

**DISK II**  
**FLOPPY DISK SUBSYSTEM**  
INSTALLATION AND OPERATING MANUAL



Apple Intelligent Subsystems



**DISK II**  
**FLOPPY DISK SUBSYSTEM**  
**INSTALLATION AND OPERATING MANUAL**

**PRELIMINARY**

**Apple Intelligent Subsystems**

## TABLE OF CONTENTS

| TITLE  | PAGE |
|--|------|
| INTRODUCTION   | 1    |
| Introduction and Unpacking Instructions                |      |
| INSTALLATION   | 3    |
| How To Install the Disk II Controller                  |      |
| Installing Multiple Disk Drives                        |      |
| If It Doesn't Work                                     |      |
| HANDLING   | 5    |
| Inserting and Removing Diskettes                       |      |
| Care of Diskettes                                      |      |
| PREREQUISITES  | 7    |
| Prerequisites To Using the Disk Operating System       |      |
| BOOTING  | 8    |
| Starting the System                                    |      |
| Initializing New Diskettes                             |      |
| The Initialization Procedure (with one drive)          |      |
| INITIALIZING   | 10   |
| Fancier Initialization                                 |      |
| INITIALIZING   | 12   |
| Using the Init File More Aggressively                  |      |
| SAVING   | 13   |
| Loading and Saving Programs in Integer Basic           |      |
| Now: Hear This!  |      |
| EXAMPLES   | 15   |
| WRITE-PROTECTION                                       | 16   |
| Write-Protecting a Diskette                            |      |
| RECOVERING   | 17   |
| Recovering From Accidental Resets                      |      |
| Renaming Files   |      |
| How to Invoke Applesoft                                |      |
| RUNNING  | 19   |
| Running Programs                                       |      |
| DOS-A  | 20   |
| A Summary Of the Commands and Procedures               |      |
| The Apple II Disk Operating System (DOS)               |      |
| Restarting and Getting Into and Out Of the Monitor     |      |
| Initializing a New Diskette                            |      |
| Commands   |      |
| Use Of the Disk Operating System From Within a Program |      |
| DISK II APPLICATION NOTE                               | 34   |



## APPLE DISK II

### INTRODUCTION AND UNPACKING INSTRUCTIONS

The APPLE DISK II is a "floppy" disk unit which allows you to store and retrieve information much more quickly and conveniently than you can with tape. The information is stored and retrieved from a "diskette". The diskette is a small (5 inch diameter) plastic disk coated so that information may be stored on and erased from its surface. The coating is similar to the magnetic coating on recording tape. The diskette is permanently sealed in a square plastic case which protects it, keeps it clean and allows it to spin freely. This package is never opened.

The term "floppy" comes from the fact that the diskette is flexible. Older computer information storage devices that worked on similar principles used rigid disks. While the diskette (and its plastic case) are somewhat flexible, actually bending the diskette will damage it.

One of the most important advantages to using DISK II is that information is stored and retrieved by a name under which it is filed. A program that catalogs phone numbers might be saved with an instruction such as

SAVE PHONE NUMBERS

and retrieved with an equally simple command. The name under which something is filed is the file name.

A single diskette can hold over 116,000 characters of information. Typically, files are much shorter than this, and a number of files (up to 82) can be stored on a single diskette. To help you (and the APPLE II) find files, a catalog of all the files currently residing on the diskette is kept on each diskette. The programs that automatically maintain this catalog, and that save and retrieve information (as well as do a multitude of other housekeeping tasks) are called the Disk Operating System, or DOS.

There are six items that comprise your DISK II system. The disk drive (the main box), a circuit card (the controller) that plugs into the APPLE II, a "DISK II MASTER" diskette, a blank diskette, the warranty card, and this manual. The disk drive is connected to the controller by a flat ribbon-like cable. The cable must be attached to the upper set of pins on the controller. They are marked "DRIVE 1". Connect it very carefully as the pins are easily damaged. One possible error is to not center the connector. Be careful and see that all the pins are going into holes, and not to one side of the connector or the other. The connector can only go on one way, as the cable prevents it from being put on backwards.

If you have purchased a drive only (for example, as a second drive on your controller) your system will include the disk drive, the diskettes, a warranty card, and this manual. You will connect your second drive to the lower set of pins on your controller. Take the same care attaching this connector as you did with the first.

Save the packing material in case you wish to transport your disk--or in the unlikely event it has to be returned to the dealer or to the factory. Send in the warranty card--not only does this put the warranty in effect, but it allows us to keep you informed of updates and new products.

## HOW TO INSTALL THE DISK II CONTROLLER

To install the DISK II, you will simply plug the controller board into a socket inside the APPLE II, and then replace the top so as to hold the cable in place, as follows:

1. Turn the APPLE II off. This is important to prevent damage to the computer.

2. Remove the cover from the APPLE II. This is done by pulling up on the cover at the rear edge (the edge farthest from the keyboard) until the two corner fasteners pop apart. Do not continue to lift the rear edge, but slide the cover backward until it comes free.

3. Inside the APPLE II, across the rear of the circuit board, there is a row of eight long, narrow sockets called "slots". The leftmost one (looking at the computer from the keyboard end) is slot #0, and the rightmost one is slot #7. BE SURE THE POWER IS OFF BEFORE YOU INSERT OR REMOVE ANY BOARD FROM THE COMPUTER. Insert the "fingers" portion of the controller into slot #6, one socket to the left of the rightmost socket. The "fingers" portion will enter the socket with some friction and will then seat firmly. Since the fingers make electrical contact, it is a good idea to keep your fingers from touching them. They may be cleaned with rubbing alcohol before installation. The controller board may be placed in any slot except slot #0, the leftmost. However, APPLE's standard location is slot #6, and most APPLE software (and this manual) is written with that location in mind.

4. Replace the cover of the APPLE II, remembering to start by sliding the front edge of the cover into place. Press down on the two rear corners until they pop into place. The cover will hold the cable.

5. The DISK II controller is now installed. Place the disk drive in a convenient location, usually alongside of or on top of your APPLE II, which may now be turned on.

#### INSTALLING MULTIPLE DISK DRIVES

If you have a second drive, it is better to attach it to the lower set of pins (marked DRIVE 2) on the controller board than to use a second controller. The third drive would be attached to the DRIVE 1 position on the second controller, the fourth to the DRIVE 2 position on the second controller and so forth.

If you have multiple drives, it is a good idea to label each with its slot and drive number since your programs will refer to the disks by those numbers.

#### IF IT DOESN'T WORK

This isn't likely, but if your unit was shipped in an old truck or some such, the connectors inside the disk drive may have worked a bit loose. If you are at all squeamish about handling the insides of your drive, your dealer will be glad to check it out. Your dealer is also the person to see if first aid doesn't get things working. One last trick, that cures 90% of all problems--re-read the manual carefully.

If you enjoy getting your fingers into the works (and don't mind voiding the warranty), you can turn the computer off, and disconnect the drive from the controller. Loosening the four screws on the bottom of the drive allows the mechanism to slip forward out of the case. Tighten the connectors by pushing them gently onto the circuit boards. Re-assemble the unit and it will probably now work.

#### INSERTING AND REMOVING DISKETTES

While there is nothing difficult about using a disk drive (it is far easier than using a cassette recorder), some care is necessary to protect the diskettes. The drive itself must also be handled with some care. The drive door is opened by pulling up its bottom edge. The diskette is then slipped in the slot with the label upwards. This should be done gently until the diskette is entirely into the drive. Do not bend the diskette! If it is pushed in too hard it can be permanently damaged. Close the door by pushing it down again. Two metal fingers (which can be seen just inside the slot) should just clear the diskette as the door closes.

A diskette is removed by opening the door and pulling it carefully out of the drive. However NEVER REMOVE A DISKETTE WHILE THE "IN USE" LIGHT IS ON. This can permanently damage the diskette, and is almost sure to destroy the information on the diskette even if the diskette is still useable.

#### CARE OF DISKETTES

Diskettes hold a tremendous amount of information. An individual bit of information on the diskette is therefore very small. An invisible scratch on the surface of the diskette, or even a fingerprint, can cause errors. Never let anything touch the brown surface of the diskette itself. Handle the diskette by the black plastic case only. When a diskette is not in use, keep it in the paper sleeves that it is supplied with. These sleeves are treated to minimize static build-up which would attract dust. If you wish to write on a diskette label, use a FELT TIP pen. Do not press hard. It is better to not write directly on the diskette package, but to use separate labels. Write on these labels and then stick them on the diskettes.

Keep diskettes away from magnetic fields. This means to keep them away from electric motors, magnets, and that they should not be placed on electronic devices such as television sets. They may be temporarily laid on the APPLE II or DISK II. Keep diskettes out of the sun, and away from other sources of heat that can cause them to warp. Car trunks on hot days can be diskette killers. Do not place diskettes on dirty or greasy surfaces. Do not let them collect dust. It is best

if they are stored vertically between use.

This may seem like a lot of "don'ts", but with reasonable care, each diskette will give you almost unlimited service. With just a little bit of carelessness, they may give you no service at all.

The DISK II drive, unlike the APPLE II, is a mechanical device. It has motors and moving parts. Therefore it is somewhat more delicate than is the computer. Rough handling, such as dropping the drive, or having things drop on it, can cause it to malfunction. The drive should not be used on top of a TV set for the same reason that diskettes should not be placed on or near a TV set. The strong magnetic fields put out by TV's can cause damage to the magnetic properties of the drive.

## PREREQUISITES TO USING THE DISK OPERATING SYSTEM

The next few sections assume that you are familiar with using the APPLE II computer. If this is your first use of the APPLE II computer, begin by learning how to use the BASIC language described in the APPLE II BASIC PROGRAMMING MANUAL. You do not have to know how to program to use the disk, but the material in the first chapter should be mastered so that you are familiar with the keyboard and with the commands to get the BASIC prompt character (>).

Many of the facilities of the DISK II will not be of any great utility (nor will they make much sense) if you have not learned the capabilities of the computer first.

Most of the tasks that the disk can do are similar to those the cassette interface can perform. Reading the section in the BASIC PROGRAMMING MANUAL on saving programs (page 76) is especially helpful. It is even more helpful if you've read the material that comes before that section.

## STARTING THE SYSTEM

To begin with, you have to get the Disk Operating System (DOS) started. In computer jargon, this is called "booting the system". This procedure places a program (which is on the MASTER DISKETTE) into the APPLE II. This program adds the disk commands to BASIC.

To boot the DOS, insert the DISK II MASTER DISKETTE into the drive (or into Drive 1 if you have more than one drive) and close the door.

Using the familiar {RESET}{CTRL}B{RETURN} sequence, put the APPLE II into Integer BASIC. Assuming that you've put the controller in slot 6, as described above, type PR#6{RETURN} the "IN USE" light will come on, and the disk drive will make clicking and whirring noises. After a few seconds this message will appear:

```
DISK II MASTER DISKETTE  VERSION 3
                          29-JUN-78
COPYRIGHT 1978  APPLE COMPUTER INC.
>
```

The version number or date may be different, as improvements are continually made in the system. At this point, DOS is booted and you are in Integer BASIC. To demonstrate that the DOS is booted type the instruction

CATALOG

and a list of the programs and data files that are on the MASTER DISKETTE will be written on the screen. Type the instruction

LIST

and you will see the program that writes the heading for the MASTER DISKETTE. It is just a few PRINT statements.

## INITIALIZING NEW DISKETTES

You have just booted from the MASTER DISKETTE that was initialized by APPLE COMPUTER. When a new diskette is purchased it is not usable by the computer. It must be initialized. If you try to boot from a diskette that has not been initialized, you will get the message:



\*\*\*DISK: DISK I/O ERROR

Once you have booted the DOS from the MASTER DISKETTE, you can use the APPLE II to initialize new diskettes. These "slave" diskettes can then be used to save data and programs.

Before beginning the initialization process, there should be a BASIC program in the APPLE II. It need not do much, it can be as simple a program as this one:

NEW

10 PRINT "SLAVE DISKETTE CREATED 9-8-78"

20 END

This program is called the "INIT" program as it will be executed when you boot the slave diskette, and present the message on the screen. This has two purposes: when you see the message, you know that the DOS has been booted successfully; the message lets you know that you are using a slave diskette.

#### THE INITIALIZATION PROCEDURE (with one drive)

Remove the MASTER DISKETTE from the drive. Select a name for your INIT program. "HELLO" is a customary name since it is the first program to greet you when you boot. Place the blank diskette in the drive. Type the instruction

INIT HELLO, V254

or

INIT whatever name you choose, V whatever volume number you wish

When you press {RETURN} the disk is set up with a catalog, and the INIT program, and the entire surface is initialized. The process takes nearly two minutes. The usual chatter is heard from the disk drive during initialization.

When the process is complete, the disk makes one last click, and the prompt character (>) reappears. Remove the new diskette and label it as having been initialized. It is a good idea to label it with the date as well.

## FANCIER INITIALIZATION

There are a number of options that are useful during initialization (and with other disk operations). The first option allows you to operate with more than one drive. Each controller has the ability to control either one or two disk drives. Normally, instructions refer to drive 1. This is the default drive selection. If you wish to specify drive 2, you use the notation D2 separated from the file name by a comma. For example, to initialize a diskette in drive 2, you would use the instruction

```
INIT HELLO, V254, D2
```

After drive 2 has been specified, all further disk commands refer to drive 2 unless drive 1 is specified.

If more than two drives are in use, then additional controllers are required. These are placed in different slots than the first controller (which is customarily in slot number 6). You can specify slot n (where n is a digit from 1 to 7) with the notation Sn separated from the other disk options by a comma. The default slot number is the one you used when booting the DOS. For example, to initialize a diskette in drive 1 attached to a controller in slot 5, you would use the instruction

```
INIT HELLO, V254, S5, D1
```

Once a slot number has been set, it becomes the default slot number until it is explicitly changed.

If you accidentally use a diskette with the wrong volume number, the system will reject it with the message

```
***DISK: VOLUME MISMATCH ERROR
```

Volume mismatch errors cannot occur when you ask to see the CATALOG. In case you wish to know the volume number of a diskette, it is given at the top of the CATALOG.

Specifying a volume number of zero disables checking of volume numbers.

Once a volume number has been set, that volume number becomes the default volume number until it is explicitly changed, or a CATALOG command is given. The CATALOG command changes the default volume number to the volume number of the diskette

CATALOGed. It is recommended that you write the volume number on every diskette that you initialize. The system diskette is volume 254.

To initialize a diskette using the name "START UP" for the INIT file, where the diskette is in drive 2 of a controller in slot 5, and to assign it a volume number of 19, you would use the command  
INIT START UP, D2, S5, V19

The drive number, slot number and volume number options may appear in any order. The above command is equivalent to  
INIT START UP, V19, S5, D2  
and to  
INIT START UP, S5, V19, D2  
and so on.

#### USING THE INIT FILE MORE AGGRESSIVELY

Let's say that you wanted the computer to run the COLOR DEMOs (provided on cassette tape) program whenever you booted Disk II. When the time came to create the INIT program you would put the COLOR DEMO tape into the cassette recorder and type LOAD

When the program had loaded from tape, you would be well advised to give it a try. If everything proved to be in working order you would put the new diskette into the drive and type  
INIT HELLO, VO

Thereafter, whenever that diskette is booted, it will automatically LOAD the COLOR DEMO program and RUN it.

The INIT file allows you to make the APPLE II into what is called a "turnkey" system. If a doctor wanted to do his or her office accounting on an APPLE II, it would be programmed with suitable accounting programs. These accounting programs would be used as the INIT file. Then, whenever the computer was turned on, all that the office staff would have to know is to turn on the APPLE II, type

{RESET}6{CTRL}P

and the doctor's programs would appear. The staff wouldn't need to know BASIC or anything else about the APPLE II. The computer would become an accounting system, its internal characteristics unimportant since all the staff would be familiar with is the modus operandi of the accounting program.

This is the essence of the idea of a "turnkey" system: from the user's point of view the computer is a device that does only a particular task, and getting the system started is as simple as turning a key in a lock. In this case, the "key" is simply pressing four buttons on the keyboard. It does not require computer expertise to be able to do that.

## LOADING AND SAVING PROGRAMS IN INTEGER BASIC

Loading and saving programs on the disk is similar to using the cassette (except that programs are referenced by file name), everything goes at least ten times faster, and you never need to press buttons to play, record or rewind. It is all automatic. There are many additional abilities that the disk brings as well, such as the catalog of programs and the automatic running of programs without user intervention. Saving of data is also very easy.

It is recommended that you hang on to your cassette tape system for trading programs and as back-up storage for vital programs and data (although experience shows that disk storage is even more reliable than cassette storage of programs and data).

If you have a program in Integer BASIC, and you wish to call it HENRY, then the command

SAVE HENRY

would save it on the diskette. If you have more than one drive it normally would be saved on the drive from which you booted. You can specify drive number, volume number and slot number as with the INIT command. For example, to SAVE a file called AGATHA on drive 1 of the controller in slot 2, where the volume number of the diskette is 214, you would use the command

SAVE AGATHA, D1, S2, V214

as before, the three options can be put in any order.

Program names are file names, and may be up to 32 characters long. They may include any characters you can type except commas. There is a danger here: a control character accidentally typed into a name will not appear to be part of the name in the catalog. If you hit Control T instead of plain T in "AGATHA", the catalog listing would appear to be "AGAHA"--which is what you would say if you tried to get at that file again. This can make the file appear totally unresponsive. So be careful not to put control characters in file names. Here are some valid names for files:

CHECKBOOK

THE QUALITY OF MERCY

HIRES34

NOW: HEAR THIS!

To LOAD a program named AGATHA, use the command  
LOAD AGATHA  
and the program of that name, if there is one in the catalog,  
will be loaded. To test if AGATHA is loaded, see if it can  
walk along a straight line.

If, when you try to SAVE a program, you get the message  
\*\*\* SYNTAX ERROR  
either you have made a typing error, or DOS isn't booted. If  
DOS isn't booted--DON'T BOOT IT. You will lose your program.

First, save the program on tape (using the usual  
SAVE  
command)  
and then boot DOS. Next, using the usual  
LOAD  
command, LOAD the program from the tape. Now you will be  
able to SAVE it on disk.

If a diskette is bad (perhaps someone tried to staple it into  
a notebook), or if the diskette is not initialized, or if  
there is no diskette in the drive, or if the door is open or  
if the diskette is "write-protected" (see the next section),  
the message

\*\*\* DISK: DISK I/O ERROR  
will appear. Check all the items listed, and correct the  
problem. Try again.

If you use the command  
LOAD HENRY  
and HENRY is not the name of a program on the diskette in the  
drive, then you get this message  
\*\*\* DISK: NOT BASIC PROGRAM ERROR

To eliminate any file that you would rather not have on your  
diskette, the command  
DELETE

can be used. For example, the command  
DELETE EXCESS, V34, D2, S1  
deletes a file named EXCESS from a diskette with volume  
number 34, which is placed in drive 2 of the controller in  
slot 1.

AN EXEMPLARY DIALOGUE SHOWING THE USE OF SAVE, LOAD, CATALOG  
AND DELETE INCLUDING A BOOT

This is a dialog as it might appear on the screen of your  
APPLE II.

>PR#6

The screen clears and you see the following (parts you type  
are underlined)

DISK II MASTER DISKETTE

VERSION 3

29-JUN-78

COPYRIGHT 1978

APPLE COMPUTER INC.

>CATALOG

DISK VOLUME 254

I 006 HELLO

\*I 043 APPLESOFT

A 005 SPIRODEMO

\*B 039 CHURCHILL

>NEW

>10 PRINT "JABBERWOCK"

>20 END

>SAVE DEMO

>CATALOG

DISK VOLUME 254

I 006 HELLO

\*I 043 APPLESOFT

A 005 SPIRODEMO

\*B 039 CHURCHILL

I 002 DEMO

>NEW

>RUN

\*\*\* NO END ERR

>RUN DEMO

JABBERWOCK

## WRITE-PROTECTING A DISKETTE

[Picture of Diskette with write-protect cut-out called out]



Certain diskettes, such as the Master Diskette, and any programs or data that you would not want to accidentally get written over (and thus lost), can be protected by having a cut-out on the left side covered over. This is done by using a piece of adhesive backed material. These stick-ons are supplied with purchases of boxes of diskettes, but in a pinch any piece of sturdy tape will do.



#### RECOVERING FROM ACCIDENTAL RESETS

In Integer BASIC, if you accidentally strike the RESET button, you can recover intact by using {CTRL}C. The DOS also has recovery procedures that will usually preserve your program and data.

If you hit RESET you get an asterisk (\*) prompt. Type 3DOG{RETURN} to get back to DOS. This will not work, of course, if you have not already booted the system. If you use {CTRL}C to get back to BASIC, and then you find that you want to re-activate the DOS, you can use the command

CALL 976

which does the same task. If this recovery to DOS does not work, and the program still LISTs, you can use a tape save.

#### RENAMING FILES

It is possible to change the name of a file. If you wished to change the name of the file called "AGATHA" to the name "SESAME"

then you would type

RENAME AGATHA, SESAME

a check of the CATALOG will show the file to have been renamed.

#### HOW TO INVOKE APPLESOFT

If you wish to write a program in APPLESOFT (or just use the computer as a calculator with floating point numbers (numbers with decimal points)), you must first invoke APPLESOFT BASIC. Use the command

FP

(that's all there is to it)

and in a few seconds APPLESOFT will be up and running. The FP stands for "Floating Point", of course. Do not try to RUN the APPLESOFT file. You will get a grand collapse of the DOS after everything seems to be running OK.

To get back to integer BASIC, use the command INT.

## RUNNING PROGRAMS

Programs in Integer BASIC, APPLESOFT BASIC and machine language can be LOAded and RUN with a single disk command. If SPEEDY is a BASIC program (in either specie of BASIC) then the command

RUN SPEEDY

will cause that program to be LOADED from the diskette (assuming that the system has been booted, and a program named "SPEEDY" is on the diskette). If SPEEDY is in APPLESOFT, it will take a bit longer as the DOS knows to first LOAD APPLESOFT, and then to LOAD SPEEDY.

Once the program is loaded, it is automatically executed. Thus the disk command

RUN SPEEDY

is exactly equivalent to the following two commands:

LOAD SPEEDY

RUN

Machine language programs can be LOAded and executed with the command

BRUN

which will be discussed in detail along with the commands for loading and saving machine language files, BLOAD and BSAVE, in a later section.

A SUMMARY OF THE COMMANDS AND PROCEDURES  
THE APPLE II DISK OPERATING SYSTEM (DOS)

COPYRIGHT 1978, APPLE COMPUTER INC.

ABBREVIATIONS

- {CTRL} holding down the key marked "CTRL" while  
another key is pressed.
- {RETURN} pressing the key marked "RETURN"  
The required {RETURN} after commands is  
not shown.
- {RESET} pressing the key marked "RESET"
- f a file name. This is from one to 30 characters.  
Any typeable character except the comma may  
appear in a file name. The first character must  
be a letter of the alphabet.  
Examples: CHESS  
RECIPE  
SUM OF SQUARES  
NEW45  
HOW-ABOUT-THIS
- g another file name.  
Example: SEPARATOR WITH LOW VELOCITY
- s number of the slot in which the controller has  
been placed. s initially defaults to the slot  
from which DOS was booted. It subsequently  
defaults to the last value specified for  
this parameter. s has to be in the range  
1 to 7.  
Examples: 7  
2  
The controller is usually put in slot 6.  
Additional controllers are placed ad libitum.

v volume number of a diskette. v initially defaults to the volume number of the diskette from which the system was booted. It subsequently defaults to the latest value specified for this parameter, or implicitly specified by a CATALOG command. v has to be in the range 0 to 254 inclusive.  
Example: 101

Note: A diskette's volume number may not be 0. Specifying a volume number of 0 in a disk command is a "wild card" and tells the DOS to ignore the volume number on the diskette.

d drive number (either 1 or 2). d initially defaults to one. It subsequently defaults to the latest value specified for this parameter.  
Example: 2

r record number. r defaults to 0 for the first READ or WRITE. It defaults to the next record to be read or written thereafter. r points to a record within a file. r must be in the range 0 to 32767.

b byte number. b defaults to 0. b points to a byte within the record pointed to by r. b must be in the range 0 to 32767. For most applications b is in the range 0 to the current record length.

j length specifier. It sets up the number of bytes in a record. j must be in the range 1 to 32767. j defaults to 1.

Not all commands permit all of these parameters, but those parameters that are permitted in a given command may appear in any order.

A simple notation is used to specify the syntax of each DOS command. Items in brackets ( [ ] ) are optional. CAPITAL letters and commas must be typed as shown, lower case letters stand for items that you must supply. For example, the DOS command that is notated

INIT f ,Vv [,Ss] [,Dd]  
can be interpreted as  
INIT HELLO ,V17,D2

by the following process: The keyword "INIT" is in upper case, and must be exactly as shown. "f" is lower case and stands for a file name--and "HELLO" is a legitimate file name. The "v" is required and must be as shown. The "v" stands for a volume number, and 17 was chosen arbitrarily for this example. The notation ",Ss" is optional and omitted. The notation ",Dd" becomes ,D2 in this example. The ",D", being a comma and capital letter must appear as shown, and the lower case "d" stands for the drive number, which in this example is 2.

Any numerical constant in the DOS can be entered in hexadecimal notation by preceeding the hex digits with a dollar sign.

#### BOOTING (starting the system)

From Integer BASIC:

>PR#s

Example: >PR#6

From APPLESOFT:

]PR#s

Example: ]PR#6

From the Monitor you can use any of the following:

\*Cs00G

\*s{CTRL}K

\*s{CTRL}P

Examples: \*C600G  
          \*6{CTRL}K

#### RESTARTING AND GETTING INTO AND OUT OF THE MONITOR

This ability is necessary if {RESET} is pressed accidentally, or if you are alternately using the monitor and BASIC.

\*3D0G (preserves current BASIC program)

\*3D3G (does not preserve the current BASIC program,  
loses APPLESOFT (unless in ROM).)

To get to the monitor from BASIC without deactivating DOS, use the commands:

>POKE 72,0

this can be done in

>CALL -151

APPLESOFT as well

Do not hit {RESET} to get the monitor as that de-activates the DOS.

To get back to Integer BASIC again use {CTRL}C. If you have not previously been in Integer BASIC you must use {CTRL}B.

To get back to APPLESOFT BASIC type 3D0G.

You can use DOS commands from within the monitor, but after using them you must issue the monitor command

\*48:0

to insure that the monitor will operate again correctly.

## INITIALIZING A NEW DISKETTE

You should have an Integer BASIC program in memory when a diskette is initialized. This program will be executed whenever the diskette being initialized is booted.

>INIT f ,Vv [,Ss][,Dd]

Example: INIT HELLO

The parameter v assigns a volume number to the diskette being initialized. v must be in the range 1 to 254.

To create a sub-master diskette run the file MASTER.CREATE on the MASTER DISKETTE supplied with the DISK II. Follow the instructions in that file. A notation such as <PROGRAM NAME> means the same thing as does a lower case letter in this write-up. Lower case letters are not usually available on the APPLE II, so this alternative was chosen.

## COMMANDS

The following commands may be issued if there is an Integer BASIC prompt (>) or if there is an APPLESOFT prompt (>). The commands must begin immediately after the prompt. Blanks are ignored except within file names.

LOAD f [,Ss][,Dd][,Vv]

Example: LOAD DOW JONES, V19, D1

Attempts to find a file with name f on the specified drive. If the volume numbers match and there is such a file, that program will be LOAded into the computer. It can then be LISTed, or RUN, or SAVED as with any program. LOAD closes any open files.

If the file is an APPLESOFT BASIC program, and APPLESOFT is not in memory, APPLESOFT will be LOAded and RUN automatically before the program is LOAded. However if there is a syntax error in the LOAD command after the file name, the system will hang, your program will be lost, and you will have to re-boot the DOS.

The instruction LOAD, without any parameters, will LOAD a program from cassette tape.

SAVE f [,Ss] [,Dd] [Vv]



Example: SAVE COLOR DEMOS, V56

If there is no file with the specified file name, a file is created on the specified drive and the current BASIC program is stored under the given file name. If there already is a file with the specified file name, its contents are lost and the current BASIC program is saved in its place. No warning is given.

DELETE f [,Ss][,Dd][,Vv]

Example: DELETE TEST

Removes the file named f from the diskette. There is no way to retrieve a DELETED file. If f was open, this command closes it.

RENAME f,g [,Ss][,Dd][,Vv]

Example: RENAME SEPERATE, SEPARATE, S4, D1, V0

Finds the file named f, and changes its name to g. The file's contents are unaffected. The INIT file's name should not be changed. (The next boot will garner a "\*\*\*DISK: FILE NOT FOUND ERROR" message.) If f was open, it is closed.

CATALOG [,Ss][,Dd]

Example: CATALOG

Displays on the screen the volume number of the diskette and a list of every file on the indicated diskette. The default volume number is changed to match that of the indicated diskette.

With each file an indication of its type and length is given.

The types are:

- I The file is an Integer BASIC program.
- A The file is an Applesoft BASIC program.
- T The file consists of Text--i.e. it was produced by a WRITE command.
- B The file is a bit-for-bit image of a portion of memory.

The number after the type is an indication of the length of the file in sectors (256 character groups). The minimum length of a file is 2 sectors due to the room needed for storing information such as the file name, and other housekeeping data. An asterisk indicates that the file is LOCKed. If s or d refer to nonexistent disks, the system will stop and any program in memory will be lost. You will have to re-boot the DOS.

VERIFY f [,Ss][,Dd][,Vv]

Example: VERIFY SAM

Performs a check that the data on the disk is self-consistent. If a file VERIFYS, it is probable that it hasn't been damaged or written incorrectly.

RUN f [,Ss][,Dd][,Vv]

Example: RUN ANNUITY, D2

Performs a LOAD, then also RUNs the program loaded. All OPEN files are closed. If no parameters are given, the program in memory is RUN. If f is an APPLESOFT file and there is a syntax error in the other parameters, the system will stop, and you will have to re-boot the DOS.

CHAIN f [,Ss][,Dd][,Vv]

Example: CHAIN PART TWO, D1, S7, V0

Used from within an Integer BASIC program, it LOADs and RUNs the program named f, but does not clear the values of variables, so that the program f can operate on the results of the previous program, and can leave data for any following program. You cannot CHAIN APPLESOFT programs.

EXEC f [,Rr][,Ss][,Dd][,Vv]

Example: EXEC UTILITY, R5, V221

Similar to RUN, except that f is a file containing commands (including BASIC statements) as they would be issued from the keyboard. This allows you to set up files that control the APPLE much as you would yourself.

There can only be one EXEC command in effect at a time.

The parameter r specifies that the EXEC file should start executing from its r-th record (a record is a line, terminated by a carriage return), ignoring all those commands preceeding the r-th record. r defaults to zero. The remainder of the parameters are as for the other instructions. When an EXEC file has completed all its commands, it CLOSEs itself and stops. If f does not exist, a file of that name will be created.

LOCK f [,Ss][,Dd][,Vv]

Example: LOCK RECIPES, V31

This command allows you to make a file safe from accidental deletion or erasure. If f does not exist, a file of that name will be created.

UNLOCK f [,Ss][,Dd][,Vv]

Example: UNLOCK RECIPES, V31, D2

If you change your mind, and want to change a LOCKed file, this command allows such a change. If f does not exist, a

<2>DOS-F Page 28

file of that name will be created.

MAXFILES n

Example: MAXFILES 6

n is an integer from 1 to 16. This specifies the number of files that can be active at one time. When this command is executed 600 bytes of memory are reserved for each file. When you boot the system, n defaults to 3, so that you will have 1800 bytes reserved for file buffers and will be allowed a maximum of 3 files open simultaneously.

All DOS commands except PR#, IN# and MAXFILES require a file buffer. Thus if you have MAXFILES 1, and one file is OPEN, an attempt to perform a DOS command (such as CATALOG) will cause the message

\*\*\*DISK: NO FILE BUFFS AVAIL ERROR

APPEND f [,Ss][,Dd][,Vv][,Lj]

Example: APPEND FOO

This command performs an OPEN, but sets up pointers such that the next character written will follow the last sequentially written character presently in the file. If OPEN were used, the file would be overwritten from its beginning.

BSAVE f, Aa, Lj [,Ss][,Dd][,Vv]

Examples: BSAVE PICTURE, A16384, L8192

BSAVE PICTURE, A\$4000, L\$2000

Creates a file named f, and stores an image of a segment of the APPLE II's memory. The segment is specified by the starting address a, and the number of bytes to be stored j.

The examples store a high resolution picture, from the second high-resolution picture area. They are operationally identical.

BLOAD f [,Aa][,Ss][,Dd][,Vv]

Example: BLOAD PICTURE, A8192

This returns a file to the memory location from which the image was made by BSAVE. If a is specified then the data is placed in memory so as to begin at address a. Note that a machine language program may no longer be executable if so moved.

The example would place a high resolution picture into the first high-resolution picture area. It would also clobber the RAM version of APPLESOFT.

BRUN f [,Aa][,Ss][,Dd][,Vv]

Example: BRUN SUPER, A\$COA, V75

BLOADs the file f so as to begin at location a. If A is omitted, the file is BLOADEd to the same location from which it was stored. Once BLOADEd, the file (which should be a machine language program) is started by a machine language jump (JMP) to location a.

OPEN f, [,Lj][,Ss][,Dd][,Vv]

Example: OPEN SESAME, L2

Allocates a buffer of 600 bytes to the file f, and prepares the system to write or read from the beginning of the file. This command is used with the READ and WRITE commands, below.

j is the number of bytes in each record. This is only necessary for explicit random access to the specified file. must be in the range 1 to 32768, and defaults to 1.

If there is no file f, one is created.

CLOSE [f][,Ss][,Dd][,Vv]

Example: CLOSE WINDOW, S6

Deallocates the buffer associated with the file f. If f is not specified, all OPEN files will be closed, with the exception of the EXEC file. (There can only be one EXEC file OPEN at any time. When another is implicitly OPENed, the existing EXEC file, if any, is automatically closed)

Files that have been allocated by an OPEN statement must be CLOSED. Failure to CLOSE a file that was OPENed and written to (by a WRITE) can result in loss of data.

READ f [,Rr][,Bb]

Example: READ SESAME, R16, B3

This command causes any INPUT statements (and GET statements in APPLESOFT) executed subsequent to executing the READ to refer to the specified file rather than to the APPLE's keyboard. If r is given, it directs the program to begin READING after the r-th record in the file. r defaults to 0. If b is specified, READING will begin with the b-th byte after the r-th record.

The READ is cancelled by the printing of any DOS command. A null DOS command (just a {CTRL}D) at the beginning of a line) will cancel the READ.

It is not particularly useful to issue a READ in immediate mode (i.e. not as part of a program) since the first attempt to get a character from the file will simply CLOSE it.

WRITE f [,Rr][,Bb]

Example: WRITE ADDRESS.DATA, R3

The parameters have the same effect as in a READ, except that all characters that would normally be displayed on the screen are sent to the diskette instead.

The WRITE is cancelled by the printing of any DOS command. The null DOS command will do. It is as useful to issue a WRITE command as it is to issue a READ command in immediate mode.

POSITION f [,Rr][,Ss]

Example: POSITION ADDRESS.DATA, R277

Similar to a WRITE or a READ, except that it only positions a pointer to the indicated line, specified by r, counting from your current position in the file. This is a relative, and not an absolute position. A line is a sequence of characters terminated by a {RETURN}. Subsequent READs or WRITEs will proceed from that point in the file f.

NOMON [C][,I][,0]

Examples: NOMON C  
          NOMON I,C

All disk commands and all information sent between the computer and the disk are normally displayed on the screen. This command allows you to disable some or all of this display. If C is specified then disk commands are not displayed. If I is specified, then information being sent to the disk will not be displayed. If 0 is specified, then information being sent from the disk will not be displayed.

Note: in this command (and the MON command) at least one of the three parameters must be present, or the command is ignored. The parameters may appear in any order, separated by commas. These parameters do not appear in any other commands.

Further note: NOMON remains in effect until a MON command, a boot, or a restart (3DOG or 3D3G). Even RUNning a program won't cancel a NOMON.

MON [C][,I][,0]

Examples: MON 0  
          MON C,I,0

This command enables the display of information disabled by the NOMON command. See the Note on the NOMON command.

FP [,Ss][,Dd][,Vv]

Example: FP

This command puts you in APPLESOFT BASIC. Any BASIC program (in either BASIC) is lost. To get this ability onto a newly initialized disk, first LOAD APPLESOFT from the MASTER diskette, then (without RUNning or LISTing the file) SAVE APPLESOFT on an initialized diskette. You must use the name APPLESOFT for this file.

INT

Example: INT

This command puts you into Integer BASIC. Any BASIC program (in either BASIC) is lost. There are no parameters to this command (the typesetter is not getting lazy).



#### USE OF THE DISK OPERATING SYSTEM FROM WITHIN A PROGRAM

Any DOS command can be used from within a BASIC program. This is done by PRINTing a string that consists of a {CTRL}D followed by the command.

The recommended way of using this command is to first create a string D\$ which consists only of a {CTRL}D, and then to use BASIC statements such as:  
200 PRINT D\$;"WRITE CHESS"

A DOS command from a program must begin with the first character on a new line. DOS commands must end with a {RETURN}.

You cannot use the IN# and PR# commands in a program with either specie of BASIC without exiting from DOS. (They may be used in immediate mode.) To use an IN# or PR# (for example: to use a printer or modem through an interface board from a program) use the commands:

```
PRINT D$;"IN#n"  
PRINT D$;"PR#n"
```

where n is a numeral from 0 to 7. If the disk's slot is specified, a boot will occur, and any program in memory will be lost.

The following corrections and comments should help you use the APPLE DISK II.

1. If the cable from the disk drive to the controller card is not plugged into the controller's printed circuit card correctly, considerable physical damage can be done to the disk drive unit and its electronics. When the cable is plugged into the controller card correctly, the cable should exit from its connector on the side of the connector that is away from the controller's printed circuit card. You can tell that you are doing it incorrectly if the cable has to be jammed between the connector and the controller card. Secondly, make sure that all the pins of the cable's connector are going into holes in the controller card's connector. The best way to assure correct assembly is to plug the disk cable into the controller card before placing the controller card into the computer. That way you can actually see that all the pins are going into the holes correctly.

2. When DOS uses APPLESOFT BASIC, READ or WRITE statements on lines whose numbers are 256 or greater are not executed. If you have this problem in an APPLE II with more than 16K of memory, you can fix it with the following commands (you must temporarily remove the write-protect tab from the disk, if it has one):

(boot your system)

```
>UNLOCK RAWDOS
>BLOAD RAWDOS
>{RESET}
*25D6:4C D5 3F
*25DC:2E
*3FD5:E8 F0 01 60 4C DD 25
*3D0G
>BSAVE RAWDOS, A$1B00, L$2500
>LOCK RAWDOS
```

Any master or slave diskette created from a corrected diskette will now work properly with respect to this problem. If your APPLE II contains 16K or less memory, this fix will not work. Use a subroutine located at low line numbers for **READING** and **WRITEing**.

3. As stated in the manual, you must not type the command

**RUN APPLESOFT**

to get APPLESOFT BASIC. You must use the command

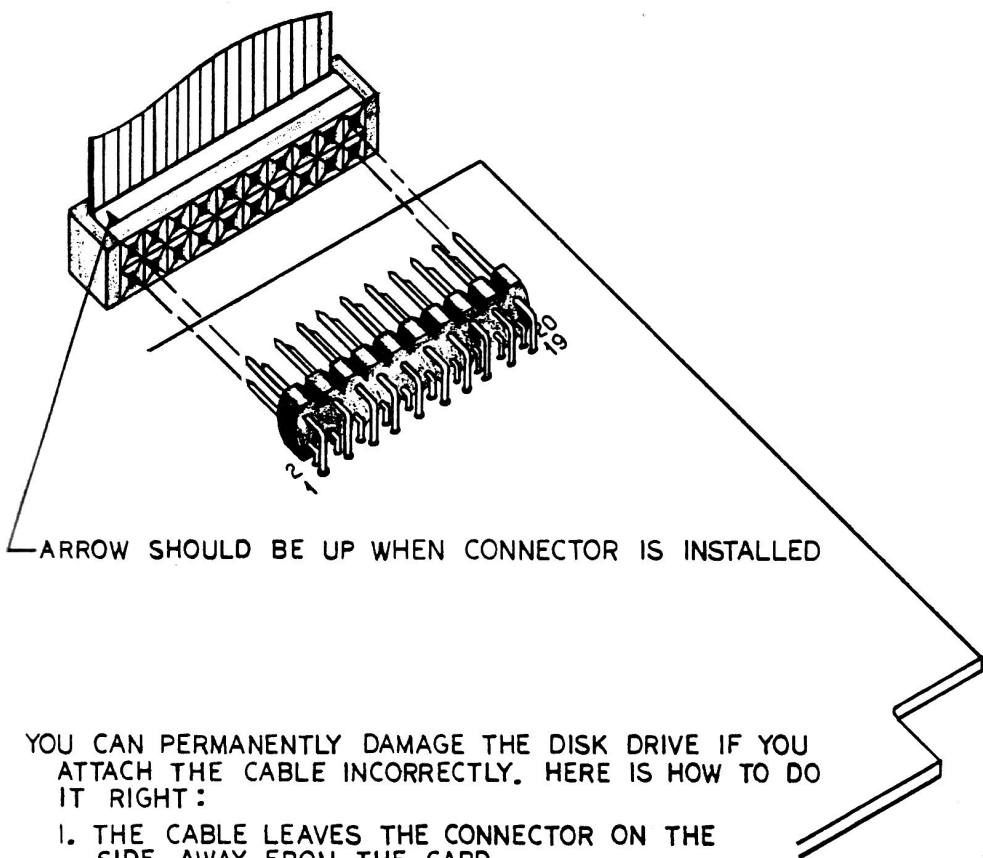
**FP**

which is not only easier to type, but works.

4. The COPY program will not work with just any pair of disk drives. It turns out that, while the speed of every disk shipped by APPLE is well within tolerance, the COPY program works only if the speeds of the two drives are matched to something like one-half of one percent. There is no easy cure if your drives are not matched this closely.

## WARNING !

PIN 1 **MUST** MATE AS SHOWN BELOW  
WHEN CONNECTING CABLE TO INTERFACE CARD



YOU CAN PERMANENTLY DAMAGE THE DISK DRIVE IF YOU  
ATTACH THE CABLE INCORRECTLY. HERE IS HOW TO DO  
IT RIGHT :

1. THE CABLE LEAVES THE CONNECTOR ON THE  
SIDE AWAY FROM THE CARD
2. MAKE SURE ALL PINS GO INTO THEIR MATCHING HOLES

5. The DOS takes up about ten thousand bytes of memory, with 600 additional bytes for each potential open file. The system normally sets aside room for three such potential files. To set the number of potential files, use the MAXFILES command. To gain a little more program operating room in memory, set MAXFILES to either 1 or 2, saving 1200 or 600 bytes respectively. With MAXFILES set to 1, you are rather limited as to what you can do: e.g. if you LOAD a program from one diskette, this opens one file, then if you try to INIT another diskette, the system tries to open a second file. Oops.

6. A Master Diskette, as supplied by APPLE, will run on any APPLE II with 16K or more memory. If you INIT a diskette, that diskette will only be able to boot on systems that contain at least as much memory as the system on which the diskette was INITed. Such a diskette is called a "slave" diskette. It can be used on systems of different memory size (for example, to LOAD and SAVE programs) with just two restrictions: it will not boot on systems smaller than the one on which it was INITed; and, when booted on larger systems, it will not take advantage of the full memory space. It will only use an amount of memory equal to that in the system on which it was INITed.

However, if you RUN MASTER.CREATE and use the INIT from within that program (in accord with the instructions provided on the screen when you RUN MASTER.CREATE), the diskette so created is a "master" diskette which will boot on any size system (16K or larger), and use all available memory. Therefore, it is generally a good idea to initialize all new diskettes with the MASTER.CREATE program.

7. A "master" diskette during boot will clobber the memory areas \$300 to 3FF, \$800 to 9FF, and \$1B00 to 3FFF. A "slave" diskette during boot will clobber the memory areas \$300 to 3FF, \$800 to 9FF, and the top 10K of memory where the DOS and file buffers normally reside.

7.5 A diskette holds slightly more than 100,000 bytes of user information.

8. The numbers that appear in the CATALOG for each file have only a dim relationship to the actual length of the program or data stored.

9. If your version of MASTER.CREATE does not work unless the controller is in slot 7, the following changes will fix the problem (you must temporarily remove the write-protect tab from your diskette, if it has one):

```
>UNLOCK MASTER.CREATE
>LOAD MASTER.CREATE
>97 X = PEEK (1528)
>100 PRINT "{CTRL}D BLOAD RAWDOS, A$1B00"
>105 POKE 14313, X: POKE 14327, X
>110 CALL 6912: END
>SAVE MASTER.CREATE
```

>LOCK MASTER.CREATE

If your MASTER.CREATE works fine, don't bother.

10. Do not use MAXFILES from within a program. It changes memory pointers, and a GOTO, GOSUB, or other instruction can get lost.

11. Remember that when READ or WRITE are set up for a particular record, they use the current byte number to look within the record. Usually, you want the byte pointer to be zero. Therefore, you should be careful to set it to zero explicitly.

12. The COPY program does not detect if the diskette being written to is write-protected. Everything will seem OK, but nothing will have happened: nothing copied, no error message. Sometimes, the COPY program may "hang," letting you know something is wrong.

13. The first disk command after a DISK: WRITE PROTECT ERROR will also garner the same error message whether or not the disk is now write-protected. You will get the error message even if the DOS operation attempted is not a write. The next legal disk command will execute normally.

14. When READING a string from a disk file, if the string is longer than 248 characters, a beep will sound for each character beyond the 248th. Ignore the beeping as much as you can.

15. CHAIN does not work with APPLESOFT.

16. CHAIN does not always work properly after you change the number of potential files (using MAXFILES).

17. When a file is LOCKed, the "length" number on the first program in the catalog may change. Ignore this change.

18. If a program is LOADED from disk into APPLESOFT (on the APPLESOFT firmware card), and you execute PRINT FRE(0) in immediate mode before you RUN the program, the APPLE goes bananas. Best to open the disk doors before you try this, as it can wipe out disks.

19. If an APPLESOFT program was SAVED on disk from a system not containing the APPLESOFT firmware card, it will not run on a system which does contain an APPLESOFT firmware card. To correct the program, perform the following steps (use the actual file name where the term "program name" appears):

```
]LOAD program name
]CALL 54514
]SAVE program name
```

From now on, it will run correctly. Similarly, an APPLESOFT program SAVED

on disk from a system with the APPLESOFT card will not run on a system which is not so provided. To correct the program, just

```
]LOAD program name  
]CALL 3314  
]SAVE program name
```

20. No deferred-execution DOS commands are carried out while in TRACE mode.

The information contained in this manual is believed to be correct at the time of publication, but Apple Computer Inc. assumes no liability arising from the use of this material.



10260 Bandley Drive  
Cupertino, California 95014  
(408) 996-1010