

---

# Commodore Business Computer

---

## User's Guide

---

Series 8000

---



 **commodore**

---

# Commodore Business Computer

---

## User's Guide

---

Appropriate for use with:

- Series 8000, 16K or 32K

Part Number 320894

---



© 1980 Commodore Business Machines, Inc.

# PREFACE

Suitable for use with BASIC Version 4.0

and

Models 2040, 3040, 4040

Dual Drive Floppy Disk Drives,

- DOS Version 1
- DOS Version 2

Suitable for use with BASIC Version 4.0

and

the Model 8050 Dual Drive Floppy Disk.

- DOS Version 2.5

# Table of Contents

Chapter 1	Introduction . . . . .	1
Chapter 2	The Commodore Business System . . . . .	5
	Preparing The CBM For Use . . . . .	9
	Turning The Computer On . . . . .	9
	Care Of Your CBM . . . . .	10
Chapter 3	Exploring the Keyboard . . . . .	11
	Character Reverse Key . . . . .	13
	RUN/STOP Key . . . . .	13
	Cursor Control Keys . . . . .	14
	Cursor Control Modes . . . . .	16
Chapter 4	Taking Command . . . . .	17
	Run Command . . . . .	18
	List Command . . . . .	19
	New Command . . . . .	19
	Save and Dsave Command . . . . .	20
	Verify Command . . . . .	20
	Load and Dload Command . . . . .	20
Chapter 5	Peripherals. . . . .	21
	IEEE-488 Interface . . . . .	21
	Parallel User Port. . . . .	23
	Memory Expansion Port . . . . .	23
	Commodore's Dual Drive Floppy Disk . . . . .	23
	Disk Commands . . . . .	25
	Formatting Your Disk . . . . .	25
	Header Command . . . . .	25
	Directory Command . . . . .	26
	Initializing Your Disk . . . . .	27
	2040 Disk Drive . . . . .	27
	8050 Disk Drive . . . . .	28

	Using the Disk Commands. . . . .	28
	Dsave "fn", Dx . . . . .	28
	Verify "dr:fn", 8. . . . .	29
	Verify "*", 8. . . . .	29
	Dload "fn", Dx . . . . .	30
	Dload "*" . . . . .	30
	Quick Load For Program Files . . . . .	31
	Relative Files. . . . .	31
	Creating A Random File . . . . .	34
	Accessing A Random File . . . . .	34
	User's Quick Reference: Commands . . . . .	34
	Cassette Tape Interface . . . . .	36
	Tape Commands . . . . .	37
	Save "fn" . . . . .	37
	Save "fn", 2 . . . . .	38
	Verify "fn" . . . . .	38
	Load "fn" . . . . .	39
	Load "fn", 2. . . . .	41
	Commodore Printer. . . . .	41
	Special Printer Associated Commands. . . . .	43
	Open . . . . .	43
	CMD . . . . .	43
	Print# and close . . . . .	44
Chapter 6	Error Messages. . . . .	47
	Errors Encountered While Trying To Load A Program . . . . .	48
	Device not Present. . . . .	48
	Load and Status Errors . . . . .	48
	Read Error . . . . .	48
	File not Found . . . . .	49
	DOS Mismatch . . . . .	49
	Disk ID Mismatch . . . . .	49
	Device Not Ready . . . . .	49
	Errors Encountered While Trying to Run A Program . . . . .	49
	Syntax Error . . . . .	49
	Input Error . . . . .	50
	Redo from Start . . . . .	50
	Device Not Ready . . . . .	51
	Errors Encountered While Trying To Save A Program . . . . .	51
	Verify Error . . . . .	51
	File Exists . . . . .	52
	Write Project On . . . . .	52
	Device Not Ready . . . . .	52
	User's Quick Reference: Error Messages . . . . .	52

# List of Illustrations

Figure	Title	
1	The Commodore Business System . . . . .	5
2	The Series 8000 Computer . . . . .	7
3	Rear View of Series 8000 Computer . . . . .	8
4	The Series 8000 Keyboard . . . . .	11
5	Series 8000 Character Display . . . . .	11
6	IEEE Peripheral Device Contact Points . . . . .	22
7	IEEE Interface Cables . . . . .	22
8	Model 2040 Dual Drive Floppy Disk . . . . .	24
9	Model 2040 Dual Drive Floppy Disk, Rear View . . . . .	24
10	Model C2N Tape Cassette . . . . .	36
11	Commodore Model 2022 Printer . . . . .	42
12	Commodore Model 2023 Printer . . . . .	42

# List of Tables

Table	Title	
1	Suggested Reading List . . . . .	2
2	Function Keys . . . . .	12
3	User's Quick Reference: Commands . . . . .	35
4	User's Quick Reference: Error Messages . . . . .	53

# Chapter 1

## INTRODUCTION

Congratulations on selecting the Commodore Series 8000 Computer. You will find that your computer is a flexible and easy-to-use tool—quite suitable for whatever applications you have in mind.

Commodore is the leading innovator and advocate in bringing personal computers into the business office—delivering both professional capability and results. Your computer has been thoroughly tested and certified by Commodore prior to leaving the factory. You will find that your computer can be immediately on-line and ready to go to work.

This guide discusses the computer's features in an easy-to-understand non-technical manner. This approach will enable you to become familiar with your computer's basic capabilities without being intimidated by technical sounding jargon.

This guide is comprised of the following chapters:

- INTRODUCTION
- THE COMMODORE BUSINESS SYSTEM
- EXPLORING THE KEYBOARD
- TAKING COMMAND
- PERIPHERALS
- ERROR MESSAGES

The information presented in these chapters will allow you to:

- unpack and install the computer,
- attach computer peripherals,
- use commercially prepared software, and
- recognize and correct error conditions.

Commodore makes no assumptions as to the user's expertise, but does recommend the SUGGESTED READING LIST, Table 1, which, should the user wish to pursue it, can provide detailed programming and technical information. Additionally, your Commodore dealer can assist you in expanding your knowledge and understanding, thereby enhancing your proficiency.

The best teaching device for learning how the computer works and what it can do is the computer itself. In some cases you will be instructed to use the computer in conjunction with specified procedures in addition to reading what to do. Every effort has been made to provide sufficient documentation to allow you to get started. Essentially, then, the rest is up to you. By constructing similar, and then progressively more sophisticated procedures and examples on your own, any questions which might arise will in all probability be answered by the machine itself.

The Series 8000 is a complete computer and its potential is virtually limitless. Information is entered via the keyboard and displayed on the CRT. The computer is programmable using a language called Beginners All-purpose Symbolic Instruction Code (BASIC). BASIC is an English-like computer language designed to permit you and the computer to understand and communicate with each other in a friendly and non-threatening way.

At the heart of your computer is a microprocessor. The microprocessor totally controls the operation of the CRT, keyboard and assorted peripherals. Compatible Commodore peripherals include:

- Model 2040 Dual Floppy Disk Drive
- Model 8050 Dual Floppy Disk Drive
- Model 2022 or 2023 CBM Printer
- Model C2N Cassette Recorder

Table 1. Suggested Reading List

Entering Basic J. Sack and J. Meadows, <i>Science Research Associates</i> , 1973
BASIC: A Computer Programming Language C. Pegels, <i>Holden-Day, Inc.</i> , 1973
BASIC Programming J. Kemeny and T. Kurtz, <i>Peoples Computer Co.</i> , 1967 1010 Doyle (P.O. Box 3100), Menlo Park, CA 94025
BASIC Albrecht, Finkle, and Brown <i>Peoples Computer Co.</i> , 1973, 1010 Doyle (P.O. Box 3100), Menlo Park, CA 94025
A Guided Tour of Computer Programming in BASIC T. Dwyer, <i>Houghton-Mifflin Co.</i> , 1973
Programming a Time Shared Computer in BASIC E. H. Barnett, <i>Wiley-Interscience</i> L/C 72-175789
Programming Language #2 <i>Digital Equipment Corp.</i> , Maynard, MA 01754
101 BASIC Computer Games Software Distribution Center, <i>Digital Equipment Corp.</i> , Maynard, MA 01754
What to Do After You Hit Return <i>Peoples Computer Co.</i> , 1010 Doyle (P.O. Box 3100), Menlo Park, CA 94025



Table 1. Suggested Reading List (cont.)

Basic BASIC

J. S. Coan, *Hayden Book Co.*, Rochelle Park, NJ

Workbooks 1-5

*T.I.S.*, (P.O. Box 921), Los Alamos, NM

BASIC With Style

*Hayden Book Co.*, Rochelle Park, NJ

Programming Proverbs

*Hayden Book Co.*, Rochelle Park, NJ

PET and the IEEE Bus (GPIB)

*Osborne/McGraw-Hill*, 630 Bancroft Way, Berkeley, CA 94710

## NOTES

## NOTES

# Chapter 2

## THE COMMODORE BUSINESS SYSTEM

Your computer is the brain of the Commodore business system, and, in conjunction with the peripherals is designed to:

- Store information
- Find stored information
- Update stored information
- Calculate
- Manipulate information
- Print information

Your computer is constructed in such a way that it cannot be damaged via the keyboard. The operating system cannot be destroyed because the computer software, or operating instructions, are contained in a fixed memory called a Read-Only-Memory (ROM). This feature allows the user to use the computer with impunity.

Just what is the Commodore business system? See Figure 1. The system consists of the following components:

- Computer
- Storage Device
- Printer
- Software
- Data file

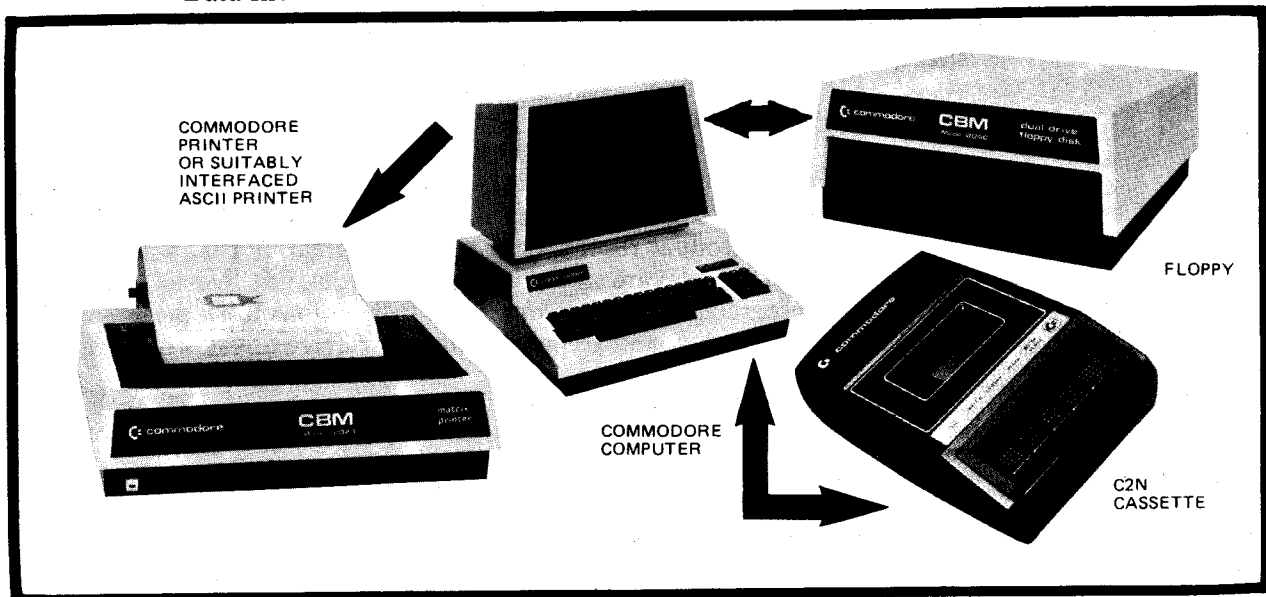


Figure 1 — The Commodore Business System

Let's examine each of these components in more detail.

**Computer**

At the heart of the system is the computer. It has two functions. First, it controls the other devices in the system. Second, it performs calculations, manipulates information, and displays the results on a CRT video screen in front of the user. See Figure 2.

**Storage Device**

There are two types of storage devices available for the system — a cassette tape drive and a floppy disk drive. Both of these devices have two functions. They are used for both the storage and retrieval of information under program control. More will be said about these devices in the chapter titled "Peripherals".

**Printer**

A printer is used to output information onto paper. Commodore offers two types of printers for the business system. These will be discussed more in the chapter titled "Peripherals".

**Software**

Software is the term used to describe the collection of programs which make the system perform business oriented work. A program is basically a series of instructions which is given to the computer to make it perform specific tasks. So, we write a program for each business task we want the system to perform. The collection of all of these business programs is the system's business software.

Where does this software come from? You can invent your own, but this requires a lot of work and a great deal of knowledge about computers and writing computer programs. Usually, you will purchase prepared software. (Commodore offers a wide selection of software for its business system.) This software usually comes on a storage medium that is compatible with your system's storage device.

**Data Files**

Data files are simply stored information. It can be information that a program needs to produce results, or it can be information that a program produces. Data files are stored on storage media, just like software. Unlike programs, data files contain no instructions to be performed by the computer and are not commonly referred to as programs.

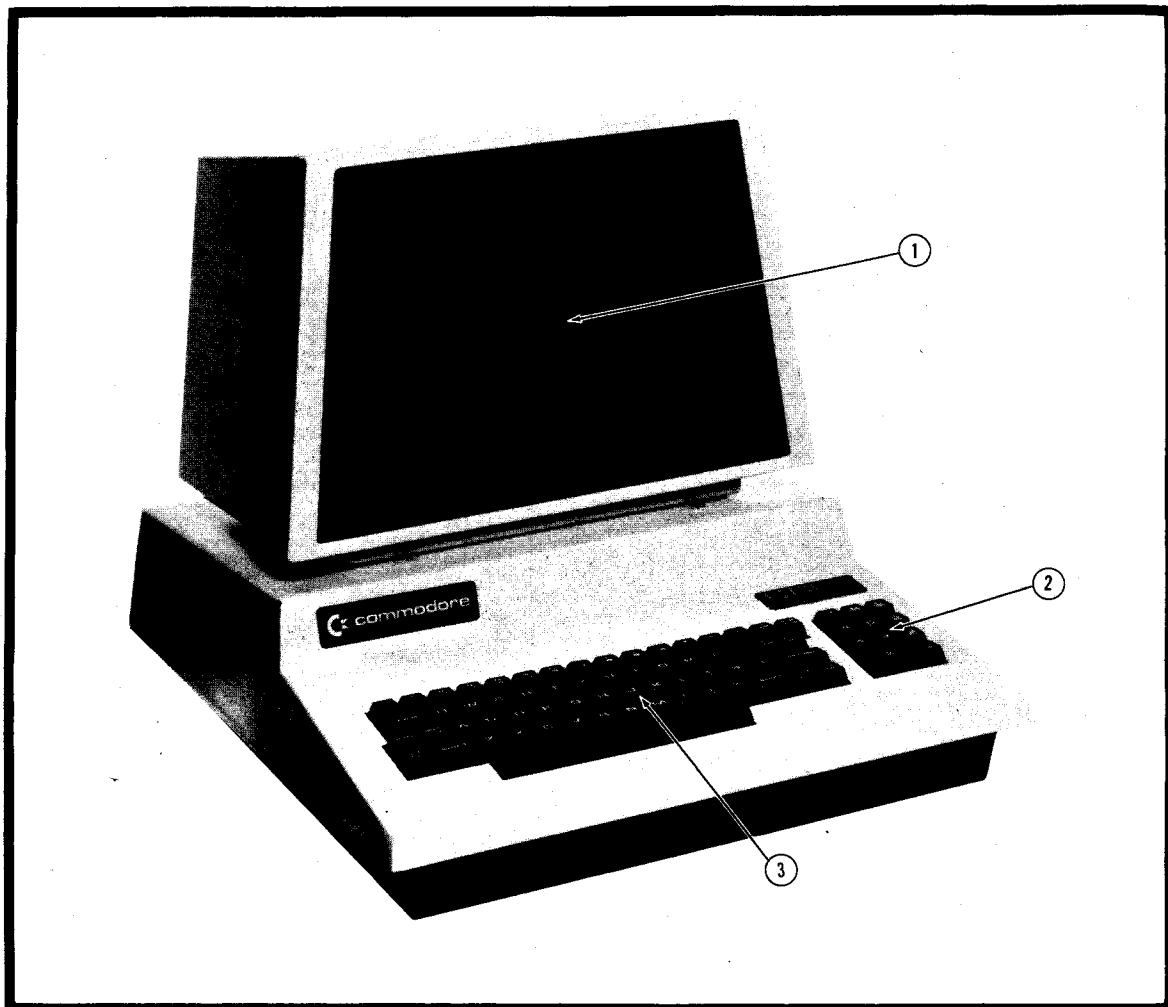


Figure 2 — The Series 8000 Computer

1. CRT Screen—Twelve inch green-and-black video display presents up to 25 lines of information, each line being 80 characters long.
2. Numeric Keypad—For quick entry of numbers.
3. Typewriter-style Keyboard—Standard typewriter keyboard for easy entry of upper and lower case alphanumeric characters.

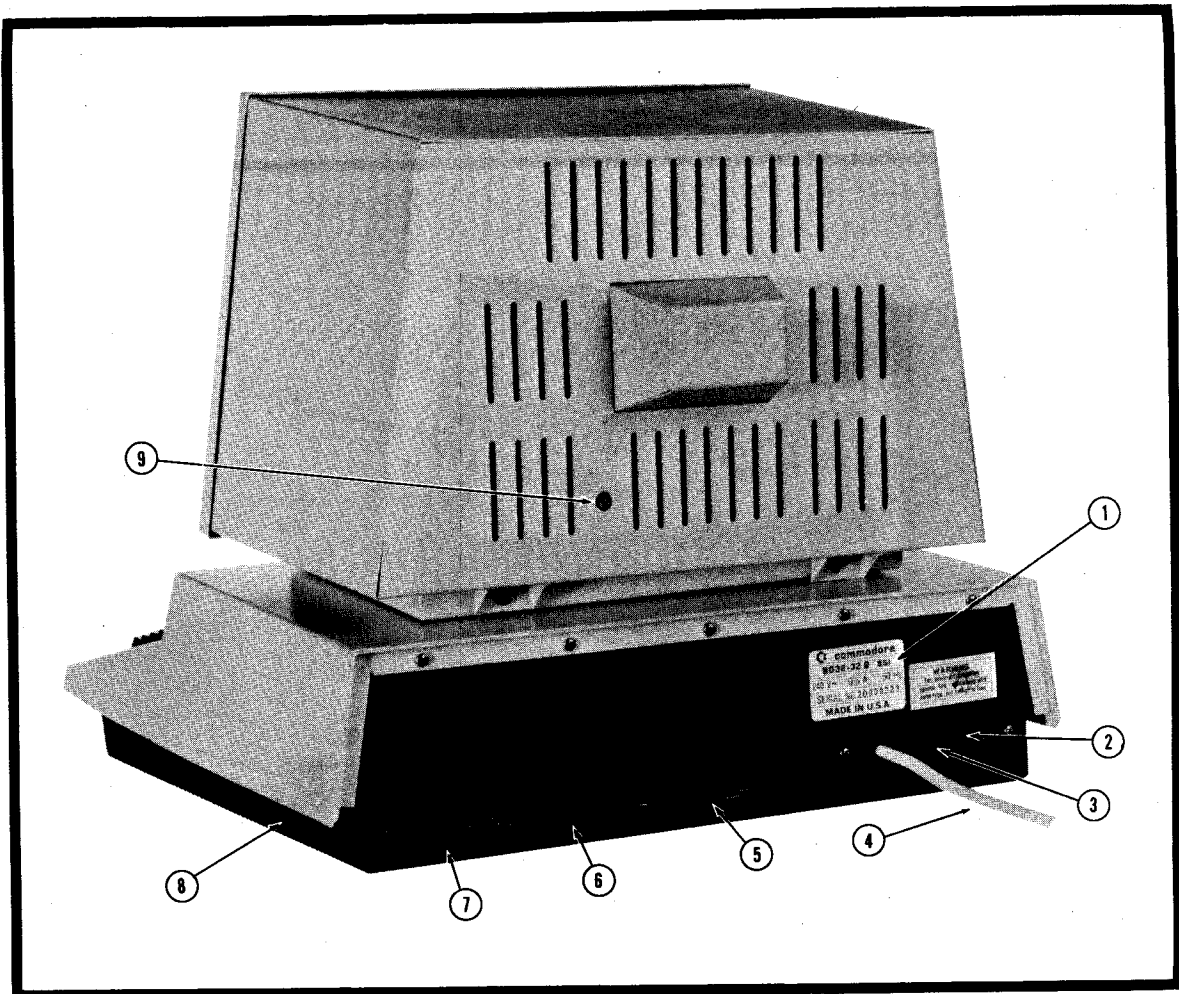


Figure 3 — Rear View of Series 8000 Computer

1. Serial Number and Electrical Specifications—Valuable information.
2. Power Switch—Turns the computer off and on.
3. Fuse—Protects the computer.
4. Three-wire AC Power Cord—Provides safe grounded power to the computer.
5. IEEE 488 Interface—For connecting intelligent IEEE 488 peripherals.
6. Parallel User Port—For connecting non-intelligent peripherals and user-built devices.
7. Primary Cassette Interface—For connecting a tape cassette recorder to save or load programs and data.
8. Secondary Cassette Interface—For connecting a second tape cassette recorder for more advanced data handling capability.  
Memory Expansion Port—For adding additional memory to the computer.
9. CRT Brightness Adjustment—Makes the screen brighter or dimmer.

## PREPARING THE CBM FOR USE

### Unpacking and Installation

- STEP 1: Check the carton for any special unpacking instructions.
- STEP 2: Open the top of the carton and carefully remove as much of the protective styro-foam material as possible.
- STEP 3: Remove the computer from the carton and place it on a sturdy, level surface such as a table or desk.
- STEP 4: Carefully remove any protective plastic covering which may be taped to the computer.
- STEP 5: Examine the computer for any damage. If anything is amiss, report it immediately to both the place of purchase and the shipping agent.
- STEP 6: Locate the power switch. Make sure it is in the "OFF" position. If there is a dot visible on one end of the switch then it is in the "ON" position.
- STEP 7: Ensure that the voltage of your outlet matches the computer's requirement of 115-volts.
- STEP 8: Unwrap the power cord and plug it into a 3-wire, grounded AC outlet.

#### NOTE

For safety, a 3-wire, grounded outlet must always be used.

## TURNING THE COMPUTER ON

When the computer has been properly installed, turn the power switch on the back to the "ON" position. After a few seconds you will see a collection of random letters and symbols on the CRT. This is normal. When the computer is turned on, its electrical circuits have to wake up slowly before they can clear the screen. The screen will soon clear and the following message will appear on the screen.

```
### commodore basic ###  
  
31743 bytes free  
  
ready  
█
```

If the screen does not clear or any other messages appear, turn the power switch to "OFF", wait 5 seconds, and turn the switch "ON" again. If the appropriate message still does not appear, turn the computer "OFF" and notify the place of purchase immediately.

The three-line message that appears on the screen when power is applied is interpreted as follows:

- Line one indicates that the computer will respond to BASIC language demands.
- Line two indicates the amount of memory space that the computer has to store information.
- Line three means the computer is "ready" to work.

The blinking square at the bottom is called the "cursor" and indicates where the next character will appear on the screen.

## CARE OF YOUR CBM

The computer requires almost no care. Only occasional cleaning of the case, keyboard and screen should be necessary. Use a damp, not wet, cloth to wipe off any dirt. Do not use spray cleaners, since the mist from the nozzle may get inside the computer and cause damage.

If anything should ever go wrong with your computer, take it to an authorized Commodore dealer.

## NOTES



# Chapter 3

## EXPLORING THE KEYBOARD

You can not communicate effectively with the computer until you have explored the keyboard. Actually, the computer has two keyboards. One looks very much like a typewriter keyboard, the other is a calculator-style keyboard used to rapidly enter numbers. See Figure 4.

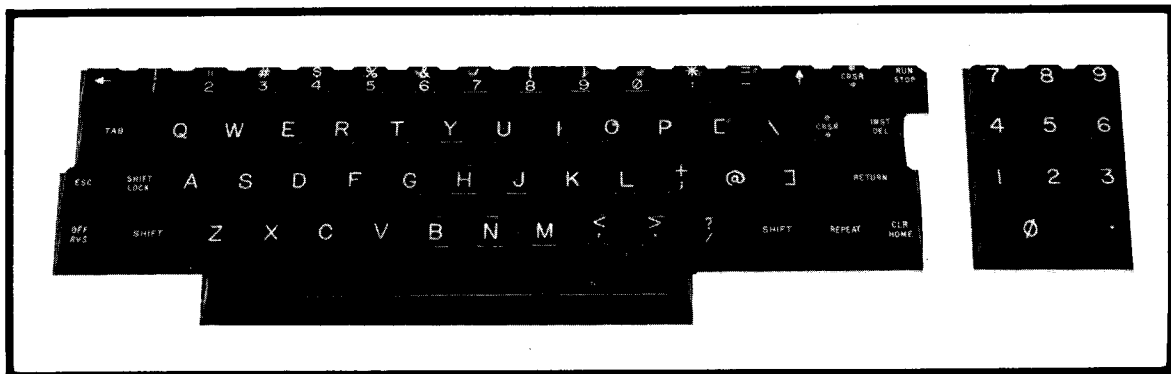


Figure 4 — The Series 8000 Keyboard

Most of the keys on the typewriter-like keyboard look and function just like their typewriter counterparts except they cause characters to be displayed on the CRT rather than printed on a piece of paper.

Figure 5 shows the different characters the keyboard causes to be displayed on the CRT.

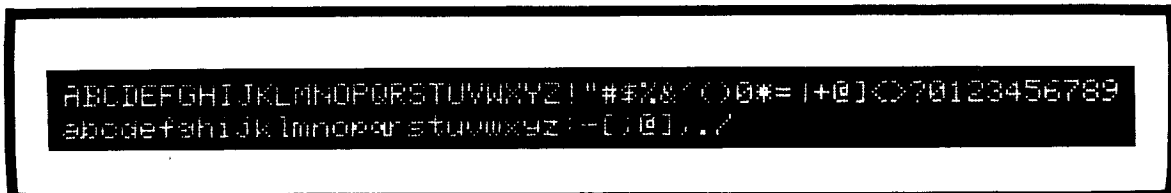


Figure 5 — Series 8000 Character Display

As mentioned earlier, most of the keys function like those on a typewriter. The exceptions to this are the function keys. The function keys are described briefly in Table 2.

Table 2. Function Keys

KEY	NOT SHIFTED	SHIFTED
<b>CLEAR/ HOME</b>	Moves cursor to upper left corner of screen.	Clears display screen and moves cursor to upper left corner of screen.
<b>CURSOR UP/DOWN</b>	Moves cursor down one line. When cursor is at the bottom of the screen, print will move off top of screen. Does not affect characters as cursor passes over them. Automatically repeats if held down.	Moves the cursor up one line. Will not go off top of the screen. Does not affect characters as the cursor passes over them. Automatically repeats if held down.
<b>CURSOR LEFT/RIGHT</b>	Moves cursor one space to the right. Will move to beginning of next line when it hits the edge of the screen. Does not affect characters as the cursor passes over them. Automatically repeats if held down.	Moves cursor one space to the left. Moves to the end of the preceding line when it hits the edge of screen. Does not affect characters as the cursor passes over them. Automatically repeats if held down.
<b>INST DEL</b>	Deletes the character immediately to the left of cursor. All characters to right of deletion are moved one space to the left. Automatically repeats if held down.	Inserts a space at cursor position. All characters move one space to the right. Automatically repeats if held down.
<b>OFF RVS</b>	Prints black characters on green background.	Resets characters to normal green-on-black printing.
<b>RETURN</b>	Sends information on screen preceding cursor to the computer for processing and moves cursor to the beginning of the next line.	Moves the cursor to the beginning of the next line without sending information to the computer.
<b>RUN STOP</b>	Stops the function of the computer, whatever it is doing (i.e., loading, running).	Loads and runs the first program from drive 0 on the disk.
<b>TAB</b>	Move cursor to next set tab space.	Set or clear tab at cursor position.
<b>ESC</b>	ESCAPE from quote or insert mode.	Same as not shifted.
<b>REPEAT</b>	Causes any key held down simultaneously to repeat.	Same as not shifted.
<b>SPACE</b>	Prints a blank space at the cursor position. Automatically repeats if held down.	Prints a shifted space at cursor position. A non-printing graphic character appears on screen. Automatically repeats if held down.

## CHARACTER REVERSE KEY

Normally, the screen displays a lighted character (green) on a dark background (black). Using the OFF/RVS key you can highlight information on the screen by displaying black-on-green, rather than green-on-black. Try the following example.

**STEP 1:** Turn the computer ON.

**STEP 2:** Clear the screen using SHIFT, CLR/HOME.

**STEP 3:** Type: AAA.

**STEP 4:** Press the OFF/RVS key.

**STEP 5:** Type: BBB. (Notice that the Bs are displayed in reverse: black-on-green.)



**STEP 6:** Hold the SHIFT key down and press the OFF/RVS key. This returns the display to its normal green-on-black. To check this:

**STEP 7:** Type: CCC.



As stated earlier, a program is a series of instructions that the computer performs to produce a specified result. Just as a chef mixes ingredients following a recipe step-by-step to create a meal, so does the computer execute programmed instructions, step-by-step, to produce meaningful results. The order in which the computer performs the instructions is contained in the program itself since each instruction has a line number preceding it. The line number serves the function of identifying the exact order in which to execute a program. The line number also tells the computer to store the information in memory as part of a program. Each new program line is put in proper numerical sequence with the program lines previously stored in memory. Program lines are stored until the user tells the computer to execute the program. If an instruction is not preceded by a line number, the computer will execute it immediately and not store it in the program text area.

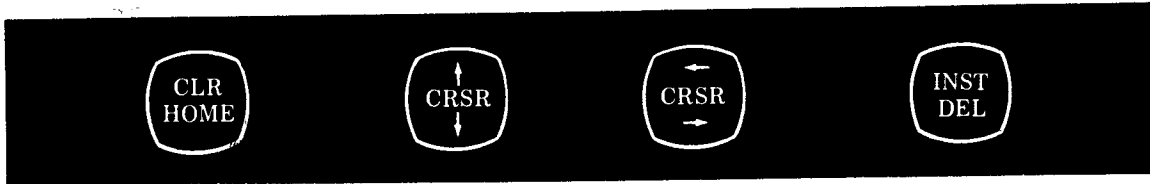
## RUN/STOP KEY

When shifted, the RUN/STOP key is used to load a program into the computer's memory from a diskette rather than entering instructions, line-by-line, from the keyboard. The use of this key will be more fully discussed in the chapter titled "Peripherals".

Unshifted, this key functions as a break key. That is, it interrupts whatever the computer is doing, such as loading a program, or executing one, and returns it to user control. The screen will display "ready." and the cursor will await your command.

## CURSOR CONTROL KEYS

Commodore Computers have a feature called "screen editing" which uses cursor control keys to move the cursor to any position on the screen and insert, delete, or retype characters. Characters which are already on the screen will not be affected by the cursor's movement. Four of the function keys are used in screen editing. These are:



All of these keys have two functions. Their action depends on whether or not the SHIFT key is pressed in conjunction with the screen editing key. Let's examine the screen editing keys and their functions through the use of examples.

**STEP 1:** Turn your computer ON.

**STEP 2:** Press the CLR/HOME key. Notice that the cursor moves to the top left corner of the screen.

**STEP 3:** Hold the SHIFT key down while pressing the CLR/HOME key. This action erases all of the characters that were present.

**STEP 4:** Now type the alphabet followed by the numbers 0 through 9. End the line with asterisks—\*. Notice that after you type the last "\*", the cursor moves to the beginning of the next line. This is called "wrap around". You'll notice that there are 80 characters on the first line—26 alphabetic, 10 numeric, and the remainder asterisks.

If you make a mistake while typing, press INST/DEL to remove the error and retype the correct character.

```
ABCDEFGHIJKLMN OPQRSTUVWXYZ0123456789*****  
*****
```

You'll notice that there are 80 characters on the first line—26 alphabetic, 10 numeric and the remainder asterisks.

**STEP 5:** Now hold the SHIFT key down and press the CRSR LEFT/RIGHT key. This moves the cursor to the left. Notice how the cursor "wraps around" back to the end of the first line. When the cursor reaches the end of the first line, release the CRSR LEFT/RIGHT key. Press the INST/DEL key and delete all the asterisks on the first line. Now move the cursor left until it rests on the zero. As the cursor moves, it does not affect the characters it passes over.

```
ABCDEFGHIJKLMN OPQRSTUVWXYZ0123456789  
*****
```

**STEP 6:** Now hold the SHIFT key down and press the INST/DEL key four times. This will insert four spaces before the zero and move the text four spaces to the right of the cursor.

```
ABCDEFGHIJKLMN OPQRSTUVWXYZ 0123456789
*****
```

**STEP 7:** Type four asterisks into the space created by the INST/DEL key. We have now inserted the arithmetic operator symbols between the alphabet and the numbers.

```
ABCDEFGHIJKLMN OPQRSTUVWXYZ**** 123456789
*****
```

**STEP 8:** Press the CRSR LEFT/RIGHT key until the cursor wraps around to the next line. Position the cursor just after the last “\*”.

```
ABCDEFGHIJKLMN OPQRSTUVWXYZ****0123456789
*****
```

**STEP 9:** Use the INST/DEL key to delete the asterisks on the second line.

```
ABCDEFGHIJKLMN OPQRSTUVWXYZ0123456789

```

**STEP 10:** Clear the screen and move the cursor to the top line by holding the SHIFT key down and pressing the CLR/HOME key twice.

**STEP 11:** Type: “This is the top.” Press a shifted RETURN (so the computer won’t think you’re trying to tell it something) and type: “This is the next line.”

**STEP 12:** We’ve just confirmed that 80 characters can be displayed on each line. Let’s see how many lines there are. Counting each keystroke as one, press the CRSR UP/DOWN key. The cursor will move down the screen, reaching bottom on the 24th keystroke. To prove this, press the CRSR UP/DOWN key a 25th time. You will see that the top line of the screen moves up and off the screen. The line “This is the next line” is now the top line. This action of inserting lines at the bottom and moving old text up is called “scrolling”.

```
THIS IS THE TOP
THIS IS THE NEXT LINE
```

We started on line one and used 24 keystrokes to reach the bottom of the screen. This means the screen will display 25 lines at one time before scrolling.

**STEP 13:** Move the cursor back down to the very bottom of the screen and type: “This is the bottom”.

**STEP 14:** Press the CRSR UP/DOWN key several times. You will observe that the text you typed at the bottom of the screen will move up toward the top of the display. The screen only scrolls from the bottom up. Prove this by pressing CLR/HOME to move the cursor to the top of the screen. Try scrolling the screen down by holding the SHIFT key down and press the CRSR UP/DOWN key repeatedly. Notice that this does not affect the screen.

THIS IS THE BOTTOM ■

## CURSOR CONTROL MODES

There are two modes of cursor control:

- direct, and
- programmed control

While under direct control, the cursor can be moved or editor function altered as soon as a key is depressed. While under programmed control, cursor movement can be programmed to occur during the execution of a program.

Programmed control is achieved by either of two means: insert mode or quote mode. Typing an odd number of quotation marks before the cursor places the screen editor into quote mode. While in quote mode, all cursor control keys with the exception of RETURN and DELETE keys will print a reverse field character when keyed. Limited editing can still be performed with the DELETE key. Exit from quote mode is accomplished by typing another quote or pressing the RETURN key. The ESC key, if activated on your computer, will also allow an exit back to direct cursor control from either quote or insert mode.

Entry to insert mode is gained by keying INST one or more times. No editing functions are available for direct control while in insert mode and the number of characters that can be inserted are limited to the number of times INST was keyed. Exit from insert mode is accomplished by pressing either the RETURN or ESC key.

Twenty control keys and eleven control functions in screen editor provide a powerful instruction set that allows the user to perform editing techniques such as:

- changing the limits of the scrolling window
- deleting entire lines
- ringing a carriage return bell at the end of a line
- using programmed cursor movement to create graphic figures on the screen
- modifying programs by simply typing over the listed line

and more.

## NOTES

# Chapter 4

## TAKING COMMAND

Certain BASIC commands are used to manipulate programs in the computer. Normally, a command is used after the computer has printed "ready" on the screen followed by the cursor. The commands are entered into the memory by typing the command and pressing the RETURN key.

The computer is very sensitive about how the commands are worded. In BASIC programming language, words must be spelled and used correctly or they make no sense. The grammatical rules governing the correct usage of the BASIC language are called syntax.

Syntax dictates the correct spelling and usage of the BASIC instruction set. If an operating system command is entered into the memory which does not obey the BASIC language syntax, the computer will reject it as useless data by printing "? syntax error" on the screen. Try the following:

**STEP 1:** Clear the screen. (By simultaneously pressing the SHIFT and CLR/HOME keys.)

**STEP 2:** Type: CBM.

**STEP 3:** Press RETURN.

Since CBM is not one of the system operating commands, the computer will respond:

```
CBM
?syntax error
ready.
█
```

In order to correct a syntax error, you must re-enter the same command using correct syntax.

Now let's look at the eight operating system commands and learn their forms and function. The commands are:

run	new
list	verify
dload	load
dsave	save

NOTE: Commands must be entered using lower case (unshifted) characters.

Each command will be defined in terms of its function. An example will be given for each command to clarify its function and syntax. These examples are in the form of a short program that you can enter into the computer via the keyboard. Type each line of the example exactly as it is worded, remembering the importance of correct syntax to the computer. Use the screen editing functions to correct each line if necessary. After each line has been typed correctly, press RETURN to enter the line into the computer's memory. For example, if you are asked to:

Type: print a

you would type "print a" and press RETURN.

## RUN COMMAND

The RUN command begins the execution of a program stored in the computer's memory.

STEP 1: Clear the screen.

STEP 2: Type: 1 print "Hello!"

STEP 3: Type: 2 print "I'm your new CBM."

STEP 4: Type: 3 print "So long for now."

STEP 5: Type: 4 print "\*\*\*\*\*"

STEP 6: Type: run

At this point the computer will execute the program starting with line 1. The screen will display:

```
1 print "Hello!"
2 print "I'm your new CBM."
3 print "So long for now."
4 print "*****"
```

```
run
```

```
Hello!
I'm your new CBM.
So long for now.
*****
```

```
ready.
```





## LIST COMMAND

The LIST command will display the entire content of the program contained in the computer's local memory. However, in order to use the LIST command to review a program contained in either a disk or tape file, you will first have to bring the contents of those files into the computer's local memory using either the LOAD or DLOAD command, discussed below.

STEP 1: Clear the screen.

STEP 2: Type: list

The computer will display the program you entered earlier.

```
list
1 print "Hello!"
2 print "I'm your new CBM."
3 print "So long for now."
4 print "*****"

ready.
█
```

## NEW COMMAND

The NEW command erases the current program from the computer's local memory, but not from the memory of any peripheral device such as the floppy disk drive or cassette tape drive. If you do not use the NEW command, a program will remain in local memory until the power is turned off, or until new information is loaded in from the keyboard or peripheral device.

STEP 1: Clear the screen.

STEP 2: Type: new

STEP 3: Type: list

The program we have been able to display using the LIST command has now been erased by the NEW command. When we try and list the program we find it is no longer resident in computer local memory:

```
new
ready.

list

ready.
█
```

## SAVE AND DSAVE COMMAND

The SAVE command is important to the user since the computer forgets everything in its memory when the power is turned off. That's a good reason to attach peripheral storage devices such as tape or disk drives in order to save valuable programs and data files.

When storing data on the cassette tape drive we use the SAVE command. The DSAVE command is used when you wish to store information on a diskette or, as it is more commonly known, a floppy disk. Without this command, you would have to reenter information from the keyboard each time you wanted to use it. This command, when properly used, will permit you to save information and have it available for recall at any time.

The computer can store two types of information: programs and data.

- Programs have already been defined as a series of BASIC instructions.
- Data are information that the computer may need for a BASIC program such as a series of numbers or information that the computer has produced as the result of a program.

Data must be saved through program control while programs may be saved with either the SAVE or DSAVE command.

The computer stores both types of information as an information file: either a program file or a data file. In addition, data files may be organized as sequential or relative access files. The SAVE command will be discussed more fully when we talk about peripheral devices.

## VERIFY COMMAND

The VERIFY command ensures that information transmitted from the computer to a storage medium has been copied exactly as it was stored in computer local memory. It does this by comparing the information in the computer's memory to that information recorded on the storage medium and verifies that it is correct. Since VERIFY command is used in conjunction with peripheral storage devices, it will be more fully described in the next chapter.

## LOAD AND DLOAD COMMAND

The LOAD command directs the computer to transfer a program from tape into the computer's memory. The DLOAD command enables you to enter programs of information into the computer's memory directly from the floppy. These commands save you the time and effort required to manually enter the program into the computer using the keyboard. The LOAD and DLOAD commands are useful for entering software which someone has already written and has been made available on either tape or disk medium. Since these commands deal with peripheral devices, it will be discussed in greater detail in the next chapter.

# Chapter 5

## PERIPHERALS

Peripheral devices, or peripherals, are simply pieces of equipment that are attached to a computer to allow it to perform some specific function. Examples of peripherals are information storage devices, printers, heat sensors, and TV cameras.

In order to make the CBM computer as flexible and versatile as possible, it has been designed to connect to many different types of peripherals. Different peripherals require different types of connectors and electrical signals to make them work correctly. A kind of electrical bridge must be provided between the computer and its peripherals. This bridge is called an "interface". Your CBM has four types of interfaces:

- IEEE-488 Interface
- Parallel User Port
- Tape Cassette Interface
- Memory Expansion Port

### IEEE-488 INTERFACE

The computer can be connected to IEEE-488 devices through the IEEE-488 interface. This interface is located on the back of the computer. See Figures 6 and 7. IEEE-488 is the designation given to a particular interface method by the Institute of Electronic and Electrical Engineers (IEEE).

All interface cables connect to the main logic printed circuit board located inside the computer housing. Openings in the rear and righthand side panels of the computer housing expose the press fit connection points. See Figure 6.

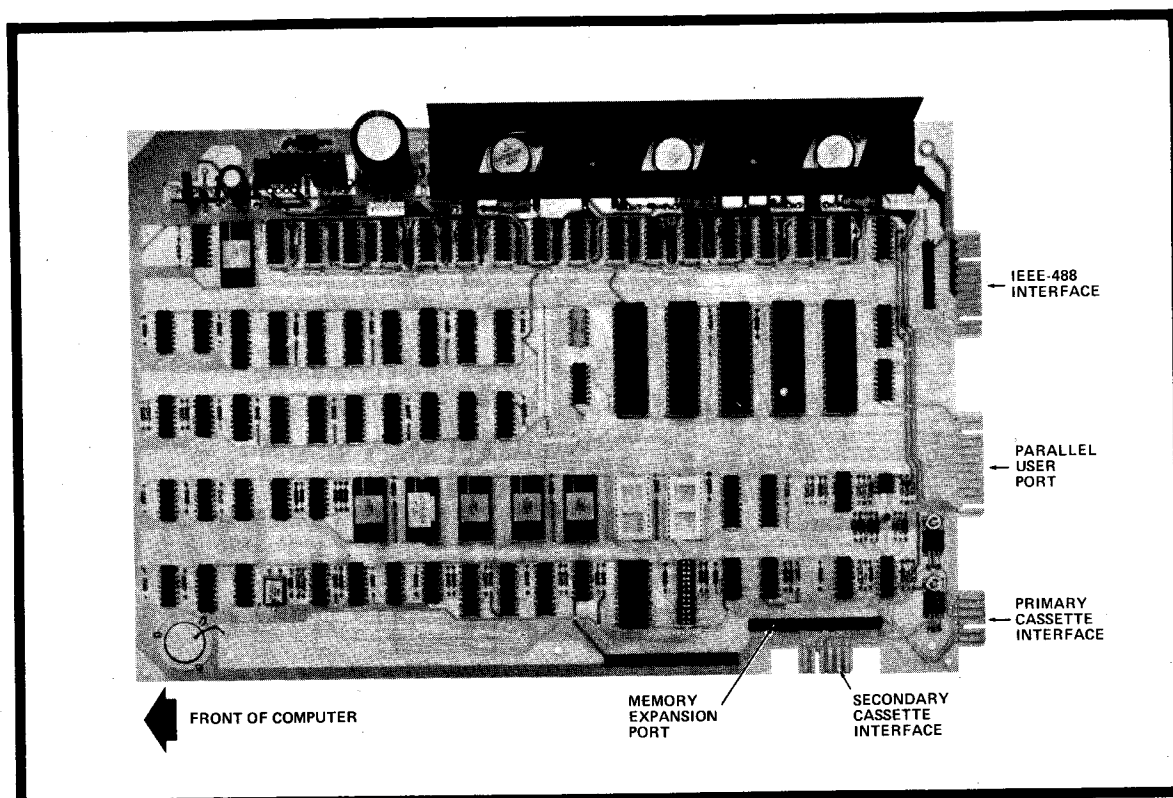


Figure 6 — IEEE Peripheral Device Contact Points

The IEEE-488 interface can be used to attach up to 15 IEEE-488 devices to the computer. Each IEEE-488 peripheral has a unique device number assigned to it. The computer “talks” or “listens” to devices connected to the IEEE-488 interface by addressing them by their device number. For example, suppose two devices are connected to the computer’s IEEE-488 interface. One has device number 4 and the other has device number 8. If the computer sends a signal to device 4, both peripherals receive the signal, but only device 4 would respond.

To connect several peripherals to one plug, a special PET-to-IEEE-488 cable is used. It is pictured in Figure 7(B). This cable can be plugged into the computer and have an IEEE-to-IEEE cable plugged into it piggy-back fashion. In this way, up to 15 IEEE-488 peripherals may be attached to the computer at the same time.

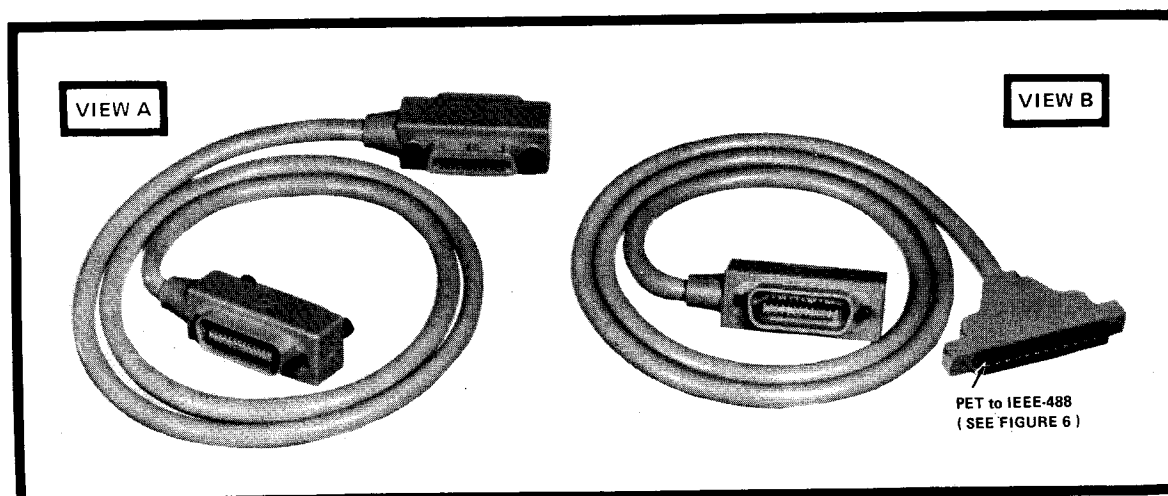


Figure 7 — IEEE Interface Cables

A PET-to-IEEE cable (P/N 320202), Figure 7(B), must be used to connect a peripheral to the computer. This interface cable is available through your Commodore dealer.

NOTE: The IEEE-to-IEEE (Figure 7(A)) cable can not be used to connect the computer to a peripheral. It is used only to connect peripherals to each other.

## PARALLEL USER PORT

User built devices and many non-IEEE-488 devices can be connected to the parallel user port. See Figure 6. Normally, the port will accept only one peripheral at a time. The interface software for the user port must be supplied by the user.

## MEMORY EXPANSION PORT

The memory expansion port can be used to interface additional memory to the computer. See Figure 6.

## COMMODORE'S DUAL DRIVE FLOPPY DISK

The Commodore dual drive floppy disk unit (Model 2040) is IEEE-488 device number 8. It is used for storing information from and entering information into the computer's memory. Its purpose, then, is much like the cassette drive. In function, however, it is vastly different. Instead of using a cassette tape as the storage medium, it uses floppy disks.

Floppy disk drives have several advantages over tape cassettes. First, they are very fast. Where it may take up to five minutes or more to find and load a file from a cassette, the floppy disk drive can accomplish the same thing in seconds. Another advantage is that a floppy disk can store more information than a cassette tape. The floppy disk drive is pictured in Figures 10 and 11.

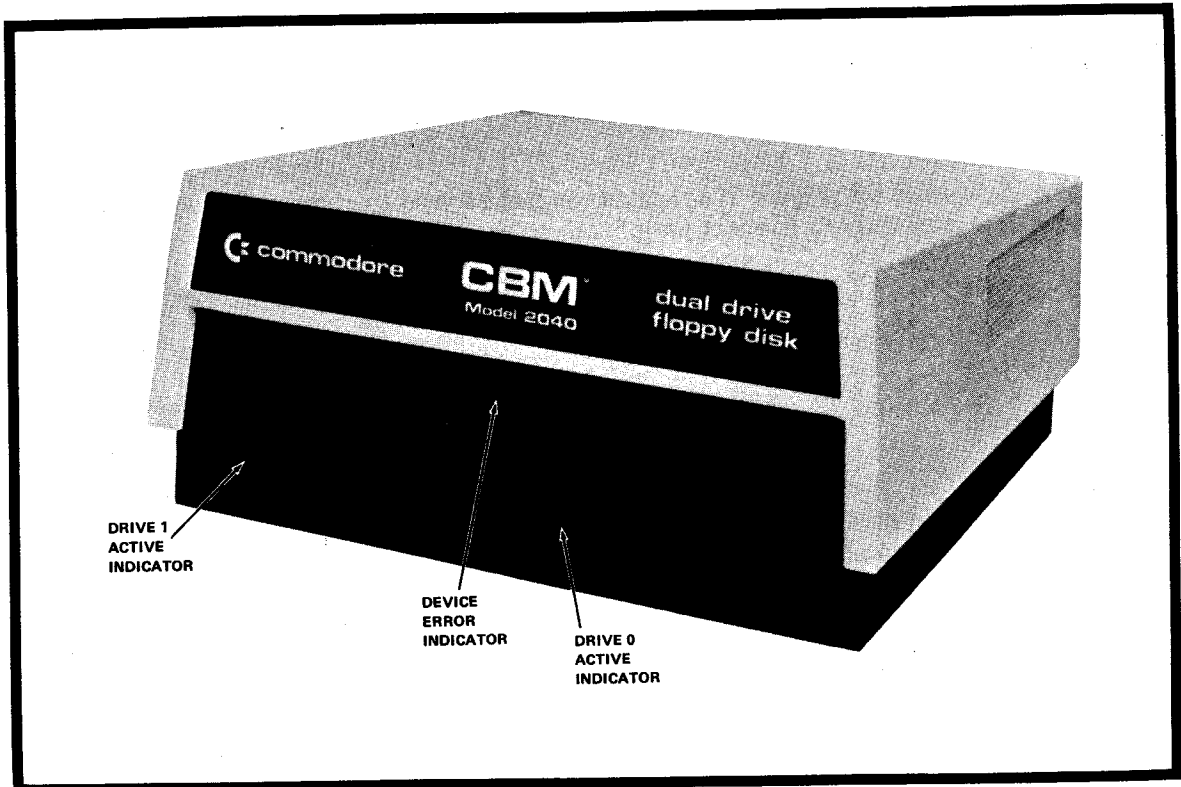


Figure 8 – Model 2040 Dual Drive Floppy Disk

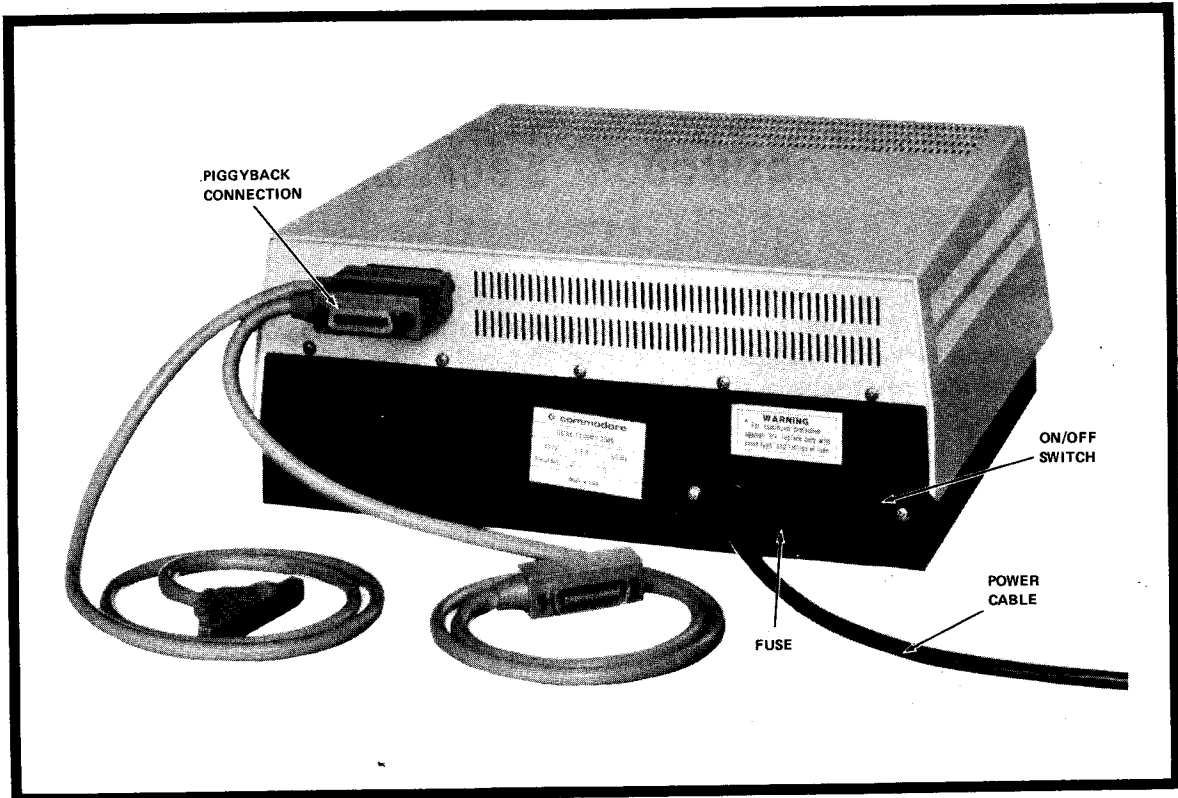


Figure 9 – Model 2040 Dual Drive Floppy Disk, Rear View

For more details on how the drive works and how to attach it to your computer, please refer to the "Commodore CBM Dual Drive Floppy Disk Model 2040 User's Manual".

## Disk Commands

Now let's look at some of the commands you'll need to use the disk drive. For more information on disk commands read the floppy disk drive user's manual. There are five commands used to write data to or read data from a floppy disk. Their formats are:

```
DSAVE"fn",Dx      VERIFY"*",8      DLOAD"*",8
VERIFy"dr:fn",8   DLOAD"fn",Dx
```

where fn=file name supplied by user  
dr=disk drive number (1 or 0)  
x=1 or 0: both Dx and dr default to 0

8=device number (8 for disk, 2 for second cassette, 4 for printer)

All upper-case characters shown in format are absolutely essential for proper execution of command and must be typed by the user. However, these commands must be entered from the keyboard using unshifted characters only. On a Business Model they will appear in lower case. The only exception is when typing the short form of the DIRECTORY command which requires a shifted R.

Version 3.0 BASIC commands are upward compatible with Version 4.0 BASIC commands.

## Formatting Your Disk

### HEADER Command

Each of these commands will be defined along with a brief example to illustrate their use. If your computer has a dual drive floppy disk attached, please try the examples. Before we can use a new floppy disk, it must first be formatted. This will prepare the disk to receive data from the computer. Care should be exercised when formatting a disk since this process will destroy all data previously stored on the disk. The improper use of the HEADER command could destroy useful data already stored in a diskette file. Therefore, a disk should be formatted once when new and possibly formatted again only if the resident data is no longer needed. In addition to removing all the data, formatting will also remove the old disk name and ID and replace them with a user-designated name and ID. Let's take time now to prepare a disk for use in the examples that follow.

**STEP 1:** Insert an unused disk into the left-hand drive and close the door. Insert the disk as shown in the User's Manual.

**STEP 2:** Clear the screen.

**STEP 3:** Keeping in mind that you are only to use lower case characters when entering commands type: header"new disk",d1,i99 then press RETURN.

("new disk" is the name of the disk label selected for this example. The disk name can be up to 16 characters long but should begin with an alphabetic character. The identifier, or disk ID, is the two digit number following the "i" in the header command and can be any unique two character number. Since the disk has been loaded into drive one we use "d1" following the disk name.)

NOTE: Never use the same ID twice. Use 99 for this example but never use these same two characters again to format another diskette. Each diskette should have its own distinctive ID.

STEP 4: Type: y then press RETURN  
(in response to the question:Are you sure?)

Remember, you do not want to format a disk that contains valuable data files since the formatting process will destroy your data. That's the reason for this question: to make sure you carefully consider your actions. If you do not wish to format the disk, this command will not execute without a "y" or "yes" followed by a RETURN.

The screen will display:

```
header"new disk",d1,199

are you sure? y
ready.
█
```

In about 80 seconds the computer will have formatted the disk for use.

Please leave your disk in drive 1

Do not remove it yet

## DIRECTORY Command

You may now wish to check that the proper ID and disk name are on your newly formatted disk. This can be accomplished in the following manner through the use of the DIRECTORY, as follows:

STEP 1: Leave your newly formatted disk in drive 1. If you have removed the disk from the drive, proceed to STEP 3 and then return to this step.

STEP 2: Type: diRd1 then press RETURN

(if diskette was inserted in drive 0 command would be diRd0)

NOTE: If you wish, you may type: directory d1 which uses the full word spelling method but we have shown the shorter method using the first two characters in lower case with a shifted R since this saves time.



The screen will display:

```
dirD1
1 new disk  99
670 blocks free

ready.
█
```

By addressing the directory and observing the display, you have now confirmed that the computer has assigned a name to your newly formatted disk, "new disk" and that 99 is the ID. The directory is where you will also find the name of each disk file that is saved on the disk as well as the number of blocks available for storage of information.

## Initializing Your Disk

### 2040 Disk Drive

**STEP 3:** If you have removed the disk from drive 1 for any reason, it will have to be initialized in order for the computer to know that there is a data storage device in drive 1. Initialize your disk as follows:

Type: open 1,8,15  
print#1,"i1"

Your disk is now both formatted and initialized. Remember that formatting is usually a one-time operation and that re-formatting a disk will destroy the data previously stored on the diskette. Initialization, however, is necessary each time the disk is inserted into a drive since the computer must know this information in order to load or save a disk file. Attempting to load or save a file onto a disk which has not been initialized may result in an error message to the display indicating "Disk ID Mismatch". The disk will then have to be initialized before it can be used in the system.

The INITIALIZE command aligns the read/write head with track 1 on the specified diskette. It then moves to track 18, reads the disk label and ID, and loads this information into the Disk Operating System (DOS) memory. Since the 2040 initialization function depends upon a change of ID to detect a change of diskette, inserting a diskette with ID identical to one previously used may lead to a loss of data. This will happen because the computer will reference an internal map of the previous disk's available blocks for storage of data since the ID's are identical. In fact, since the ID's are identical, the computer assumes that there was no change of diskettes at all since it compares ID's to detect a change. A DSAVE command would now cause new data to be written over good data already resident on the disk because the computer will use the last disk's map of available storage area, and not the current one. After all, the ID's do match. The resultant chaos from over-writing data is usually sufficient to convince anyone to use separate, distinct ID's when using the HEADER command, and always to initialize a diskette when inserting it into a 2040.

The 8050 Floppy Disk Dual Drive detects the removal of a diskette, so it is not necessary to initialize since this function is taken care of automatically.

## Using The Disk Commands

DSAVE“fn”, Dx

This command will save a file named “fn” on the floppy disk in drive number 0 or 1. The file name, “fn”, can be any name of 16 characters or less.

There are two drives on the dual disk drive, the one on the right is drive number 0. The drive on the left is drive number 1.

**STEP 1:** Your formatted floppy disk should still be in drive 1. If you have removed it, insert the disk in drive 1 and initialize as follows:

```
open 1,8,15  
print#1,“i1”
```

**STEP 2:** Clear the screen.

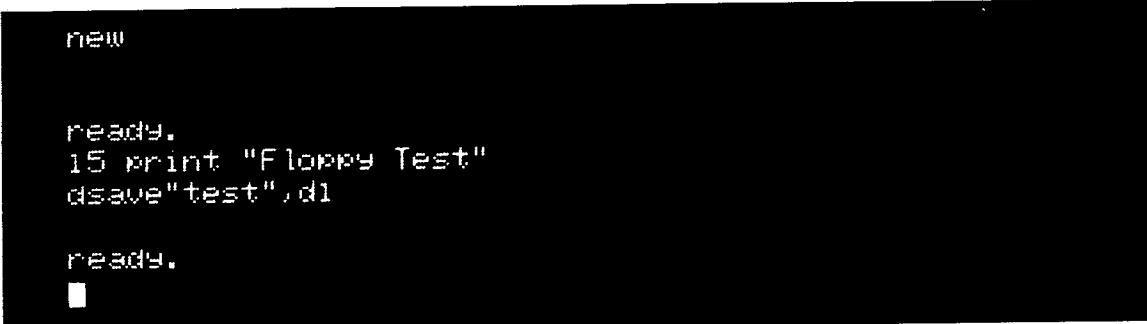
**STEP 3:** Type: new

This step will clear local computer memory and makes sure that only the following information typed from the keyboard is saved to the disk file.

**STEP 4:** Type: 15 print “Floppy Test”

**STEP 5:** Type: dsave“test”,d1

The screen will now display:



```
new  
  
ready.  
15 print "Floppy Test"  
dsave"test",d1  
  
ready.  
█
```

If the light between the two drives lights and stays lit, an error has occurred. When this happens, press the RUN/STOP key, and repeat Step 5.

**DO NOT TURN OFF THE MACHINE OR USE THE “NEW” COMMAND AT THIS POINT.**

## VERIFY“dr:fn”,8

In the exercise above, you saved a file named “test” to the disk in drive 1. Now, through the use of the VERIFY command you can assure yourself that the information you selected to be saved has actually been stored in disk memory.

This command verifies that a file named “fn” stored on a floppy disk contains the same information which is stored in the computer’s memory. This command is similar to the VERIFY command used with the cassette. Once again dr refers to the drive number, either zero or one. Note that the syntax of this command specifies that the drive number be placed before the filename. The 8 at the end of this command is the device number (8 for disk, 2 for second tape cassette, 4 for printer).

**STEP 1:** Make sure the disk with “test” on it is in drive 1 and that the door on the drive is closed.

**STEP 2:** Clear the screen.

**STEP 3:** Type: verify “1:test”,8

Once verified, the screen displays:

```
verify "1:test",8
           searching for 1:test
verifying
ok

ready.
█
```

If a verify error occurs, repeat Step 1. If the error persists, reSAVE the program and verify it again.

## VERIFY“\*”,8

This form of VERIFY verifies the last file saved and saves you the trouble of re-typing in the filename.

**STEP 1:** Make sure that the disk having “test” on it is in drive 1 and that the door on the drive is closed.

**STEP 2:** Clear the screen.

**STEP 3:** Type: verify “\*”,8

**STEP 4:** Proceed as with VERIFY “dr:filename”.

## DLOAD“fn”,Dx

This command is used to load a file named “fn” from drive number x into the computer’s memory.

**STEP 1:** Make sure the disk having “test” on it is in drive 1 and that the door on the drive is closed.

**STEP 2:** Clear the screen.

**STEP 3:** Type: new

**STEP 4:** Type: dload “test”,d1

The computer will now search for “test” and load it.

```
new
ready
dload"test",d1
searching for 1:test
loading
ready.
█
```

If a load error occurs, press RUN/STOP and repeat Step 4.

## DLOAD“\*”

This command is valid for use with DOS 2; if operating under DOS 1 use LOAD“\*”,8 since DLOAD command does not perform auto-initialization of disk drive 0.

This command initializes a disk previously inserted in drive 0 and loads the first file on the disk into memory. To utilize this command, it must be the first disk command given after a cold start.

**STEP 1:** Achieve a cold start by turning the computer OFF, then ON.

**STEP 2:** Make sure the disk having “test” on it is in drive 0 and that the door on the drive is closed.

**STEP 3:** Type: dload “\*”

The computer will initialize the disk in drive 0, search for the first program and load it.

```
### commodore basic ###
```

```
31743 bytes free
```

```
ready.  
dload"*"
```

```
searching for *  
loading  
ready.  
█
```

## Quick Load For Program Files

It is also possible to utilize the quick load feature of your Series 8000 computer by loading a diskette into drive 0 on your 2040 and pressing both the SHIFT and RUN/STOP keys. The first file on the disk in drive 0 will be loaded into computer memory, ready for further processing by the user.

## Relative Files

As the number of items stored in a data file increase, the time needed to access this data also increases until a point is reached where the disk is said to be "disk bound"; that is, the Disk Operating System (DOS) cannot handle the inquiries as fast as the user would desire due to the excessive "look up" time required to find a file and retrieve its contents. This is a basic, inherent drawback to the use of sequential disk files which retrieve and save data in much the same manner as a tape cassette: by sequentially searching through each and every file, by track and sector, until the desired information is found. Of course, if a file has an identifiable file name which has been stored in a directory, the task of retrieval is greatly simplified. However, it is not uncommon for disk files to contain information which is not stored under a file name and, hence, not easily found through the use of a directory. Examples of this type include ticket reservation systems and various types of personnel information files. Under these circumstances the disk drive's major advantage of greater file retrieval speed over the tape cassette becomes less of a factor, particularly when the DOS becomes saturated with a great many demands for file searches. A partial cause of this problem lies within the architecture of those DOS which only permit a sequential search of disk files. Clearly, a need has existed for a more efficient method of seeking and retrieving disk files and for a method which was also easy for the programmer to use and understand.

Both the CBM 8050 Dual Drive Floppy Disk (DOS 2.5) and the CBM 2040 (DOS 2) are capable of relative file handling: a direct access method which allows the programmer to rapidly access any record relative to the beginning of that file. Through the use of relative files, a programmer can significantly reduce the amount of time required to retrieve a record stored on disk.

The two main components of a relative file are the side sector chain of blocks and the data block chain. Both are linked together through forward pointers similar to those used in a sequential file. Record sizes, while fixed in length, may range from one to 254 bytes. The number of records is limited to the capacity of the disk but for practical purposes should not exceed 65,535.

The side sectors do not contain record information, but do contain locations of the data blocks. The record size dictates where the pointer is placed when a record number is referenced because the record size is used in an algorithm to compute where the pointer is placed when a record number is given through the RECORD command. The side sector also contains a table of pointers to all of the other side sectors within the file. In order to move from one side sector to another, the pointer is referenced through the appropriate DOS command, and the corresponding side track and sector read into memory. By using the information contained in the referenced side sector, the data block pointer can be located and used to read in the actual data block containing the record. The relative file data block pointers in the side sectors allow the DOS to move from one record to another within two disk read commands — a considerable savings in the amount of time required to find a desired data block when compared to sequential methods.

When a relative record is opened for the first time, the DOS will generate one data block and one side sector. If the record size, for example, is 10 then 25 records will be automatically generated upon the opening. A file may contain up to six side sectors and each side sector may contain pointers to 120 data blocks. Therefore, the largest file on the 2040 Dual Drive Floppy Disk would be 182,880 bytes (120 pointers/side sector \* 6 side sectors \* 254 bytes/block) which happens to be larger than the capacity of this particular disk. The increased file storage capacity of relative files is a perfect match for the storage capabilities of the CBM 8050 Dual Drive Floppy Disk (DOS 2.5) which has a capacity of over 500,000 bytes per diskette, making it ideally suited for relative file handling.

The relative file design architecture is such that the programmer need not specify record number when printing to it since the relative file can be treated as a sequential file of certain record lengths. Whenever a write or read operation through an input or print statement is performed, the next referenced record will be the sequential record. For example, after reading record ten, record eleven will become available for reading or writing. This feature could be used to simulate variable record size in a relative file by using a small record length such as ten and referring to the beginning of a variable record and "N" number of records thereafter. Each of these groups of records could be referred to as a logical record. Each time the DOS positions to a record through either the RECORD command or a previous I/O statement to a record, it performs a read ahead function. For example, if record three was referenced through the RECORD command, then the data block following that which holds record three will be read for further sequential action — if required. This "look ahead" feature is transparent to the programmer, since the disk drive performs all I/O functions independent of the computer.

To expand a relative file, a programmer may reference the last record number generated through the RECORD command and print to that particular record. The intermediate records from the point of the current end of the file to the reference record number will be automatically generated by the DOS. This includes any side sectors and all data blocks necessary to contain a file, regardless of size, but within the capacity limits of the diskette. For example, if the current size of the relative record is one data block long and the record number referenced would expand it to 125 blocks, then an additional side sector would be generated by the DOS since one side sector can only represent 120 data blocks.

Spanning is a key feature of relative files which aids in reducing the number of disk read/write operations required to find and retrieve data. Before explaining how this feature of DOS 2.5 improves time efficiency, we need to examine how I/O channels are utilized by relative files:

When a channel is opened to a previously existing file, the DOS will position to the first record provided that the given parameters match properly. The record length variable is not necessary on the OPEN if the file is already in existence, but the DOS causes a check to be made against the record size that was originally made in the DOPEN statement creating the file. If these do not match, then error 50 — record not present — will be generated.

The relative channel requires three memory buffers from the system, whereas sequential files only require two. Since there are twelve channels in the system and two of these are used in directory searches and internal functions, only three relative channels can be open at one time. The highest number of buffers that can be used is ten, which limits the total number of channels which can be open at any one time.

If a record was found to be on the boundary between two data blocks, that is, starting in one data block and finishing in another, then the DOS would read the first segment as well as any following records in the second data block. In practice, the records of most relative data files will span across data blocks. The only exceptions are record size 1, 2, 127, and 254. These divide evenly into the 254 size of the data block and spanning is unnecessary. This method of spanning has the advantage of requiring no system memory overhead aside from the side sector blocks in the relative files. When a record is written upon through the PRINT# statement, the data block is not immediately written out. It is only written out when the DOS moves beyond the particular data block in which that record resides. This can occur through successive printing to sequential records, or when positioning to another record outside of that particular block.

Because of the spanning feature, it is recommended that two channels not be open to a relative file at the same time if either channel will be writing to the same file. An update may be made in the channel's particular memory buffer area, but the change may not be made on disk until the DOS moves that particular data block. There is no restriction on this, however, and in certain instances where the file is only read from, it may be advantageous to have more than one channel open to a single relative file.

The DOS terminates printing to a record by detecting the EOI signal which is generated with each PRINT# statement. If the PRINT statement goes over the maximum record size, error 51 — record overflow — will be generated. Any data overflow will be truncated to fit the number of characters specified by the record size and the DOS will position to the next record in sequence. If the print statement contains less characters than the record size, the remaining positions within that record will be filled with nulls. Consequently, when positioning to a record for input the EOI signal is generated from the DOS to the computer when the last non-zero is transmitted. Should the programmer desire to store binary information, a record terminator such as carriage return will have to be used and the record size increased by one character to accommodate the terminator.

When the DOS is generating new data blocks for relative files, the requested record number is compared to the number of data blocks left on the diskette. If the resulting number of data blocks is greater than what is left on the diskette, then error 52 — file too large — is generated.

## Creating A Random File

The following steps are required to create a random file:

1. OPEN the file for random access. Write is implied by the logical record length:

```
DOPEN #1, "RANDOM", L20
```

2. Write data to each record using PRINT# :

```
PRINT #1, A$
```

## Accessing A Random File

The following steps are required to update and access random records:

1. OPEN the file. Read is implied:

```
DOPEN #1, "RANDOM", L20
```

2. Use the RECORD command to select the record:

```
RECORD #1, I
```

3. INPUT the data:

```
INPUT #1, A$
```

## User's Quick Reference: Commands

The User's Quick Reference, table 3, is intended to assist the user in becoming familiar with the various commands used in both Version 3.0 and 4.0 BASIC.

In order to make Version 4.0 BASIC easier to use, disk commands have been incorporated into the language. For example, with Version 4.0 BASIC:

- DSAVE and DLOAD commands eliminate the need to specify device number each time you store and retrieve disk files.
- Directory display is now a one-step procedure and no longer interferes with the program in memory.
- It is no longer necessary to write a program to read the error channel. The variable DS\$ contains the error message.
- Formatting is now a one-step procedure through the use of HEADER command.
- Initialization is unnecessary on the 8050 Disk Drive. It is now an automatic feature of that device.

Commands in Version 3.0 are upward compatible with Version 4.0 BASIC. That is, if you are familiar with Version 3.0, those commands will still work on your 80-column Series 8000 Computer furnished with Version 4.0 BASIC. Tape commands are the same in both versions of BASIC.

The User's Quick Reference has been designed as an aid to the user and it is suggested that a copy be placed near your computer until you become familiar with the various commands.



Table 3. User's Quick Reference: Commands

DISK COMMANDS	
<p><i>40 column display</i> Version 3.0 BASIC</p> <p>SAVE"dr:fn",8 LOAD"dr:fn",8 LOAD"*",8 VERIFY"dr:fn",8 VERIFY"*",8</p>	<p><i>80 column display</i> Version 4.0 BASIC</p> <p>DSAVE"fn",Dx DLOAD"fn",Dx DLOAD"*",Dx VERIFY"dr:fn",8 VERIFY"*",8</p>
<p><i>directory:</i></p> <p>LOAD"\$dr",8 LIST</p>	<p><i>directory:</i></p> <p>DI&lt;shifted&gt;R Dx DIRECTORY Dx</p>
<p><i>formatting:</i></p> <p>OPEN 1,8,15 PRINT#1,"Nx:fn,xx"</p>	<p><i>formatting:</i></p> <p>HEADER"fn",Dx,Ixx</p>
<p><i>initialize:</i></p> <p>OPEN 1,8,15 PRINT#1,"Ix"</p>	<p><i>initialize:</i> (not necessary with the 8050)</p> <p>OPEN 1,8,15 PRINT#1,"Ix"</p>
<p><i>error messages:</i></p> <p>10 OPEN 1,8,15 20 INPUT#1,A,B\$,C,D 30 PRINT A,B\$,C,D</p>	<p><i>error messages:</i></p> <p>PRINT DS\$</p>
TAPE COMMANDS	
<p><i>40 column display</i> 1st cassette:</p> <p>SAVE"fn" LOAD"fn" VERIFY"fn"</p>	<p><i>80 column display</i> 1st cassette: (same as Version 3.0)</p> <p>SAVE"fn" LOAD"fn" VERIFY"fn"</p>
<p>2nd cassette:</p> <p>SAVE"fn",2 LOAD"fn",2 VERIFY"fn",2</p>	<p>2nd cassette: (same as Version 3.0)</p> <p>SAVE"fn",2 LOAD"fn",2 VERIFY"fn",2</p>
<p>NOTE: All Version 3.0 commands are upward compatible with Version 4.0 BASIC. fn=file name supplied by user x=dr=disk drive number (1 or 0): both Dx and dr default to 0 8=device number (8 for disk, 2 for second cassette, 4 for printer) xx=two character, numeric id supplied by user A=error message number B\$=error message C=track D=sector</p> <p>All upper-case characters shown in format are essential for proper execution of command. These must be entered from the keyboard as unshifted characters. On a Business Model they will appear in lower case. The only exception is the short form of the DIRECTORY command which uses a shifted "R".</p>	

## CASSETTE TAPE INTERFACE

A primary tape cassette interface is provided at the extreme left on the back of the computer for attaching a cassette tape drive connector. A secondary cassette tape interface is also available for attaching a second cassette drive connector. Commodore supplies cassette drives with a compatible cassette-interface connector.

The Commodore Tape Cassette (Model C2N) is used as a storage device. It can record information from the computer's memory onto a cassette tape and can play previously recorded information back into memory. See Figure 10.

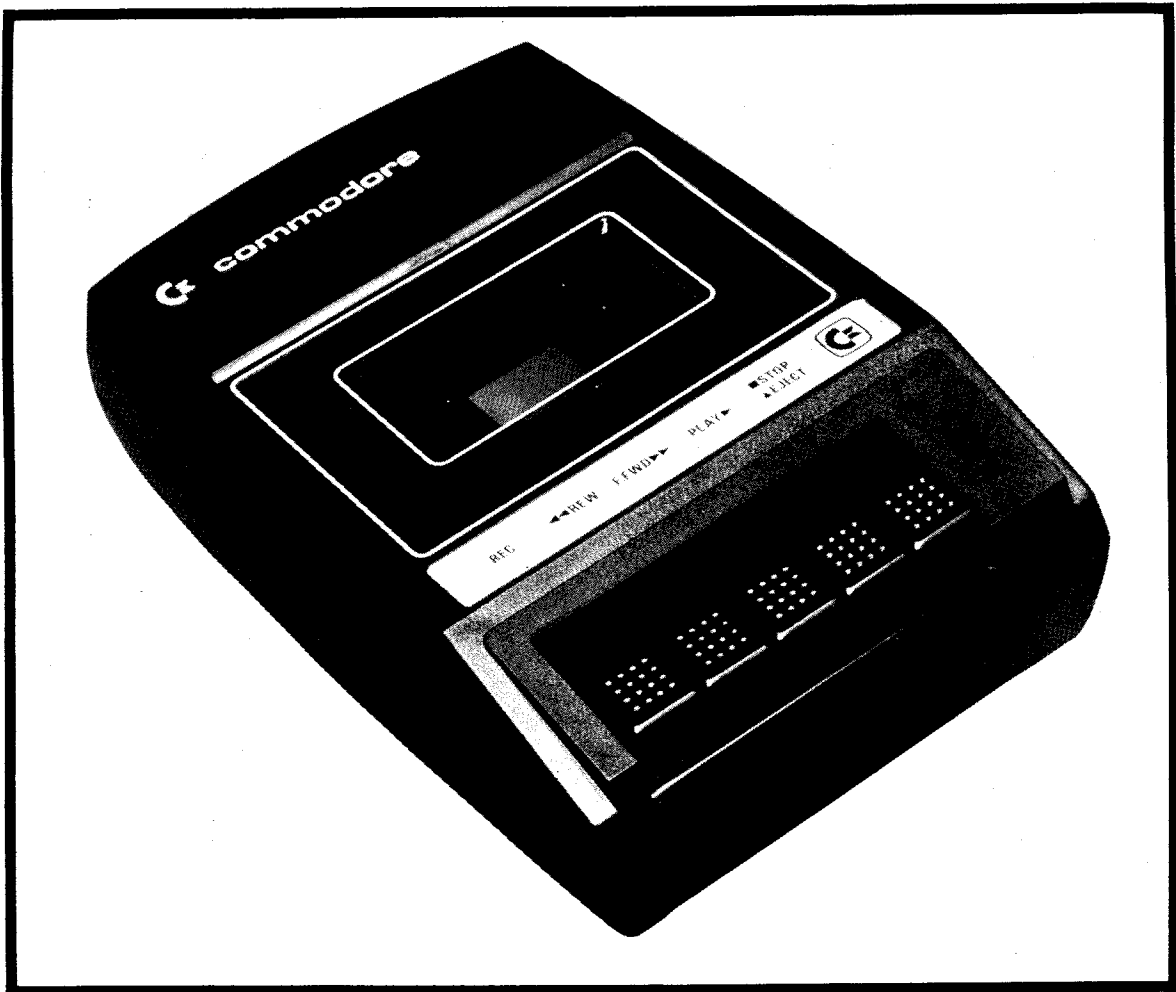


Figure 10 — Model C2N Tape Cassette

To attach a single cassette drive, simply plug its connector into the primary cassette interface on the back of the computer (Figure 3). If you require a second cassette for more advanced data handling capability, refer again to Figures 3 and 6 for the location of the secondary cassette interface.

## Tape Commands

There are five commands that are used to write information onto or read information from a cassette tape. They are:

```
SAVE "fn"          LOAD "fn"
SAVE "fn",2       LOAD "fn",2
                VERIFY "fn"
```

where fn=file name supplied by user

Tape commands are the same in both Version 3.0 and 4.0 BASIC.

NOTE: Commands must be entered using lower case characters.

Each of these commands will be defined along with a brief example to illustrate their use. If your computer has a cassette attached, please try the examples.

### SAVE "fn"

The SAVE "filename" command is used to write a file named "filename" onto a tape in the primary cassette unit. The file name does not need to be specified.

STEP 1: Insert a blank cassette into the primary cassette drive and rewind it. Use only low-noise music quality cassette tape for the best results.

STEP 2: Type: new

STEP 3: Clear the screen.

STEP 4: Type: 15 print "This is a test"

STEP 5: Type: save "test"

We have now asked the computer to save the one line program we wrote and to call it "test". We could have named it anything we wanted as long as the name was not more than 16 characters long. At this point the screen displays:

```
15 print "This is a test"
save "test"

press play & record on tape #1
```

STEP 6: Press and lock the RECORD and PLAY buttons on the cassette. The reels will start to spin indicating that the program is being saved on tape. We will later verify this to be true.

The display now shows:

```
ok
writing "test"
ready.
█
```

The computer has saved the program on cassette tape.

**STEP 7:** Press the STOP button and REWIND the tape. Leave the tape in the drive for now.

**DO NOT TURN THE COMPUTER OFF OR USE THE "NEW" COMMAND AT THIS TIME.**

**SAVE "fn",2**

This command is exactly the same as the SAVE "fn" command except that it writes the file on the secondary cassette, identified by the "2" following the comma.

**VERIFY "fn"**

This command verifies that a file named "fn" on the primary cassette drive contains the same information stored in the computer's memory. It does this by reading the file off the cassette tape and comparing it to the information in memory. This command ensures that your tape copy of the information is accurate.

**NOTE:** If the first name is not specified the computer will use the first file it comes to on the cassette tape in the verification process.

**STEP 1:** Make sure that the tape with "test" on it is in the primary cassette drive and that the tape is rewound.

**STEP 2:** Clear the screen.

**STEP 3:** Type: verify "test"

The computer will display the request:

```
verify "test"
press play on tape #1
```

**STEP 4:** Press PLAY. The computer will search the tape for "test".

```
verify "test"
press play on tape #1
ok
searching for test
```

Once the computer finds "test" it will begin verification.

```
verify "test"

press play on tape #1
ok

searching for test
found test
verifying
```

Once the file has been verified, the screen will display:

```
verify "test"

press play on tape #1
ok

searching for test
found test
verifying
ok

ready.
█
```

If the computer finds that the recorded file is not the same as the information in memory, it will display:

```
verify "test"

press play on tape #1
ok

searching for test
found test
verifying
?verify error
ready.
█
```

If this occurs, repeat Step 1 and try again. If you still get "verify error", reSAVE your information and verify it again.

## LOAD "fn"

The LOAD "fn" command is used to enter a file named "fn" into the computer from the primary cassette player. If a filename is not specified, the computer will load the first file it finds on the tape.

**STEP 1:** Press STOP. Rewind the tape with "test" on it.

**STEP 2:** Clear the screen.

**STEP 3:** Type: new. Now "test" is no longer in the computer's memory.

**STEP 4: Type: load "test"**

The computer will display:

```
new  
  
ready.  
load "test"  
  
press play on tape #1
```

**STEP 5: Press PLAY. The computer will search for "test".**

```
new  
  
ready.  
load "test"  
  
press play on tape #1  
ok  
  
searching for test
```

When it finds the file it will load it.

```
new  
  
ready.  
load "test"  
  
press play on tape #1  
ok  
  
searching for test  
found test  
loading  
ready.  
█
```

Sometimes the computer will fail to load a file correctly. When this occurs it displays:

```
new  
  
ready.  
load "test"  
  
press play on tape #1  
ok  
  
searching for test  
found test  
loading  
?load error  
ready.  
█
```

If this occurs go back to Step 1 and try again.

**STEP 6:** Type: list.

The program "test" will now be displayed. Note that it is identical to the program you originally typed in.

```
new
ready.
load "test"

press play on tape #1
ok

searching for test
found test
loading
ready.
list
15 print "This is a test"

ready.
█
```

**LOAD"fn",2**

This command has the same function as the LOAD "fn" command, only it loads the named file from the secondary cassette.

## COMMODORE PRINTER

The Commodore printers, either Model 2022 or 2023, are IEEE-488 device number 4. See Figures 11 and 12. They are used to print information stored in the computer's memory onto paper for permanent reference. The printer may also be used to print results generated by a program.

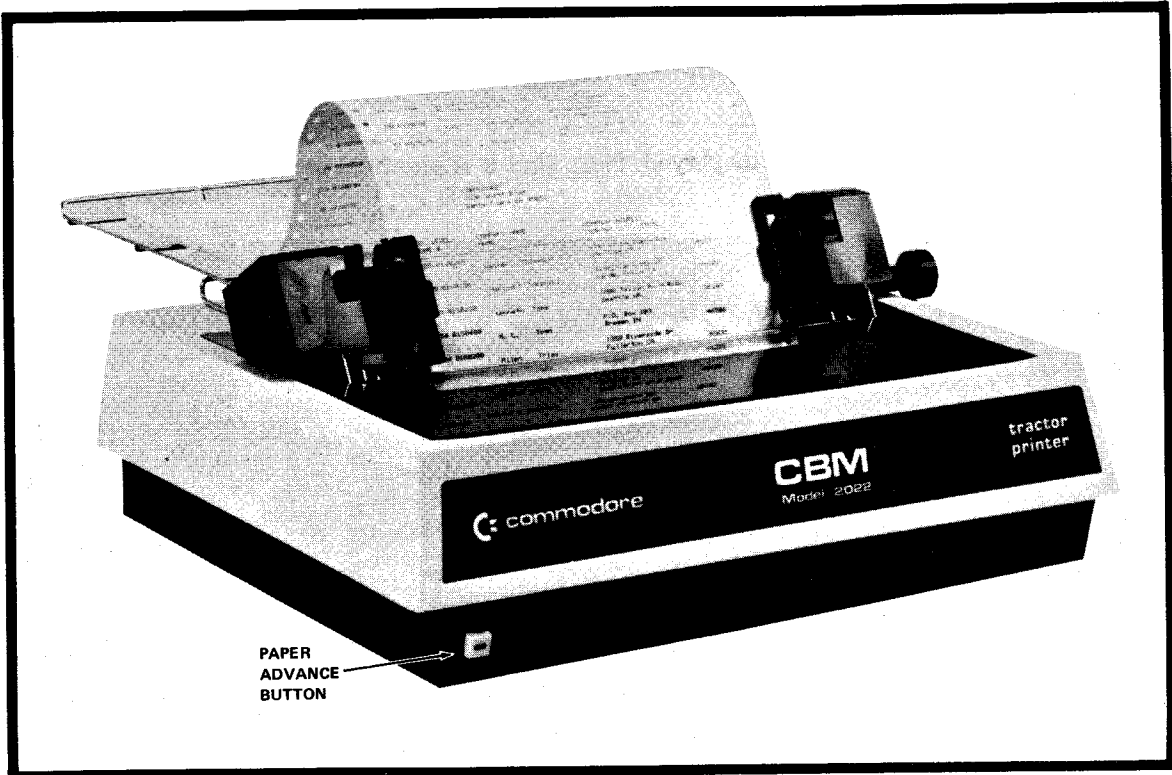


Figure 11 — Commodore Model 2022 Printer



Figure 12 — Commodore Model 2023 Printer



## Special Printer Associated Commands

Most of the programs that you will use already have the necessary printer commands incorporated into them, so we will concentrate, then, on the sequence of commands that list programs from the computer's memory to the printer. For most information on printer commands, and attaching the printers to the computer, see the user's manuals for the Commodore Model 2022 and 2023 printers.

### OPEN

The sequence of commands for listing programs on the printer, opening and closing the communications channel, and the function of the CMD command will be presented in the format of the following example. If you have a printer attached to your computer, try the following:

**STEP 1:** Type: new

**STEP 2:** Clear the screen.

**STEP 3:** Type: 15 print "This is a printer test."

**STEP 4:** Type: open, 4,4

This last statement opens a communications channel between the computer and printer. The first digit after OPEN identifies the logical file number of your file and can be any number from 1 to 255. You must be consistent in using the same logical file number throughout your set of printer commands. This command opens file four and prepares it to be sent to device four, the printer. The printer is always designated by internal computer hardware as device four as identified by the second digit in the OPEN command. You may elect to have your printer's device number changed if you desire to have more than one printer attached, but it is a handwired option and must be done by an authorized Commodore technician. Formatting of the printer output could also be accomplished with this command by appending a secondary address after the device number, separated by a comma. As defined above in this example, the secondary address is zero. There are seven secondary addresses available for use with the OPEN command and the zero in this example instructs the printer to print data exactly as received.

**Note:** Files opened with this command must also be closed with the appropriate sequence of commands.

### CMD

The CMD command transfers control from the computer to the printer. Again, the logical file number, four, must be the same as in the OPEN statement above. As you will observe, the CMD statement will cause the red light in the paper feed button to light and the printer to print READY to indicate that device is ready to receive further commands. In this example we will follow the CMD command with a LIST statement and all output will be directed to the printer. Once the CMD statement is used it is necessary to use the PRINT# command to close down the connection between computer and printer as shown in Step 7 below.

**STEP 5:** Type: cmd 4

This opens the printer up to receive data. The printer will print:

ready.

The red light in the printer paper feed button will light up to indicate the printer is awaiting further commands.

**STEP 6:** Type: list

The listing of the program is routed to the printer rather than the screen:

```
ready.  
15 print "This is a printer test."  
ready.
```

**PRINT# and CLOSE**

**STEP 7:** Type: print#4:close 4

This closes the communications channel between the printer and the computer and reroutes future listings to the display screen.

You may not exceed ten open files so it is a good practice to close previously opened files when finished in order to have the maximum number available for use. The CLOSE statement must be preceded by a PRINT# command in order to properly close a file that has been opened for output to the printer because CMD command will not close the line between computer and output device.

The previous example was printed in direct mode; that is, commands were entered directly from the keyboard. It is also possible to control the printer from within a BASIC program. The following example will illustrate this procedure:

**STEP 1:** Type: new

**STEP 2:** Clear the screen

```
STEP 3: Type: 10 open 3,4  
20 cmd 3  
30 print"program control"  
40 list
```

**STEP 4:** Type: run

Here is the output at the printer:

```
ready  
  
program control  
  
10 open 3,4  
20 cmd 3  
30 print"program control"  
40 list
```

Take a look at the light in the paper feed button on the printer. It is still lit because line 20 of the program used the CMD to open up the channel from the computer to printer and it is still open. You can close it by typing:

```
print#3:close 3
```

The red light will go off indicating that your command was correct and the CLOSE statement closes logical file three.

## NOTES

## NOTES

# Chapter 6

## ERROR MESSAGES

Whenever the computer cannot understand what it was told to do, it will print an error message on the screen and wait for you to do something about it. The advantage of having this immediate response to error conditions is that you can use the screen editing function to quickly fix the problem when it occurs. In many cases the problem will be obvious to you. For example, when a disk error occurs the red error light between the two disk drives will light up. Typing PRINT DS\$ followed by RETURN will cause the error message to be printed on the screen. Refer to Table 4, User's Quick Reference—Error Messages, to interpret its meaning.

When an error occurs, the computer will stop any program that is running, print an error message or light the error light on the disk drive, put the cursor on the screen, and wait for the error to be fixed. The error messages which are covered here are those which commonly occur when learning to LOAD and RUN a program, or SAVE it on disk or tape.

Errors encountered while trying to LOAD a program—

- DEVICE NOT PRESENT
- LOAD and STATUS ERRORS
- READ ERROR \*
- FILE NOT FOUND \*
- DOS MISMATCH
- DISK ID MISMATCH \*
- DRIVE NOT READY

Errors encountered while trying to RUN a program—

- SYNTAX ERROR
- INPUT ERROR
- REDO FROM START
- DRIVE NOT READY

Errors encountered while trying to SAVE a program—

- VERIFY ERROR
- FILE EXISTS \*
- WRITE PROTECT ON \*
- DRIVE NOT READY

\*Disk Errors

While the computer has many error and diagnostic messages, we will specifically examine these thirteen errors—their causes and cures—because they are the ones most frequently encountered when using the LOAD, RUN, and SAVE commands.

## **ERRORS ENCOUNTERED WHILE TRYING TO LOAD A PROGRAM**

### **Device Not Present**

This error message means that the computer has tried to communicate with an IEEE-488 device that either is not attached to the computer or is not turned on. To correct the error, attach the appropriate device to the computer IEEE-488 port and turn it on. Be sure that the cables are connected properly. Exercise caution here: you must first remove any diskettes from the 2040 Dual Disk Drive before turning power OFF or ON. Take care to ensure that the device number of the device being addressed corresponds to that device. That is, don't attempt to address the printer as device number eight when it is actually device number four. (8 for disk, 2 for second cassette, 4 for printer)

### **Load And Status Errors**

If a fatal error occurs while loading a program from tape, this message appears on the screen:

?load error

This means that the computer could not load the program after repeated tries and has quit trying. Rewind the tape and reload the program. Make sure you are searching for a program that actually is on the tape you are using. Tapes can sometimes be aligned by running the whole tape through on FAST FORWARD.

Marginal load errors, which do not cause complete failure to load, can be determined by checking status—type PRINT ST followed by RETURN. If ST=0 then the program has not loaded correctly.

### **Read Error \***

\*This is a disk error, signalled by the red light between the disk drives. Obtain the error message by typing: PRINT DS\$ followed by a RETURN.

This error means that the disk controller is unable to read the requested file from the disk. If this occurs upon loading a program into computer memory, re-initialize the disk and repeat your loading procedures. If this error message persists, your disk may have become unreadable due to physical damage, exposure to magnetic fields, or some other cause. If a read error is encountered on a large proportion of disks, or on one drive and not the other, contact your Commodore distributor since this error may indicate a hardware-related problem.

NOTE: If you attempt to read either a:

- 2040 disk with an 8050 disk drive or an
- 8050 disk with a 2040 disk drive,

A READ ERROR WILL RESULT.

## **File Not Found \***

\*This is a disk error, signalled by the red light between the disk drives. Obtain the error message by typing: PRINT DS\$ followed by a RETURN.

This error message is an indication to you that the file requested doesn't exist on the disk you addressed. Be sure that you are using the correct file name, that it is spelled and punctuated correctly, and that the correct disk has been placed in the correct drive. Trailing spaces may cause this error. Try ending the filename with an asterisk (\*) to ignore further characters.

## **DOS Mismatch \***

\*This is a disk error, signalled by the red light between the disk drives. Obtain the error message by typing: PRINT DS\$ followed by RETURN. The error message is: "73 CBM DOS V2.5 8050."

Versions 1.0 and 2.0 of the 2040 DOS (Disk Operating System) are read compatible, but not write compatible. That is, disks may be interchangeably read with either DOS, but a disk formatted on one version cannot be written upon with the other version because the format is different. Whenever an attempt is made to write upon a disk which has been formatted in a non-compatible format, this error is displayed.

## **Disk ID Mismatch \***

\*This is a disk error, signalled by the red light between the disk drives. Obtain the error message by typing: PRINT DS\$ followed by RETURN.

This error occurs on the 2040 when a disk has not been initialized. Initialize the disk by typing:

OPEN 15,8,15, "i0" followed by RETURN

and try the disk operation again.

NOTE: Use "i1" instead of "i0" if your disk is in drive 1.

## **Drive Not Ready \***

\*This is a disk error, signalled by the red light between the disk drives. Obtain the error message by typing: PRINT DS\$ followed by RETURN.

An attempt has been made to access the 8050 Dual Drive Floppy Disk without any diskettes present in either drive.

## **ERRORS ENCOUNTERED WHILE TRYING TO RUN A PROGRAM**

### **Syntax Error**

When you type something that the computer does not recognize, it prints:

?Syntax error

This is usually caused by incorrect punctuation or misspelling. To correct the error, retype the incorrect command or line. The computer will help you find a syntax error in a program by printing the line number in which the error occurs.

```

15 print "hello"
25 print "goodbye"
run
hello

?syntax error in 25
reads.
list 25

syntax error
reads.
list 25

25 print "goodbye"
reads.
25 print "goodbye"
run
hello
goodbye

reads.

```

## Input Error

When this error occurs the computer will display "??". This error does not stop the program that is being executed. It simply means that the computer has asked you for two or more inputs and you have given it less than requested. The computer will continue to request the missing data with a double "?" until it has all of the data it asked for originally.

The computer needs one more entry so you type in the second variable.

```

10 input "What is your name and age"; A$,B
20 print A$;B
run

What is your name and age? CBM

?? 3

CBM 3
reads.

```

Now you have satisfied the requirements of the program's input statement which requested two items of data.

## Redo From Start

This message appears on the screen following an input:

```
?redo from start.
```



This means that the computer has asked for a number and you gave it an alphabetic character. The computer will print the error message followed by a “?” and wait for the correct information.

“r” is not a number

When you give it a number it continues with the instructions.

```
10 input "What is your age"; X
20 print "You are "X" years old"
run

What is your age?  r
?redo from start
? 25

You are 25 years old

ready.
```

### Drive Not Ready \*

\*This is a disk error, signalled by the red light between the disk drives. Obtain the error message by typing: PRINT DS\$ followed by RETURN.

An attempt has been made to access the 8050 Dual Drive Floppy Disk without any diskettes present in either drive.

## ERRORS ENCOUNTERED WHILE TRYING TO SAVE A PROGRAM

### Verify Error \*

This error occurs when the computer can not verify a file stored on a cassette or a floppy disk with the information in the computer's memory. The screen displays:

?verify error

If you are using a CASSETTE:

1. Rewind the tape.
2. List the program stored in memory to be sure it is the correct program.
3. Make sure that you are using the correct cassette.
4. Re-VERIFY

If you are using a FLOPPY DISK:

1. List the program stored in memory to make sure it is the correct program.
2. Make sure you are using the correct disk.
3. Re-INITIALIZE the diskette and,
4. Re-VERIFY

If the error persists, re-SAVE the program.

## **File Exists \***

\*This is a disk error, signalled by the red light between the disk drives. Obtain the error message by typing: PRINT DS\$ followed by RETURN.

When saving a file or program on disk, it is good practice to make sure that you are using a new file name that does not already exist. If the file name already exists in the directory of the disk presently being used, this error message will result. If you do wish the new file to replace an old one of the same name, SCRATCH the old file before saving the new one. Always keep BACKUP DISKS.

## **Write Protect On \***

\*This is a disk error, signalled by the red light between the disk drives. Obtain the error message by typing: PRINT DS\$ followed by RETURN.

If a diskette has a write protect tab over the notch it cannot be written upon unless the tab is first removed. Attempting to write to a disk which has the write protect will result in this error message. The write protect features on both the tape and disk media are designed to protect valuable files and this error message serves as a reminder to exercise caution when attempting to save files on disk.

## **Drive Not Ready \***

\*This is a disk error, signalled by the red light between the disk drives. Obtain the error message by typing: PRINT DS\$ followed by RETURN.

An attempt has been made to access the 8050 Dual Drive Floppy Disk without any diskettes present in either drive.

## **USER'S QUICK REFERENCE—ERROR MESSAGES**

The User's Quick Reference is provided to assist the user in becoming familiar with the various commands and most probable error messages to be encountered. Table 4 presents some of these error messages. It is suggested that a copy be placed near your Series 8000 Computer until you become familiar with the meaning of these error messages and the proper method of recovery.

TABLE 4: User's Quick Reference — Error Messages

Errors encountered while trying to:	Disk Media	Tape Media
LOAD a program—	<p><b>DEVICE NOT PRESENT:</b></p> <ul style="list-style-type: none"> <li>• Appropriate device not attached to computer or not turned on.</li> </ul> <p><b>NOTE:</b> Before turning the disk drive on or off take care to ensure that any diskettes are first removed.</p> <ul style="list-style-type: none"> <li>• Addressed device doesn't correspond to accepted device number protocol.</li> </ul>	
	<p><b>* READ ERROR:</b> Disk controller unable to read requested data.</p> <p><b>* FILE NOT FOUND:</b> Requested file not on indicated drive.</p> <p><b>* DISK ID MISMATCH:</b> This error occurs upon failure to initialize the disk.</p> <p><b>* DEVICE NOT READY:</b> An attempt has been made to access the 8050 Dual Drive Floppy Disk without any diskettes present in either drive.</p>	<p><b>LOAD ERROR:</b></p> <ul style="list-style-type: none"> <li>• More than 31 errors exist in first tape block.</li> <li>• Errors in exactly the same corresponding positions of both blocks.</li> </ul> <p><b>STATUS ERROR:</b> Non-fatal load errors of various types. Note: Obtain after LOAD is complete by typing PRINT ST followed by RETURN.</p>
RUN a program—	<p><b>REDO FROM START:</b></p> <ul style="list-style-type: none"> <li>• Computer asked for a number and you gave it an alphabetic character.</li> <li>• Computer will continue to print an error message until correct data is entered.</li> </ul> <p><b>INPUT ERROR:</b></p> <ul style="list-style-type: none"> <li>• More data is needed to satisfy program demands.</li> <li>• This error does not stop program execution. The computer waits for more data to be entered.</li> </ul> <p><b>SYNTAX ERROR:</b></p> <ul style="list-style-type: none"> <li>• Computer cannot recognize command.</li> <li>• Incorrect spelling or punctuation.</li> </ul>	
	<p><b>* DEVICE NOT READY:</b> An attempt has been made to access the 8050 Dual Drive Floppy Disk without any diskettes present in either drive.</p>	<p><b>DEVICE NOT PRESENT:</b> Device not connected to computer or not turned on.</p>
SAVE a program—	<p><b>VERIFY ERROR:</b> Stored file does not correspond with information in computer memory.</p>	
	<p><b>* FILE EXISTS:</b> File name of newly created file already exists on diskette.</p> <p><b>* WRITE PROTECT:</b> Write protect tab over notch on diskette prevents any data being written to disk.</p> <p><b>* DISK ID MISMATCH:</b> This error occurs upon failure to initialize the disk.</p> <p><b>* DOS MISMATCH:</b> This error occurs whenever a disk with wrong format is written upon.</p> <p><b>* DEVICE NOT READY:</b> An attempt has been made to access the 8050 Dual Drive Floppy Disk without any diskettes present in either drive.</p>	<p><b>DEVICE NOT PRESENT:</b> Device not connected to computer or not turned on.</p>

\* NOTE: Disk errors are first indicated by red light in center of disk drive. You may obtain error message by typing PRINT DS\$ followed by RETURN. The screen will then display the appropriate error message.





Commodore Business Machines, Inc.  
3330 Scott Boulevard  
Santa Clara, California 95051

Made in Germany