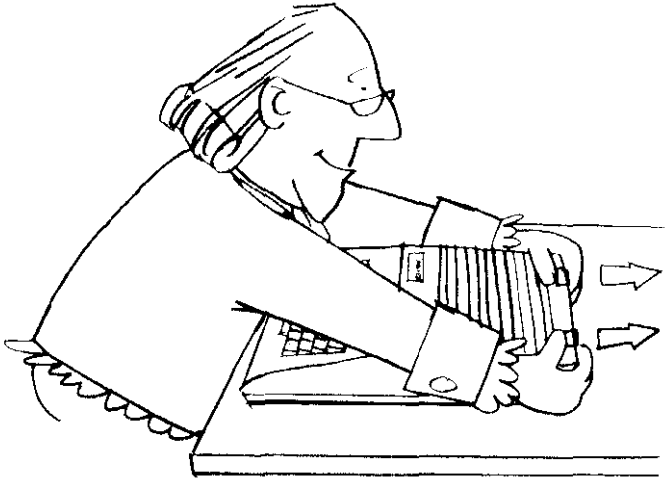
Getting It Ready

Make sure the power is off before you put anything together. If you haven't tried to connect the power cable yet, you should be OK. If you have connected it, disconnect it now.

Getting The Lid On And Off

The lid of the ACE 1000 is designed to pop off and snap back on. To get it off, stand in front of the unit facing the keyboard. Reach back and grasp the corners of the back of the lid, one corner in each hand. With your thumbs, push back the two latches that are about an inch and a half in from the corners, holding the lid in place. At the same time, lift the lid up. It should come right off.

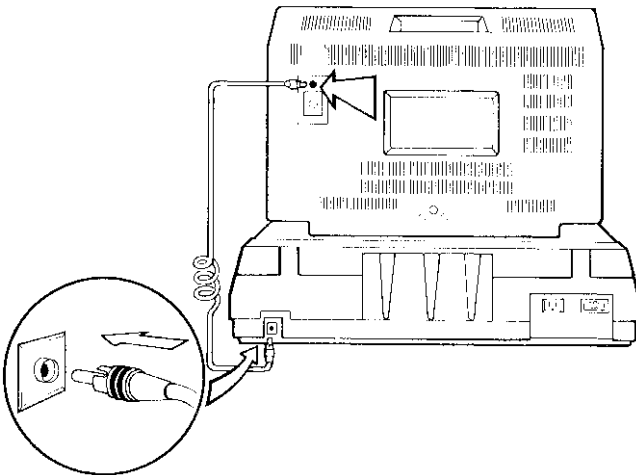
To get it back on, face the keyboard and look at the underside of the front of the lid. There's a piece of plastic screwed onto it,



forming a groove or slot at the front edge of the lid. Put the front of the lid down onto the main unit so that the lip of the case just above the keyboard slips into that groove or slot. Center the lid, lining it up with the bottom piece of the case, and then press the back of the lid down firmly. The two latches should snap into the indentations on the rear of the lid.

Attaching The Cables — The Video Cable

The video cable is the skinnier of the two round cables shipped with the ACE, the one with a little male jack on each end. Find the one inch square opening in the back right-hand side of the unit, near the bottom. In it is a small female connector. Plug either end of the cable into it.



The other end of the cable connects to your display device. If you're using a video monitor, the cable plugs right into a female connector similar to the one at the back of the ACE. If you're

using a TV with an RF modulator as a display device, things are slightly more complicated. To hook it up, follow the instructions provided with the modulator.

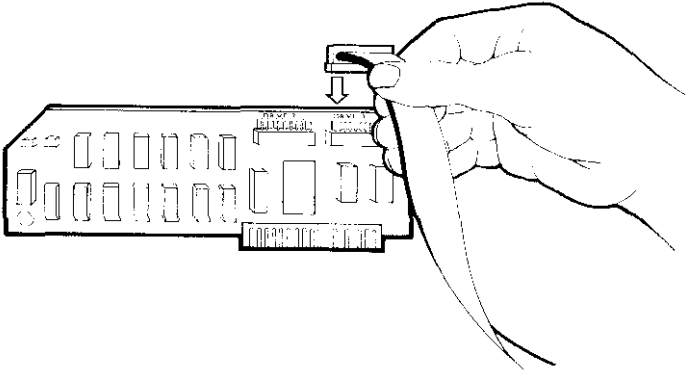
The Disk Drives

The disk drives are only slightly harder to connect than the video cable. Locate the disk controller card that came packaged with your disk drives. This is a rectangular piece of plastic eight inches by three inches, covered with electronic components. You have to plug this card into one of the connectors on the main electronics card inside the ACE, so pop the lid off.

Look inside, if you haven't already. On the floor of the main unit is a board strewn with electronic components. Toward the back, right up against the wall, you'll see eight black connectors. These are called slots. You'll see numbers printed in large white numerals on the main board behind the slots. These are the slot numbers, as you might have guessed. You're going to plug the disk controller card into slot 6. But not quite yet.

First, look at the disk drive cable, a wide, thin cable that looks a lot like a ribbon. Notice that one side is black (or possibly red). Now take a look at the controller card and find the two connectors that are marked DRIVE 1 and DRIVE 2.

With the colored (red or black) edge of the cable lined up underneath the "D" of the DRIVE 1 label, plug the cable into the two rows of pins on the connector. Be sure both these rows are plugged into the holes on the cable connector because if they're not, you may damage the ACE, the disk drive, or both.



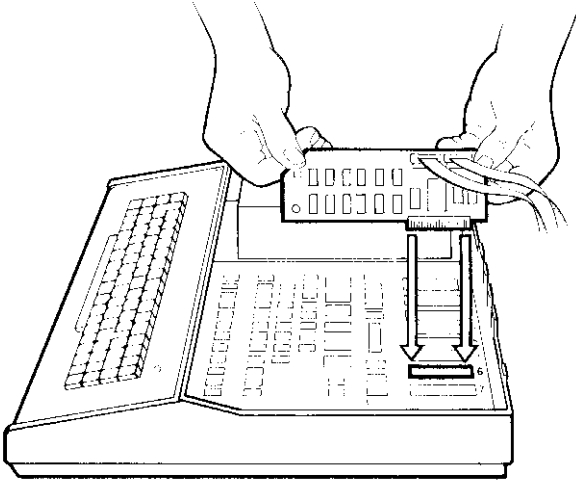
If you're at all uncertain about this procedure, ask someone who has done it before to help you.

If you have a second disk drive, attach its cable to the connector on the disk controller card labeled DRIVE 2, following the same procedure. If you have two drives, it doesn't particularly matter which drive is connected to which connector on the disk controller card, but you've got to remember which is drive 1 and which is drive 2 after you make the connection. The computer, the card, and the drive could care less about how you decide to number them as long as you make sure they're connected properly and remember which is which.

Now you're ready to plug the disk controller card into slot number 6. The card has a little plastic tab sticking out from one edge. This tab has many small gold-plated electrical contacts on it. They look something like tiny fingers.

Plug this tab into slot 6 so that the disk drive cable connectors are on the side of the card nearest the back of the main unit. Route the disk drive cables to the very back of the main

unit and out the back through one of the three vertical openings in the wall. Bending the cables won't hurt them but do make sure that the cables won't interfere with closing the lid.



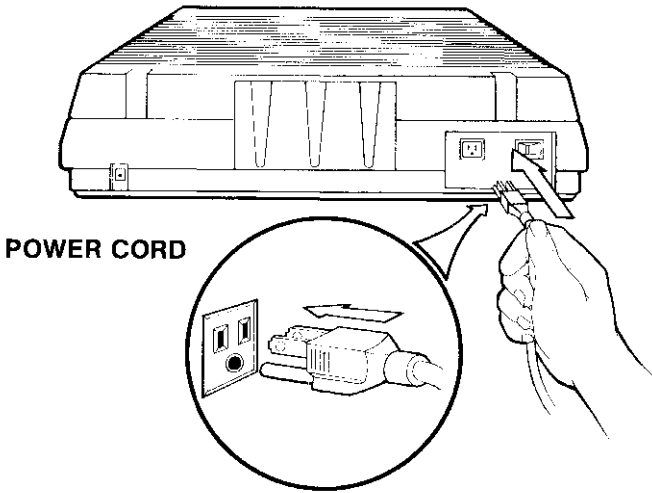
That's all there is to it.

Your disk drives are now installed.

The Power Cord

As soon as you make sure the ON/OFF switch is in the OFF position, you can plug in the AC power cord. It's the six-foot cord with a standard male plug for an electrical outlet on one end. In the back of the unit, on the left-hand side, there is a three-pronged connector. Plug the female end of the power cord into

the male connector on the ACE. Plug the other end into your wall socket and you're in business.



Turning It On – What It Should Do

Let's try the ACE without any diskettes first, just to see if it's working properly. Turn the unit on using the ON/OFF switch on the back. Here's what should happen:

1. The red light at the top right corner of the keyboard comes on.
2. The display screen clears.
3. On the screen you see:

(X.X is a version number which will change from time to time.)

4. You hear a beep.

5. The red light on the front of disk drive 1 lights up.
6. You hear strange noises from disk drive 1 for a few seconds and then a constant whirring sound that continues until the power is turned off.

At this point, the ACE is trying to read something from a diskette that isn't there. It will try to read the nonexistent diskette forever until it's told to stop. (The ACE can't tell if there's a diskette in a drive or not. It takes futility right in stride.) For the moment, just turn the machine back OFF.

If all this happens, the ACE is working fine. If it doesn't, don't despair.

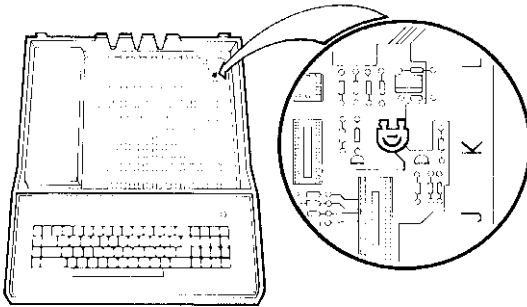
Why It Might Not Be Working

This checklist may help you find the problem.

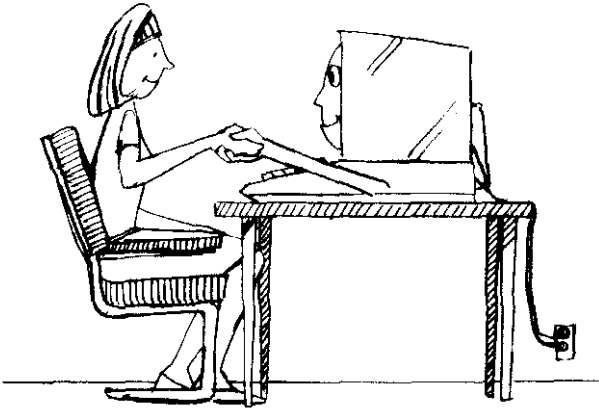
1. Is the light in the upper right corner of the keyboard on? If it's not, then the machine isn't getting power. Check the power cord connection, both to the ACE and to your electrical outlet.
2. Is your display screen doing something strange? Check your video cable connection. If you're using a modulator to a TV, check your modulator connection. If the screen is "rolling" or "breaking up," then you might have to adjust your display device. Play with the horizontal and vertical hold adjustments on the screen. If the display screen does nothing, check to see if it's turned on and plugged in.
3. If the disk drive light didn't go on, check the disk drive cable connection. Also make sure that the disk controller card is firmly plugged into slot 6 on the main electronics

board inside the unit.

4. Is the display screen very dim? You can adjust it a bit inside the ACE. In the back right-hand corner of the main electronics board, there's a small half moon disk with a slot in it. It's near the video connector. To adjust video brightness, turn this disk using a small screwdriver, preferably a plastic one. Be careful not to let your screwdriver touch anything else on the electronics board because you can damage the components.



If none of these measures solve the problem, then there might be something wrong with the unit. Don't attempt to fix it yourself. Take it back to your dealer or to a company authorized repair shop and have it checked.



The Communications Connection

Say Hello To Your Keyboard

Since the keyboard is going to be your primary means of communicating with the ACE, an introduction is in order. You're going to spend a lot of time pounding away at it, so you should know what it can do.

Lower Case and Alpha Lock

In most ways, the ACE keyboard works exactly like a standard typewriter. If you type characters without depressing the **SHIFT** key, you'll get lower case. When you simultaneously depress the **SHIFT** key and a character key, you'll get an upper case character.

Unlike a standard typewriter, however, the ACE has no SHIFT LOCK function. Instead, it has an ALPHA (short for alphabet) LOCK, which is slightly different.

In a typewriter's SHIFT LOCK mode, all the characters on the keyboard shift to either an upper case letter or the top character on the punctuation keys. In the ACE's ALPHA LOCK mode, only the alphabetic characters shift to upper case; the punctuation keys don't shift to their top characters.

The tiny red light on the **LOCK** key indicates that you're in ALPHA LOCK mode. To get out of ALPHA LOCK, just hit the **LOCK** key and the red light will go out. Touch the **LOCK** key again, and you're back in ALPHA LOCK.

Although the ability to generate lower case is a distinct advantage, there are times that it causes some problems. Certain programs don't handle lower case characters properly.

The reason for this program weakness is that they were written for use on an APPLE, which normally can't generate lower case characters. Some programs will actually modify every signal sent by the keyboard so that all characters become upper case. The result is that, using some programs, you can't type lower case characters with your ACE.

There's no easy solution to this problem. You can enter ALPHA LOCK mode and only type upper case characters. Or, if you can lock in upper case with one of the commands that the program provides, you can sometimes type lower case characters, strange as that may seem. You see, when the program is locked into upper case mode, it thinks that the keyboard can generate only upper case characters, so it won't bother to modify the keyboard characters.

If you still can't type lower case characters, then contact the program's manufacturer and ask for either a revised version of the program or a "patch" that you can apply to your version to allow lower case.

Numeric Pad/Editing Pad

The group of twelve keys on the right of the keyboard is called the numeric pad. It consists of the ten digits, the "greater than" sign (>), and the decimal point. If you do a lot of numerical calculations, you'll find using the numeric pad faster than using the numbers on the top row of keys.

Tapping the **PAUSE** key while pressing the **CTRL** key changes the numeric pad into an editing pad. Some word processing programs use this pad to let you place characters wherever you want them on the screen.

When you're using the editing pad, pressing the **2** key positions the next character you type one line down, while the **8** key places it one line up. The **4** and **6** keys move the character to be typed one space to the left or right, respectively. The **7** key places the next character at the beginning of the previous word. The **9** key places the character at the first letter of whatever would be the following word. You can move backwards and forwards one screen full of text at a time, using the **1** and **3** keys.

Pressing the **.** key deletes characters one at a time. To reshape a paragraph after making changes in it, press the **>** key to align text within the margins you've set. The **5** key displays other command options open to you.

Typing **CTRL PAUSE** lets you switch back and forth between the numeric pad and the editing pad functions. You can also use the editing keys by touching the **CTRL** key and a numeric key at the same time. Release the **CTRL** key and you're back to numeric functions. Try experimenting with your word processing program to see if it uses these editing keys. If it doesn't, consult the installation section of the manual that came with your program to see what program modifications you can make in order to use the editing pad.

Special Keys

Tapping the **BREAK** key will perform a "break" function in many programs. This usually means that the program aborts whatever it's doing. Whenever a program requires that you type **CTRL C**, you can hit **BREAK** instead.

The **PAUSE** key normally makes a program "pause" in whatever it's doing. For example, hitting **PAUSE** will stop the display on the screen indefinitely. Typing **PAUSE** again will restart the program.

The minus sign, the plus sign, and the asterisk are located to the left of the numeric pad for easy access when you're using programs that process financial data. Some of these programs are so popular that this section of the keyboard was designed with them in mind.

Keycode Differences

If you're interested in using programs that were designed for the APPL.E, you'll want to know about the six keys that must

be typed differently on the ACE. The following chart lists these keys and shows how they are typed on the APPLE and on the ACE. (The reference to Hex Code is for the benefit of those who are interested in programming.)

<u>APPLE KEYS</u>	<u>ACE KEYS</u>	<u>HEX CODE</u>
SHIFT M]	DD
SHIFT N	^	DE
SHIFT P	@	C0
CTRL SHIFT P	CTRL @	80
CTRL SHIFT M	**NOT AVAILABLE ON ACE	
CTRL SHIFT N	**NOT AVAILABLE ON ACE	

Any manual or documentation for a program specifically written for the Apple will refer to those APPLE keys at times. To type them on the ACE, type the corresponding key from the ACE column of the chart. Unfortunately, if the program requires you to type **CTRL SHIFT M** or **CTRL SHIFT N**, you can't type that key on the ACE. Once again, the solution to this problem is to contact the program manufacturer for help.

Just Where Is The Reset Switch?

The **RESET** switch on the ACE 1000 is designed to be easy to reach if you really must, but difficult to bump accidentally. It's also designed to be out of sight and hopefully out of mind. You'll find it in the front of the unit, underneath the keyboard, on the left-hand side.

The DOS Diskette

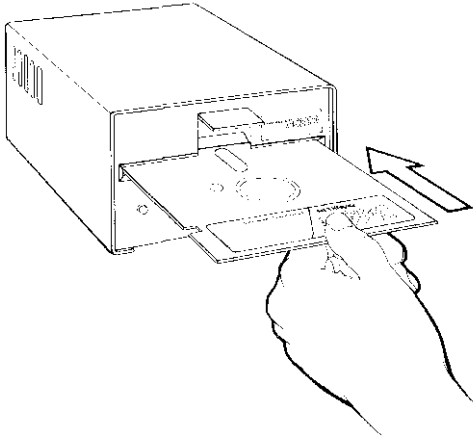
Franklin disk drive controller cards (optional on some models in the ACE 1000 line) come with a **DOS** diskette labeled “ACE XXXX DOS Diskette Version X.X.” The “X.X” is a version number that changes from time to time, but the diskette’s purpose remains the same: to allow you to use **DOS 3.3** applications programs on your ACE.

Most of the programs on the **DOS** diskette are so invisible that you can barely tell they’re functioning. If you were to look at a table of contents of the diskette, they wouldn’t even be listed. They constitute an operating system known as **DOS 3.3**, a disk operating system.

Besides providing you and the computer with control over the disk drives, the **DOS** diskette also contains a collection of programs that help you update, sort through, and shuffle the data files you create as you do word processing, financial management, data base management, or any other application of your ACE’s powers requiring **DOS 3.3**.

Giving The DOS Diskette A Boot Up

Before explaining everything that’s on the **DOS** diskette, let’s power up the ACE with this diskette in drive 1. Here’s how to put it in:



1. To open the disk drive door, put your finger under the latch in the middle of the front of the disk drive and lift up.
2. Grasp the diskette on the label side and hold it flat, label facing the ceiling.
3. Insert the diskette into the horizontal slot on the front of the disk drive. The diskette should go all the way in, but don't force it after it stops.
4. Push the disk drive door down until it snaps shut.

Now flip the power switch in the back of the ACE to turn it on. You'll see everything you saw when there was no diskette in the machine, plus some other things that happen because the ACE now has the **DOS** diskette to read. Here's what happens:

1. After the disk drive starts reading the diskette (noises), the screen clears for a split second. The character (a bracket called a "prompt") appears on the bottom left corner of the screen for a few seconds.

2. The screen then clears again and you see the title and version message mentioned earlier.
3. Then a second message appears. It tells you to wait while a program (one called **INTEGER BASIC**) goes from the diskette to the machine. This takes about fifteen seconds.
4. Finally, a third message appears. It lets you know that your system is all ready for you and provides programmers with some information that's of interest to them. The character appears in the bottom left corner of the screen, along with a square that blinks on and off. The square is called a "cursor" because it marks the place where the next character will appear.

What just happened is called a disk "boot," an absolutely essential and (happily) automatic process. When you see the , you know that the boot is complete.

In the early days of personal computers (a few years ago), you had to type in a series of complex commands to start up a program stored on a diskette. Nobody but computer people could remember the necessary commands. This upset a lot of folks, especially salespeople, who couldn't sell the machines because nobody could remember how to make the machine get programs from the diskette. That's why the automatic boot was created. Now you don't have to type anything in.

Now What?

Now that you've booted the **DOS** diskette, what can you do with your ACE? Not much . . . yet. The **DOS** diskette contains a good many programs, but they're not designed to apply the

computer's capabilities to any particular task. Instead, these programs perform more general functions in giving you the means to work with the applications programs you buy. In order to do word processing, financial management, data base management or use any other **DOS 3.3** application program, you'll need the programs on the **DOS** diskette.

But before this all gets too abstract, let's practice with the programs on the diskette just to get a feel for how they work.

How To Use The Programs On the DOS Diskette

You'll never learn how to use the programs on the **DOS** diskette unless you sit down at the computer and try each operation as it's described. You can learn by reading and doing but not by reading alone. Familiarizing yourself with the way these programs work is a hands-on process, not an intellectual exercise.

To get programs to work for you, you've got to call upon them and then tell them exactly what you want done. As long as you type accurately and follow the directions given here and on your display screen, the programs will usually obey your commands. Some will even make things easier by asking what you want them to do once you've called them from the diskette into the computer.

Although it may not be readily apparent (shortcuts can be tempting), there are good reasons to try things in the sequence in which they're presented here. An example may make this a little clearer.

Before you try anything else, you should make a backup of your **DOS** diskette. Since the program called **COPY** transcribes programs and files from one diskette to another, you'll need it to make this backup. When you're done, you'll have a perfect copy of all the programs on the **DOS** diskette, even **COPY** itself. You'll be able to put the original **DOS** diskette somewhere out of harm's way.

How To Use The Copy Program

To get the **COPY** program from the **DOS** diskette into the ACE, you've got to boot the diskette as described earlier. When the prompt and cursor appear, **DOS** is in the computer and you can call on **COPY**. To do so, type:

RUN COPY

and hit RETURN.

You've now given **DOS** your first command. The **COPY** program loads into the computer and starts running (operating in the computer). The monitor screen then clears and you see:

DISK COPY PROGRAM vX.X
FRANKLIN COMPUTER CORP

For a moment you see the message:

Loading . . .

on the screen. The **COPY** program is reading additional

information from the **DOS** diskette so that it can perform the copy operation.

After five to ten seconds, Loading . . . disappears and your screen looks like this:

```
DISK COPY PROGRAM vX.X
FRANKLIN COMPUTER CORP

ENTER SLOT AND DRIVE #
SOURCE DISK SLOT 6

PRESS RETURN FOR DEFAULT VALUE
```

When you use **COPY**, you have to indicate which slot on the main electronics board contains the disk controller card. Since the card is usually installed in slot 6, **COPY** assumes that's where the card is located. As long as it's there, all you have to do is press **RETURN**. This is known as the default value, the response you give when you don't specify something other than what the machine expects.

COPY then displays drive 1 on the screen to the right of the number 6 that's already there and asks you for the source drive number. This is the number of the disk drive holding the diskette that you want to copy, drive 1 or 2. Make a habit of copying from the diskette in drive 1. **COPY** assumes this is what you're doing, as you can tell by the 1 flashing on the screen. Hit **RETURN**, and **COPY** will be set up to copy whatever is on the diskette in drive 1.

COPY then displays:

```
Dest disk slot 6
```

Dest is an abbreviation of the word destination. The **COPY** program is just checking to make sure that you haven't connected more disk drives and put controller cards somewhere besides slot 6. Again, just hit **RETURN** unless you've done something other than what the manual has said to do.

Next you're prompted for the destination drive number. The **COPY** program wants to know where to send the copy that it's about to make. Your response to this prompt depends on whether you have one or two disk drives. If you have only one, you don't have any choice; you must type **1**. If you have two drives, you can type **1** or **2**, but **2** is preferable, as you can see from the screen. There's a good reason for this.

Copying data from one diskette to another poses a small problem if you have only one disk drive. You can't just read data from a diskette in one drive and write it onto a diskette in another drive.

With a single drive system, you have to read one diskette, take it out of the drive, insert the other diskette, and then write the data onto the second diskette. You have to do this a number of times. This diskette "swapping" is both time consuming and error prone. If you have a two drive system, you can simply specify drive 2 as the destination drive and leave the original diskette in drive 1. **COPY** can then do its job without any diskette swapping help from you.

If you have only one drive, don't worry about when to switch diskettes back and forth. It's always going to take time, but a little thought and care should keep you from making errors. On your display screen, you'll see that the **COPY** program guides you along the way by telling you to insert

either the source diskette (the original) or the destination diskette (the one receiving the copy) at the appropriate times. Just be careful to insert the diskette requested. If you insert the wrong one, you'll destroy data on one or both of the diskettes.

After entering the slot and drive numbers for the source and destination disk drives, you're ready to make a copy of your **DOS** diskette. If you're using a single disk drive system, what you read here in the text will differ slightly from what appears on the screen. The process of making a copy is the same, but the procedures you must follow are a bit different, so follow the steps as they appear on the screen. If you're using a two drive system, there should be no discrepancies between what you read here in the text and what the screen displays.

On the screen you should see the question phrase:

ok to start copy (y or n)?

Make absolutely sure that you have your original in the source drive and the blank diskette in the destination drive. After you've checked, type **[Y]** for yes. If you type **[N]** for no, you're again asked to enter the slot and drive numbers.

The message

copying disk

appears next, along with an arrow on the source disk line indicating that the **COPY** program is reading the source diskette, your **DOS** diskette. After about ten seconds, the message

formatting

appears and the arrow moves to the destination disk line. After another twenty seconds or so,

copying disk

appears again and remains on the screen as programs on the **DOS** diskette are alternately read (from the source diskette) and written (to the destination diskette, the blank diskette).

If an error occurs while the copy's being made, an error message appears. You have the choice of abandoning the process or starting over. If you choose to begin again (as you should), remove the diskettes from the disk drives and simply reinsert them, a procedure known as "reseating." Try to make a copy again.

If reseating the diskettes doesn't eliminate the problem, you may have a bad diskette. Replace it with a new one if it's the destination diskette. If it's the source diskette, you may have lost one or more files on the diskette. To try to recover them, look up the **FUD** utility program's **C**function (copy files) in the index and follow the procedure given there.

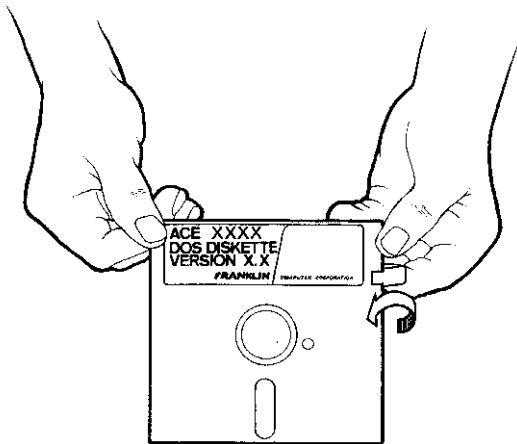
If all goes well, the message

copy finished

appears and you know that you now have an identical copy of the **DOS** diskette. Take it out of the drive and answer **Y** to the question that now appears on your screen about making additional copies. You're asked the same questions once again, and you should respond the same way you did the first time around.

When you have two or three backup copies of your **DOS** diskette, answer **[N]** to the prompt on the screen. Take your original out, put it in a safe place, and use only copies from now on. That way you'll have a good copy of the **DOS** diskette, despite any disaster that might befall the copy in use.

So that you don't run the risk of mistakenly using your **DOS** diskette for storing data, take one of the small gummed labels (usually called "write-protect labels") and cover the rectangular notch that's on the side of the diskette. A small piece of cellophane tape does the job just as well, but you should have no trouble finding labels since they're usually provided with blank diskettes.



If you should ever want to add or delete a program on a write-protected diskette, all you have to do is take off the label. As a matter of fact, you won't be able to make any changes unless you do.

If you want to make a copy of a diskette other than the **DOS** diskette, you first load the **COPY** program into the machine from the **DOS** diskette just as you did in the previous example

and then substitute whatever diskette you want to copy for the **DOS** diskette in the source drive. You see, once **COPY** is in the machine, the **DOS** diskette is no longer necessary to the copying process unless it happens to be the diskette you're copying, as it was in the example just given. As soon as you see **DISK COPY PROGRAM vX.X**, etc., you can insert whatever diskette you want to copy into the disk drive you declare as the source.

One last note: When you've finished making copies and have typed **[N]** in response to the prompt, the cursor and the **J** reappear on the screen. You can turn off your **ACE**, completely confident that you're not hurting any program or data. Everything you'll need again is on diskettes and retrievable.

The applications programs you use will tell you how to store your programs and data on diskettes so that you can turn off the machine without worrying about losing the program or what you've entered. With so many different applications programs on the market, it's impossible to come up with a general rule for when it's safe to turn the machine off that isn't contradicted by the peculiarities of at least one program. While using the **DOS** diskette, follow the guidelines mentioned in the previous paragraph.

How To Use DOS 3.3

When the **DOS** diskette is in drive 1 and you turn on the power, **DOS 3.3** — the program that enables you to use your disk drives — is pulled into the computer automatically. Only those aspects of the **DOS** program that are particularly useful to nonprogrammers are described here and in an abbreviated form at that. You won't find all the possible options and parameters

for every command.

For example, the command formats shown here will work provided that the disk controller card is plugged into slot 6 on the main electronics board and that you have a one or two drive system. If this isn't the case, you may have to provide some additional information when you type in a command.

At some point in the future you may want to learn **BASIC** programming, add a third disk drive, or make any one of a number of changes to your system. If this happens, your dealer can recommend books that describe **DOS 3.3** in detail. For now, follow along using your **ACE** and a copy of the **DOS** diskette so that you can get some practice with the procedures used to make **DOS** work for you.

First of all, the **DOS** program has to be in the computer before you can give it commands. Unfortunately, because it's such an automatic, invisible program, it's not always easy to determine whether or not **DOS** is running, i.e., in the machine. The simplest way to be sure is to boot the **DOS** diskette (insert the **DOS** diskette in drive 1 and turn the power on). As soon as you see either of the prompt characters] or >, you know that **DOS** is running, and you can enter **DOS** commands.

If you type a **DOS** command when **DOS** isn't running, the machine will probably beep and you'll see an error message on the screen. You've done no harm.

As long as you boot up with the **DOS** diskette, you can type commands in either upper case or lower case. To execute a **DOS** command once it's typed in, hit the **RETURN** key.

If you make a mistake and type an incorrect character, hit the

backspace key (the one with the arrow pointing left) in order to back up the cursor over your error. Correct it by typing in the appropriate character. If you make so many mistakes on a line that you want to forget the whole thing, cancel the command by holding down the **CTRL** key and typing the **X** key at the same time (called **CTRL X**). Then you can type the same command again or try a new one.

You're about ready to try the **DOS** commands, so make sure that you're using a backup copy of the **DOS** diskette and then boot up. The only way to get a feel for all this is to use it.

CATALOG (Displaying The Table Of Contents Of A Diskette)

As soon as you know you're running **DOS**, type:

CATALOG **RETURN**

You should see:

DISK VOLUME NNN

*A 004 HELLO

*B 050 INTBASIC

*B 050 FPBASIC

*B 003 CHAIN

*B 007 BOOT13

*B 027 FUD

*A 016 COPY
*B 003 COPY.OBJ
*B 006 DIAGNOSTICS
*B 004 PROMTEST.OBJ
*B 007 KYBDTEST.OBJ
*B 008 MEMTEST.OBJ
*B 004 PADLTEST.OBJ
*B 008 DISKTEST.OBJ
*B 004 HRESTEST.OBJ
*B 034 DIAGNOSTIC DATA-DON'T "RUN"
*A 003 PRETTY PATTERN

The first line of information presented is the message DISK VOLUME NNN, where NNN is the volume number of the diskette. On the **DOS** diskette, the volume number tells you which version of the diskette you happen to be using. The number changes as Franklin produces new versions.

Generally speaking, data and programs are stored on diskettes in files, with program names or titles used synonymously with the names of the files where they're stored. Since most commands oblige you to specify the name of some file as a part of the command procedure (this is the case with most **DOS** commands), you'll have to name any files you create using applications programs.

File names appear under the volume message, each on its own line. For example, on a typical file line such as *A 004 HELLO, **HELLO** is the name of a file, probably the first file on your **DOS** diskette.

There's a lot of other information here if you know what you're looking at. The asterisk at the beginning indicates that the file is "locked" and can't be altered. If the file is "unlocked," no asterisk appears in this space. This type of file can be altered. You'll find out how you can lock and unlock files a little later.

The letters after the asterisk indicate different categories of files, most of which are represented on the **DOS** diskette. You have to know what type a particular file is because each file type requires a different **DOS** command to get it from the diskette into the machine. Although you may never want or need to know the difference between **Floating Point BASIC**, **INTEGER BASIC**, and binary, the computer does notice which is which and categorizes program files accordingly. Some commands work with only certain categories of programs, other commands on other categories.

A indicates that the file is a program written in **Floating Point BASIC**. File type I is a program written in **INTEGER BASIC**. Type B files contain programs written in assembly language (B stands for binary). T indicates that the file is not a program file, but a text file containing some sort of textual information.

The last bit of information on the line before the file name is the size of the file. This is important because although diskettes have a lot of space, there are limits. And the computer has limits as well. In the example, 004 means that the file is four sectors long. Each sector can have up to 256 characters in it. 004 sectors means that the file containing the **HELLO** program consumes 256 times 4 sectors or 1024 characters of diskette space.

This isn't the exact size of the **HELLO** file, only an approximation. It really means that **HELLO** has more than 768 characters (256 x 3) but is less than or equal to 1024 characters

(256 x 4) in size. (Keep in mind that this is an example only. The **HELLO** file on your version of the **DOS** diskette may have more or less than four sectors). What this means to you is that **HELLO** is a relatively small program that takes up a small amount of space on the diskette and a small portion of the machine's memory, considering that the ACE has room for at least 64,000 characters.

Sometimes a diskette will contain more files than the screen can display. When this happens, **DOS** displays the first eighteen file names on the screen and then stops to let you see what they are. To see the next eighteen file names, hit the space bar. You can continue looking at each succeeding group of eighteen until you've seen the names of all the files on the diskette. If **DOS** is waiting for you to hit the space bar for the next group of files, you'll see no prompt character] or > at the end of the file list.

One more point about **CATALOG**. As long as you have **DOS** running in the computer, you can use this command to find out what files you have on any diskette created using **DOS 3.3** (as opposed to diskettes created using **CP/M**). You simply boot the **DOS** diskette, remove it from the drive when you see the prompt and the cursor, and then insert another diskette that runs on **DOS 3.3**. You then type:

CATALOG

and follow the steps given in this section.

RUN (Running A BASIC Program)

You can run a type A or type I program using the **RUN** command. **HELLO** has a file type of A, so you can use the **RUN** command to

make it work. Type:

```
RUN HELLO RETURN
```

The red light on the disk drive comes on, and you should hear noises from the disk drive. On the display screen, you see the same sequence of messages that appeared when you booted up the **DOS** diskette. **HELLO** is the name of the program that displays those messages on the screen during the boot up process.

What you've just done is to run a program manually that runs automatically when you boot the **DOS** diskette. When you see the] prompt character at the bottom of the screen, **HELLO** has finished running and you can continue. Or, if you want to take a break, you can shut off the machine.

Don't worry about hurting anything by trying to use the **RUN** command for a program file that isn't type A or I. **DOS** will recognize the mistake and simply display the error message: FILE TYPE MISMATCH. No harm is done. You just have to try again.

BRUN (Running A Binary Program)

Use the **BRUN** command to run a program with file type B. You use it exactly the same way that you use **RUN**. Again, don't worry about hurting anything by trying to **BRUN** a file that is not type B. You'll just get the FILE TYPE MISMATCH error message.

To try out the **BRUN** command, type:

```
BRUN FUD RETURN
```

This will run the program **FUD**, a type B program written in binary, a machine or assembly language. When **FUD** runs, it displays a menu of possible operations. Since **FUD** is explained later, just hit the **Q** key to leave the **FUD** program and return to **DOS**.

LOCK (Protecting A File)

The **LOCK** command will “protect” a file so that it can’t be changed in any way. Once it’s locked, you can’t delete a file from the diskette, change its name, change any data on it, or add more data to it. It’s protected from modification. You *are* permitted to read a locked file, though. For example, you can run a program in a file that’s locked (as you did with **HELLO**) or copy the file to another diskette, but if you try to modify a locked file, you’ll get the error message **FILE LOCKED**.

To lock a file, you type the command **LOCK** and the name of the file you want to protect on the same line and then hit **RETURN**. From that point on, an asterisk will appear at the beginning of the line in which the name appears in the catalog of the diskette. The file is locked.

All of the files on your **DOS** diskette are locked, so it serves no purpose to try the command at this point. It doesn’t hurt anything to lock a file that is already locked, though. If you want, you can type:

LOCK HELLO

You'll hear the disk drive whir and see its red light come on as the **LOCK** command locks the **HELLO** file on the diskette. Nothing actually changes, but no harm is done either. Just look at it as computer calisthenics.

UNLOCK (Removing The Protection From A File)

If you want to change a locked file, you have to unlock it first. Type:

UNLOCK

and the name of the file on the same line and then hit the key. Try it on **HELLO**. Type:

UNLOCK HELLO

Now type:

CATALOG

You'll see **HELLO** has no asterisk in front of it. It's unlocked. Lock it again by typing:

LOCK HELLO

RENAME (Renaming A File)

At some point, you may want to change the name of a file. You can use the **RENAME** command to do this. For example, let's rename **HELLO**. Suppose you want to call it **GOODBYE**. Type:

```
RENAME HELLO, GOODBYE [RETURN]
```

What happened? If you followed the instructions, you should have gotten the error message **FILE LOCKED**. Because the file was locked, **DOS** couldn't perform the **RENAME** function.

Do a **CATALOG** (i.e., type **CATALOG [RETURN]**) to check this. **HELLO** should still be named **HELLO**. Unlock **HELLO** as explained earlier under the **UNLOCK** command and then try the **RENAME** command again. This time it should work. Do a **CATALOG** to check. **HELLO** should now be named **GOODBYE**.

Before going on, change it back to **HELLO** by typing:

```
RENAME GOODBYE, HELLO [RETURN]
```

If you don't do this, the diskette will not boot up correctly later. While you're at it, you should also lock it again by typing:

```
LOCK HELLO [RETURN]
```

DELETE (Deleting A File)

The **DELETE** command removes a file from a diskette. Once you delete a file, you can't get it back. It's gone forever. To delete a

file, type **DELETE** and the name of the file on the same line and then hit **RETURN**. You can't delete a locked file; if you try, you'll get an error message.

Since all of the files on the **DOS** diskette are still necessary, there's no way you can practice using the **DELETE** command. Here's a hypothetical example of how you might use it. Let's say you've been using a word processing program and you've created files of text that you've since revised to the point where there's no reason to keep the originals. With the **DOS** diskette in drive 2, you could turn the power on, boot the **DOS** diskette, and then, as soon as you saw the prompt and the cursor, you could use the **DELETE** command with one of the file names from the diskette in drive 2. You'd have to put a comma after the file name and **D2** after the comma, but that's all that would be dissimilar from what you've done so far.

INIT (Initializing A Diskette For Use)

Let's say you're using a personal financial management program, entering all the information relevant to your income taxes for the last five years. After you've recorded everything for year one, you run out of space on your diskette, so you put a blank one in the disk drive. You continue to enter data as usual and then, as the computer's memory gets full and it starts to transfer information onto the diskette, an error message appears on your display screen.

What went wrong?

Nothing, really, except that blank diskettes are so empty that the computer can't use them until it puts a sort of organizational

plan on them. This way the computer can easily find out what files are on the diskette, where they're located, and where there's unused space. Putting this organizational scheme on a diskette is called initializing or formatting.

When you do it, you'll also be putting a version of **DOS**, the disk operating system, on the diskette. There's an advantage to this. No matter what you store on it, you'll be able to boot the diskette and use **DOS** commands any time without involving the **DOS** diskette.

You should know a few more things before you try the **INIT** command. The version of **DOS** that **INIT** puts on the blank diskette is generally called a "slave" **DOS**. Your diskette will work fine on your computer or on any other ACE or ACE-compatible machine that has exactly the same memory size as yours. But if you intend to use the diskette on another machine with a different memory capacity, you'll have to do one more thing to the diskette — make it a "master" using **FUD's M** command, which is described a little later.

You might be wondering why you didn't have to initialize or format the blank diskette you used to make backups of your **DOS** diskette. If you think back to what you saw on the display screen, you may remember seeing formatting while **COPY** was at work. The **COPY** program automatically formats blank diskettes before it transfers all the programs and data.

To initialize a diskette, take the **DOS** diskette out of drive 1 and insert a new diskette. Make sure that you're using a blank diskette or one that you don't need when you try this **INIT** example. If you forget and leave the **DOS** diskette in, you'll erase all of the programs and data on it.

Now type:

INIT HELLO RETURN

The ACE will begin initializing the diskette in drive 1. This will take ten to fifteen seconds. You'll know it's done when you get the prompt character on the next line (either the] or the >). The diskette is now initialized, but you're not quite done yet.

When you initialize a diskette you can specify that any type A program be run automatically when the diskette is booted. Once you start using application programs, you'll find type A programs that you'll want to run automatically because you can save the time and typing you'd have to spend calling upon the program and getting it to run. The **DOS** diskette, for example, was initialized with the **HELLO** program so that every time you boot it, the **HELLO** program automatically displays:

ACE XXXX
DOS Diskette

Please wait a few seconds while I
load INTEGER BASIC.

Okay, everything is ready to go!
You are now in Floating Point Basic.

You haven't gotten your new diskette to this stage yet. Even though you've just typed INIT HELLO, the **HELLO** program won't run and display the version messages when you boot your new diskette. You'll only see a prompt or two and a blinking cursor. You have an operating system on there and the word HELLO will appear if you type CATALOG and hit RETURN,

but **HELLO** won't run. The program itself isn't on the diskette yet.

There are two ways you can get it there. You can use the **FUD** program to copy the **HELLO** program from the **DOS** diskette onto the new diskette. For now, though, try this second method.

Put the **DOS** diskette in drive 1 and type:

LOAD HELLO

The **HELLO** program will go from the diskette into the computer's memory. Remove the **DOS** diskette and put the new diskette back in drive 1. Type:

SAVE HELLO

The **HELLO** program is now on the new diskette. Boot it and you'll see the usual version message (with one modification that isn't particularly relevant for a storage diskette).

If you now type **CATALOG** , you'll find the **HELLO** program listed. It'll be the only one listed.

This manual won't explain the operation of the **LOAD** and **SAVE** commands because such details are useful only to a **BASIC** programmer. If you ever want this information, you can find it in any **BASIC** programming book.

You probably won't want to include **HELLO** as a part of the **INIT** process on every diskette, but you must include some type A file. Although you have a very limited repertoire to draw from at this point, you can follow the same procedure, substituting

PRETTY PATTERN for **HELLO**. This is one way to spruce up your otherwise dreary financial records. After **PRETTY PATTERN** has filled your screen, tap the space bar a few times. The program itself will appear on the screen along with the cursor and prompt. Ignore the program gibberish and start back on your financial records.

A Few Generalizations About File Names

File names can't be more than thirty characters long. The first character must be a letter, and none of the characters in the name can be commas, because they're used to separate file names from other things such as disk drive numbers. The commas you see in the examples must appear in the command exactly as they're shown. When **D1** and **D2** appear, they must be typed exactly as they're shown too, because they tell **DOS** which drive to go to in order to perform the command.

There's one more thing to remember about disk drive numbers. As in the examples shown, you can enter commands without specifying a drive number. In those cases, **DOS** automatically uses what's called the default drive number, the number of the last drive mentioned in a **DOS** command. If you're going to execute a number of commands on one particular drive, you can type the drive number once as a part of a command and forget about it until you want to perform a command on a different drive.

How To Use FUD

The letters **FUD** stand for Franklin Utility for Diskettes. Some of the routines in this program may seem similar to those you saw when you were experimenting with **DOS** commands. **FUD's** commands provide a little more flexibility and versatility than do most **DOS** commands. Both are useful, however.

FUD lets you remove old, unused files, make backup copies of files, determine if your files are all still readable, and perform other mundane but essential functions.

Imagine what your filing cabinet would look like if you couldn't throw things away or at least move them around. That's what your diskettes would be like without **FUD**.

Although the step by step cookbook approach was useful for familiarizing you with the **ACE** and **DOS**, it does get a bit tiresome. You won't be given such specific directions for using **FUD**. Still, you should practice and experiment as you read. The **FUD** program itself does a good job of keeping you on track.

Basically, **FUD** has three groups of functions: diskette management, file management, and quitting. (You shouldn't do the latter, at least not yet.)

To run the **FUD** program, boot the **DOS** diskette. (If your machine is on, turn it off, put the **DOS** diskette in drive 1, and then turn the computer back on.) When you see the] prompt at the bottom of the screen, the boot is complete. As a matter of fact, you can call upon **FUD** any time you see the] prompt. Type:

BRUN FUD

After several seconds, you should see the **FUD** title and version number on the screen:

FUD version X.X
by
Franklin Computer Corp

Below the version and title, you should see:

MAIN MENU

S-how files on diskette

C-opy files

D-elete files

L-ock files

U-nlock files

V-erify files

M-ake MASTER diskette

F-ormat diskette

Q-uit

Please press letter for
function desired:

This is the main menu, the list of options open to you. Remember the name, because it's where you return for a fresh start and you'll hear it mentioned often.

The screen line where the words MAIN MENU now appear always lets you know what you're doing. As you use each of the

commands, what you see displayed on this line changes so that it corresponds to the function you're performing.

FUD commands work differently from **DOS** commands. To specify which operation you'd like to perform, all you have to do is type the letter just to the left of the hyphen in each of the items that appears on the main menu. You don't even have to hit **RETURN** in order to begin the procedure.

Let's take a look at what each of those options does.

Diskette Management Commands

FUD has two diskette management commands: The **F** command, which formats diskettes, and the **M** command, which transforms formatted or initialized diskettes into "masters." Each of these commands has an impact on the entire diskette.

Selection F (Format A Diskette)

As you probably remember from the description of the **INIT** command in the **DOS** section, every diskette must be initialized or formatted before you can use it. The **FUD F** command is almost the same as the **DOS INIT** command, except that **INIT** obliges you to put at least a file name on the diskette, like it or not. If you follow through on the entire **INIT** procedure, you'll have a type A program such as **HELLO** on there, too. The **F** command formats the diskette and transfers a slave version of **DOS** to it. This provides an advantage over the **INIT** command, if

you're space conscious. For storing text files, you'll probably want as little else on the diskette as possible.

There's another advantage to the **F** command. If you happened to be using **FUD** for copying files and you discovered that you needed a formatted diskette, with only **INIT** to call upon, you'd have to leave **FUD**, put a blank diskette in the drive, go through the initialization process, get back into **FUD**, type **C** for copying files, etc. The **F** command eliminates most of that.

When you select **F** from the main menu, the screen displays:

FORMAT

What volume number should I put on the
diskette, press RETURN for normal 254:

The volume number is an identification number placed on each diskette when it's formatted. Although the number can range from 1 to 254, most people just use 254 all the time, since few users need a sophisticated reference system for their diskettes.

Now the computer asks:

Which drive should I format?
(press RETURN for drive #1)

FUD wants to know where to find the diskette that it's supposed to format. Press **RETURN** for drive number 1, or press **2** for drive number 2. You can press your selection before you put a blank diskette in the drive.

This message now appears on your screen:

Please place the diskette to be formatted into drive # and press RETURN to continue. If you wish to quit, then press BREAK.

If you've changed your mind about formatting this diskette, you can hit the **BREAK** key and you'll see **FUD's** main menu again. Otherwise, be sure that you put the correct diskette into the drive specified. Formatting a diskette destroys all information on it. Be sure there's nothing important on the diskette!

When you're sure, press **RETURN**. The formatting operation takes about a minute. You see this message on the screen below the previous message:

Formatting diskette, relax a minute

You get this message when the format operation is finished:

The diskette is completed. It is formatted and contains a slave DOS.
Press RETURN to format another disk,
M to make a master disk
or BREAK to exit.

The diskette now can be used to store programs or data. Press **RETURN** to format another diskette or **BREAK** to return to **FUD's** main menu.

If the message above didn't appear, you probably got an error message such as:

The disk is write protected. Please remove the write protect tab. Press RETURN to try again, or BREAK to quit.

The error message is self-explanatory, but before you take that paper or plastic covering from the notch, try to remember what's on that diskette. Better yet, hit **BREAK** to return to the main menu and then type **S** to see what's on it. Remember that formatting will destroy whatever is on the diskette.

If you don't want to format another diskette, then press **BREAK** to return to the **FUD** main menu. Or you can format another diskette by putting it into the drive and pressing **RETURN**.

If you didn't get a write-protect error, then you may have gotten this message:

Sorry but I had an I/O error. Try reseating the diskette. Press RETURN to try again, or BREAK to quit.

I/O is a catch-all phrase. Sometimes it even appears instead of the write-protect message when the diskette is write-protected. It means "something went wrong." Exactly what went wrong can be hard to determine. You may be able to correct the problem by reseating the diskette. Make sure the red light is off, then pull the diskette out, put it back in, and close the door again. Strange as it may seem, this simple reseating usually works. If it doesn't or if you'd just as soon start over anyway, press **BREAK** to go back to **FUD's** main menu.

Selection M (Make A Master Diskette)

The **M** command creates a diskette with a rather intelligent version of **DOS**, the operating system, on it.

Why would you want to do that? After all, when you **INIT** a blank diskette or use **FUD's F** command, you put a version of **DOS** on it, don't you? Well, yes, but a diskette created with the **INIT** command can't be used on other ACE-compatible machines that have a memory different from your ACE. That means that the machine you have at home must be identical to the one you have at school, at the office, or at Aunt Minerva's if you want to use your programs and files different places, as many people do.

With a master **DOS** on the diskette, you'll be able to use it on ACE-compatible machines.

To create this kind of diskette, you need a copy of the **DOS** diskette that came with the ACE and another diskette that has already been initialized using the **DOS INIT** command or the **FUD F** command but is otherwise blank.

The first thing that the **M** command asks you is:

What drive should I get the DOS image
from? (Press RETURN for drive #1)

The program wants to know which drive contains the **DOS** diskette. If it's in drive 1, just press **RETURN**. If it's in drive 2, type **2**. The screen then shows this message:

Please wait a minute while I load the DOS image into memory.

The drive containing your **DOS** diskette starts to spin. In a few seconds it stops and this message appears:

What file should I run at boot time?

This message may appear instead:

I'm sorry, but there isn't a DOS image on that diskette.

Press BREAK to exit, or put a MASTER disk into drive #x and press RETURN to try again.

This probably means that you don't have the **DOS** diskette in the same disk drive as the one you specified. Put the **DOS** diskette into drive number x (look at the message on the screen to tell what x is) and press **RETURN**.

Let's assume that everything worked the way it should and you got the "What file should I run..." message. Just type in the name of the file you want to run (**HELLO** is commonly used) and press **RETURN**. You then see this message:

At boot up time, should I:

RUN the file (press RETURN),

BRUN the file (press B), or

EXEC the file (press E)?

If the program that you named in your last answer is written in **BASIC** as **HELLO** is, then you want to **RUN** it, so press **RETURN**. If you specified a binary file, then press **B**. **E** is useful only if you want a text file to run when you boot the diskette. For now, pressing **RETURN** is the best response.

After you answer that question, you see:

What drive am I writing a MASTER to?
(Press RETURN for drive #1)

FUD wants to know which disk drive holds the initialized or formatted diskette that will get the master **DOS**. If you haven't inserted the diskette yet, don't worry. **FUD** won't start to do anything at this point. Go ahead and hit the **RETURN** key or a **2** for the destination disk drive. The next message reads:

Please insert the disk to be mastered
into drive #x. Press RETURN to continue

The number x is the number of the drive that you said contains the initialized diskette that's going to get the master **DOS**. Insert the diskette into the specified drive, if you haven't already done so, and then press **RETURN**. If you change your mind about putting a master **DOS** on the diskette at this point, press the **BREAK** key to quit.

If you press **RETURN**, the disk drive comes on for a few seconds, then the screen prints:

Writing DOS to disk, Relax a minute

And then:

The diskette is now finished. If you wish to MASTER another diskette, press RETURN. If you wish to exit this program, then press BREAK.

Congratulations. You've just put a master **DOS** on a diskette. If you press **RETURN**, you can do the same thing to another formatted or initialized diskette. Pressing the **BREAK** key at this point will return you to **FUD's** main menu.

If for some reason the machine can't write **DOS** onto the diskette, an error message appears. If the destination diskette is write-protected (the little notch on the side of the diskette covered), then you get this error message:

The disk is write protected! Please remove the write protect tab and try again.

Be careful. The diskette was probably write-protected for a reason. Don't take off the write-protect tab if the diskette has valuable information on it.

The other possible error message is:

Sorry, but I can't write to this disk.
It probably needs to be formatted.

Usually **FUD** is correct in its diagnosis, but sometimes this message appears even if the diskette is properly initialized or formatted. To make sure you haven't just got a bad diskette, reformat it and try putting a master **DOS** on it again.

After either error, this message appears on your screen:

At this point, you have two options:

BREAK to exit this program, or...
RETURN to try again

If the problem can be fixed immediately, as by removing a write-protect tab, then you'll probably want to press **RETURN** to try again. But if the error was caused by something like an unformatted diskette, you'll have to stop and correct the problem before you can put a master **DOS** onto that diskette. Pressing **BREAK** will return you to **FUD's** main menu.

File Management Commands

FUD has six file management commands: **S** (show the files on diskette), **C** (copy files), **D** (delete files), **L** (lock files), **U** (unlock files), and **V** (verify files). Each command can operate on either single files or groups of files.

As you probably recall, whenever programs or data are stored on a diskette, each program or collection of data is stored in a file that has its own particular name. Programs that you buy off the shelf come with file names already assigned. When you create your own files, you must create the names.

From working your way through the **DOS** section, you already know the conventions for specifying file names in commands, so operating on single file names should present no problem. But since **FUD** lets you operate on more than one file at a time, you'll need some special naming procedures if, for example, you want to lock all the files on a diskette.

How can you specify a group of file names in a single command? It's not hard, really. This is where you'll notice the versatility of the **FUD** program.

Whenever a file management function requires a file name, you can enter a single file name so that the command operates on that file alone, or you can enter what's known as an ambiguous file name so that the command operates on a group of files. With carefully chosen ambiguous file names, more than one file can meet the specifications of the name you've created.

Now for the conventions. A file name is ambiguous if it has either of the characters **?** or **=** in it. These two characters have special meanings to **FUD**. The **?** character means "any character is acceptable." Consider the following purely hypothetical examples. The file name **A?E** refers to all file names three characters long beginning with **A** and ending with **E**. It doesn't matter what the middle character is. The file names **ACE**, **APE**, and **AXE** all fit the description **A?E**. The file names **ICE** and **APPLE** don't fit this description, the first because its first character is not an **A**, and the second because it has more than one character between the **A** and the **E**.

The **=** character is a little trickier. It means "any string of characters is OK." For example, a file name of **A=** means any file name starting with the letter **A**. **ACE**, **APPLE**, and

ANTAGONISTIC all fit that description. The hypothetical file name **BLARF=STUFF** refers to any file which starts with the characters **BLARF** and ends with the characters **STUFF**. **BLARF=STUFF** matches files like **BLARFINGSTUFF** or **BLARF ME OUT STUFF** or even **BLARFSTUFF**, in which the = matches no characters at all.

Any number of ?s and =s can be used in a file name, as long as you can keep straight what the thing you've created really means. For example, **A?C=F=D** means (take a deep breath): any file name starting with **A** whose third letter is **C**, whose last letter is **D**, and which contains an **F** somewhere between the **C** and the **D**. As you can see, ambiguous names can become very confusing very quickly. What's not so obvious is the way they can get you into trouble.

It's possible to create an ambiguous file name that embraces file names you didn't intend to include in the operation of the command. For example, by using an overly ambiguous name in a delete command (a command that deletes files), you could eliminate all the files on the diskette. Every file name might match your ambiguous description.

FUD tries to protect you against such calamities. Whenever you specify an ambiguous file name, **FUD** poses the question: "Ask before processing each file?" before it begins to execute the command. If you answer **N**, **FUD** shows you the file names as they are processed, but you won't be able to prevent the command from being performed on each file. It's a good idea to say **Y** to **FUD**. Then, if a file shows up that you don't want to be affected by the command, you can exclude that file from processing.

If you answer **Y**, a question mark will appear to the right of

each file name included in the ambiguous file name you've specified. Three options are open to you: press **Y** and the command will be performed on that file, press **N** and the command will not be performed on that file, or press **BREAK**.

There may be times when using **FUD** that you need to specify a particular drive number. As you recall from the **DOS** section, the default drive number is usually the last specified drive number. So, for example, if you last performed a command on files stored on a diskette in drive 1 and now you want to do something on drive 2, you have to specify the new drive number along with the file name. You specify a drive number by typing either ,D1 or ,D2 after the file name to specify either drive 1 or drive 2.

Selection S (Show The Files On Diskette)

This command allows you to see what files are on the diskette. It's similar to the **CATALOG** command described in the **DOS** section, except that it lets you restrict the list of file names displayed to a specific file or a specific group of files by using ambiguous file names. Limiting what you see may seem to defeat the purpose of a catalog, but there may be times when you want to see how much space on the diskette a particular file or group of files consumes. **FUD's S** command does this and gives you the file type as well.

When you select **S**, the screen clears, then displays this

message:

 Show diskette catalog
 which file(s)?
 RETURN for all files
 BREAK to abort
 ESC for help
 Defaults: slot 6, drive 1, volume 0
 :

Start by pressing **RETURN**. Disk drive 1 should start up, since you haven't specified anything out of the ordinary. The names of the files on the diskette in drive 1 then appear on the screen. The information shown about each file is the same information that is shown when you type CATALOG **RETURN**.

Instead of hitting **RETURN** in response to what's displayed on the screen, you can type a file name or an ambiguous file name instead. (Here come the ifs!) If you type a file name, **FUD** will display only that file in the catalog display. If you type an ambiguous file name, **FUD** will display all files on the diskette that match that description. If the file name is not on the diskette or if no files match the ambiguous file name description, you'll get the message:

no files found.

If **FUD** must display more files than will fit on the screen, the flashing cursor appears. Press any key except **BREAK** to have the additional files displayed, or press the **BREAK** key to stop the file display.

After all files are displayed, the total of all their file sizes is shown, along with the number of unused 256 character sectors

remaining on the diskette. The question

all done?

appears. If you answer **[N]**, you're again asked which files to display. If you answer **[Y]**, the main menu of **FUD** appears on the screen.

Selection D (Delete Files)

The **D** command in **FUD** performs the same function as the **DELETE** command described in the **DOS** section, except that with **D**, you can specify ambiguous file names. When the main menu is displayed on the screen, all you have to do is type **[D]** and a new display appears, giving you your options. You answer it the same way as you do when you want to display files. This time, however, you're specifying the files to be deleted.

The **D** command will not delete locked files. If you try to do that, you'll get an error message.

WARNING: Be particularly careful when using this command with ambiguous file names. It's easy (and excruciatingly painful) to wipe out entire diskettes this way. You'll have a chance to decide whether you want the files displayed individually so that you can decide in each case if you really do want to delete the file. **FUD** tries to take good care of you and your files.

Selection L (Lock Files)

Selection U (Unlock Files)

The **L** files and **U** files commands perform the same function on files as the **LOCK** and **UNLOCK** commands described in the **DOS** section. The **FUD** versions let you use ambiguous file names, however.

Let's say that in using a word processing program you've created a number of files on a diskette. You're satisfied with the text in each file and you want to be sure that you don't accidentally delete one of them. To lock them, you put the **DOS** diskette in drive 1, and boot up.

Type:

BRUN FUD

Next use the **S** command, specifying drive 2 where you have your files on diskette. Then create an ambiguous file name to cover all the files you want to lock. You then type to go back to the main menu, and to begin the locking procedure. From that point on you simply follow the directions displayed on the screen.

Selection V (Verify Files)

The **v** command checks whatever files you select to make sure that they're still readable. If your files have been damaged, the **V** command can be the bearer of the bad tidings. If the **FUD**

program can read all your files without any errors occurring, you can be fairly confident that all the data is still in good shape.

Use the command the same way you use the other commands on **FUD's** main menu; respond to the questions and options as they're displayed on the screen.

Selection C (Copy Files)

The **c** function (copy files) lets you copy a file or group of files from one diskette to another. As with all the rest of the functions on **FUD's** main menu, you'll be guided every step of the way.

You're first asked which files you want to copy. If you enter a single file name, only that file is copied. If you specify an ambiguous file name, all files that match the ambiguous description are copied.

You're then asked which drive you want to copy the file or files to. If you want to change the drive number from the one that the screen displays, simply enter the appropriate number (e.g., ,D2) instead of pressing **RETURN** .

You can also change the normal slot number or volume number at this point, although you'd seldom have any reason to. Should that ever be necessary, just enter the information after a file name. For example, S4,D2,V33 would specify a copy to slot 4, disk drive 2, volume 33.

If your source drive, the one holding the diskette you're copying from, is different from the destination drive, the one

holding the diskette you're copying to, then the diskettes may remain in the drives throughout the copying process.

If you have only one disk drive, you must specify that drive as both the source and the destination drive. **FUD** guides you through the procedure, but make absolutely sure you don't accidentally place the wrong diskette in the drive. You could ruin the diskette and you'll definitely ruin the copy of that file.

If a file being copied or even the name of a file being copied already exists on the destination diskette, the message

already exists, replace?

appears. If you answer N, no copy takes place. But, if you answer Y, the file is deleted from the destination diskette, even if it's locked. Be careful.

FUD Error Messages

If something goes wrong during one of the **FUD** procedures, you're likely to see an error message displayed on the screen. These messages may seem a bit cryptic, but each has meaning. There are even ways to recover from most mistakes.

oops, diskette is write protected

Remember the write-protect notch on the edge of a diskette? If the notch is covered, you can't write anything onto the diskette

under any circumstances. It's physically impossible. If you really want to modify the diskette, remove the plastic tab to uncover the notch.

oops, wrong volume id

You see an error message when the volume identification number you specified is different from the one that is actually on the diskette. Specifying a volume ID of 0 works for any volume ID on the diskette. Should this message ever appear, specify a volume identification number of 0.

oops, file is locked

This appears when you try to delete a file that's locked. Unlock it and try again.

oops, no more room on diskette

If you see this, the destination diskette you're copying to is full. Abort the copy by pushing **BREAK** and insert a diskette with enough room for the copy. You can determine the size of the file and the amount of free space on a diskette with **FUD's S** function (show file). If you replace the full diskette with a blank one, be sure to initialize or format the diskette before you try to use it.

oops, can't find that file

This should never occur in normal use, but it may show up if you put the wrong diskette in the drive. Check to make sure you've inserted the right one.

oops, bad diskette or none in drive

This probably means there is no diskette in the drive, but it also can be caused by a damaged diskette. If there's a diskette in the drive, open and close the drive door and try the operation again. If it still doesn't work, the diskette may be bad. Reseat it and try again. Persist if the diskette holds important programs or data. You may reseat the diskette a dozen times and find that it works on the thirteenth try. If it's a new diskette, reformat it or throw it away. Otherwise, save it and try it on someone else's machine compatible with the ACE.

How To Use BOOT13

The **BOOT13** utility permits older **DOS 3.2** diskettes to be used on the ACE. To boot an older 13-sector diskette, first run the **BOOT13** utility by typing:

```
BRUN BOOT13 
```

The screen clears and the **BOOT13** version message appears, followed by a prompt for the slot number to boot from. Insert

the 13-sector diskette into drive 1 and then either press **RETURN** to use the default slot 6 or press the number of the slot which you want to boot from (1 through 7). The red light should glow on the drive with the 13-sector diskette in it. If it doesn't, you may have specified the wrong slot number or the diskette may be in drive 2.

In a moment, you should see the digit 0 in the upper left corner of the screen. If all's well, it counts up to 9 and then disappears. Your 13-sector diskette is then booting. If the 0 still remains, try opening and closing the door of the drive. If the 0 stays on the screen, either the diskette doesn't contain a good **DOS 3.2** system or it may really be a **DOS 3.3** diskette.

If the digit stops somewhere between 1 and 9, the diskette is probably bad. Try it again. If you still have trouble, discard the diskette.

A Synopsis Of The Programs On The DOS Diskette

Here's a list of the programs on your **DOS** diskette, just for review and reference purposes. Some of them haven't been mentioned earlier because they operate in such a way that you don't really have to know that they're there.

This isn't a permanent list of the programs that Franklin will include on its **DOS** diskettes. Franklin will upgrade and enhance the **DOS** diskette from time to time. Those mentioned here will probably always be on the **DOS** diskettes, but some may be added and others may be modified in future versions.

DOS 3.3 This is the only program on the diskette whose name does not show up in the table of contents of the diskette when you use the **CATALOG** command. **DOS 3.3** manages and controls the operation of the disk drives for all other programs. You can't use your disk drives without it or an equivalent program. (Some application programs have a disk manager that performs the same functions.) **DOS 3.3** is one of the programs that is automatically read in and run when you turn the power on and boot your **DOS** diskette. Generally, you can't even tell that it's running, but it remains in the machine until you turn the power off.

HELLO Like **DOS 3.3**, this program automatically goes from the diskette to run in the machine when you boot the **DOS** diskette. It's a very simple program. It displays the three messages that you see during the disk boot and reads the **INTEGER BASIC (INTBASIC)** program into the computer.

HELLO does its job at power-on time and is never used again.

FPBASIC This is a program called **Floating Point BASIC**. It allows you to write programs in the programming language called **BASIC**. It's an exact copy of the **Floating Point BASIC** that is always stored in the machine. The only reason it's on the diskette is for diagnostic purposes. (One of the diagnostic programs uses it to check memory contents.) You'll never actually use this copy of the program.

INTBASIC This is a program called **INTEGER BASIC**. It also allows you to write programs in the **BASIC** language, although there are slight differences in capabilities between it and **FPBASIC**. This is one of the programs that is automatically read into the machine when you boot the **DOS** diskette. You'll never have to explicitly run **INTBASIC**. Once it's read in, it usually stays in memory until the power is turned off.

FUD FUD lets you do such things as copy files, delete files, display the contents of a diskette, and perform several other fundamental file utility functions. You'll be using **FUD** often.

COPY COPY allows you to make copies of diskettes. This is the program you use to backup diskettes, and you should be using it frequently.

COPY.OBJ COPY.OBJ is a very small program that the **COPY** program uses to help it copy a diskette. **COPY** uses **COPY.OBJ** without your ever knowing it. You will never explicitly use **COPY.OBJ**.

CHAIN CHAIN is of interest only to sophisticated **BASIC** language programmers. It allows a program written in **Floating Point BASIC** to read in and run another program written in **Floating Point BASIC**. Don't worry about **CHAIN**. Most people will probably never use it.

BOOT13 **BOOT13** solves a rather infrequent problem. Diskettes have been manufactured for the ACE in two "styles." The older style is referred to as "13-sector," and the newer is called "16-sector." The 13-sector diskettes were designed to be used with **DOS 3.2**, an earlier version of the **DOS 3.3** program. 13-sector diskettes cannot be used with **DOS 3.3** because **DOS 3.3** presumes a 16-sector diskette. **BOOT13** solves this problem.

Typically, the only time you use **BOOT13** is when you buy a program that is distributed on 13-sector style diskettes. In most of those cases, the program manufacturer will explain how to use the program, including the use of **BOOT13**.

PRETTY PATTERN This program is just that. Run it and see.

DIAGNOSTICS This program allows you to run diagnostics on your machine. Diagnostics perform checks on the machine to see if it's working correctly. If it's not, then the diagnostics may be able to tell you what's wrong. Although **DIAGNOSTICS** is easy to use because it instructs you as it runs, interpreting its results requires a good deal of knowledge and experience. Most ACE users ignore this program, letting service centers handle the problems.

As a matter of fact, the same is true of the rest of the programs on this list.

PROMTEST.OBJ A program used by the **DIAGNOSTICS** program. It will test devices called EPROMS inside the ACE.

KYBDTEST.OBJ Another program used by the **DIAGNOSTICS** program. It tests the keyboard.

MEMTEST.OBJ Yet another program used by **DIAGNOSTICS**. It tests the memory devices in the ACE.

DISKTEST.OBJ It tests the disk drives.

PADLTEST.OBJ Tests the game I/O connector (paddles).

HRESTEST.OBJ Tests the graphics capabilities of the ACE.

PICTURE This is data used by the **HRESTEST.OBJ** program.

Keyboard Code Map

Monitor Program— Differences Between The ACE And APPLE II

This appendix details the program differences between the Franklin ACE computers and the APPLE II Plus computers. The versions of the ACE monitors used for the comparison are three — version 2.0, version 2.1, and version 2.2. Version 2.0 was shipped in most of the ACE 100 models, version 2.1 was shipped in the early ACE 1000's and version 2.2 is the latest version for the ACE 1000. These are all compared to the Apple monitor known as the AUTOSTART ROM.

There are three primary areas of differences:

- 1.) The sign-on name and its location have been changed.
- 2.) The cassette I/O routines in the APPLE are not present in the ACE.
- 3.) The ACE will recognize and display lowercase characters.

There was only one change made between versions 2.1 and 2.2. This was at addresses FD11-FD1A. To save space in the following table, version 2.2 is not listed in the version column except at those addresses. Everything elsewhere that is noted as applying to 2.1 also applies to 2.2.

<u>ADDRESSES</u>	<u>APPLE HAS</u>	<u>ACE HAS</u>	<u>ACE VERSIONS</u>
FAD5-FAD6	two NOP's	two bytes of 00h	2.0 only 2.1 has NOP's
FB09-FB10	title: "APPLE II"	eight bytes of 00h	2.0 & 2.1
FB64	title size byte	same but = 0Ch same but = 0Dh	2.0 2.1
FB66	title loc, low byte	same but = B2h same but = FDh	2.0 2.1
FB67	title loc, high byte	same but = FCh	2.1 only
FB69	start screen pos of title, low byte	same but = 0Dh	2.0 & 2.1
FBB3-FBC0	fourteen NOP's	title string = "ACE 100 V2.0" = C1, C3, C5, A0, B1, B0, B0, A0, F6, B2, AE, B0, 00, 00	2.0 only 2.1 has NOP's

FCC9	HEADR routine entry	RTS	2.0 & 2.1
FCCA-FCDS	HEADR routine body	twelve 00h pads	2.0 & 2.1
FCD6	WRBIT routine entry	RTS	2.0 & 2.1
FCD7-FCDE	part of WRBIT body	JSR \$FCE2 AND #\$3F ORA #\$40 RTS	2.0 & 2.1
FCDF-FCE8	more of WRBIT body	LDA \$200, Y CMP #\$E0 BCC \$FCE8 AND #\$DF RTS	2.0 & 2.1
FCE9-FCEB	end of WRBIT body	three 00h pads	2.0 & 2.1
FCEC	RDBYTE routine entry	RTS	2.0 & 2.1
FCED-FCF9	RDBYTE routine body	thirteen 00h pads	2.0 & 2.1
FCFA-FCFC	RD2BIT routine entry	three NOP's	2.0 & 2.1
FCFD	RDBIT routine entry	RTS	2.0 & 2.1

FCFE-FD0B	RDBIT routine body	fourteen 00h pads	2.0
		title string =	2.1
		“ACE 1000 v2.1” =	
		C1, C3, C5, A0,	
		B1, B0, B0, B0,	
		A0, F6, B2, AE,	
		B1, 00	
FD11-FD1A	part of RDKEY routine	LDY \$24	2.0 & 2.1
		LDA (\$28), Y	
		PHA	
		JSR \$FCD7	
		STA (\$28), Y	
		PLA	
		JMP (\$38)	
		DB 0	
		LDY \$24	2.2
		LDA (\$28), Y	
		PHA	
		JSR \$FCD7	
		NOP	
		STA (\$28), Y	

FD7E-FD83	convert to uppercase	PLA JMP (\$38)	2.0 & 2.1
FEAE-FEAF	two NOP's	six NOP's	2.0 only 2.1 has NOP's
FEC3	one NOP	two 00h pads	2.0 only 2.1 has NOP's
FEC5-FEC9	five NOP's	one 00h pad	2.0 only 2.1 has NOP's
FECD	WRITE routine entry	five 00h pads	2.0 only 2.1 has NOP's
FECE-FEEC	WRITE routine body	RTS	2.0 & 2.1
FEEF	WRBYTE routine entry	thirty-one 00h pads	2.0 & 2.1
FEEE-FEF5	WRBYTE routine body	RTS	2.0 & 2.1
FEFD	READ routine entry	eight 00h pads	2.0 & 2.1
FEFE-FF2C	READ routine body	RTS	2.0 & 2.1
FFAD-FFAF	LDA \$200, Y	forty-seven 00h pads	2.0 & 2.1
		JSR \$FCDF	2.0 & 2.1

FFCF	CTRL T vector B2h	same but = EDh	2.0 & 2.1
FFD2	CTRL S vector B2h	same but = ECh	2.0 & 2.1
FFE9	CTRL Y vector C4h	same but = C3h	2.0 & 2.1

Memory Configuration Jumpers

At location F'2 on the ACE 1000 motherboard is a sixteen-pin DIP socket. This socket allows you to change the ACE's memory configuration. These options have been preset at our factory to provide you with a normal 12K bank of RAM or ROM at \$D000 to \$FFFF. We include this information for those of you who are interested in using some unusual features of the ACE 1000.

The functions of the pins are shown on the next page.

NOTE: If peripheral slot zero is to be used for any peripheral card (especially a RAM card), cut X3 and jumper Y3 to disable the on-board 16K RAM bank.

Pins to Jump	Results
1-16	Normal mode. With this option, you have two 4K banks of RAM located at \$D000-\$DFFF. The user selects which bank is active.
2-15	If this jumper is installed, the user has only one bank of 4K RAM from \$D000 to \$DFFF.
3-14	Normal mode. The user's program can write enable the 12K RAM bank at \$D000 to \$FFFF.
4-13	This option allows the user to write into the 12K RAM bank until it is write inhibited. From that point on, the 12K RAM can not be write enabled, except by turning the power off and back on.
5-12	Normal mode. This allows the user to read enable or disable the 12K RAM bank.
6-11	This option, when memory is write inhibited, enables the 12K RAM bank. The RAM is permanently enabled if option 4-13 is in effect.
7-10	Normal mode. This allows the 12K ROM bank in the range \$D000 to \$FFFF to be enabled or disabled from the data bus.
8-9	This prevents the 12K ROM bank from ever being on the data bus when the 12K RAM bank is read enabled.

Glossary*

acoustic coupler — a device into which you place the handset of a telephone so that your computer can communicate with another computer.

application program — a program such as a word processor or accounting program that allows you to apply your computer's power to a particular task.

backup — a copy of a diskette.

BASIC — an acronym that stands for Beginner's All-purpose Symbolic Instruction Code. It is a language used to program computers.

binary file — generally, a file that holds only certain types of programs.

boot — the process of inserting a diskette in a disk drive and turning the computer on so that the disk operating system loads into the computer from the diskette.

character — generally, any number, letter, symbol, or space on the keyboard.

computer calisthenics — doing things like locking a file that's already locked.

command — a computer user's order or direction to a program running in the machine.

compatibility — the ability of different components, e.g., programs, peripherals, computers, etc., to replace one another with little or no change in capability.

CP/M — an operating system program that's different from **DOS 3.3**.

cursor — the blinking square that shows you where the next character you type will appear.

data — programs and information in their simplest useable form; electronic signals.

*These are informal, sometimes even irreverent, but accurate definitions of terms as they're used in this manual. If you need more detailed technical explanations of these terms, consult the nearest dictionary of computer terms.

default or default value — the alternative or option you choose by choosing none of the others that are open to you.

destination — the disk drive to which you're sending data, usually during the copying of a diskette or files.

disk drive — a device that reads data from and writes data on a diskette.

display monitor — a television-like device that enables you to see what you're doing as you use your computer.

"Dixie" — a song that no known video card can whistle (at least not on key).

DOS — the operating system program that enables your computer to use disk drives for storing and retrieving data.

DOS 3.3 — a particular version of the collection of programs that enables your computer to use the disk drive system for storing data.

DOS 3.2 — a version of **DOS** that is somewhat older than **DOS 3.3** but that performs basically the same function.

80 columns — twice as many columns of data as your ACE normally lets you display on your video screen.

enter — to type data into the computer.

error message — a report from a computer program that something has failed to go as planned.

file — a medium of categorizing and storing data on a diskette.

file names — names given to computer files according to certain conventions so that they are easily stored and retrieved.

file type — a classification system whereby data is categorized according to the programming language for which it is written.

Floating Point BASIC — a version of the **BASIC** language in which decimals are permissible in numbers.

floppy diskette — a pancake-like device used for storing data.

floppy disk system — the components (usually disk drives) necessary in a computer system in order to use floppy diskettes as a storage medium for computer data.

format — to add an organizational scheme to a diskette.

initialize — to add **DOS**, a program of your choosing, and an organizational scheme to a diskette.

INTEGER BASIC — a version of the **BASIC** language in which the variables used in writing the programs can only be integers.

interface cards — (modem, video, disk drive controller, etc.) — printed circuit boards that enable you to connect your computer to disk drives, telecommunications devices, and a number of other peripheral devices.

I/O — the place on your ACE's main electronics board where you plug in your paddles or joysticks.

Itzhak Perlman — a violinist.

joysticks — devices useful for playing computer games.

jug of cider — what no known program can make jump off the table.

load — to bring a program or data from a diskette into your computer.

machine — a word commonly used synonymously with computer.

main electronics board — the large piece of plastic-like material that houses the majority of the electronic components in your computer.

master — a diskette with a version of **DOS** that will operate in a compatible computer with a memory size the same as or different from that of your ACE.

memory — generally, the place in the computer where data stays before, during, and for a short time after processing.

menu — a list of options — usually operations — provided by a program so that you can choose what you want to do from what you see on your display screen.

microprocessor — the principal component in your computer, central to all of the computer processes that take place.

modem — a device that allows you to connect your computer to the telephone lines in order to communicate with another computer.

nibble copiers — programs that make copies of other programs that would otherwise be impossible to copy.

paddles — gadgets useful for playing games.

peripherals — devices such as disk drives and display monitors whose main place of residence is outside the computer but without which you'd have no good means of communicating with the computer.

program — a set of instructions that directs the operation of a computer; to write such a set of instructions.

prompt — a character, word, phrase, or statement which indicates either that something is missing or that the program is ready to accept some further information, usually a command.

read — to introduce data into the computer from a diskette in a disk drive.

reseating — the process of opening the disk drive door, removing and reinserting the diskette, and reclosing the disk drive door. This sometimes facilitates the functioning of a diskette in a disk drive.

RESET — the button you might consider using if your program has gone crazy but which you must avoid at all costs if your computer is reading or writing data on a diskette.

RF modulator — a gadget that lets you use your television set as a display monitor for your computer.

run — to execute or operate a program in the computer.

save — to send whatever data is in the computer out to a diskette for storage.

slave — a diskette with a version of **DOS** that works only on computers that are compatible with your ACE and that have exactly the same memory size.

slots — elongated sockets at the back of the main electronics board. Slots hold peripheral cards such as the disk drive controller card.

source — the disk drive holding the diskette that's providing the data to the computer.

Terrine Maison — meatloaf.

utility program — a program you use to sort through files, reorganize the contents of a diskette, and perform a number of other housekeeping functions.

Winchester — a device that uses a hard disk rather than a floppy disk for storing data.

write — to transcribe data onto a diskette in a disk drive from the computer itself.

write-protect tab — a piece of adhesive plastic or paper (approximately $\frac{3}{8}$ " x 1") used to cover the rectangular notch on the side of a diskette so that it's impossible to write anything on the diskette.

zap — to blast, obliterate, pulverize, demolish, efface, erase, expunge, cancel, rub out, blot out, scratch out, or otherwise destroy some foe in a computer game.

Index

- Acoustic coupler, 1-37
- Ambiguous file names, 4-37 through 4-43
- Application program, 4-1, 4-11, 4-14, 4-23, 4-48. *See also* Program
- Backup, 1-31, 2-6 through 2-10, 4-5, 4-10, 4-49
- BASIC**, 4-12, 4-16, 4-34, 4-49, 4-50
- Binary file, 4-15, 4-17, 4-18, 4-34
- Boot, 4-1 through 4-3, 4-12, 4-13, 4-17, 4-20, 4-22 through 4-24, 4-33, 4-48, 4-49
- Boot13**, 4-46, 4-47, 4-50
- BRUN**, 4-17, 4-18, 4-33
- CATALOG**, 4-13 through 4-16, 4-19, 4-20, 4-23, 4-24, 4-39, 4-48
- Character, 3-11 through 3-14, 4-13, 4-15, 4-37
- Command, 4-5, 4-11 through 4-15, 4-27 through 4-44
- Compatibility, 1-18, 1-19, 1-38, 1-39, 4-32
- Copy (**COPY, C-opy**), 4-5 through 4-11, 4-22, 4-36, 4-43, 4-49
- CP/M**, 1-39, 4-16
- Cursor, 4-3, 4-11, 4-13, 4-40
- Default (default value), 4-6, 4-25, 4-39
- Delete (**DELETE, D-elete**), 4-18, 4-20, 4-21, 4-36, 4-38, 4-41, 4-44, 4-45
- Destination, 4-6 through 4-9, 4-35, 4-43 through 4-45
- Diagnostics, 4-50, 4-51
- Disk drive, 1-29 through 1-31, 3-5, 3-6, 4-1, 4-2, 4-6 through 4-8, 4-11, 4-12, 4-25, 4-34, 4-39, 4-43, 4-44, 4-46, 4-48, 4-51.
See also Destination; Source; Specifying drive numbers
- Display monitor, 1-24, 1-25, 1-45, 3-4, 3-5
- DOS** (diskette), 1-39, 4-1 through 4-51

DOS 3.3, 1-39, 4-1, 4-4, 4-5, 4-11 through 4-13, 4-16, 4-47, 4-48, 4-50

DOS 3.2, 4-46, 4-47, 4-50

Eighty columns, 1-33, 1-34

Error message, 4-9, 4-12, 4-17, 4-18, 4-20, 4-21, 4-30, 4-31, 4-35, 4-36, 4-41, 4-44 through 4-46

EXEC, 4-33, 4-34

File, 1-30, 4-1 through 4-51

File name, 4-14, 4-25, 4-36 through 4-43

File type, 4-15 through 4-18, 4-23, 4-24, 4-28, 4-42

Floating Point BASIC, 4-15, 4-49, 4-50

Floppy diskette, 1-14, 1-29, 1-30, 1-31

Floppy disk system, 1-14, 1-31

Format (**F-format**), 4-22, 4-28 through 4-32, 4-34, 4-39, 4-45

FUD, 4-9, 4-18, 4-22, 4-24, 4-26 through 4-46, 4-49

HELLO, 4-14 through 4-17, 4-19, 4-20, 4-23, 4-24, 4-33, 4-34, 4-48, 4-49

Initialize (**INIT**), 4-21 through 4-24, 4-28, 4-29, 4-32, 4-34, 4-45

INTEGER BASIC, 4-15, 4-48, 4-49

Interface card. *See* Printer interface card; Telephone card

I/O, 1-44, 4-31

Joystick, 1-43 through 1-45

Keyboard, 3-11 through 3-14, 4-51

Keycodes, 3-14. *See also* Appendix A

Lid, 2-10, 3-3, 3-4

Load (**LOAD**), 4-5, 4-6, 4-24, 4-51

Lock (**LOCK, L-ock**), 4-15, 4-18, 4-19, 4-36, 4-41, 4-42, 4-44, 4-45

Main electronics board, 3-5, 4-6, 4-12

- Main menu. *See* Menu
- Master (**M-ake Master**), 4-22, 4-28, 4-32 through 4-36
- Memory, 1-14, 1-15, 1-24, 1-29, 4-22, 4-51
- Menu, 4-18, 4-27, 4-30, 4-31, 4-41
- Microprocessor, 1-24
- Modem, 1-36 through 1-38
- Monitor, 1-25, 1-26, 1-45
- Nibble copier, 2-9, 2-10
- Paddles, 1-43 through 1-45, 4-51
- Paging, 1-32
- Peripherals, 1-25
- Power cord, 3-2, 3-7
- PRETTY PATTERN**, 4-25, 4-50
- Printer, 1-26, 1-27
- Printer interface card, 1-27, 1-28
- Program, 1-9, 1-40, 1-41 through 1-43, 4-14 through 4-47. *See also*
Appendix B: Application program; *names of individual programs*
- Prompt, 4-2, 4-11, 4-12
- Read, 1-29, 4-8, 4-9, 4-18, 4-43
- RENAME**, 4-20
- Reseat (a diskette), 4-9, 4-31, 4-46
- RESET switch, 2-4 through 2-6, 3-15
- RF modulator, 1-26, 1-45
- Run (**RUN**), 4-5, 4-12, 4-34
- Save (**SAVE**), 4-24
- Scrolling, 1-32
- Sector, 4-15, 4-46, 4-47, 4-50

Slave, 4-22, 4-28, 4-36
Slot, 3-5, 4-6 through 4-8, 4-12, 4-43
Show (**S-how**), 4-31, 4-36, 4-39, 4-41, 4-45
Source, 4-6, 4-8, 4-9, 4-11, 4-43, 4-44
Specifying drive numbers, 4-39, 4-42, 4-43
Telephone interface card, 1-36 through 1-38
Text file, 4-15, 4-30, 4-34
Turning (the machine) on, 3-9
TV, 1-25, 1-26, 1-45
Utility program, *See* **DOS; FUD**
Unlock (**UNLOCK, U-nlock**), 4-15, 4-19, 4-20, 4-36, 4-42, 4-45
Verify (**V-erify**), 4-36, 4-42, 4-43
Video card, 1-31 through 1-33, 1-34
Video cable, 3-2, 3-4, 3-5
Volume number, 4-14, 4-29, 4-43, 4-45
Window vision, 1-27
Winchester, 1-31
Write, 1-29, 4-9
Write-protect tab, 4-10, 4-31, 4-35, 4-36, 4-44, 4-45

DOS And FUD Command Formats

The COPY Program

Type: RUN COPY

DOS 3.3

Type: CATALOG

Type: RUN (The name of some Type A file)
or

RUN (The name of some Type A file) ,D1
or

RUN (The name of some Type A file) ,D2

Type: BRUN (The name of some Type B file)
or

BRUN (The name of some Type B file) ,D1
or

BRUN (The name of some Type B file) ,D2

Type: LOCK (The name of any file)
or

LOCK (The name of any file) ,D1
or

LOCK (The name of any file) ,D2

Type: UNLOCK (The name of any file)
or

UNLOCK (The name of any file) ,D1
or

UNLOCK (The name of any file) ,D2

Type:	RENAME	(The name of any file) or		<input type="text" value="RETURN"/>
	RENAME	(The name of any file) or	,D1	<input type="text" value="RETURN"/>
	RENAME	(The name of any file)	,D2	<input type="text" value="RETURN"/>
Type:	DELETE	(The name of any file) or		<input type="text" value="RETURN"/>
	DELETE	(The name of any file) or	,D1	<input type="text" value="RETURN"/>
	DELETE	(The name of any file)	,D2	<input type="text" value="RETURN"/>
Type:	INIT	(Insert a new diskette) (The name of any Type A file)	,D1	<input type="text" value="RETURN"/>
	LOAD	(Replace new diskette with DOS diskette) (The name of any Type A file)	,D1	<input type="text" value="RETURN"/>
	SAVE	(Replace DOS diskette with the new diskette) (The name of any Type A file)	,D1	<input type="text" value="RETURN"/>

The FUD Program

Type: BRUN FUD

Then Type: F
or
M
or
S
or
C
or
D
or
L
or
U
or
V

The BOOT13 Program

Type: BRUN BOOT 13