SOFTWARE MANUAL

AMOS

USER'S GUIDE

DWM-00100-35
REV. B00

alpha micro

SOFTWARE MANUAL

AMOS

USER'S GUIDE

DWM-00100-35
REV. B00

alpha micro

FIRST EDITION

April 1979

REVISIONS INCORPORATED	
REVISION	DATE
A00	April 1979
A01	15 October 1979
B00	1 January 1983

THE INFORMATION CONTAINED IN THIS MANUAL IS BELIEVED TO BE ACCURATE AND RELIABLE. HOWEVER, NO RESPONSIBILITY FOR THE ACCURACY, COMPLETENESS OR USE OF THIS INFORMATION IS ASSUMED BY ALPHA MICRO.

This document reflects AMOS versions 4.6A and later,
and AMOS/L versions 1.0A and later.

This document may contain references to products covered under the following U.S. Patent Number(s): 4,530,048

ALPHA MICROSYSTEMS
3501 Sunflower
P.O. Box 25059
Santa Ana, Ca 92799

PREFACE

If you've already read the Introduction to AMOS, DWM-00100-65, you have a good idea of what goes on inside your Alpha Micro computer, and a basic understanding of the theory behind the operation of computers in general. Now, in this manual, you will have the opportunity to put your knowledge to practical use.

If you are already familiar with the operation of a computer system, but have not read the Introduction to AMOS yet, you might want to skim over that book anyway before continuing with this one. This way you'll be able to pick out the differences between the Alpha Micro computer and whatever other computers you may already be familiar with.

The AMOS User's Guide contains instructions for the operation and use of your Alpha Micro computer system. Not only does it contain guidelines for operating the various pieces of equipment attached to your system, but also rules for entering the many available AMOS commmands, creating your own commands, and manipulating files and data sets.

When you're through with this book, you can move on to more specialized documentation (on programming languages, word processing, and much more) listed in the final chapter.

Table of Contents

PART II MAKING THE SYSTEM DO THE WORK

CHAPTER 6 USING WILDCARD SYMBOLS

CHAPTER 7 COMBINING AND SORTING FILES

CHAPTER 8 BACKING UP YOUR FILES

CHAPTER 9 ACCOUNTS AND PASSWORDS

CHAPTER 10 COMMAND FILES, DO FILES, AND CONTROL FILES

CHAPTER 11 USING ALPHA MICRO PROGRAMMING LANGUAGES

CHAPTER 12 WHERE TO GO FROM HERE

APPENDIX A AMOS ERROR MESSAGES

CHAPTER 1

INTRODUCTION TO THIS MANUAL

The first book in this series, the Introduction to AMOS, gave you the theoretical background you need to understand why the computer acts the way it does, and why you have to treat it the way you do. This manual gives you the opportunity to put the theory into practice.

The topics you'll read about in this chapter are:

- The Contents of This Manual
- A Comparison of AMOS and AMOS/L
- A List of Common Abbreviations and Symbols

1.1 WHAT'S IN THIS BOOK

This book is a step by step guide to the Alpha Micro Operating System, known as AMOS. It is necessarily general since there is such a wide variety of possible ways to configuration your Alpha Micro computer system. But whatever size computer you have, and whatever types of equipment you have attached to your system in the way of terminals, disks units, tape units, etc., the same basic principles apply.

First, we'll tell you how to start up the computer in the morning and turn it off at night. Then, how to use your terminal, how to mount and unmount disks, how to request information from the system, and how to use the various commands that come with the system. Next we'll describe how you can store, manipulate, and retrieve data in the form of files.

In Part 2, you'll find out how to make things even easier for yourself by using the wildcard features of certain commands, how to rearrange the data in your files, and how to create duplicate or backup copies of your data. We also describe how you can use different disk accounts to your advantage, how to protect them from unauthorized eyes, and how to construct special files of system commands.

You will also get a brief introduction to several programming languages you can use to write your own programs, and finally, we'll point you in the direction of additional documentation you can read to get more in-depth information on many of the topics we've covered only briefly in this manual.

1.1.1 AMOS VS AMOS/L

A computer can do nothing without a set of instructions to tell it what to do. The main set of instructions, or program, that controls the entire operation of the system is the System Monitor. As you learned in Introduction to AMOS, all the other programs that run on your computer are governed by this monitor program.

Alpha Micro supports two types of system monitors, AMOS and AMOS/L. We've tried to make them act as much alike as possible in spite of the differences in their processing capabilities. This manual is designed for users of both types of systems, and for this reason you may find that your system's displays are slightly different from the ones we use as examples. Other differences, such as the file extensions used for certain files, are pointed out as we go along.

So whenever we refer to AMOS in our discussions of the system monitor and system commands, we are really talking about both the AMOS and AMOS/L operating systems.

1.2 SYMBOLS AND ABBREVIATIONS

Like all the other manuals in the Alpha Micro Documentation Library, this manual contains a number of standardized symbols and abbreviations that make the examples easier to read and understand.

PPN	Project-Programmer Number. This abbreviation represents an account on a disk that you can store files and data in. Project-programmer numbers usually look like [100,2] or [1,4], and are sometimes shown as [p,pn].

devn:

Device-Name. The "dev" is the three letter physical device code, and the "n" is the logical unit number. Examples of device names are DSK0:, DSK5:, HWK1:, and MTU0:. Usually, device names indicate disk units, but they can also refer to magnetic tape drives and video cassette recorders.

filespec

File Specification. A file specification identifies a specific file within an account. A complete filespec is made up of the devn:, the filename, the file extension, and the project-programmer number. For example:

```
devn:filename.ext[p,pn]
      -or-
DSK0:SYSTEM.INI[1,4]
```

{ }

Braces are used in some examples to indicate optional elements of a command line. In the example:

```
.DIR{/switch}
```

the braces tell you that "/switch" is not a required portion of the DIR command line.

______

Underlined characters in our examples are those that AMOS sends to your terminal. The part you have to enter yourself is not underlined. Most of the time you'll see the Monitor prompt shown as an underlined dot (.). The prompt is the signal AMOS gives you that it's ready to perform your commands.

RET

Return. The carriage return symbol tells you when to press the RETURN key on your terminal to terminate a command line.

^

Caret. This symbol in front of a capital letter means that the letter is a control character. That is, you held down the CONTROL key on your terminal keyboard while you typed in the letter. For example, when you type a Control-C, it appears on your screen as ^C. (^C is the control character used to cancel most programs and return you to AMOS command level.) You'll find out more about control characters in Chapter 2.

But before you can start writing your own programs, entering your own commands, and creating your own files, you have to know how the equipment operates. The next chapter tells you everything you need to know.

PART I

GETTING YOUR HANDS ON THE SYSTEM

CHAPTER 2

THE EQUIPMENT

In this chapter, you will find out how to get started using your Alpha Micro computer. Before you can begin using the computer for programs and text processing, you have to know a bit about the equipment you are going to use.

Computers are generally discussed in terms of hardware and software. The term "hardware" refers to the machines and devices - the computer, your terminal, the printer, disk drives, etc.

"Software" refers to the sets of coded instructions that tell the hardware what to do. A set of instructions that performs a specific task or a series of related tasks is a program. Software is further divided into system software which supervises the general operation of the equipment and controls the running of other programs, and application software which performs the routine processing of data under the control of the system software.

This chapter will show you how to:

- Turn your system (your hardware) on and off.
- Use your terminal.
- Log on and off the computer.
- Use your keyboard.
- Change disk cartridges and floppy diskettes.

Before you begin to use your computer system, make sure that the person designated as your System Operator has installed the system software so that it operates in accordance with your system's specific hardware configuration.

Ask the System Operator to assign to you an account or accounts on the system. These accounts will be yours to use when you log into the system

and when you are creating files. Your System Operator will assign you as many accounts as you need.

Now you're ready to turn on your system.

2.1 TURNING THE SYSTEM ON

Below are explanations for turning on most of the equipment you will need.

> NOTE: Check with your System Operator or refer to the instructions that came with each device for information on how to turn the devices on and off. The procedure is different for different brands of devices. In this manual, we cannot detail all of the procedures for all of the equipment that we support, but each device should come with complete instructions for use.

The procedure for turning on your system will depend on what types of devices are on your particular system. Since Alpha Micro supports many different types of storage devices (such as Winchester disks, Floppy disks or video cassettes), you will have to adjust your procedure according to the needs of the specific devices you have.

When you come to the point of turning on your disk drive, use the instructions for the type of system that you have. Generally, you should turn the devices on in the following order:

YOUR TERMINAL

There are several things you can do to extend the life of your terminal. First, you can make your terminal last longer and be more reliable if you don't turn it off and on frequently. The surges of electricity that occur when it's turned on and off might eventually cause certain components to wear out. Also, you can turn the screen contrast down so that the characters don't become etched on the screen. It is a good idea, too, to clear the screen of characters when you are going to be away from the terminal for any length of time. To do this, make sure that you are at AMOS command level, and press the "CLEAR" or "CLEAR ALL" key, or press RETURN until the screen clears.

If you do turn your terminal off, it is best to do so after the computer itself has been shut off, and to turn it on again before turning on the computer. Turning a terminal on or off while the computer is powered on can sometimes cause electrical "noise" on the connecting cable that might interfere with your computer.

CHAPTER 2

THE EQUIPMENT

In this chapter, you will find out how to get started using your Alpha Micro computer. Before you can begin using the computer for programs and text processing, you have to know a bit about the equipment you are going to use.

Computers are generally discussed in terms of hardware and software. The term "hardware" refers to the machines and devices - the computer, your terminal, the printer, disk drives, etc.

"Software" refers to the sets of coded instructions that tell the hardware what to do. A set of instructions that performs a specific task or a series of related tasks is a program. Software is further divided into system software which supervises the general operation of the equipment and controls the running of other programs, and application software which performs the routine processing of data under the control of the system software.

This chapter will show you how to:

- Turn your system (your hardware) on and off.
- Use your terminal.
- Log on and off the computer.
- Use your keyboard.
- Change disk cartridges and floppy diskettes.

Before you begin to use your computer system, make sure that the person designated as your System Operator has installed the system software so that it operates in accordance with your system's specific hardware configuration.

Ask the System Operator to assign to you an account or accounts on the system. These accounts will be yours to use when you log into the system

and when you are creating files. Your System Operator will assign you as many accounts as you need.

Now you're ready to turn on your system.

2.1 TURNING THE SYSTEM ON

Below are explanations for turning on most of the equipment you will need.

> NOTE: Check with your System Operator or refer to the instructions that came with each device for information on how to turn the devices on and off. The procedure is different for different brands of devices. In this manual, we cannot detail all of the procedures for all of the equipment that we support, but each device should come with complete instructions for use.

The procedure for turning on your system will depend on what types of devices are on your particular system. Since Alpha Micro supports many different types of storage devices (such as Winchester disks, Floppy disks or video cassettes), you will have to adjust your procedure according to the needs of the specific devices you have.

When you come to the point of turning on your disk drive, use the instructions for the type of system that you have. Generally, you should turn the devices on in the following order:

YOUR TERMINAL

There are several things you can do to extend the life of your terminal. First, you can make your terminal last longer and be more reliable if you don't turn it off and on frequently. The surges of electricity that occur when it's turned on and off might eventually cause certain components to wear out. Also, you can turn the screen contrast down so that the characters don't become etched on the screen. It is a good idea, too, to clear the screen of characters when you are going to be away from the terminal for any length of time. To do this, make sure that you are at AMOS command level, and press the "CLEAR" or "CLEAR ALL" key, or press RETURN until the screen clears.

If you do turn your terminal off, it is best to do so after the computer itself has been shut off, and to turn it on again before turning on the computer. Turning a terminal on or off while the computer is powered on can sometimes cause electrical "noise" on the connecting cable that might interfere with your computer.

CHAPTER 2

THE EQUIPMENT

In this chapter, you will find out how to get started using your Alpha Micro computer. Before you can begin using the computer for programs and text processing, you have to know a bit about the equipment you are going to use.

Computers are generally discussed in terms of hardware and software. The term "hardware" refers to the machines and devices - the computer, your terminal, the printer, disk drives, etc.

"Software" refers to the sets of coded instructions that tell the hardware what to do. A set of instructions that performs a specific task or a series of related tasks is a program. Software is further divided into system software which supervises the general operation of the equipment and controls the running of other programs, and application software which performs the routine processing of data under the control of the system software.

This chapter will show you how to:

- Turn your system (your hardware) on and off.
- Use your terminal.
- Log on and off the computer.
- Use your keyboard.
- Change disk cartridges and floppy diskettes.

Before you begin to use your computer system, make sure that the person designated as your System Operator has installed the system software so that it operates in accordance with your system's specific hardware configuration.

Ask the System Operator to assign to you an account or accounts on the system. These accounts will be yours to use when you log into the system

and when you are creating files. Your System Operator will assign you as many accounts as you need.

Now you're ready to turn on your system.

2.1 TURNING THE SYSTEM ON

Below are explanations for turning on most of the equipment you will need.

> NOTE: Check with your System Operator or refer to the instructions that came with each device for information on how to turn the devices on and off. The procedure is different for different brands of devices. In this manual, we cannot detail all of the procedures for all of the equipment that we support, but each device should come with complete instructions for use.

The procedure for turning on your system will depend on what types of devices are on your particular system. Since Alpha Micro supports many different types of storage devices (such as Winchester disks, Floppy disks or video cassettes), you will have to adjust your procedure according to the needs of the specific devices you have.

When you come to the point of turning on your disk drive, use the instructions for the type of system that you have. Generally, you should turn the devices on in the following order:

YOUR TERMINAL

There are several things you can do to extend the life of your terminal. First, you can make your terminal last longer and be more reliable if you don't turn it off and on frequently. The surges of electricity that occur when it's turned on and off might eventually cause certain components to wear out. Also, you can turn the screen contrast down so that the characters don't become etched on the screen. It is a good idea, too, to clear the screen of characters when you are going to be away from the terminal for any length of time. To do this, make sure that you are at AMOS command level, and press the "CLEAR" or "CLEAR ALL" key, or press RETURN until the screen clears.

If you do turn your terminal off, it is best to do so after the computer itself has been shut off, and to turn it on again before turning on the computer. Turning a terminal on or off while the computer is powered on can sometimes cause electrical "noise" on the connecting cable that might interfere with your computer.

If your terminal has a REMOTE/LOCAL switch, the switch must be turned to REMOTE in order to communicate with the computer. If it has a HALF DUPLEX/FULL DUPLEX switch (sometimes labeled just HALF/FULL), turn that to FULL DUPLEX.

THE PRINTER

If you are going to be using a printer, turn it on. Like terminals, you might have problems if you turn your printer on or off while the computer is on. You may want to turn off your printers if you also turn off your computer.

If it has a REMOTE/LOCAL switch, turn the switch to REMOTE. If it has an ONLINE button or switch, turn that to ONLINE.

OTHER DEVICES

If you have any other devices on your system, such as magnetic tape drives, video cassette recorders, modems, etc., turn them on now. Check the instructions that came with the device to find out the procedure for turning them on.

POWERING UP DISK DRIVES

There are many different configurations of Alpha Micro computers available. In some of these configurations, the disk drives are powered up when you turn on the power to the computer main enclosure; other configurations require that you power up the disk drives separately.

If your disk drives power up separately from the computer, turn them on before you turn on the computer. In fact, with many disk drives, you may wish to keep them powered up all the time. For example, we strongly recommend that you leave power applied to a Phoenix disk drive so that the blower runs continuously. This drive contains a filter that cleans the air inside the unit. If you turn off the power to this drive, the air within it is no longer being filtered, and contamination damage to the drive could result.

Do not "spin-up" the drive - that procedure should come after the computer is on. (Spinning up a disk involves setting the removable and fixed disks in motion so that the mechanism inside the cabinet can access the data on the disk surface.) If your disk has a removable disk cartridge, make sure that the cartridge in the drive is the one that you want to use; if it is not, you may change it at this time following the instructions supplied with your drive. (Also, see Section 2.5, "Changing Disk Cartridges and Floppy Disks.")

If you are going to be using floppy-disk drives in addition to your hard-disk drives, turn them on now.

NOTE: As a general rule, floppy disk drives are the last components of the system to be powered up, and the first items to be shut off.

THE COMPUTER

Turn on the power to your computer. With most Alpha Micro systems, you have to hold down the RESET button while you turn on the power. In all cases, the RESET button is on the computer's front panel.

NOTE: The exception is the AM-1000 computer-- just turn the on/off switch to "on." If you hold in the RESET button on the AM-1000 while you turn on power, you will enable the Diagnostic (testing) Mode.

In many Alpha Micro systems, your Winchester or floppy disk drives will be powered up automatically when you turn on the power to your computer.

You may find that you do not want to turn your computer off very often. Since the Video Cassette Recorder software and the Task Manager are time controlled, you may find it valuable to schedule tasks and file-backups to run in the evening.

It generally does not harm the computer to leave it on for long periods of time. And, in fact, turning power on and off frequently to electronic circuits creates more wear and tear than if you leave them on continuously.

HARD DISK DRIVES

After the computer and hard disk drives are powered up:

1. If you have an AM-1000 and your system software resides on the Winchester disk drive, you have to press the RESET button so the computer can signal the drive to start up.

2. Now, perform the steps needed to get the drives up and ready; follow the instructions provided by the manufacturer of your disk drive.

   For example, in the case of the Hawk Drive (the disk used with the AM-500 Hard Disk Controller) you must push in the START/STOP button. This button will then light up, telling you the disk has started spinning.

When the light behind the READY button comes on, the disk is spinning at full speed and is ready to use. This process usually takes a minute or so.

FLOPPY DISK DRIVES

After the computer and floppy disk drives are powered up:

If your system boots from a floppy-disk drive, insert your System Disk into drive zero (the System Drive). (The System Disk is the disk that contains the programs that comprise AMOS.) Then close the drive door and push the RESET button.

If you have more than one drive in your disk unit, you can go ahead and insert other floppy disks that you are going to use. (Refer to Section 2.5, "Changing Disk Cartridges and Floppy Disks.")

Never turn the power on or off while there are floppy disks in the drive.

2.2 TURNING THE SYSTEM OFF

We recommend that you do not turn off the computer during normal day-to-day use. However, you should spin down your Winchester disk drives by use of the MOUNT command with the proper option. If you have a floppy disk unit, you should remove the diskettes when the system is not in use. (You should remove and insert floppy diskettes only when power is on.) These measures help to protect your stored data from being damaged, in case of power surges or failures.

If you do want to turn off your system, follow the procedures below (which also tell you how to spin down the disk drives).

NOTE: One of the reasons for turning off the power is if you are going to move the computer equipment. You MUST turn off the power any time you move the equipment, or you may damage your disk storage and/or your Central Processing Unit. Users of the AM-1000 systems should especially take note of this warning, since the AM-1000 is easy to move.

Before you turn off your computer or disk drives, make sure that everyone using the system is at AMOS command level, and make sure that AMOS is not in the process of transferring data between the computer and the disk.

The procedures for shutting off both Hard disk and Floppy-disk systems follow. Use the instructions for whichever type of system you have.

HARD DISK SYSTEMS

1. On some disks, such as the Winchesters, you have to spin down the disk. Use the MOUNT command on your terminal (with the proper option for your system) to unmount the disk. Refer to the instructions shipped with your drive for the exact power-down procedure for your disk drive.

2. If your disk drive has a removable cartridge and you want to remove and store the cartridge, do so now. (Follow the instructions supplied by the manufacturer of your drive.)

3. If you are going to turn off the power to your drive, power down the hard-disk drive now. (Remember that Phoenix disks should be left on continuously to filter the air and keep dust from settling inside.) Usually this procedure consists of more than just turning the power switch of the device off. For example, on the Hawk drive, you must first release the START/STOP button; and then wait for the light behind that button to go off. You may then turn off the power to the drive. It takes 60 seconds or so for the light to go off.

   In some cases, powering down the drives and shutting off the power to your computer are done in the same step. If this is true for your system, just go on to the next step.

4. Turn off the power to your computer.

FLOPPY DISK SYSTEMS

1. Remove and store the floppy disks that are in the drives.

2. Turn off the power to your floppy-disk drives.

3. Turn off the power to your computer. In many cases, turning off the power to the disk and the computer is done in one step.

When these major components of your system have been powered down, you can turn off your terminals and printer.

2.3 WHAT IS A TERMINAL?

A terminal is the device you use for communicating with AMOS. It has a keyboard, upon which you type commands and enter data to the computer. The terminal also has some method of displaying to you what you type in and what is printed out by the computer.

If the terminal displays its output on a video screen, we call it a Cathode Ray Tube Terminal (CRT), or a Video Display Terminal (VDT).

If the terminal displays its output by printing it on paper, we call it a hard-copy terminal (a teletype, for example). Section 2.4.1 will explain how to use your terminal.

2.3.1 Prompt and Cursor Symbols

When your system is turned on, you should see a prompt symbol on the left side of your terminal screen. The prompt symbol is a period (.), and indicates that AMOS is waiting for a command.

If you do not see this prompt symbol, type a Control-C (hold down the CONTROL key on the keyboard, and type a C).

You should now see the AMOS prompt on the screen.

If you still do not see the AMOS prompt, check to see that your terminal is on, and that the cables running between the computer and the terminal are firmly in place. If these are okay and you still do not see the prompt symbol, then consult the System Operator for advice.

When the AMOS prompt is displayed, we say that you are at "AMOS Monitor command level"; that is, that you are communicating directly with AMOS.

At other times you may see other prompt symbols that indicate that you are communicating with a program that AMOS is executing.

For example, when you use the text editing program, AlphaVUE, you see the AlphaVUE prompt symbol: >. At that point you must enter commands that the AlphaVUE program understands.

When you exit AlphaVUE, you are returned to AMOS command level (you see the AMOS prompt (.) again). The various prompt symbols remind you which program you are communicating with.

Remember: To enter AMOS commands, you must be at AMOS command level.

If you are using a CRT or VDT terminal, you also see another symbol (the cursor) next to the prompt symbol. The cursor may be a small white rectangle, a triangle, a blinking line, etc., depending on the type of terminal you have.

The cursor always marks your current screen position. Any characters that you type appear at that position on the screen. As you type, the cursor moves forward as the new characters appear on the screen.

2.4 LOGGING ON AND OFF THE SYSTEM

Now you are ready to sit down at your terminal and begin working. The first procedure you need to know is how to log onto and log off of the computer.

Once you see the AMOS prompt, you are ready to log onto the system. Type LOG, the device and account number you are going to use, and press the RETURN key. For example:

```
.LOG DSK2:[200,5] RET
```

The device this account is on is DSK2:, and The 200,5 is the Project-Programmer Number (PPN) of the account. Your System Operator should have already assigned you one or more of these account numbers for your use. After you have typed the above command, you will see:

```
Logged into DSK2:[200,5]
```

Now you are ready to begin processing data on your computer system. But before we get into using the computer, you should know how to log off. You should log off your computer whenever you are going to be away from your terminal for an extended period of time. If you leave your terminal logged into an account while you are gone, someone else could gain access to your files.

The LOGOFF command looks like this:

```
.LOGOFF RET
User 200,5 logged off
```

You are now off the system. If you try to perform most commands or access any programs when you are not logged in, you will see:

```
?Login please
```

To get back on the system, use the LOG command again.

Once you are logged in, try typing a few characters on the keyboard. As you type, the system displays the characters on your terminal display.

If you should see each character displayed twice on the screen as you type, check your terminal for a switch labeled FULL/HALF or FULL DUPLEX/HALF DUPLEX; turn the switch to FULL or FULL DUPLEX.)

Now you are ready to begin entering commands and instructions on your keyboard.

2.4.1 Using your Keyboard

The first step in communicating with AMOS is to be able to type your instructions on the terminal keyboard. The keyboard is very similar to that of a standard typewriter, but you will find a few extra keys that have special functions.

Take a moment to look at the keyboard of your terminal so that you can easily locate these keys later:

RETURN (or RET)

RETURN is the carriage return key. Just as you type a carriage return on a typewriter to begin a new line on the page, a RETURN tells the system that you are ending a line of input and that you want to begin a new line.

The monitor will not process an instruction from you until you type a RETURN to let it know that you are finished with that line.

RUB (or DELETE, DEL, or RUBOUT)

RUB is the deletion key; it backspaces AND deletes.

If you make a mistake while typing an instruction to AMOS, you can erase it by using the RUB key.

Press the RUB key -- if you are using a CRT terminal, you will see the cursor move to the left, erasing the character in its new position.

Keep pressing the RUB key until the incorrect characters are gone, then type the correct characters. (If you are using a hard-copy terminal, rubouts usually appear as a backslash (\) followed by the characters you've deleted and another backslash. For example, deleting the word THE appears as: \EHT\.)

SHIFT

The shift key on most keyboards acts much like the shift key on a typewriter. By holding down the shift key, you can type upper case letters and the symbols that occur on the upper half of the keys that bear two symbols.

ALPHA
(or CAPS LOCK)

While the ALPHA key is locked down, all of the letters you type appear as upper case letters. The keys that have two symbols on them, however, are not affected by the ALPHA key on most keyboards.

That is, to type a %, you must hold down the shift key while you press the 5 key, even if the ALPHA key is down. (NOTE: Not all terminals have ALPHA keys.)

ESC (or ESCAPE
or ALT MODE)

The Escape key is used with several applications on the system to signal the end of input, or to switch between command modes. However, you do not use Escape at AMOS command level.

CONTROL (or CTRL)

The Control key is used in conjunction with other keys to give you the ability to enter a different kind of character-- a Control-character.

To enter a Control-character, hold down the CONTROL key while you press another key. For example, to type a Control-C, you hold down the CONTROL key while you type a C. (See below for a list of the important Control-characters that AMOS recognizes.)

2.4.2 Control Characters

Control characters allow you to perform special functions. The following list contains the most important control characters. See Appendix D for a list of all of the Control characters and their function.

Control-C

The Control-C is the system interrupt command. Use it to interrupt whatever program is in progress and return to AMOS command level. After typing a Control-C to interrupt a program, you cannot resume execution of that program; you must start it over from the beginning.

You may usually interrupt programs even when they are displaying data.

Some programs, such as AlphaVUE, do not recognize a Control-C as an exit command; instead you must use the exit command for that program if you want to return to AMOS command level.

Other programs do recognize a Control-C. However, if an exit command exists for a program, it is usually better to use that command than to enter a Control-C. Many programs perform various closing functions when you use their normal exit commands and would not have a chance to perform those procedures for an orderly exit if you bypass them by using a Control-C.

Control-U

At AMOS command level, you may erase the line you are typing by entering a Control-U.

On CRT Terminals, the erased line of characters simply disappears, and the cursor moves to the left edge of the display. On a hard-copy terminal, you see a ^U, and the terminal readies itself for a new line of input.

Control-I

Control-I is a tab character. (Many terminals have a TAB key that you can use instead of typing Control-I.) A tab moves the cursor to the next tab stop on your terminal display.

Control-S

If you have a CRT or VDT Terminal, it often happens that a program or command displays more data on your terminal than will fit on one screen-page. To stop the screen display, type a Control-S.

You may now read the data on the screen at your leisure. Not only does the display freeze, but AMOS actually stops sending data to your terminal until you type a Control-Q (see below); at that point AMOS resumes sending information where it left off.

While a Control-S is in effect, AMOS stores (but does not act upon) anything that you type except for a Control-Q. There is, however, a limit to how much can be stored. The exact number of characters depends upon your initial system set-up.

Control-Q

When you type a Control-S (described above) to freeze the screen display, you must type a Control-Q to resume the screen display. If you have typed anything while the Control-S was in effect, a Control-Q tells AMOS that it can now go ahead and act upon that input.

Try this sample. Enter a Control-S, then type DIR and press RETURN, and then PRINT and press RETURN. The commands aren't displayed on the screen and it appears that nothing happened. Now use Control-Q to release the display, and you will see first a list of the files in your account printed on the screen, and then a display of the files waiting to be printed. (You'll learn more about the DIR and PRINT commands later on.)

Control-R

This control function shows you what commands are in your command buffer (the command buffer is an area of memory where the computer stores commands that have been entered, but not yet executed). If you type commands faster than the system can process them (for instance, you type a command to "PRINT" while AMOS is still processing your last command to "TXTFMT"), your command is stored in the command buffer.

Since AMOS is still working on the one command, the characters you typed when you entered your latest command will be intermixed with the output of the command that is processing.

Some commands, like TXTFMT and COMPIL, take several seconds to process, so you may want to type in the next command or commands that you will be using while you are waiting. If you want to see the commands you have already typed but which haven't been been processed yet, you can use Control-R, which will display them to you.

2.4.3 Correcting Typing Mistakes

If you make a mistake while typing a command line, you may always correct any mistakes in that line if you have not yet typed a RETURN. Use the RUB key to erase single characters, or type a Control-U to erase the entire current line. (See the paragraphs above for an explanation of the RUB key and Control-U.) You may also use a Control-C to tell AMOS to ignore the current line.

If you press the RETURN key before correcting your mistake, and the command you entered was not a valid AMOS command, AMOS tells you that it did not recognize the command:

```
.PRIMT  RET
?PRIMT?

.
```

(You meant to say PRINT.) After letting you know that it does not understand PRIMT, AMOS displays its prompt symbol. You are now free to try again.

2.5 CHANGING DISK CARTRIDGES AND FLOPPY DISKS

To change disk cartridges or floppy disks, follow the instructions that accompanied your disk drives. You do not need to reset the computer in order to change a disk; but before you cycle it down, you should consult other users who might be accessing the same device.

DO NOT turn off the disk drive units or the computer when changing a disk.

After changing a disk cartridge or a floppy disk, you must always "mount" that new disk by using the MOUNT command. (AMOS always mounts the System Disk (DSK0:) for you when you first turn on or reset the system.)

2.5.1 The MOUNT Command

Type MOUNT and the name of the device which contains the new disk. (Remember to include the colon.) Then press the RETURN key on your terminal keyboard. For example:

```
.MOUNT AMS1:  RET
```

This command mounts the floppy disk that is in drive AMS1:.

Using a MOUNT command is the only way you have of letting AMOS know that you have changed a disk. If you do not use MOUNT after changing a disk, AMOS assumes that the disk that was previously in the drive is still there.

AMOS stores information in an area called a bitmap that tells it where data and files are located on a disk. Each disk, having different numbers and lengths of files saved on it, has a different bitmap.

Therefore, if AMOS is assuming that a certain disk is in the drive, it writes to and reads from that disk according to the bitmap of the disk. If the disk in the drive was not mounted when it was loaded, AMOS will be using the wrong bitmap when it reads from and writes to the disk.

This can cause your data to be overwritten or destroyed. When you use the MOUNT command, AMOS reads the bitmap from the new disk, and uses that when reading from and writing to the disk.

> WARNING: Do not make any typing mistakes when entering the MOUNT command. If you accidently attempt to mount a disk that someone else is already using, you could cause damage to the bitmap of the disk, which will probably cause loss of data. It does not harm a disk if it is already mounted and you use the MOUNT command, unless someone else is using that disk.

On AMOS/L systems, you will get a warning that other users are accessing the disk, and asking you if you want to go ahead with the MOUNT. On the AMOS systems, you will not get this warning - either way, make sure the other users are aware of what you are doing before you proceed.

If you should incorrectly enter the name of the device you want to mount, you see a message something like this:

```
.MOUNT ASM1: RET
?Cannot mount ASM1: -device does not exist
```

If you get this message, it means that you misspelled the device name or that the device is not defined on your system. To find out what devices are defined, use the DEVTBL command:

```
.DEVTBL RET
```

and you may see something like this:

```
DSK0 (sharable) (alternate) (physical unit 0, logical unit 0)
DSK1 (sharable) (alternate) (physical unit 0, logical unit 1)
DSK2 (sharable) (alternate) (physical unit 0, logical unit 2)
DSK3 (sharable) (alternate) (physical unit 0, logical unit 3)
LPR0
VCR0
HWK0 (sharable)
HWK1 (sharable)
```

These are the devices that are currently defined for your system. If you do not have a Winchester disk drive, the display you see may be substantially different from the first four lines of the example above.

Now that you are familiar with the system hardware, we can show you some of the wide variety of commands and programs you can use to retrieve information and process data. In the next chapter we'll show you a group of commands that display valuable system information.

CHAPTER 3

REQUESTING INFORMATION FROM THE COMPUTER

Some of the commands that you can use on the system are designed so that if you enter only the command itself (with no specifications), AMOS will display information pertaining to that command. With others, AMOS will display instructions for the command if you leave out vital information. The rest will give you some kind of error message if you do not enter the command properly, or leave out vital information.

The AMOS System Commands Reference Manual, DWM-00100-49, and the AMOS/L System Commands Reference Manual, DSS-10004-00, contain information about all of the commands available.

This chapter describes some of the commands that will help you get information about the system. You can experiment with these commands to familiarize yourself with using your terminal and entering commands. The following topics are discussed in this chapter:

- The HELP command
- The LOG command
- The STAT and SYSTAT commands
- The JOBS and JOBALC commands

3.1 THE HELP COMMAND

The purpose of the HELP command is to give information about the system and the system commands to a user who is new to the system.

You do not have to be logged into the system to use HELP. To find out what topics the HELP command knows about, type HELP followed by a RETURN:

```
.HELP RET
```

Depending on the selection of topics on your particular system, you may see a display that looks something like this:

```
Help is available for the following:

APPEND  BAUD  COPY  DEL  ERASE  LOAD
```

To see information about one of the topics listed, type HELP followed by the name of the topic:

```
.HELP COPY RET
```

The screen clears and the terminal displays information on that topic. If you ask for information on a topic that HELP doesn't know about, it will display the subjects it does have information about.

Of course, there are more HELP files than in the display above, and you may create your own HELP topics by using one of the system editors to create text files with .HLP extensions in the HELP library file on DSK0:[7,1]. (See Chapter 4 for a discussion of file extensions.)

The HELP command automatically includes the .HLP files that you create in its list of topics, and displays your HELP files on command.

3.2 FINDING OUT WHAT ACCOUNT YOU ARE LOGGED INTO (LOG)

If you don't know or cannot remember which account your terminal is logged into, type LOG and a RETURN, and AMOS will tell you:

```
.LOG RET
Current login is DSK0:[321,10]
```

If you are not logged in, you will see the message:

```
Not logged in
```

3.3 THE STAT AND SYSTAT COMMANDS

The System Status commands give you a quick summary of tasks the computer is currently performing. The displays list all the jobs on your computer system and tell you what programs they are running. Jobs, you recall from Introduction to AMOS, are subdivisions of you computer's memory and each is usually attached to a terminal.

SYSTAT displays on your terminal screen one line for each job on the system. Each line gives the name of the job, the name of the terminal that the job is using, the account that the job is logged into, the octal base address in system memory where the system maintains information about the job, the status of the job, the name of the last program run by the job, the number of bytes (decimal) in the memory partition being used by the job, and its base address in memory (octal).

Below the job information lines is a list of all storage devices currently mounted on the system and the number of blocks free on those devices.

To see the SYSTAT display, enter:

```
.SYSTAT RET
```

The STAT command is much like the SYSTAT command, except that the display is updated continuously until you use Control-C. This can be very useful if you wish to watch the progress of a task, or to see what other users or devices are doing. To see the STAT display, enter:

```
.STAT RET
```

Consult the Reference Sheets for SYSTAT and STAT in the AMOS System Commands Reference Manual, DWM-00100-49, or the AMOS/L System Commands Reference Manual, DSS-10004-00, for examples and further explanations.

3.4 FINDING OUT THE STATUS OF YOUR JOB (JOBS AND JOBALC)

To interpret the SYSTAT and STAT displays mentioned above you will need to know your job name. Also, if you wish to use the SEND command (explained in Section 8.6, SEND), all users on the system will need to know the names of their jobs.

Because of differences in the two systems, the two commands discussed below, JOBS and JOBALC, have different functions. On the AMOS/L system, the commands will function as we show them below. On the AMOS system, there is only the JOBS command, but it shows you the same display as the JOBALC command does on the AMOS/L system.

To find out your job name, type JOBALC (JOBS on AMOS systems) and a RETURN:

```
.JOBALC RET
YOUR JOB NAME IS JOB1
```

The JOBS command on AMOS/L systems shows you how many jobs are allocated on your system, and also how many are currently assigned. The command and display look something like this:

```
.JOBS RET
13 jobs allocated, 13 currently assigned
```

Now that you know the basics of how to operate your terminal and keyboard, and how to use a few commands to extract information from the system, we can move on to the structure of more complex commands. The next chapter tells you how to arrange various pieces of data when you enter commands and how the arrangement affects the commands' behavior.

CHAPTER 4

COMMAND FORMATS, SWITCHES, AND DEFAULTS

Every AMOS command is actually an individual assembly language program or command file stored on the disk that processes data according to the arguments and switches you specify when you enter the command name. Most commands require that you enter these arguments and switches in a relatively fixed sequence.

One measure of your system's versatility is that, in addition to the many commands supplied with your Alpha Micro Operating System, you can write your own commands, and we'll tell you how to do that in Chapter 10, "Command, Do, and Control Files."

In the previous chapter we discussed several commands that did not need additional information to perform their job. You merely entered the command name and pressed RETURN. However, most commands require certain data, entered in a certain sequence, to produce the desired results.

In this chapter, we'll talk about how commands expect to receive this additional information. Here are the topics this chapter contains:

- Command Syntax
  (How to fit the pieces of a command together.)

- Command Switches
  (Special options that change the way a command works.)

- Command Defaults
  (What AMOS assumes if you leave something out.)

First, let's look at how the elements of a command are arranged.

4.1 FORMAT

Although some commands consist of the command name alone, many commands need additional information to do what you want them to. How you arrange this additional information is determined by the format (syntax) of the individual command.

Each command requires different types of information depending on what the command does. Some pieces of data are required, while others are optional. Also, the amount of data you give on a command line and the order you enter it in determines how the command performs.

Here is a command format that shows what order the options must be in.

```
.COMMAND {outfilespec=}{infilespec1{...,infilespecN}}{/switch} RET
```

The command name (six characters or less) is required, but everything else (as indicated by the {} symbols) on this sample command line is optional. In this case, the "outfilespec=" is followed by the "infilespecs" and the "/switches." In this example, the phrase {...,infilespecN} means that you can specify any number of infilespecs, each separated by commas.

An infilespec is an input file specification-- a file the command takes and manipulates in some way. An outfilespec is an output file specification-- a file the command creates or a file where it places its results. We'll explain switches in just a moment.

Notice that the outfilespec comes before the equal sign, and the infilespecs come after it. If you're used to scientific equations, this arrangement may seem strange to you. But computers usually treat an equal sign as an "is replaced by" sign, rather than "is equal to."

That's why you'll often hear the phrase "replacement statement" used instead of "equation" when talking about programming instructions or system commands. All you have to remember is that the output of a command or instruction comes before the equal sign and the inputs or factors the command uses to get that result come afterward.

You've already been introduced in Chapter 1 to the abbreviations and symbols we use in our examples, but here's a quick review.

. AMOS Prompt symbol. When the cursor is positioned next to this dot, it means that AMOS is ready for your next command. In all our examples, this dot is underlined to indicate that AMOS places it on your screen.

{ } Braces. Anything enclosed in braces in a command syntax example is optional.

/ Slash. A command switch is always preceeded by a slash. Some commands allow you to string several switches after a single slash, while others require a separate slash ahead of each switch you use.

RET Return. This symbol appears at the end of a command line and tells you when to press the RETURN key.

4.2 SWITCHES

Switches are indicators that turn on and off various command options. They are always preceeded by a slash. The command reference sheets in the AMOS or AMOS/L System Commands Reference Manual tell you how to specify switches for each command.

You can specify switches in commands by their full name (such as /QUERY or /KILL), or you can use their abbreviation (such as /Q or /K). Many switches have counterparts that reverse their effect. For example, the /NOQUERY switch reverses the effect of the /QUERY switch, and can be abbreviated /NOQ.

4.3 DEFAULTS

You don't always have to specify complete file specifications because every command assumes certain values unless you tell it otherwise. These assumed values are called defaults, and they enable you to use the command in many situations without having to type in a long string of additional information.

Let's look at the DIR command for a moment. This command displays a list of the files in certain accounts, and we'll discuss it more fully in the next chapter. For now, just enter the command name on your terminal, like this:

```
.DIR RET
```

The DIR program assumes that you want to see a directory of all the files in the account you're currently logged into. This is the most general usage of DIR.

If you want to see a directory listing of just the files in your account that have a file name of MEMO, you can enter the specific filename to limit the display.

```
.DIR MEMO RET
```

The DIR command still assumes you want to see the files in your own account. But if you want to see a list of the MEMO files in another account (we'll talk more about accounts and PPNs in Chapter 9), all you have to do is add an account number. Like this:

```
.DIR MEMO[220,15] RET
```

DIR assumes the account is on the same logical device as your own account. So if you want to see a list of the MEMO files in a certain account on another device, add a device specification. For example:

```
.DIR HWK1:MEMO[200,15] RET
```

As you can see, the more information you give a command, the more easily you can control the results. As we explain the use of each command, we'll tell you the default values each command uses and what other information each command assumes.

The next step is to create some files to use these commands on. The next chapter introduces you to AlphaVUE, an extremely useful and powerful program that enables you to create and modify files. You'll also learn about a number of system commands that make it easy for you to handle these files once you've created them.

CHAPTER 5

CREATING AND USING FILES

Files are the basic units for organizing and storing related data. All the data you work with on the computer is eventually stored on your disk in the form of files, so before we do anything else, let's find out how to create files and retrieve data from them.

This chapter covers the following topics:

- Creating a new file in your disk account
- Finding out what files you already have
- Which file extensions have reserved meanings
- Displaying a file on your terminal
- Making another copy of a file
- Restoring a file from a backup device
- Changing the name of a file
- Erasing a file
- Printing a copy of a file on your printer

Let's start off by creating a new file.

5.1 CREATING FILES WITH ALPHAVUE

To create a new file, you must first select a file name six or fewer characters long. Every file name also has an extension of three or fewer characters. The extension is useful for identifying the type of data a file contains and is separated from the file name by a dot, like this:

```
DATA.LST
```

Duplicate extensions are acceptable within an account as long as the file names preceding them are different. And you can have files with the same name and extension in different accounts, but not in the same account.

Although you can theoretically use any file extension you wish, there are a number of file extensions that have taken on special meanings.

5.1.1 Reserved File Extensions

A file extension can be any combination of letters you choose. For many files that you need to process routinely, there are certain file extensions that AMOS automatically generates or expects to receive.

Appendix B contains a brief description of each reserved file extension and where it's used. For example, AlphaVUE uses the extension .BAK. AlphaBASIC uses .BAS, .RUN, .SBR, and .DAT. TXTFMT uses .TXT and .LST. Other reserved extension include .HLP, .CMD, .PAS, and .LSP. Glance over this appendix to familiarize yourself with the various extensions that have special significance.

After you've selected a filename and extension, you might wish to find out if the name you've chosen is already being used.

5.1.2 Finding Out What Files are in Your Account (DIR)

To see if the file name is in use, you can request a display of your account directory with the DIR command. Entering DIR by itself causes a single column list of the files in your account to appear on your screen. If this list is longer than one screenful, you can use Control-S to stop the display and Control-Q to start it up again.

Perhaps an easier way to view the account directory is by using the /W switch with the DIR command. This switch widens the display to four columns. Just enter:

```
.DIR/W  RET
```

The display that appears on your screen now is much easier to read, especially if your account contains a large number of files.

5.1.3 Creating a Sample File

Let's say you are assigned to write a letter to a customer, and the file you want to create is called LETTER.TXT. Enter:

```
.VUE LETTER.TXT  RET
```

AMOS loads the AlphaVUE program into memory, and AlphaVUE searches for the file name you entered. If there were already a file by that name in your account, you'd see it displayed on your screen. (To get back to AMOS command level from here, you could press the ESCAPE key, then press the letter Q, and finally press RETURN. Section 5.1.4 goes into this procedure in more detail.) But since it's a file that doesn't exist yet, AlphaVUE responds:

```
LETTER.TXT does not exist, do you wish it created?
```

You do, so enter a Y for yes and press RETURN. (AlphaVUE asks this question just in case you make a typing mistake when asking for an existing file.) Your screen clears and fills with asterisks, and the cursor moves to the upper left corner, ready for you to start typing your note. If a different display appears-- one where the cursor appears next to a > sign-- press the Escape key (ESC) to see the screenful of asterisks.

Now you can enter the sample letter, just to get the feel of creating a document. Just type it in, and don't worry about mistakes. It might seem a bit different, especially if you are used to a typewriter. Later on, you'll learn more about AlphaVUE from the AlphaVUE/TXTFMT Training Guide, DSS-10000-03, and the AlphaVUE User's Manual, DWM-00100-15. These manuals will show you how useful and easy AlphaVUE can be for entering, correcting, and editing documents.

Here is the sample letter for you to type:

Mr. John R. Baxter
Autonomic Neurosystems
P.O. Box 12
Commerce, CA 92714

Dear Mr. Baxter:

Your recent order helped us surpass our sales goal for
last month, and we wish to express our sincere appreciation.
Thank you for your confidence in our products. We are looking
forward to your continued patronage.

Sincerely,

George Morton

5.1.4 Getting Back to AMOS Command Level

Now that you have typed your note into memory, you need to store it on your disk and return to AMOS. Pressing the ESCAPE key brings you to AlphaVUE command level. The AlphaVUE prompt is a right angle-bracket (>) and if you enter an F (for Finish) and a RETURN, your file is stored on the disk. As AlphaVUE stores your file, it prints one dot on your screen for each block of the file it transfers.

Your cursor appears next to the AMOS prompt indicating that you have left AlphaVUE and are once again talking to AMOS. The whole operation looks like this:

```
>F...
.
```

If you decide that you don't want to save what you just typed, you can still return to AMOS by entering a Q (for Quit) at AlphaVUE command level instead. Like so:

```
>Q RET
.
```

You can do a lot more things with AlphaVUE than we've described here, and you can learn more about both AlphaVUE and TXTFMT by reading the AlphaVUE User's Manual, DWM-00100-15, the TXTFMT User's Manual, DWM-00100-07, and the AlphaVUE/TXTFMT Training Guide, DSS-10000-03.

5.2 DISPLAYING FILES (VUE, TYPE)

Once your files are stored in your account on the disk, you can look at them and modify them whenever you wish by using the VUE command or the TYPE command.

The VUE command is the same one you used to create the file in the first place. Here's how to look at that note you wrote earlier:

```
.VUE LETTER.TXT RET
```

Since it now exists in your account, AlphaVUE has no trouble locating it and displays it on your screen. If your file is bigger than one screenful, you can page back and forth using Control-R and Control-T. To go directly to the end of the file, use Control-E; and to return to the front of the file, use Control-^ or the HOME key.

When you're through looking at your file or changing it, you use the same method of storing it on the disk as you did when you created it. That is, you use the Escape key to get to the command level, and enter "F" to write the file to the disk and bring you back to AMOS.

> NOTE: On the AMOS/L system, AlphaVUE won't let you look at files with certain extensions, either because they contain data your terminal can't display (such as .RUN files), or because they should not be modified (such as .BAK files). If you try to look at a file you shouldn't, AlphaVUE sends you an error message. For example:
>
> ```
> ?Cannot VUE or UNYANK file with .BAK extension
> ```

Another way to look at a file is with the TYPE command:

```
.TYPE LETTER.TXT RET
```

You can't modify a file with the TYPE command the way you can with the VUE command, and instead of showing you only one screenful at a time, TYPE scrolls through the entire file from beginning to end without stopping.

The only way you can halt the display is with a Control-S. To start it up again, use Control-Q. (On AMOS/L systems, you can use the /P switch with the TYPE command to display one page of data at a time.)

Also, TYPE does not allow you to go backward in the file or jump around the way the VUE command does. Although you'll probably use the VUE command most often, TYPE is useful when you just want to verify the contents of a particular file.

5.3 COPYING FILES (COPY)

If you need to send the same letter you wrote at the beginning of this chapter to a different customer, you can easily make another copy of it and just change a few of the words. The COPY command lets you do this with a minimum of bother. Decide on another file name that does not already exist in your account and enter the following command.

```
.COPY NOTE.TXT=LETTER.TXT/NOD RET
```

where NOTE.TXT is the name of the new file, and LETTER.TXT is the name of your original letter. Now you have two copies of the same thing and you can use AlphaVUE to change one or the other.

The /NOD on the end stands for /NODELETE. It means that if, by chance, the file name you chose (NOTE.TXT) was already in use in your account, the file you copy (LETTER.TXT) will not wipe out whatever was in that file before. Rather, AMOS will inform you that the file NOTE.TXT exists, and not do any copying. You can then choose an unused name for your file.

5.4 RETRIEVING FILES FROM BACKUPS

There are several different backup devices available where you can store files that you are not actively using. Storing your inactive files on floppy disks, video cassettes, or streaming tape cartidges keeps your disk from becoming cluttered, and also prevents the files from being accidentally modified or erased. We'll talk about backing up your files in Chapter 8, "Backing Up Your Files."

Each backup device has special commands that enable you to store and retrieve data. Floppy disks that are used with Winchester technology disks use the commands WINFLP, FLPWIN, and FLPDIR to copy files back and forth from one device to the other, and to display a directory of the files stored on a particular floppy disk.

1/2" magnetic tape drives use TAPFIL, FILTAP, and TAPDIR commands for the same procedures.

The video cassette recorder uses the commands VCRSAV, VCRRES, and VCRDIR to perform the same functions. And the 1/4" streaming tape drive uses the commands STRSAV, STRRES, and STRDIR.

You can learn more about these devices and commands by reading the appropriate AMOS or AMOS/L documents listed in Chapter 12: Video Cassette Recorder Backup Software, 1/4" Streamer Backup Software, and the System Operator's Guide.

5.5 RENAMING FILES (RENAME)

There are times when you may wish to change the name of a file, and the RENAME command was developed for this purpose. It's a good idea to verify that the new file name you picked is not already being used; RENAME sends you an error message if it is.

If you suspect the file name is in use and you don't wish to save the contents of the file if it is, you can use the /D switch which deletes the file before renaming your current file to the new name. Here's how it works:

```
.RENAME/D MISSIV.TXT=NOTE.TXT RET
```

where MISSIV.TXT is the new name given to the current file, and NOTE.TXT is the current file name. When the command is finished, the file NOTE.TXT no longer exists and you can use that file name again for another file.

5.6 PRINTING FILES (PRINT)

Now that you've finished your letter, you need a printed copy of it to put in the mail. To use the PRINT command, enter the name of the command and the name of the file:

```
.PRINT LETTER.TXT RET
```

The command above causes your letter to be sent directly to the printer. If you have more than one printer attached to your system, your printing will automatically go to whichever one the System Operator has defined as the default printer. If you want to use a printer other than the default, you must specify the printer name in your PRINT command. If it's a Diablo printer, for example, your PRINT command might look like this:

```
.PRINT DIABLO=LETTER.TXT RET
```

The file automatically begins printing. See the System Operator for a list of printer names on your system.

To check on the progress of a file being printed, enter just the command:

```
.PRINT RET
```

The resulting display tells you how much of your file is left to print.

5.7 ERASING FILES (ERASE)

When you want to delete a file from your account, you can use the ERASE command to get rid of it. Suppose you want to get rid of the LETTER.TXT file. Just enter:

```
.ERASE LETTER.TXT RET
```

The file is gone and does not appear on the directory listing of your account. If you want to create another LETTER.TXT file, you can do so by following the same steps you used to create the first one.

> NOTE: Depending on how your system is set up, you may see an error message if you try to ERASE, RENAME, or access a file. It's possible that the file has been protected during system initialization from being accessed. If you need to access such a file, contact your System Operator for help.

These first four chapters have given you a broad, basic knowledge of how the AMOS and AMOS/L computer systems work, and how you can use them to your advantage. The next six chapters tell you how to add leverage to the commands you've already used, and introduce you to some new commands that make your job even easier.

PART II

MAKING THE SYSTEM DO THE WORK

CHAPTER 6

USING WILDCARD SYMBOLS

So far, you've learned the basic operation of several system commands, but useful as these commands are, they can become tedious if you have to repeat them for a large number of files. If you have to make duplicate copies of twenty AlphaBASIC programs, or rename all your .TXT files, or erase the .BAK files in your account one by one, you could become extremely frustrated if it weren't for wildcard symbols.

A select group of the most often used commands recognize these special symbols and allow you to process groups of files without having to specify each and every file name. Many commands do not recognize these wildcard symbols, so if you're ever in doubt, check the appropriate command reference sheet in the AMOS or AMOS/L System Commands Reference Manual.

This chapter contains the following information:

- Definitions of the various wildcard symbols
- Some rules for using wildcard symbols in account specs
- Examples of how wildcarding works

Before we do anything else, we should define wildcard symbols.

6.1 WHAT ARE WILDCARD SYMBOLS?

In some card games, you can use a joker (or some other card) as a wild card which can be used in place of any other card in the deck. Likewise, you can substitute wildcard symbols for various parts of file and account specifications in certain AMOS commands and enable the command to process several files with a single file specification.

The list below explains what each of the wildcard symbols means.

*   Asterisk. This symbol stands for any group of characters in file names, file extensions, and account numbers. For example, *.BAS means every file that has the extension .BAS. Also, PR*.BAS means all files that start with PR and have .BAS extensions like PRIVAT.BAS, PROOF.BAS, PRT23.BAS, and PR.BAS.

?   Question mark. This symbol stands for a single character in file names, file extensions, and account numbers. For example, EX?PT.LST includes all files whose names begin with EX and end with PT (with only one character in between), and have .LST extensions, such as EXMPT.LST, EXRPT.LST, and EX9PT.LST, but not EXERPT.LST or EXPT.LST. A series of question marks at the end of a file name indicates that many or fewer characters.

ALL: All Logical Devices. This symbol stands for any file-structured, mounted device. For example, ALL:START.DO means all START.DO files on all logical devices.

dev: All logical devices whose names begin with the code "dev" where "dev" is the device abbreviation. If you leave off the unit number from a device specification, wildcard commands search for the files you specify on all units of the specified device type. For example, HWK:BAKUP.CMD stands for any file named BAKUP.CMD on any logical device whose name begins with "HWK" (such as HWK0:, HWK1:, HWK2:, etc.).

[]   Brackets. This symbol stands for any account number (PPN). For example, SORT.HLP[] means all SORT.HLP files in any account. [] is the same as [*,*].

The system commands that recognize these wildcard symbols as valid arguments allow you to manipulate numerous files with a single command. But before we give specific examples of how each of these wildcard commands operates, there are a couple of general rules that you need to keep in mind when using wildcard symbols in account numbers.

6.2 SOME RULES FOR USING WILDCARD SYMBOLS

First, when you use an asterisk (*) in an account number, you can either use it to represent the entire project or programmer number (as in [*,5] and [45,*]), or you can use it as a partial project or programmer number with one or more numbers following it (as in [*41,12] and [22,*2]). The catch is that you cannot put numbers ahead of the asterisk in a partial project or programmer number. (This wildcard account number is illegal: [25*,6].)

Here are some examples of valid wildcard account numbers:

FOLIO.LST[*,12] covers FOLIO.LST[5,12] and FOLIO.LST[120,12]

PROG.BAS[15,*] covers PROG.BAS[15,1] and PROG.BAS[15,34]

WWTIRE.TXT[*3,*] covers WWTIRE.TXT[3,0], WWTIRE.TXT[23,5], and WWTIRE.TXT[103,14]

When you use an asterisk as part of a file specification, just the opposite is true. You can still use the * to represent the entire file name or extension (as in *.TXT or PRGRAM.* or *.*), but you cannot put any characters behind the asterisk in a partial file name or extension. (These are valid wildcard file names: PAY*.LS* and X*.B*, but these are invalid: F*M.*AS and *RX.F*L.)

The question mark symbol (?) at the beginning of a project or programmer number stands for that many or fewer numbers. (For example, EASY.PRG[??1,??] covers EASY.PRG[1,2] and EASY.PRG[21,14].) If you use the ? symbol in the middle or at the end of a project or programmer number, the exact number of characters must match. (Thus, FINDIT.LST[1?1,2?] covers FINDIT.LST[101,20], FINDIT.LST[141,24], and FINDIT.LST[111,26], but not FINDIT.LST[11,2] or FINDIT.LST[1,245].)

6.3 WILDCARD FILE SPECIFICATIONS

The way you specify files in wildcard commands determines the way the commands perform. The paragraphs below describe how the placement of input file specifications (infilespecs) and output file specifications (outfilespecs) alters the results of these commands.

6.3.1 Input File Specifications

Most commands, wildcard commands included, use infilespecs. These are the file specifications that the command works upon to perform its designated function. In a wildcard command, you are able to use one infilespec to represent a group of files. For example, if you want to refer to DSK0:MCSAM.RUN and DSK1:MCELI.RUN, you could condense them into a single filespec: DSK:MC*.RUN.

> NOTE: There are certain situations, however, when wildcard commands should be used with caution since you might unintentionally include more files in the wildcard filespec than you bargained for. Such is the case when you are ERASEing or RENAMEing files.

6.3.2 Output File Specifications

A few of the wildcard commands require you to supply an outfilespec under certain circumstances. The outfilespec, if used, always comes immediately after the command name itself, ahead of any infilespecs, and indicates the destination or result of the command's designated function. For example, the RENAME command requires both an outfilespec and an infilespec:

```
.RENAME *.SAV=*.BAK RET
```

In this command, *.SAV is the outfilespec and *.BAK is the infilespec. All the files having .BAK extensions in the account you're currently logged into are automatically renamed with .SAV extensions.

If we changed the command to say:

```
.RENAME *.SAV=BRUT.BAK,SEC.BAK RET
```

...only the two specified files in the account you're currently logged into would be renamed with .SAV extensions.

Another way to use wildcard symbols in outfilespecs is this:

```
.RENAME AR*=*.FIN RET
```

This RENAME command takes all the files with .FIN extensions in the account you're currently logged into and changes the first two letters of the filename to AR.

6.3.3 Ersatz Device Specifications

The more data you have to enter on your screen, the easier it is for you to make a typing mistake. The less you have to type, the better. So for your convenience, several abbreviated substitutes have been invented for the device and account specifications that you use most frequently. These ersatz specifications are only four characters long and reduce the possibility of error a great deal.

The following table lists the ersatz specs and the devices and accounts they correspond to:

ERSATZ	DEVICE SPEC	LIBRARY NAME
RES:	System Memory	
MEM:	User Memory	
OPR:	DSK0:[1,2]	System Operator's Account
SYS:	DSK0:[1,4]	System Program Library
DVR:	DSK0:[1,6]	Device Driver Library
CMD:	DSK0:[2,2]	Command File Library
LIB:	DSK0:[7,0]	Miscellaneous File Library
HLP:	DSK0:[7,1]	Help File Library
BOX:	DSK0:[7,2]	Mailbox Data File Library
LSP:	DSK0:[7,4]	AlphaLISP Language Library
PAS:	DSK0:[7,5]	AlphaPASCAL Language Library
BAS:	DSK0:[7,6]	AlphaBASIC Language Library
MAC:	DSK0:[7,7]	Macro Language Library

You can use these ersatz specifications anytime you want to refer to one of the library accounts listed above. For example, to see a directory listing of all the files in the Help File Library, you could enter either:

```
.DIR DSK0:[7,1] RET -or-     .DIR HLP: RET
```

6.4 WILDCARD SWITCHES

Most ordinary commands permit you to put switches only at the end of a command line. Wildcard commands give you added flexibility by permitting you to place switches anywhere in the command line. Where you place a particular switch within a command will then affect how the switch operates.

Wildcard commands recognize two types of switches: file switches that pertain to individual file specifications in the command line, and operation switches that affect the entire command line regardless of their placement.

6.4.1 File and Operation Switches

A file switch usually applies to a single file specification, and you can place it anywhere in a command line depending on the results you want. We'll talk more about file switches and where to put them when we begin setting switch defaults.

You can place an operation switch anywhere in the command line and its effect is always the same. For example, the /K switch in the PRINT commands below is an operation switch:

```
        .PRINT/K MEMO.LST,REPORT.LST,LETTER.LST RET
```

is the same as...

```
        .PRINT MEMO.LST/K,REPORT.LST,LETTER.LST RET
```

and is the same as...

```
        .PRINT MEMO.LST,REPORT.LST,LETTER.LST/K RET
```

The description of each command will tell you whether its switches are File Switches or Operation Switches.

6.5 WILDCARD DEFAULTS

We discussed command defaults and rules of syntax back in Chapter 4, and those same basic rules apply to wildcard commands as well. But there are some significant additional features of wildcard command syntax that increase their power tremendously.

Setting the Device Defaults:

If you do not specify the device or account in an infilespec, most commands default to the device and account you are logged into. Wildcard commands allow you to reset these default values within a command line. Let's take an example:

```
  .ERASE *.BAK,HWK1:*.LST,*.BAK,*.CMD RET
```

The first thing this ERASE command does is get rid of all the files with .BAK extensions in the account and device you are currently logged into, say DSK1:. Then the device specification changes to HWK1:, so ERASE deletes all the files with .LST extensions on that device, but still with the same account number as the account you are logged into.

For the next filespec, ERASE stays on HWK1: (it's now the default device), and deletes all the files with .BAK extensions, and *.CMD extensions.

Setting the Account Defaults:

But that's not all. Here's another example:

```
  .ERASE DSK0:SAVE.*,RESTOR.*,[200,1]PRT04.LST,MGMT.LST,DSK1:*.RST RET
```

This ERASE command sets the default device right away to DSK0:, and deletes all files named SAVE and RESTOR from the account on that device with the same number as the account you're currently logged into. So far, so good.

Now, notice that there is an account number ahead of the next filespec. Placing the account number in this position resets the default account number, so that the files PRT04.LST and MGMT.LST are both deleted from account [200,1] on DSK0:. The last filespec contains a different device name, but since the account number default has been changed to [200,1], ERASE deletes all the files with .RST extensions from that account on DSK1:.

That example might seem complicated at first, but if you analyze it one part at a time, you'll see how much you can accomplish with a single command.

Setting the Project-Programmer Number Defaults:

Here's another default feature you can use.

```
.ERASE [24,0]TAXI.BAS,JRNL.LST[,5],[211,3]GRT.PGM,SRT101.SLR[212,] RET
```

If you look closely at the account numbers in this command, you'll notice that one is missing a project number and one is missing a programmer number. If you leave one or the other of these numbers out of an account number, wildcard commands default to the currently set default project or programmer number. In this example, the project number for JRNL.LST and the programmer number for SRT101.SLR default to the previously set project and programmer numbers. This command could also have been written:

```
.ERASE TAXI.BAS[24,0],JRNL.LST[24,5],GRT.PGM[211,3],SRT101.SLR[212,3] RET
```

As your use of files becomes more extensive and your need to manipulate a lot of files quickly increases, you'll find that the ability to condense commands into the shortest possible space (especially in Command and Do files which we'll discuss in Chapter 10), while having them process the maximum number of files, is extremely important.

Setting File Switch Defaults:

If you put a wildcard file switch right after the command name like this:

```
.ERASE/Q TEXT1.TXT,RPT23.TXT,SUMX.TXT,FILE.TXT,GAME.TXT RET
```

the switch applies to everything on the command line. (In this case, the Query switch will cause the ERASE command to pause after each file for your confirmation before actually erasing it.)

If the switch comes after one of the file names,...

```
...RPT23.TXT/Q,SUMX.TXT,FILE.TXT,...
```

the command only stops for confirmation for that one file, since it is a file switch.

When you place the switch ahead of the file specification, it remains in effect for the rest of the command line. Such as:

```
...RPT23.TXT,/Q SUMX.TXT,FILE.TXT,...
```

From these examples, you can see that the placement of the switches in the command line can greatly alter the outcome of the command.

The preceeding examples have given you just a small taste of how wildcard symbols operate. Now we can look at some of the wildcard commands themselves to see how they operate and how they can save you hours of work.

6.6 WILDCARD SYMBOLS IN COMMANDS

All the commands described in the remainder of this chapter recognize wildcard symbols. There are countless ways to use these commands and we'll try to show you as many uses for each one as we can. It would be impossible for us to give examples of all the possible combinations of options and wildcard symbols, so we encourage you to experiment and use your imagination.

The ultimate source of information on the features and use of any command is, of course, the Command Reference Sheet for that command. Each reference sheet contains a complete listing of command characteristics, switches, defaults, etc., so refer to your *AMOS* or *AMOS/L System Commands Reference Manual* for all the details on each command.

6.6.1 DIR - File Directory Display

You've already seen a few of the things the DIR command can do for you, but now let's explore some of this command's other capabilities. Before we go any further, though, here is a list of the command's syntax and defaults.

COMMAND SYNTAX:

```
.DIR {{listfilespec}=}{filespec1{,...filespecN}}{/switch{/switch}} RET
```

COMMAND DEFAULTS:

listfilespec	DIRECT.LST, and the device and account you're currently logged into.
filespec	*.*, and the device and account you're currently logged into.
/switch	/WIDE:1/NOBASE

How to Create a Directory File

The first thing you probably noticed about the syntax of this command is that it contains an optional listfilespec. This corresponds to the outfilespec you read about earlier in this chapter. If you specify a filename followed by an equal sign (or just an equal sign if you want to use the default DIRECT.LST), DIR creates for you a file containing the directory display you requested. Here are some examples:

```
.DIR/W = RET
```

(Creates a file named DIRECT.LST containing a four column wide list of the files in the account you're currently logged into.)

```
.DIR/W:3 FILES.LST=*.TXT,*.LST RET
```

(Creates a file named FILES.LST containing a three column wide list of the files in your current account that have .TXT and .LST extensions.)

```
.DIR TOC.LST[20,2]=MY??.*[24,1] RET
```

(Creates a file named TOC.LST containing a single column directory list in account [20,2] on your current device. This file contains a list of all the files in account [24,1] on your current device whose names contain four characters or fewer, start with the letters MY, and have any extension.)

How to Print a Directory List on the Printer

If you ever need to retain a permanent, printed record of a directory display, you can do it two ways.

a. Use the DIR command with a listfilespec to create a disk file that contains the display you want to print, and then print it with the PRINT command (more on PRINT later), or...

b. Substitute the name of the printer (or any other terminal) for the listfilespec in the DIR command. Ask the System Operator what the terminal identification of your printer is, or type TRMDEF followed by a RETURN to see a list of all the terminals attached to your system. Then to send a list, say, of all the .CMD files in all the accounts on the device you're logged into, to the printer TI810, enter:

```
.DIR TRM:TI810=*.CMD[]
```

The prefix TRM: tells DIR that the following name applies to a terminal rather than a disk file.

How to Find a Lost File

If you're looking for a specific file, but you can't remember where it is, you can use the DIR command to locate it for you. For example, if the file you're searching for is called LOST.FIL, enter:

```
.DIR ALL:LOST.FIL[] RET
```

This DIR command searches through every account ([]) on every disk device (ALL:) and displays the location of every LOST.FIL it finds.

How to Use Switches with DIR

The DIR command allows you to use a variety of switches. You can see a complete list of these switches and their capabilities in the DIR command reference sheet, but here are some highlights.

1. /DATA - (an operation switch) This switch causes DIR to list the complete file specification for each file it selects. Instead of displaying just the file name, for example: TARIFF.AMT, the /DATA switch tells DIR to display DSK5:TARIFF.AMT[143,10]. If you create a list file of a directory display using the /DATA switch, your AlphaBASIC

programs can read the file to access these file specifications and determine which files to open for data transfer.

2. /KILL - (an operation switch) Normally, when you use DIR to create a file containing a directory list, DIR won't create the file if one already exists with the same name in that particular device and account. The /KILL switch tells DIR to go ahead and destroy the previously created file, if there is one.

3. /WIDE - (an operation switch) This switch tells DIR to display the list of file names in four columns instead of one. You can specify a different number of columns by placing a colon and the desired number after the switch. Like so:

```
.DIR/WIDE:3 RET
```

6.6.2 COPY - Transferring Files

Many times, it's more convenient to make a duplicate copy of a file and change it to suit a new purpose than it is to create a new file from scratch. It's also desirable to make backup copies of important files frequently to prevent their loss in case of accident or emergency. The COPY command, by virtue of its wildcard features, is suited perfectly to this task.

COMMAND SYNTAX:

```
.COPY {outfilespec}={infilespec1{,...infilespecN}}{/switches} RET
```

COMMAND DEFAULTS:

outfilespec

., and the device and account you're currently logged into. If the System Operator uses COPY from account [1,2] on any disk, the same defaults apply except the default account number is the wildcard symbol []. This means that unless the device and account are explicitly stated, the files the System Operator copies are placed on the device the System Operator is logged into, in the account number corresponding to the one they came from.

infilespec	*.*, and the device and account you're currently logged into.
/switch	/DELETE/NOQUERY

How to Copy a File Within Your Account

To make a duplicate copy of a file, you can use the COPY command. Choose a filename that's not already being used in your account (use the DIR command to see the file names), and enter the command like this:

```
.COPY NEWFIL.DAT=OLDFIL.DAT RET
```

where OLDFIL.DAT is the existing file and NEWFIL.DAT is the file you want to create. Now you have two copies of the same file under two different names, and you can modify one or the other to suit your needs.

How to Copy a File from Another Account

Many times the file you want a copy of is in a different account. You can copy it into your own account under the same name (if that name is not used already in your account), or you can give it a new name. Note that if a file already exists in your account with the same name, the copy is written over it and the file's former contents are lost. Here's an example of copying a file from another account:

```
.COPY MINE.TXT=DSK3:YOURS.TXT[120,2] RET
```

In this example, the file YOURS.TXT was copied from DSK3:, account [120,2] into a file called MINE.TXT on the device and account you're currently logged into.

If you want to copy several files into your account keeping their names the same, the command might look like this:

```
.COPY =DFCLT.*[120,2],HWK0:EASY.*[50,6] RET
```

Here we copied all files named DFCLT from account [120,2] on the device you're currently logged into, and all the files named EASY from account [50,6] on HWK0:, into the account you're currently logged into keeping the same names they had before.

How to Copy a File Between Accounts

Most of the time, you'll be copying files to and from the account you're logged into. You can also copy files between accounts without being logged into either one as long as they share the same project number as the account you are logged into.

The System Operator has the added ability to copy files to and from accounts that he or she is not logged into, regardless of project number. The System Operator must be logged into account number [1,2] on any device to do this, however. If someone needs a copy of all the program files in a certain account written in AlphaBASIC, the System Operator could enter a command like this:

```
.COPY DSK2:[44,10]=DSK0:*.BAS[12,3] RET
```

The System Operator's account should have special protection so no one can log into it without knowing the special password. If you need something like this done, submit your request to the System Operator.

Also, if the System Operator is logged into [1,2] and tries to copy files into an account that hasn't been established on the disk, COPY automatically creates the new account, and gives it the same password as the original, if any.

How to Use Switches with COPY

Like the other wildcard commands, COPY allows you to use a number of switches which enable you to alter the effect of the command. Check the COPY command reference sheet for a complete list of switches.

1. /QUERY - (a file switch) This switch tells COPY to pause before copying a file and wait for you to enter a Y (Yes) or an N (No). You don't need to press RETURN.

   If you enter Y, COPY goes ahead and copies the file, but if you enter N, it doesn't. You can deactivate the /QUERY switch with the /NOQUERY switch. Here's an example:

   ```
   .COPY/Q *.MAJ=*.MIN,*.INT/NOQ,*.KEY,*.FLT,*.SHP RET
   ```

   This command pauses before copying all the files with .MIN, .KEY, .FLT, and .SHP extensions, but copies the .INT files without stopping.

2. /NODELETE - (a file switch) The /NODELETE switch tells COPY not to delete a file if it has the same file name and extension as the outfilespec in the command line. This

prevents you from losing a file that might have a duplicate filename. The /NODELETE switch counteracts the /DELETE switch, the default, which causes the COPY command to erase an existing file before copying one with the same filename and extension to that same account and device.

6.6.3 ERASE - Deleting Files

After you're finished using a particular file, there is no need to keep it on your disk any longer. You might as well get rid of it and make room for new files. This is especially true if you already have a backup copy of the file on a removable disk or video cassette, so here is the ERASE command. We used ERASE in several examples near the beginning of this chapter, so you already know a lot about this command.

COMMAND SYNTAX:

```
.ERASE {outfilespec=}filespec1{,...filespecN}{/switch} RET
```

COMMAND DEFAULTS:

outfilespec	*.*, and the device and account you're logged into.
filespec	The account and device you're logged into. The extension defaults to a null extension (one that's zero characters long).
/switch	/NOQUERY

How to Delete Files from Another Account

You can erase files from your own account or from another account as long as the project number is the same as your own, even if the accounts are on different devices. For example, if you are logged into account [200,2] on DSK5:, you can enter the following command:

```
.ERASE *.BAK,DSK4:CRT?.*[200,5],VCR?.*[200,6] RET
```

In this command, ERASE deletes all the files having .BAK extensions from the account you're logged into, [200,2]. DSK4: sets the default device for the rest of the command line, and ERASE deletes all files with four character names that start with CRT from account [200,5] and all files with four character names that start with VCR from account [200,6], both on DSK4:.

If you try to delete a file that doesn't share the project number of your own account, you see an error message like this:

```
    ?cannot delete FILESPEC - protection violation
```

The System Operator, when logged into DSK0:[1,2], can use the ERASE command to delete any file in any account.

How to Delete Duplicate Files

When you specify an outfilespec using wildcard symbols, the ERASE command compares the infilespecs to the outfilespecs and if there is a match, it deletes the matching outfile. Here's how it works:

```
    .ERASE *.GRT=*.NOM[105,0] RET
```

ERASE selects all the files with .NOM extensions in account [105,0] and compares them to all the files with .GRT extensions in the account you're currently logged into. If there are two files with the same filename, ERASE deletes the one with the .GRT extension from your account.

You can use this feature within your own account to get rid of duplicate files you may no longer need.

How to Use Switches With ERASE

The only switches available with the ERASE command are the /QUERY and /NOQUERY switches. Both are file switches. The query switch forces ERASE to pause before erasing the files you specify. ERASE waits for you to enter either Y (Yes) or N (No) before erasing the file and proceeding on to the next. It's not necessary to press RETURN after the Y or N. Suppose you're logged into account [20,3]...

```
    .ERASE *.BAK,/Q A*.BAS,DEBIT.*,CREDIT.*,SUM??.*/NOQ RET
```

In the command above, ERASE deletes all the files with .BAK extensions in the account you're logged into. The /Q switch sets the default to QUERY for the next group of files. ERASE pauses before erasing all the .BAS file that begin with the letter A, all the DEBIT, and all the CREDIT files. Finally, the /NOQ switch counteracts the /Q switch for the last group of files, all those with five character filenames that begin with the letters SUM.

6.6.4 RENAME - Changing File Names

The RENAME command operates much the same as the COPY command. The difference is that RENAME gets rid of the file under the old name. It's really a combination of the COPY command and the ERASE command, but it saves you from having to perform those two separate steps to change the name of a file or group of files.

COMMAND SYNTAX:

```
.RENAME outfilespec=infilespec1{,...infilespecN}{/switches} RET
```

COMMAND DEFAULTS:

outfilespec — The default device and account is ALL:[], and the default filename is *.*.

infilespec — *.*, and the device and account you are currently logged into.

NOTE: You must specify at least a partial outfilespec and partial infilespec for this command to work correctly.

/switches — /NODELETE/NOQUERY

How to Change Filenames in Your Own Account

Occasionally, you may wish to rename a group of files in your account. If you should want to change the extension of all your .BAK files to something else, like .SAV, you could enter:

```
.RENAME *.SAV=*.BAK RET
```

If you want to organize a variety of files with different extensions into a group with the same extension, you could say:

```
.RENAME *.GRP=HEADER.TXT,AR*.SAV,MKT??.LST RET
```

Here, we renamed the HEADER.TXT file, all the files whose names start with AR and have .SAV extensions, and all the files whose names begin with MKT, are five characters or fewer, and have .LST extensions. Now they all have .GRP extensions.

How to Change Filenames in Other Accounts

It's possible for you to rename files in accounts that have the same project number as the one you're logged into. The other account can be on the same device as your account, or on a different device, as in this example:

```
.RENAME MFG*.*=DSK1:[55,2]BOM?.BAS,WRK??.LST,COG.L? RET
```

Assuming your project number is also 55, this rename command changes the names of all the files in account [55,2] on DSK1: whose four character names start with BOM and have .BAS extensions; all the files in that account whose five character names start with WRK and have any extension, and all the files in that account named COG whose two character extensions start with L.

These files all stay in the same account you found them in, and they keep their original extensions, but now the first three characters of all their names are MFG.

You can't move a file to a different account using the RENAME command, so don't specify an account in the outfilespec. If you do, an error message like this appears on your screen:

```
?cannot rename FILESPEC - protection violation
```

The System Operator can rename files in any account, regardless of project number, when logged into DSK0:[1,2].

How to Use Switches with RENAME

The RENAME command recognizes two pairs of switches.

1. /QUERY - (a file switch) Depending on where you place this switch in the command line, this switch forces RENAME to pause before renaming certain files and wait for you to respond either Y (Yes) or N (No).

   This switch enables you to selectively rename files within a large group without having to enter each filename in a RENAME command line. The effect of this switch can be reversed by using the /NOQUERY switch.

2. /DELETE - (a file switch) If you specify this switch, RENAME deletes an existing file if you are renaming another file to the same name. You can turn off this switch by using the /NODELETE switch.

6.6.5 PRINT - Getting a Printed Listing

Nine times out of ten, you'll want to get a printed listing of your file. Often it will be the final copy of a letter or memo, a listing of your AlphaBASIC or AlphaPASCAL program, or perhaps a copy of the report that your program generates.

Since a printer can only print one file at a time, a special program known as the Spooler maintains a list of the surplus files until the printer finishes printing the file it's currently working on. The list of pending files waiting to be printed is called the "print queue." And as soon as one file is finished printing, the spooler checks to see if there is another file waiting in the queue to print. If there is, it automatically sends this next file to the printer.

COMMAND SYNTAX:

```
   .PRINT {printerspec=}{infilespec1{,...infilespecN}{/switches}} RET
```

COMMAND DEFAULTS:

printerspec — The default printerspec is the printer having the fewest blocks of data currently in its print queue or, more likely, the printer set as the default by the System Operator.

infilespec — The default infilespec is the account and device you're currently logged into, and extension of .LST, and a null filename.

switches — The default switches are set up by the System Operator when the Printer Spooler is installed on your system. Check with your System Operator for the current switch defaults.

How to Check the Print Queue

To find out how many files are currently waiting to be printed, or to find out how many blocks of the currently printing file remain, you can use the PRINT command without any arguments:

```
          .PRINT RET
```

If the print queue is empty, a message on your screen will tell you so.

How to Print Multiple Files

When you have a series of files to print, you can use a single PRINT command to do so. If the files in that series share common elements in their names, you can make the command even simpler by using wildcard symbols. For example, if you have written a series of chapters for a certain document, and their filenames all begin with CHAP, to print them you might enter:

```
.PRINT CHAP* RET
```

This command causes all the files in the account you're currently logged into, whose filenames begin with CHAP and have .LST extensions, to be placed in the print queue.

If you have more than one printer attached to your system, you can specify which one you want the file printed on by entering a printerspec:

```
.PRINTER DIABLO=*.BAS,*.TXT RET
```

The command above places all the files with .BAS and .TXT extensions into the print queue for the Diablo printer.

Check with the System Operator to find out the names of the printers on your system.

How to Use Switches with PRINT

The PRINT command has a large number of switches that you can use to control the files you send to the printer. They are all described in detail on the PRINT reference sheet in the AMOS or AMOS/L System Commands Reference Manual. The most important ones are described below.

1. /COPIES:n - (a file switch) Prints the number of copies you specify.

2. /KILL - (an operation switch) Deletes the files specified from the print queue. If the file is currently printing, it will stop printing when it reaches the end of the current block.

3. /FORMS:xxxxxx - (a file switch) Forces the printer to pause before printing a file if the forms name is not the same as for the previously printed file. This allows you time to change the paper. The typical default is /FORMS:NORMAL if you do not use this switch. The System

Operator must specify the default form name when the printer spooler is first set up.

4. /WAIT - (an operation switch) If you want to print more files than the spooler can keep track of, the extra file requests are discarded. You can use the /WAIT switch to tell PRINT not to discard these file requests if there's no room for them in the print queue. The catch is that PRINT ties up your terminal until it's able to print enough of the earlier files to make room in the queue for the extras.

In this chapter, you've read about several useful commands that use wildcard symbols. But there are many more commands in Alpha Micro's repetoire, and in the next chapter we'll discuss two of them that can lighten your workload a great deal.

CHAPTER 7

COMBINING AND SORTING FILES

This chapter introduces you to two commands that you can use to combine several files together and organize according to conditions you specify. These two commands do not recognize wildcard symbols.

Many times, a large file can be organized into several smaller sub-files which are easier to keep track of and to work with. Then, when you're ready to process all the files together, you can combine them into one. If the records in these files need to be arranged in a certain sequence, you can sort them into the proper order. Here are the two topics we'll talk about in this chapter:

- Appending several files together
- Sorting data within files

Let's start with the APPEND command.

7.1 THE APPEND COMMAND

This command allows you to combine two or more files into one. You can use APPEND to create an entirely new file that contains the same data as its components, or you can add additional data to an existing file.

COMMAND SYNTAX:

```
    .APPEND outfilespec=infilespec1{,...infilespecN} (RET)
```

COMMAND DEFAULTS:

outfilespec — At least a partial filespec is required. The extension defaults to a null extension (that is, an extension zero characters long), the account and device default to the account and device you're currently logged into.

infilespec — At least one partial filespec is required. The account and device default to the account and device you're currently logged into. The extension defaults to the extension of the outfilespec, and each subsequent infilespec defaults to the extension of the infilespec preceeding it.

7.1.1 How to Combine Files

For certain purposes, such as formatting a group of text files, you may wish to combine the files into a completely new file rather than processing them separately. You can do it like this:

```
.APPEND BOOK.TXT=INTRO,CHAP1,CHAP2,CHAP3 RET
```

In this command, we combined four individual text files into a new file call BOOK.TXT. All these files defaulted to the device and account we were currently logged into, and their extensions all defaulted to .TXT. BOOK.TXT now contains a copy of each file in the order you specified, and the original files remain intact.

7.1.2 How to Add Files to an Existing File

You don't necessarily have to create a new file each time you append files together. You can simply append additional files to an existing one. Here's how:

```
.APPEND PRGM.BAS=PRGM.BAS,RTN1.BAS,RTN2.BAS,RTN3.BAS RET
```

Note that we used the same file, PRGM.BAS, as both the outfilespec and the first infilespec. All the subsequent files are thus added on at the end of the existing PGRM.BAS file. (We could have placed PRGM.BAS anywhere in the string of infilespecs depending on what order we wanted the files arranged in.) PRGM.BAS now contains its former self plus the contents of the other three files. Those other three files remain as they were.

7.2 THE SORT COMMAND

The SORT command arranges the records in a single sequential file in the order you specify. SORT asks you several questions to determine your criteria and arranges the records in the file accordingly. SORT only works on sequential files, and like APPEND, SORT does not recognize Wildcard Symbols.

COMMAND SYNTAX:

```
.SORT filespec  RET
```

COMMAND DEFAULTS:

filespec — SORT assumes the device and account you're currently logged into. The file extension defaults to .DAT (indicating a data file).

7.2.1 How to Use the SORT Command

Every file is made up of individual records, and each record is a collection of related information about a certain subject. SORT enables you to reorganize the records within a single file in either ascending or descending alpha-numeric sequence. Each record in your file must be terminated with a RETURN. Here's a sample file that contains customer address records:

```
         1    1    2    2    3    3    4    4    5    5    6    6    7
....5....0....5....0....5....0....5....0....5....0....5....0....5....0

001   Carlson, Ander      43 Beekman Place    New York   NY   12/24/79
002   Lorenson, Will      P.O. Box 1221       San Diego  CA   04/30/77
003   Mathews, Michael    6701 23rd Street    Denver     CO   04/27/77
004   Lincoln, Thomas     12992 Dover Place   Concord    MA   08/01/78
005   Hunt, Linda         605 Charleston St.  Las Vegas  NV   06/30/78
006   Carter, Marilyn     5443 Via Verde      Tucson     AZ   02/19/79
007   Lincoln, Jeff       4225 Larimore       Omaha      NE   06/30/81
008   O'Grady, Rosie      R.R. 1, Box 143     Goleta     CA   06/02/81
009   Walker, Joan        4 Pleasant Circle   Newport    RI   09/14/80
010   Smith, Steven       1675 Seaside Drive  Monterey   CA   01/25/82

....5....1....1....2....2....3....3....4....4....5....5....6....6....7
         0    5    0    5    0    5    0    5    0    5    0    5    0
```

(NOTE: The scales at the top and bottom are not part of the file; they're there to make it easier for you to figure out the relative positions of the data within each record.)

Each of the ten records in this file contains the same pieces of data: a customer ID number, the name of the customer, the customer's address, and the date of the customer's last purchase.

Take a minute or two to use AlphaVUE to create a file called CUST.DAT containing the ten records shown above. When you're ready to sort them, enter:

```
.SORT CUST.DAT RET
```

Now SORT asks you several questions to determine how you want the file sorted.

1. Record size: Although the records in our file are all the same length (70 characters), that might not always be the case. Some files may contain records of differing lengths, so you must enter the maximum record size in your file. If you underestimate, any records larger than the size you specify will be truncated to that length in the sorted file. The records in our file are 70 characters long, so enter that number.

2. Key size: A key is a piece of data within each record that you want the file sorted by. Let's put the file in alphabetical order by customer name. We've allowed for twenty characters in the name field of our file, so enter 20 here.

3. Key position: Now you have to tell SORT where that field begins in the records to be sorted. Looking back at the listing of the individual records, you can see that the customer name starts in position 7 of the record. Enter that number.

4. Key order: Do you want the records sorted in ascending or descending order? Answer A for ascending or D for descending.

5. Key size: If you want to specify an additional sort key, you can enter the size of a second key field here. SORT then requests the Key position and Key order for the second key. You can specify up to three key fields each time you sort a file. Or just enter a RETURN, and SORT begins rearranging your file right away.

It takes SORT only a short time to reorganize your file, and when it's done, SORT displays a group of statistics similar to the ones shown below:

```
Records sorted: 10
Runs generated from sort phase: 0
Passes over data in merge phase: 0
Record comparisons made: 20
Sector reads: 0
Sector writes: 0
```

7.2.2 Some SORTing Hints

You can sort files that are larger than available memory (as little as 6K is enough in some cases), but as a general rule the more memory you have available, the more efficiently (faster) SORT works.

If your file won't fit into memory, SORT performs the sort on disk. This takes longer, but it still gets the job done.

When you specify the dimensions of the key fields, make sure they fall completely within the record to be sorted. For example, in our little customer file, you could not specify a key size of 6 and a key position of 69 since the record is only 70 characters long.

You can specify as many as three key fields for SORT to use. The first field is the major sort key and determines how the entire file is sorted. Those records that have duplicate information in the major sort key field are arranged in order by the second key field you specify. And finally, records that have duplicate information in both the major and intermediate sort keys are arranged in order by the third, or minor, sort key field.

If you use a date as a sort key, keep the format of the date field in mind. In our customer file the date field is in the MM/DD/YY format, where MM represents the month, DD represents the day, and YY represents the year. When we sort by this field, the year is more important than the month and day. We therefore have two choices: we could either convert all the dates on our file to YY/MM/DD format, or we could split the date field into two separate keys.

If we decided to specify two separate keys, two digits for the year and five digits for the month and day, the process would look something like this:

```
.SORT CUST.DAT RET

Record size:  70 RET

Key size:  2 RET
Key position:  69 RET
Key order:  A RET

Key size:  5 RET
Key position:  63 RET
Key order:  A RET

Key size: RET
```

And the file would be arranged like this:

```
003   Mathews, Michael   6701 23rd Street    Denver     CO   04/27/77
002   Lorenson, Will     P.O. Box 1221       San Diego  CA   04/30/77
005   Hunt, Linda        605 Charleston St.  Las Vegas  NV   06/30/78
004   Lincoln, Thomas    12992 Dover Place   Concord    MA   08/01/78
006   Carter, Marilyn    5443 Via Verde      Tucson     AZ   02/19/79
001   Carlson, Ander     43 Beekman Place    New York   NY   12/24/79
009   Walker, Joan       4 Pleasant Circle   Newport    RI   09/14/80
008   O'Grady, Rosie     R.R. 1, Box 143     Goleta     CA   06/02/81
007   Lincoln, Jeff      4225 Larimore       Omaha      NE   06/30/81
010   Smith, Steven      1675 Seaside Drive  Monterey   CA   01/25/82
```

The two commands we've discussed in this chapter are extremely useful but they can also cause some confusion if you use them incorrectly. Before you start manipulating your files drastically, it's a good idea to make a backup copy that you can always go back to, in case something unforeseen happens. The next chapter explains several different ways to back up your data.

CHAPTER 8

BACKING UP YOUR FILES

If there is one lesson learned early in the computer business, it is this:

BACK UP YOUR DATA !!!

Backing up your programs and data by regularly copying them onto a spare disk, magnetic tape, or video cassette is perhaps the most important habit you can develop.

Computers are generally very reliable machines, but nobody can completely guard against such an accident as someone tripping over the power cord or spilling a cup of coffee into the disk drive. There are also natural disasters such as power outages, floods, fires, etc. that cannot be prevented, but must be planned against.

Your programs and data are irreplaceable, so keeping a current copy of them in a safe place will save you the frustration of losing several days worth of programming or data entry.

In this chapter, we will discuss the following topics:

- Backup Procedures
- Backing up a single account (COPY)
- Backing up an entire disk (DSKCPY)
- Backing up onto a VCR cassette

8.1 BACKUP PROCEDURES

We will discuss specific backup techniques in a later section of this chapter. For now, let's talk about some other things that you can do when using the system to keep data loss to a minimum.

Since Alpha Micro supports a number of types of data recording devices, we will refer to those devices as "backup media" or "media." These include such methods of data storage as Hard disks (Winchester, Hawk or Phoenix disk drives), Floppy disks, 1/2" Magnetic tape drives, 1/4" Streaming tape drives, and Video Cassette Recorders.

1. Whenever you change a disk cartridge or floppy disk, you MUST use the MOUNT command; if you do not, AMOS may well write over data that is already on the disk. (See Section 2.5.1. for an explanation).

2. If your text files begin to show strange typographical errors (that you didn't make), ask the System Operator to run a memory diagnostic test program -- you may have a memory problem. (The System Operator should be running memory and disk diagnostic tests on a regular schedule, anyway.)

3. Keep your text editing sessions short; never use one of the text editors for more than half an hour or so without exiting and saving your file.

   If that unexpected catastrophe should occur, it is better to have lost a half an hour's worth of work instead of a day's. (The changes and additions that you make while you are editing are only changed in your temporary memory, you have to save to permanent memory in order to make those changes permanent.)

4. As a general rule, never leave your terminal without logging off the system (using the LOGOFF command). Change your password occasionally (if you have one). These measures protect your system and accounts from unauthorized use.

5. Treat your backup media with respect. Whether your backup media are disk cartridges, floppy disks, magnetic tape, or video cassettes, handle them carefully and gently.

   Do NOT stack your media in tall piles on top of the disk drives, and do NOT leave them lying about where they can be knocked off counters or desks.

If at all possible, you should have several spare disks, tapes, or video cassettes that you can use for backup. You should never have just one backup media (what happens if something should really go wrong while you are in the process of backing up, and you lose the data on both your backup and original media?)

The usual procedure is to follow the "grandfather-father-son" philosophy of backup. That is, you always have several backups of varying age. The next time you back up, use the media with the oldest version of your data (the "grandfather").

By rotating your backup media, you can ensure that you always have several fairly recent backups of your data.

8.2 BACKING UP THE FILES IN YOUR ACCOUNT

The command that you will use to transfer copies to your backup media will vary depending on what the media is. COPY is one of the commands you use to copy between disks.

You can use the COPY command to make backup copies of all the files in your account, all of your accounts, and even of all accounts on the disk. The COPY command is a wildcard command, so you may want to review Chapter 6, "Using Wildcard Symbols," before proceeding.

Use the SET command to tell AMOS to notify you of any soft disk errors encountered during the backup procedure:

```
.SET DSKERR RET
```

Now, decide what files you want to back up. For example, let's say that you are logged into DSK0:[300,4], and want to copy all files in that account to the same account on DSK1:.

```
.COPY DSK1:= RET
CRMINV.BAS to DSK1:CRMINV.BAS
RESET.BAS to DSK1:RESET.BAS
CRITEM.BAS to DSK1:CRITEM.BAS
PRINT.TXT to DSK1:PRINT.TXT
Total of 4 files transferred
```

The fact that nothing appears on the right side of the equal sign in the command line above tells COPY that we want to copy ALL the files from the device and account we are logged into.

The DSK1: on the left hand side of the equal sign tells COPY to transfer copies of those files under their own names to the same account on DSK1:.

The next command:

```
.COPY DSK1:[]=[300,*] RET
FUNC.BAS[300,1] to DSK1:FUNC.BAS[300,1]
PRTYP.LIT[300,2] to DSK1:PRTYP.LIT[300,2]
IBUF.LIT[300,2] to DSK1:IBUF.LIT[300,2]
PARSER.LIT[300,10] to DSK1:PARSER.LIT[300,10]
Total of 5 files transferred
```

tells AMOS to copy all of the files from all of the accounts in Project 300 on the device you are logged into (DSK0:) over to the same accounts on DSK1:.

You must remember to include the wildcard account symbol [] on the left side of the equal sign -- if you do not, AMOS will copy every file on the device you are logged into (DSK0:) into the account that you are logged into on DSK1:. (In the example above, you must be logged into an account in Project 300 to avoid a protection violation error.)

You should perform file backup frequently if you create and change files often. Perhaps at the end of every working day you should copy those files onto a backup disk.

Creating a command or DO file to perform this function for you can make your task simpler; see BACKUP.DO" in Section 10.3.3, for an example of a DO file that backs up files from one disk cartridge to another.

8.2.1 The System Operator and the COPY Command

The System Operator (if logged into [1,2]) can use COPY to back up all of the accounts on a disk regardless of their project numbers.

For example:

```
.COPY HWK1:[]=DSK1:[] RET
```

copies all of the files in all of the accounts on DSK1: over to their corresponding accounts on the device HWK1:.

NOTE: The System Operator may omit the wildcard PPN symbol in the outfilespec-- when it is used from account [1,2], COPY assumes a default outfilespec of *.*, the device you are logged into, and an account specification of [].)

8.3 BACKING UP ENTIRE DISKS (DSKCPY)

Rather than copying over individual files from one device to another, DSKCPY makes a literal image copy of an entire disk. We will assume for the purposes of this discussion that you want to back up the System Disk.

You can only copy between devices of the same type; that is, where the first three characters of the device name are the same, (e.g., AMS1: to AMS2:, DSK2: to DSK3:, DDS1: to DDS2:, etc.).

If you are using a floppy-disk based system, place the backup disk in a drive other than the System Drive. If you are using a hard-disk based system, see the note at the end of this section for cautions.

The backup disk must have been formatted at some time in the past. (Check with the System Operator for information on what formatting program to run to format a brand new disk.) Note that DSKCPY completely obliterates any data previously on the backup disk.

To MOUNT the backup disk, enter:

```
.MOUNT DSK1: RET
```

Then enter the DSKCPY command. The DSKCPY program first asks you for the INPUT DRIVE. Enter the specification of the device you are going to back up. Then the DSKCPY program asks you for the OUTPUT DRIVE. Enter the specification of the disk you are going to copy onto (the backup device).

For example, if you want to copy DSK0: onto DSK1:, use the DSKCPY command in this way:

```
.DSKCPY RET
Input drive: DSK0: RET
Output drive: DSK1: RET
```

DSKCPY begins to make a literal image of DSK0: onto DSK1:, and displays how many blocks it is copying. After it has finished the backup, DSKCPY tells you that it has finished copying the disk; then it proceeds to verify the duplication by checking the data on the backup disk against the data on the source disk.

After this verification is complete, DSKCPY returns you to AMOS command level. The entire process looks something like this:

```
.DSKCPY RET
Input drive: DSK0: RET
Output drive: DSK1: RET
[Copying 9696 records]

[Duplication completed]
[Verification completed]
.
```

NOTE FOR HAWK HARD-DISK SYSTEM USERS:

If you have been working on a disk cartridge, and want to back up the entire disk, you can use DSKCPY.

However, this process is tricky if you only have one Hawk drive, since you must back up the System Disk, temporarily load certain programs into memory, and then erase the contents of the System Disk in order to transfer data.

Therefore, this procedure must not be undertaken lightly. We advise that the System Operator be in charge of this kind of operation.

Of course, if you happen to have more than one Hawk drive on your system, you can simply copy the cartridge from one drive onto a backup cartridge on the other Hawk drive.

8.4 BACKING UP ONTO VIDEO CASSETTE RECORDERS

Alpha Micro supports the use of video cassette recorders as backup media. VCRs are portable, relatively inexpensive, and can also be used for other business purposes when not being used with your computer.

If your VCR contains a time of day clock that allows you to program it to begin recording at a specific time, you can combine that with the VCR software to automatically run your backup at night.

The command you use to save data onto video cassettes is called VCRSAV. There are also other commands that are used with the VCR, such as VCRDIR, which display a directory of what is on your VCR tapes.

The procedure for the backup of data onto video cassettes is fully explained in our manuals, AMOS Video Cassette Recorder Backup Software, DSS-100027-00, and AMOS/L Video Cassette Recorder Backup Software, DSS-10032-00.

8.5 BACKUP ON OTHER MEDIA

We also support two kinds of magnetic tape backup systems. These are the 1/2 inch magnetic tape drives and the 1/4 inch streaming tape drives.

Information on the 1/2 inch magnetic tape drive can be found in the AMOS System Operator's Guide, DSS-10001-00, and in the AMOS/L System Operator's Guide, DSS-10002-00.

Information on the 1/4 inch streaming tape drive can be found in AMOS 1/4" Streamer Backup Software, DSS-10033-00, and in AMOS/L 1/4" Streamer Backup Software, DSS-10031-00.

Finally, there are our programs for backing up onto Floppy Disks from Winchester disks. See the AMOS System Operator's Guide, DSS-10001-00, for the AMOS system; and the AMOS/L System Operator's Guide, DSS-10002-00, for the AMOS/L system.

CHAPTER 9

ACCOUNTS AND PASSWORDS

Every user of the system has one or more accounts in which to create, store, and edit files. When you create files on a disk, the system marks those files as belonging to the account you are logged into.

Each account has a directory that lists all of the files in that account. It may seem easier to have just one account, but it can get very hard to find your files if you have a lot of them. The best way to organize your accounts is by subject. If you set up each account so that it contains the same type of files (for instance, one account for storage, one for memos, one for letters, one for inventory documents, one for purchase order records, etc.), it will be easier to find the files you are looking for.

To provide security for your programs and files, you can password protect your accounts. In this chapter, we will discuss accounts and passwords and their use. The subjects covered will be:

- Project-Programmer numbers
- Passwords
- Logging into another account
- Start command files
- Sending messages to other system users

9.1 PROJECT-PROGRAMMER NUMBERS

Accounts are assigned to each user on the system by the System Operator. When you are assigned an account, you are given the account number, which consists of a project number and a programmer number.

Here are some examples of project-programmer numbers:

[110,5] [334,7] [250,12] [200,1] [100,100]

The account number is a unique, two-part number that distinguishes your accounts from all other accounts on the same device. The first number is called the project number.

If several users' accounts have the same project number, those users are said to be in the same project. Users in the same project have certain privileges when it comes to transferring files between each others' accounts.

For instance, it is easier to copy files between accounts that are in the same project. See Chapter 6, "Using Wilcard Symbols", the COPY command, for examples.

Here is an example of several accounts that are in the same project (100):

[100,1] [100,7] [100,44]

The second number is called the programmer number, and is separated from the project number by a comma (in the example above, the programmer numbers are 1, 7, and 44). You will usually see account numbers enclosed within brackets.

Both the project and the programmer number may range from 0 to 377, octal. (For a discussion of the octal numbering system, see the manual Introduction to AMOS, DWM-00100-65; for now, the fact that the numbers are octal just means that no single digit may be greater than 7.)

A programmer number of zero usually indicates a library account for that project (e.g., the account [311,0] contains files of interest to all of the users in project 311).

Project numbers 1-77 are reserved by Alpha Micro for system software and for the AlphaACCOUNTING business package. See Section 6.3.3 for a list of the system software accounts.

9.2 PASSWORDS

The System Operator may assign you an account password. The password is for your protection; if you maintain its secrecy, other users cannot log into your account.

You do not have to enter a password if you are transferring to an account that is in the same project. That is, if you are in account

[100,2], and you want to log into account [100,1], you would not have to enter the password if account [100,1] was password protected. You would have to know the password if you were transferring from an account such as [105,5].

If an account is password protected, you will be asked to enter the password when you log in:

```
.LOG 100,3 RET
Password:       RET
```

The system does not display your password on the terminal screen as you type it. This prevents anyone from seeing what you entered. Remember that the purpose of the password is to keep unauthorized users from gaining access to the computer through your account. Keep your password a secret!

Once you type your account number and password correctly, AMOS will tell you that you are logged into the system:

```
.LOG 100,1 RET
Password:       RET
Logged into DSK0:[100,1]
```

If you make a mistake in typing your account number or password, you will see one of the following errors messages:

```
?Account number invalid

?Bad password

?Command format error
```

and you will have to re-enter the LOG command with the correct information.

9.3 LOGGING INTO ANOTHER ACCOUNT

Once you are logged into an account on the system, you can transfer to another account by using the LOG command with the account number of the account you want to tranfer to:

```
.LOG 100,3 RET
Password:       RET
Transferred from [321,10] to [100,3]
```

You can take short cuts if you are transferring to an account with either the same project number or the same programmer number. For example, if you are in account [100,1] and wish to transfer to account [114,1], you can type:

```
.LOG 114, RET
```

and AMOS will log you into [114,1] since you were already using programmer number 1. Likewise, if you are in [100,1] and you want to transfer to [100,4], type:

```
.LOG ,4 RET
```

and AMOS will log you into [100,4].

If the account you wish to log into is on another disk, you might have to specify that device in the LOG command. You have to specify the device if there is another account with the same account number somewhere else, or if the device is on a different type of disk (for instance, transferring from an account on DSK0: to one on HWK1:).

For example, if you are currently logged into DSK3:[107,3], and you wish to transfer to DSK2:[100,5], enter:

```
.LOG DSK2:[100,5] RET
```

You should enter the DSK2: here to make sure that you get to the right place. If you did not, and DSK3: also contained an account [100,5], that is where you would end up.

If you're absolutely sure that there is only one account [100,5] on the system, you don't have to enter the DSK2: -- AMOS will search the other logical units of the same physical device beginning with unit #0. However, if that account is on a different physical device, such as HWK1:, you must include the device specification in your LOG command.

9.4 THE START COMMAND FILE

Whenever you log into an account, AMOS looks for a file in that account named START.CMD. If such a file exists, the system assumes that it is a command file, and begins executing it as such.

You can use this special command file to perform certain procedures automatically every time you log into an account, such as erasing all backup files, creating a directory listing for that account, etc. It is also useful for placing a message on the screen that gives the purpose of the account, and perhaps what files are contained in it.

Command files of any other name will not execute automatically when you log into the account, only the file named START.CMD. We discuss command files in detail in the next chapter.

9.5 SENDING MESSAGES TO OTHER USERS (SEND)

The SEND command allows you to send messages to other jobs while you are logged into the system.

You simply type SEND, a space, the jobname of the person you wish to send the message to, and the message you would like to send.

The message can be one line long. The length of the line on your screen is defined in the system initialization command file. This is usually a little longer than the width of the screen. You can, of course, send more than one message, if it will not fit on one line. A typical SEND command might look like this:

```
    .SEND BILL WILL YOU BRING OVER THE SALES REPORT FOR LAST WEEK? RET
```

This message will print out on the terminal of the user whose jobname is BILL. It is usually effective, and an easy way to communicate short messages to other people on your system.

The person whose jobname is BILL will see:

```
    JOB -- WILL YOU BRING OVER THE SALES REPORT FOR LAST WEEK?
```

where JOB is the jobname of the person who sent the message-- in this case, you.

This is not a foolproof method, because sometimes the other user will be doing something that prevents the message from appearing, such as running a program. In this case, you will be given the message:

```
          ?Busy
```

or

```
          ?Guarded
```

indicating that your message did not get through. You can try again later, or use another method of contacting that person.

CHAPTER 10

COMMAND FILES, DO FILES, AND CONTROL FILES

One of the constants of everyday life is repetition. Especially in business, you find that certain tasks have to be done on a regular basis. The mail has to be opened every day, somebody has to make the coffee, and if you don't do the payroll each week, people tend to get upset.

Some of these tasks are done on your computer. One of the great strengths of computers is their ability to free people from repetitious tasks, and to speed up the completion of those tasks. However, sometimes the operation of the computer itself can be repetitious.

Many times a certain task on the computer is done by using the same string of commands and operations every time. Alpha Micro provides special files called command, DO, and control files that can contain these strings of commands and operations, making the ordinary operation of your computer easier.

This chapter will discuss the following points:

- What is a command file?
- How to create command files
- How to create DO files
- The Task Manager

10.1 WHAT IS A COMMAND FILE?

One of the important features of the Alpha Micro system is that rather than having a set of "built in" commands that AMOS recognizes, all of our commands are actually files stored on the disk.

In examples in earlier chapters, it looked like you were just typing the name of the command, but what you were really doing was telling AMOS to find and execute the file with that name.

Because of this feature, you can extend and customize the set of AMOS commands that already exist on your system by creating your own executable files.

And, you can design your own commands, combining the existing AMOS commands with special instructions. You can also erase or rename AMOS commands that do exist, if there are some commands you do not want people on your system to use, or if you want to control who knows and can use certain commands.

And the easiest way to do these things is to build command files.

A command file is a file that contains the same kinds of input that you might enter from the keyboard. You can tell AMOS to read its instructions from a command file instead of having to enter those commands and data yourself.

For example, let's say that you often erase all of your backup files, and then view the directory of your account. The process that you must go through to accomplish this task is:

```
.ERASE *.BAK  RET
MEMDOC.BAK
COMF.BAK
Total of 2 files deleted, 3 disk blocks freed

.DIR  RET
MEMDOC  TXT    12              DSK0:[100,4]
COMF    TXT     5
VAR     BAS    10
VAR     RUN     7
Total of 4 files in 34 blocks
```

The example above is a simple one; sometimes a frequently used sequence of commands may be quite long and tedious to type. An extremely powerful tool for dealing with this problem is the command file.

Let's say that you create the command file CLEAN.CMD to perform the functions in the example above. The file contains the following lines of text:

```
;This file erases BAK files and displays account directory
:T
ERASE *.BAK      ; Get rid of the backup files.
DIR              ; Display Files
```

The :T at the front of your command file is a special symbol that allows you to see the lines of your command file on your terminal display as AMOS processes them. The semi-colon (;) indicates that the text following it is a comment. See section 10.2.1 for what the comment means, and how it is used. All of these special symbols will be discussed later in this chapter.

With this command file, instead of entering what you entered in the first example each time, you merely have to enter:

```
        .CLEAN RET
```

and AMOS reads the file CLEAN.CMD and performs the process for you. With this small example, you do not save much typing, but imagine the typing and time you would save if the sequence of commands you regularly enter is ten or twenty lines (or more) long!

If the extension of a command file is .CMD or .DO, you do not have to include the extension when entering the name of the command file (you saw that you only had to type CLEAN rather than CLEAN.CMD).

Otherwise, you must specify the file extension. (That is, if your command file is called DOIT.TXT, you must include the .TXT when specifying the file.) This is why it is easiest to use .CMD (or .DO) extensions.

10.2 HOW TO CREATE A COMMAND FILE

To create a command file, use one of the text editor programs to make a text file (usually with a .CMD extension). Fill the file with the commands you would ordinarily enter from the keyboard, plus any special symbols.

A command file can contain most commands or data that you might enter from the keyboard; the file can even contain the name of another command file.

AMOS continues to read lines of text from the command file until the file ends. You may run, enter, and exit programs; supply data to programs; or perform system functions - all under the control of a single command file.

As an example, let us say that you have an AlphaBASIC program that you want to edit, compile and then test. Suppose that you also want to delete the backup file created by the editor. You might want to create a small command file to perform these functions:

WARNING: Do not use the commands MEMORY, JOBSIZ, or JOBMEM inside a command file. These commands change the memory allocation of your job; that is, they change the section of computer memory that you have been assigned. Since your command file and any other files it is using are in memory, changing that memory may cause serious problems.

```
:T
VUE AVRAGE.BAS
COMPIL AVRAGE      ; Compil the basic programs
RUN AVRAGE         ; Run the compiled program
3
12                 ; Enter data needed for the program
7
ERASE *.BAK        ; Erase backup files
```

The first line of the command file (:T) means that you see the lines of text in the command file as AMOS processes them.

The next line puts you into the text editing program, AlphaVUE, and tells AlphaVUE that you want to edit the AlphaBASIC source program named AVRAGE.BAS.

After you make some changes to the program, you leave AlphaVUE by using the AlphaVUE exit command (F). Now AMOS returns to your command file and reads its next instruction.

You are again at AMOS command level, and the next line of the command file tells AMOS to bring in the AlphaBASIC compiler and compile the AlphaBASIC program.

The compiler produces a new file named AVRAGE.RUN (the compiled version of your original source program). Line four of the command file tells AMOS to run the compiled program (AVRAGE.RUN).

The next three lines provide test data for the program, which asks for three numbers. When the AlphaBASIC program finishes, you are returned to AMOS command level.

The final command in the command file then tells AMOS to erase all backup files in your account.

10.2.1 Comments in a Command file

You may include comments within your file by preceding them with a semicolon (;). AMOS does not process the comments. If a :T symbol is used, any comments that follow it will be printed out on your terminal as the command file executes. If no :T is present, these comments will not be seen, but will remain in the file to guide anyone who looks at the file.

Comments serve as explanations of the purpose and function of a command file. Often a command file will not be used by just one person, but by many. If a person wants to do a certain task, that person might look at the contents of a command file to see if it performs the needed task or not.

If special symbols or unusual commands are in the file, the viewer might not know what they do, and so, might not understand what the command file does.

Therefore, it is a good idea to liberally comment your command files so that other users will know exactly what functions the file performs.

10.2.2 The System Initialization Command file

Of special interest to the System Operator is a unique command file called the system initialization command file that the system uses every time you turn on or reset the computer.

This command file has special properties and commands that help AMOS tailor the system software for your particular hardware system.

The System Operator can find information on the system initialization command file in the AMOS System Operator's Guide, DSS-10001-00 (for AMOS systems), or the AMOS/L System Operator's Guide, DSS-10002-00 (for AMOS/L systems).

10.2.3 Special Symbols in Command Files

In addition to normal input that you might make on your keyboard, there are several special symbols that appear only in command files that are never seen or processed by programs other than AMOS.

These symbols MUST appear at the very beginning of a command file line in order to be read as special symbols. The one exception is the comment.

:T TRACE - Causes the lines of the command file and all of the output to be displayed on your terminal screen as AMOS processes them.

Sometimes it's useful NOT to be able to see the commands and data in a command file as the file is processed; in that case, use the :R or :S symbols below.

:S SILENCE - Causes the display of command file lines and program output generated by the command file to be suppressed. When :S is used, nothing that the command file does will be displayed on the screen (except for messages). If no :R or :T symbol is specified, :S is assumed (the default).

:R RESULT - Causes the results of command file lines, but not the lines themselves, to be displayed. You can use multiple :S, :R, and :T symbols within one command file to allow the user of your command file to see some program output and command file lines, but not others.

:<...> MESSAGE - These symbols allow you to include messages within your command files that will be displayed on your terminal. The AMOS Monitor displays all the characters within the <> symbols when it reaches that point in the command file.

A command file message is not acted upon by AMOS or any other program. A message may be more than one line in length. The end of the message must be indicated by the > symbol. NOTE: Be careful! If you forget the > symbol, the command file will keep printing indefinitely.

:K KEYBOARD INPUT - The :K symbol allows the user of the command file to enter one line of data or commands to either AMOS or the program currently being executed.

When AMOS finds a :K symbol in your command file, it halts the processing of the file until the user of the file enters a line of characters that end with a RETURN. This input is then acted upon by AMOS or the program that is executing.

; COMMENT CHARACTER - the semicolon symbol marks a comment line which is not processed, but is displayed with the rest of the command file if a :T is present at the top of the file.

Certain commands, such as EXIT and LOOKUP, allow you to put a message on the command line. If you put a comment after these commands, they will be taken for these messages, and will print out on the terminal, regardless of whether :R or :S are used.

:P PARTIAL INPUT - The :P symbol is used in conjunction with the :K symbol to provide you with the ability to make a partial input. Whatever you input is added to the lines that follow the :P in the command file until it encounters a :K symbol.

For example, if you have these lines in a command file:

```
:T
:<Input the name of the file
  you wish to VUE:
>
:P
VUE
:K
```

the command file will stop and wait for your input, and then append to "VUE" whatever you enter to make one command line. If you enter TEST.TXT, the command file will execute the command VUE TEST.TXT.

Put the :P before the command you want to add to, and do not forget the :K symbol after, or it will add other command file lines also.

NOTE: The descriptions of the command file elements below are necessarily brief and do not go into details on command syntax and use. We just want you to know what is available to you. See the examples in section 10.3.3 for samples of how these are used, and see either the _AMOS System Commands Reference Manual_, DWM-00100-49 or the _AMOS/L System Commands Reference Manual_, DSS-10004-00, for the syntax of these commands.

PAUSE When PAUSE appears in a command file, the file stops and waits for the user of the command file to input a character. If the user of the file presses RETURN, the rest of the command file is then executed.

If the user of the file presses any other key, the control file is suspended, and whatever is left in the command file is placed into a file called CNT.CMD. You may also put a message on the PAUSE line, which will print out before the file pauses.

CONT CONTINUE - This command, used either inside or outside (at AMOS command level) of a command file, causes the command file CNT.CMD to execute. This command continues the execution of a command file that was interrupted by the PAUSE command.

LOOKUP The lookup command is used in conjunction with the GOTO command to cause AMOS to go to a new line in the command file if a specified file exists, or to end the command file if the file does not exist.

Enter LOOKUP followed by the filespec of the file you wish to find.

If the file is found, the rest of the command file is executed. If it is not found, the file aborts. You can add a message after the LOOKUP command that will print out on your terminal.

You can also add the symbol "/" after the filespec. This symbol means that the file will not abort if it does not locate the filespec - rather, it will execute the next line of the command file.

If "/" is specified, and AMOS finds the file, AMOS will then execute the line *after* the next line. This allows you to put a GOTO command immediately after the LOOKUP command, so that the file can branch to different sections of your command file depending on whether or not the file looked for was or was not found.

When we say "branch" we mean that you are making a "fork in the road" of your command file. AMOS looks for whatever file you specify. If it finds it, it goes to one branch of the file and does something. If if doesn't find it, it goes to the other branch. This allows your command file to choose between two different courses of action. It is almost like an "IF-THEN" statement in a BASIC program.

GOTO Transfers control of a command file to a different line. The GOTO statement is associated with a label. When the command file executes a GOTO line, it branches to the associated label in the command file and continues executing from there.

The label can be either a legal command file element, or it can be a comment. In either case, the label must be on a line by itself. If the label is a comment, it must be preceded by a semi-colon with no intervening spaces, although the argument in the GOTO statement must not include the semi-colon.

See the sample command file SEE.DO in Section 10.3.3 below for an example of GOTO and other special symbols.

EXIT When this command is encountered in a command file, it causes the command file to be terminated. You use this in command files that have LOOKUP and GOTO statements in them.

Since you want the command file to go to a specific area depending on whether or not a file exists, you have to have a way of stopping the command file after it has finished processing the lines in the "branch". Otherwise, the command file will continue to execute statements, and it will process the commands in the second branch also.

Thus, you use an EXIT command at the end of the statements that make up a branch, and it will end the command file at that point. You may add a message after EXIT on the same command line that will be displayed on your terminal.

BATCH BATCH is a command that loads frequently used commands such as GOTO, LOOKUP, EXIT, TRACE, PAUSE and LOAD into your memory partition. You can use this command either at AMOS command level, or at the start of a command file.

This may be a good idea if you are planning to use many of these commands in your command file, since those commands will execute faster if they are in your memory partition rather than on the disk.

COM COM processes files based upon their extensions. You use COM followed by a filespec. DO NOT specify an extension for the filespec. Your command file will find the specified file, and process it according to its extension.

If it is a .TXT file, it will format it using the TXTFMT command, if it is an AlphaBASIC program (with a .BAS extension), it will compile it, etc. The file must be in your account number, but you may specify a different device.

NOTE: Any commands in the command file after a COM statement will not be executed because COM does not return control to the command file after whatever processing it has done. Therefore, use COM as the last statement in a command file.

IMPORTANT: Do not use "CNT" as a name of a command file, since a PAUSE command will create a file named CNT.CMD, and write over any file of the same name that exists.

See the AMOS System Commands Reference Manual, DWM-00100-49, or the AMOS/L System Commands Reference Manual, DSS-10004-00, for more information on these commands.

10.3 HOW IS A .DO FILE DIFFERENT FROM A COMMAND FILE?

A DO file is a special type of command file that allows you to pass arguments to the file. This type of file has a .DO extension and contains exactly the same type of elements as a regular command file (including the special symbols mentioned above), but also includes some additional symbols that allow you to specify items of text to be substituted into the command file at the time that you invoke the file. (These additional parameter symbols are: $0,$1,$2,...$9.)

Because you can pass text items to a DO file, you can use DO files in many different situations in which a regular command file would be too specific.

10.3.1 How to Make a DO file

Create a DO file by using one of the text editors to build a text file with a .DO extension. The file may contain up to ten different user-defined parameters (indicated by the parameter symbols $0-$9).

When you enter the name of the DO file at AMOS command level, you also include an argument list, the items of which will be substituted for the parameter symbols in the DO file. The items in the argument list are separated by blanks. (To include a blank within an argument, enclose the argument within <> symbols-- e.g., <argument #2>.)

Each parameter symbol becomes associated with one of the argument list items (the first item with parameter $0, the second item with parameter $1, and so on). For example:

```
:T
TXTFMT HEADER,$0
RENAME/D $0.LST=HEADER.LST
PRINT $0
TXTFMT HEADER,$1
RENAME/D $1.LST=HEADER.LST
PRINT $1
DIR/W $2
PRINT
```

If we name this file DOC.DO and execute it by typing:

```
.DOC PSTINV ACTPAY *.TXT RET
```

the $0 is replaced with PSTINV, the $1 is replaced with ACTPAY, and the $2 is replaced with *.TXT. Thus, the DO file above will be executed as if it were:

```
:T
TXTFMT HEADER,PSTINV
RENAME/D PSTINV.LST=HEADER.LST
PRINT PSTINV                    ;print defaults to .LST
TXTFMT HEADER,ACTPAY
RENAME/D ACTPAY.LST=HEADER.LST
PRINT ACTPAY
DIR/W *.TXT
PRINT
```

You can use a parameter symbol to represent an entire filespec, a portion of a filespec, a command or any other piece of text inside a DO command file. For example, say you have a DO file called FORM.DO:

```
:T
;$0
TXTFMT DSK$1:$2.TXT
PRINT DSK$1:$2.LST
DIR DSK$1:$2.*
```

when you type:

```
.FORM <FILE IS PSTINV> 2 PSTINV RET
```

it will be executed as if it were:

```
:T
;FILE IS PSTINV
TXTFMT DSK2:PSTINV.TXT
PRINT DSK2:PSTINV.LST
DIR DSK2:PSTINV.*
```

If you have more items in your argument list than there are parameter symbols in the DO file, the extra items are ignored. If you have fewer items in your argument list than there are parameter symbols, the extra parameters are ignored.

However, if you do not supply a parameter, the line it appears on may still be processed, especially if there is a default for the command that was specified. For example, if a line in your DO file is DIR $0, and you do not specify that parameter, AMOS will assume the default of DIR *.*, and process the line in that manner.

Argument list items are associated with parameter symbols NOT in the order that the parameter symbols appear in the file, but in the order in which the parameter symbols are numbered. (That is, the first item in the argument list is associated with parameter $0, even if parameter $2 appears before $0 in the DO file.)

10.3.2 Special Parameter Symbols

In addition to the usual DO file parameter symbols and the special command file symbols, five special parameter symbols allow you to use DO files in a more flexible way and for a greater range of applications.

$: CURRENT DEVICE SYMBOL - Represents the device that the user of the DO file is currently logged into. For example, if the user of your DO file is logged into an account on DSK0: at the time that he or she uses your DO file, the line in the DO file:

```
LOG $:[1,4]
```

is transformed into:

```
LOG DSK0:[1,4]
```

You can use this symbol in combination with the Current PPN Symbol (below) to keep track of the current account and device of the user of your DO file; the DO file can log the user into another account to perform special functions, and then return him to his own account and device.

$P CURRENT PPN SYMBOL - Represents the account that the user of the DO file is currently logged into. For example, if the user of your DO file is logged into account [230,5], the DO file statement:

```
DIR DSK0:[$P]
```

will be executed as:

```
DIR DSK0:[230,5]
```

$ NULL PARAMETER SYMBOL - A single $ indicates a null parameter in a default parameter list, or a null argument in an argument list. This symbol allows you to designate which parameter will be associated with which argument. Take a look at the example below for an idea of how to use the $ symbol.

$$ DOLLAR SIGN SYMBOL - Represents an actual dollar sign. If you need to use a dollar sign as part of a message within a DO file, you enter it as two dollars signs to distinguish it from the null parameter symbol.

$D DEFAULT PARAMETER LIST - If you specify fewer items in the argument list than there are parameter symbols in the DO file, AMOS usually ignores the extra parameter symbols.

You can, however, supply a default argument list that AMOS will use if you omit an argument list when you invoke the DO file at AMOS command level, or if you do not supply a complete argument list.

For example, say that the AlphaVUE command's default extension is .TXT.

You can create your own VUE command (in this case named LOOK) in which the default extension is .CMD.

```
$D $ .CMD
:T
VUE $0$1
```

In the example above, the $ represents the fact that the parameter $0 is a null, that is, it does not have a default setting. The .CMD, then, is the default setting for $1.

So, if you invoke the DO file with an argument list containing just a filename (which would be $0), without specifying an extension ($1), you will see something like this:

```
.LOOK PSTINV RET

VUE PSTINV.CMD
```

This means that if you use your LOOK command without specifying an extension, it will default to .CMD. Of course, with this example, if you wanted to specify another extension, you would have to use this form:

```
.LOOK PSTINV .TXT RET
```

NOTE: If a $D line appears in your DO file, it MUST be the first line of the file (even before a :T or :R symbol).

10.3.3 SAMPLE DO FILES

Below are some examples of the kinds of DO files you can create to help you perform frequently used sequences of commands.

If you want all of the users on the system to be able to share your command and DO files, have the System Operator copy them into the System Command File Library, DSK0:[2,2]. If a command or DO file is in this account, it can be called by any user on the system without any disk or account specifications.

TFORM.DO

You can use the text formatting program, TXTFMT, to format a group of text files so that they form one document. If you always begin the filespec list with the same file or files, a DO file can be convenient:

```
:T
TXTFMT HEADER,PRPNOT,PRFACE,$0
```

For example, the DO file above tells TXTFMT to format a header file (HEADER) that contains standard formatting information (page size, line size, etc.), a file that contains an official proprietary notice (PRPNOT), and a file containing a standard preface (PRFACE).

Let us say that these three files are often formatted in front of a document. You can call TFORM.DO with a filespec that specifies the main body of the document:

```
.TFORM YEARLY (RET)
TXTFMT HEADER,PRPNOT,PRFACE,YEARLY
```

SEE.DO

This DO file allows you to enter AlphaVUE by specifying only the filename of your file. Instead of typing VUE TEXT.CMD, you will be able to type SEE TEXT. If the file TEXT has any common extension, this DO file will find it.

Remove ths Line for it to work →

```
:R
BATCH                      ; Load commands into memory
LOOKUP $0.CMD/ <CMD>
;                            Is there a CMD file with that filename?
GOTO BRANCH                ; If not, go to BRANCH
VUE $.CMD                  ; If there is, VUE it
EXIT                       ; End DO file once done with VUE
;BRANCH
LOOKUP $0.DO/ <DO>
GOTO BRANCH1
VUE $0.DO
EXIT
;BRANCH1
LOOKUP $0.CTL/ <CTL>
GOTO BRANCH2
VUE $0.CTL
EXIT
;BRANCH2
LOOKUP $0.TXT/ <TXT>
GOTO BRANCH3
VUE $0
EXIT
;BRANCH3
LOOKUP $0.LST/ <LST>
GOTO BRANCH4
VUE $0.LST
EXIT
;BRANCH4
LOOKUP $0.BAS/ <BAS>
GOTO BRANCH5
VUE $0.BAS
```

```
EXIT
;BRANCH5
LOOKUP $0.DAT/ <DAT>
GOTO BRANCH6
VUE $0.DAT
EXIT
;BRANCH6             ; If the file was not found, print:
;
:< The file does not exist with any usual extensions.
>
```

PRINTE.DO

When you are printing a long list of files (e.g., AlphaBASIC programs), it is often convenient to separate those files by sending a form-feed character to the printer after each file that you print; then each file begins at the top of a page.

When you call PRINTE.DO, give it the Filespec of the file you want to print. PRINTE prints the file, and then sends a form-feed character (FF.TXT) to the printer.

```
PRINT $0
; FF.TXT is in System Command File Library
PRINT DSK0:FF.TXT[2,2]
```

FF.TXT is a text file that contains only one symbol-- a form-feed character. To create FF.TXT, follow these steps:

1. Log into DSK0:[2,2]:

   ```
   .LOG 2,2 RET
   ```

2. Enter AlphaVUE to create FF.TXT:

   ```
   .VUE FF RET
   ```

   Answer with a "Y" when it asks you if you wish to have FF.TXT created.

3. In the blank file, type a Control-G, then a Control-L. This enters a form feed character into the file. You will see (^L). Use an escape and an "F" followed by a RETURN to exit from AlphaVUE.

TALK.DO

This DO file allows you to send every user on your system a message at the same time:

```
:T
SEND MIKE $0
SEND DAVID $0
SEND JOB1 $0
SEND JOB2 $0
SEND BILL $0
```

If you wish to make the message longer than one word, enclose the message in angle brackets <>:

```
.TALK <BE SURE TO TURN IN YOUR STATUS REPORTS BEFORE LUNCH> RET
```

Since spaces separate DO parameters, it will accept only the first word up to a space as the $0 parameter unless you use the brackets.

BACKUP.DO

This DO file transfers backup copies of the specified files in account [200,1] from one disk cartridge (or floppy disk) to the same account on another disk cartridge (or floppy disk):

```
:T
SET DSKERR      ; Report disk errors
:<Backup Account [200,1] of Working Cartridge
onto Archives Cartridge
>
LOG DSK1:[200,1]
COPY DSK0:=$0 ; Copy from DSK1:[200,1] to DSK0:[200,1]
:<Change the cartridge to Archives Cartridge;
when ready, press RETURN -
>
:K
MOUNT DSK1:
COPY =DSK0:$0 ; Copy from DSK0:[200,1] to DSK1:[200,1]
:<Change the cartridge back to Working Cartridge; when
ready, press RETURN -
>
:K
MOUNT DSK1:
:<Clear DSK0:
>
ERASE DSK0:*.*[200,1]
LOG $:[$P]      ; Return to previous account.
```

10.4 THE TASK MANAGER

Alpha Micro computers also support a system called the Task Manager, which allows you to perform tasks without tying up your terminal.

The Task Manager is a batch processing program that allows you to use real or pseudo terminals to run tasks without human control.

If some tasks that you do take a long time to run (an hour, for instance), you can have the Task Manager perform them and still be able to use your terminal for other things.

You can also schedule tasks to run at specific date and times, and you can schedule tasks to remain permanently in the queue, running at specified intervals (such as weekly, monthly, etc.)

See the AMOS/L Task Manager User's Manual, DSS-10010-00, for more complete information about the use of the Task Manager.

Check with your System Operator to see if the Task Manager is installed on your system.

10.4.1 Control Files

The Task Manager uses control files to give it directions. These control files are similar to command files, but have a different set of special symbols, and are more versatile. Almost anything you do on your terminal can be done in a control file.

Commands are entered into the control file just as they would be on the terminal.

10.4.2 Submitting Tasks

The SUBMIT command is used to submit control files to the Task Manager. The format is:

```
.SUBMIT {Control filespec} {/switch1.../switchN} RET
```

If you do not specify a control filespec, the program will display the contents of the queue file. See the SUBMIT command reference sheet in the AMOS/L System Commands Reference Manual, DSS-10004-00, for the switches and their use.

Once the control file has been submitted, the Task Manager executes the commands in the control file to complete the task. The Task Manager allows you to control exactly when the task is performed, and gives you the capability of having tasks repeat automatically at regular intervals.

CHAPTER 11

USING ALPHA MICRO PROGRAMMING LANGUAGES

In this chapter, we will discuss the use of the various Alpha Micro Programming languages. Of course, we will leave detailed instructions to the specific manuals for each language, but we will give you an idea of what each language can do, what applications each language would be best for, and how to access each language.

These are just the standard languages that are generally supported. Your Alpha Micro dealer may have other languages available. See him or her for information about optional languages and their capabilities.

The topics discussed will be:

- AlphaBASIC
- AlphaPASCAL
- AlphaLISP
- Alpha Micro Assembly Language

11.1 WHAT IS ALPHABASIC?

The BASIC language is probably the most commonly used computer language, especially for micro-computers.

The acronym BASIC stands for Beginners' All-purpose Symbolic Instruction Code. BASIC is a higher-level programming language that was designed to be versatile tool for learning computer programming, and also to provide a relatively simple language for a wide variety of applications.

BASIC is an easy to use programming language primarily because of its similarity to the English language. Don't let the word "Beginners" in its name fool you. BASIC is a good language for newcomers, but it is also a powerful language used by all types of programmers.

Over the years since its inception, BASIC has been added to and modified as new concepts of programming have emerged. Some versions of BASIC are more extensive than others; the use of these extended versions allows the programmer a wider range of applications, greater ease in programming, and/or greater efficiency and speed.

AlphaBASIC is just such an extension of the BASIC language, with several features not found in other implementations. For example, AlphaBASIC provides a method for defining sophisticated data record structures that adds flexibility and efficiency to business-oriented programs.

These features not only enhance the performance of traditional uses of the language but also make business applications easier to program.

11.1.1 How to use AlphaBASIC

You can use AlphaBASIC in one of two modes: Interactive or Compiler mode.

INTERACTIVE MODE

In Interactive mode you create, edit and test your program while it resides in temporary memory. This mode is convenient for the creation and debugging of new programs or for learning AlphaBASIC.

Type BASIC and a RETURN, and you see:

```
.BASIC RET
READY
```

There is no prompt symbol, just type in either single lines, or your program. To exit, type:

```
.BYE RET
```

Here is a small example of how an AlphaBASIC program might look in interactive mode:

```
.BASIC RET
AlphaBASIC Version xx.(xxx)

READY
10 ! Program to print name in reverse  RET
20          STRSIZ 20  RET
30 START:   INPUT LINE "Enter your name: ",NAME$  RET
40          IF LEN(NAME$) = 0 THEN GOTO START  RET
50 LOOP:    COUNTER = LEN(NAME$)  RET
60          FOR I = 1 TO COUNTER  RET
70             PRINT NAME$[COUNTER;1];  RET
80             COUNTER = COUNTER - 1  RET
90          NEXT  RET
100         PRINT  RET
110         INPUT "Another one? (Y or N): ",QUERY$  RET
120         IF UCS(QUERY$) = "Y" GOTO START ELSE PRINT "All done."  RET
130         END  RET
RUN  RET
COMPILING
Compile time was 0.008 seconds, elapsed time was 0 seconds
Enter your name: ALPHA MICRO  RET
ORCIM AHPLA
Another one? (Y or N): N  RET
All done.

CPU time was 0.043 seconds, elapsed time was 1 seconds

READY
BYE  RET

.
```

COMPILER MODE

Compiler mode is more useful for programs which are to be put into production use, or for testing programs which are too large to fit in memory in the interactive mode.

In compiler mode, you compile the program at AMOS command level and store the compiled program on the disk.

During the actual running of the compiled program, only the compiled program and a minimal run-time execution package need to be in memory, thereby conserving memory space.

To use compiler mode, use one of the text editing programs (AlphaVUE is the best for programming) to create your program. When you are back at AMOS Command level, enter:

```
.COMPIL MYPROG RET
```

Once your program is successfully compiled, you can execute it by typing:

```
.RUN MYPROG RET
```

COMPIL assumes an extention of .BAS, and RUN assumes an extention of .RUN.

11.2 WHAT IS ALPHAPASCAL?

Pascal is a newer programming language designed to meet new programming needs - to be clearer and more powerful than most older languages. It encourages programmers to use good programming style.

Style is becoming more and more important as programs and systems become larger and more complex. To make maintenance and updating of programs cost efficient, programs today need to be not only functional, but also well-designed and readable.

AlphaPASCAL is an advanced version of Pascal that has been developed specifically to run under the Alpha Micro Operating System.

11.2.1 How to use AlphaPASCAL

AlphaPASCAL programs are written in compiler mode. This means that you use one of the text-editing programs (AlphaVUE is the easiest for writing programs) to create your program file. When you are ready, you compile the file (using the AlphaPASCAL compiler command, PC instead of COMPIL) and run the compiled version. (See the explanation of compiler mode in section 11.1.1 above).

The AlphaPASCAL User's Guide, DWM-00100-08, contains detailed instructions on how to prepare an AlphaPASCAL file and how to compile it. It also contains complete instructions for programming in AlphaPASCAL.

11.3 WHAT IS ALPHALISP?

The LISP language was developed to handle symbolic expressions on a computer. It is designed to be used in programs that process large numbers of string variables, or that do large amounts of information retrieval.

One of the possible uses of AlphaLISP might be in mailing list programs, where names, addresses, and other data are associated.

11.3.1 How to use AlphaLISP

AlphaLISP is used in Interactive mode. (See the definition of interactive mode in section 11.1.1 above.) To enter the LISP program, type:

```
.LISP RET
```

The LISP Monitor will then ask you:

```
Load Extended Library?
```

Type a Y or an N (Yes or No) to answer this question. The extended library gives LISP more features, but also reduces the memory you will have in which to create your program.

You will then see an asterisk (*). This is the LISP prompt symbol, and indicates that LISP is waiting for an input. You may then enter either single statements, or the lines of your program.

To get out of LISP and back to AMOS, enter:

```
*(EXIT) RET
```

See the AlphaLISP User's Manual, DWM-00100-05, for further information on the LISP language.

11.4 ALPHA MICRO ASSEMBLY LANGUAGE

All of the programming languages discussed above are called "higher-level" languages. That means that they are one step removed from the basic machine language the computer understands. This makes them easier to use and more efficient for normal programming needs than the more hardware-oriented machine language.

However, it means that they are not appropriate for programming certain applications that require high speed execution and the ability to communicate with hardware devices.

The programming language that you use to create machine language programs is called "assembly language." Alpha Micro provides two assembly languages, one for each of its two families of computers.

For information on the AMOS/L assembly language, refer to the AM-100/L Instruction Set, DSS-10006-00 and the AMOS/L Assembly Language Programmer's Manual, DSS-10007-00.

See the AMOS Assembly Language Programmer's Manual, DWM-00100-43, and the WD16 Microcomputer Reference Manual, DWM-00100-04, for information on the AMOS assembly language.

CHAPTER 12

WHERE TO GO FROM HERE

In the previous chapters you've learned many of the most useful commands and features of the Alpha Micro Operating System, but of course there is still much more. By now, you've mastered the basic skills you need to operate your system and to create and edit your files and programs.

You know everything that all Alpha Micro users need to know, and now you can move on to the more specialized topics that apply to your particular equipment configuration and your specific business requirements.

Whether it is a programming language, a text-editing program, or a hardware technical manual, we have documentation available. This chapter will give you some guidelines for finding the right document for your needs. The points we will discuss are:

- Other useful commands
- References to other manuals
- Guidelines for the System Operator

12.1 OTHER USEFUL COMMANDS

There is a such wide variety of commands already available on your Alpha Micro computer system that you will find yourself using some commands all the time, and others not at all. To familiarize yourself with all the available commands, skim through the AMOS System Commands Reference Manual, DWM-00100-49, or the AMOS/L System Commands Reference Manual, DSS-10004-00.

The commands listed below are a few that we feel might be particularly useful.

SRCCOM - Compares the text of two files and shows you where they are similar and different. This is useful if you are not sure which of two files is the latest version.

CREATE - This command creates an empty file, which you can then access later and fill.

SET - This command has many options. It allows you to set certain characteristics of your terminal and other devices.

DSKANA - This command scans the specified disk for errors. It is a privileged program that can only be used from account DSK0:[1,2], and is usually run by the System Operator. See the Cautions in the DSKANA reference sheet before attempting to use this command.

DATE - Displays the date on your terminal. For Example:

```
.DATE RET
Friday, August 2, 1982
```

TIME - Displays the time on your terminal. For Example:

```
.TIME RET
3:00:14 PM
```

TRMDEF - Displays a list of the terminals that are defined on your system.

DEVTBL - Displays a list of all devices defined on your system.

PPN Devn: - Displays the current accounts for a specified disk or device.

12.2 OTHER REFERENCES

The following is a general list of our manuals. Since we are continually producing new products and manuals, by the time you read this many other manuals may also be available. Your local dealer or the Alpha Micro Sales Administration Department can provide you with a complete list of all documents available.

Refer to the appropriate manual for more specific information about the topics that interest you:

Introductory Texts

Introduction to AMOS	DWM-00100-65
AMOS User's Guide	DWM-00100-35
AlphaMAIL User's Manual	DSS-10000-06

Word Processing Manuals

AlphaVUE/TXTFMT Training Guide	DSS-10000-03
AlphaVUE User's Manual	DWM-00100-15
TXTFMT User's Manual	DWM-00100-04

Programming Manuals

AlphaBASIC User's Manual	DWM-00100-01
AlphaPASCAL User's Manual	DWM-00100-08
AlphaLISP User's Manual	DWM-00100-05
AlphaFIX User's Manual	DWM-00100-69
AMOS Assembly Language Programmer's Reference Manual	DWM-00100-43
AMOS/L Assembly Language Programmer's Manual	DSS-10007-00
AlphaFIX/L User's Manual	DSS-10005-00
ISAM System User's Guide	DWM-00100-06
AMOS Monitor Calls Manual	DWM-00100-42
AMOS/L Monitor Calls Manual	DSS-10003-00
WD16 Microcomputer Reference Manual	DWM-00100-04
AM-100/L Instruction Set	DSS-10006-00

Other System Manuals

Alpha Micro Integrated Systems User's Guide	DWM-00101-00
AMOS System Commands Reference Manual	DWM-00100-49
AMOS/L System Commands Reference Manual	DSS-10004-00
AMOS System Operator's Guide	DSS-10001-00
AMOS/L System Operator's Guide	DSS-10002-00
AMOS Video Cassette Recorder Backup Software	DSS-10027-00
AMOS/L Video Cassette Recorder Backup Software	DSS-10032-00
AMOS 1/4" Streamer Backup Software	DSS-10033-00
AMOS/L 1/4" Streamer Backup Software	DSS-10031-00
AMOS/L Task Manager User's Manual	DSS-10010-00
AMOS LOKSER User's Manual	DSS-10022-00

12.3 IF YOU ARE THE SYSTEM OPERATOR

On Alpha Micro multi-user systems, one user is usually designated as the System Operator. If you are the System Operator, you have a few additional responsibilities, and you must have knowledge of some things that other users don't have to worry about.

The System Operator:

1. Maintains the system initialization command file.
2. Assigns accounts and passwords to the other users.
3. Sets up the hardware (like printers, tapes, etc.)
4. Manages system memory.
5. Controls the special system libraries.
6. Periodically tests memory and disks for errors.

You can find special instructions for these tasks in the AMOS System Operator's Guide, DSS-10001-00, or the AMOS/L System Operator's Guide, DSS-100002-00.

APPENDIX A

AMOS ERROR MESSAGES

Below is a list of error messages that you might see when operating at AMOS command level. For information on the specific error messages you might see when using a particular command, see the documentation for that command.

The format of the error message may differ slightly between the AMOS and the AMOS/L systems, but the errors and their solutions are the same.

A.1 COMMON ERROR MESSAGES

The most common kind of error message that you will see on the system consists of two parts. The message begins with:

?Cannot {init, open, close, read, write, input, output, wait, delete, rename, assign, deassign, allocate record on, deallocate record on, read bitmap on, write bitmap on, lock directory on, unlock directory on, allocate random file, mount, unmount, load, or access} Filespec or Devn: -

and ends with one of the messages below (for example: Cannot read AMS3: - disk not mounted):

- BADBLK.SYS not found

The program BADBLK.SYS was not found on the disk, meaning that it was erased or somehow lost. See your System Operator for help.

- BADBLK.SYS has a bad hash total

The BADBLK.SYS program is damaged. See your System Operator for help.

- BADBLK.SYS is in wrong (unsupported) format

You will only see this error if you have an AMOS/L system and your version of BADBLK.SYS was certified on an AMOS system that was version 4.2 or earlier. The AMOS/L system will not support it. To correct this problem, your System Operator will have to transfer your files to an AMOS system that has version 4.3 or later, and then re-certify.

- bitmap kaput

Every time the system writes data to a file-structured device (e.g., DSK0:, AMS1:), it checks that device's bitmap to determine where on the disk to write the data. (A bitmap is a map of the device -- it tells the system what records on the device already contain data, and which are free for use.)

Every time the system checks the bitmap, it sums the data in that bitmap and checks it against the value it has previously computed; if there is a discrepancy, it means that something in the bitmap has changed, and you see the bitmap kaput error message.

To solve this problem, MOUNT the disk. If you still get this message, check with the System Operator for help -- he or she can run the DSKANA program that will reset the bitmap.

- buffer not INITed

Before your assembly language program can perform an open, read or write operation, the program must assign a buffer to the file DDB. (Because you can use the monitor call INIT to do this, we say that the buffer has been INITed.) If you fail to do this procedure, you will see this message.

If you should see this message within AlphaBASIC, the message may indicate memory problems; check with the System Operator.

- deadly embrace possible

You are trying to OPEN a file out of sequence from the sequence defined in the LOKGEN program. This occurs if you have the LOKSER file-locking system installed on your system. Check the LOKGEN program for the proper file-opening sequence.

- device does not exist

You've tried to access a device that the system does not recognize. For example:

```
.TYPE ASM1:FILE.TXT RET
?Cannot init ASM1:FILE.TXT - device does not exist
```

You probably mis-typed the device name. You can use the DEVTBL command to see a list of all the devices on the system.

- device error

A hard disk error has occurred; that is, the system was not able to read data from a disk. Try to perform the operation again. If you still have no success, check with the System Operator for help; this message may indicate a hardware problem.

- device full

There is not enough room on the disk to complete the data transfer. Start over again with another device, or make room on the first device by erasing unnecessary files.

- device in use

Another user is using the non-sharable device that you wish to access (e.g., a paper tape punch). Wait and try again later.

- device not mounted

You have tried to access a valid system device, but that device is not mounted. Use the MOUNT command to mount the device, and try again.

- device not ready

You are trying to access a device that is not ready. For example, you will see this message if you try to mount a disk before that device is powered up and ready.

- disk not mounted

The disk drive that the command tried to access is not mounted. Use the MOUNT command to mount that disk, then try again.

- file already exists

You've tried to create a file that already exists. For example, if the file NEWCPY.MAC already exists in your account, and you try to rename another file to that name:

```
.RENAME NEWCPY=WRKFIL RET
WRKFIL.MAC to NEWCPY.MAC
?Cannot rename WRKFIL.MAC - file already exists
```

- file already open

Your assembly language program tried to open a file that is already open. Check your program to see if you are opening the file twice.

- file cannot be DELETEd

You tried to delete a file that was specified as undeletable in LOKGEN. This only occurs if you have the LOKSER file locking system on your system.

- file cannot be RENAMEd

You tried to rename a file that was specified as unrenamable in LOKGEN. This only occurs if you have the LOKSER file locking system on your system.

- file in use

You tried to OPEN a file which is being used by another user. Either you or the other user requested an exclusive lock on the file. This only occurs when you have the LOKSER file-locking system on your system.

- file not found

AMOS cannot find the file you've specified. For example:

```
.TYPE LABDAT RET
?Cannot open DSK0:LABDAT.LST - file not found
```

Check your spelling, and make sure that you've specified the correct device and account.

- file not open

Your assembly language program has tried to access a file that is not open for input. Check your program to see if you are accessing the correct file.

- file specification error

You did not properly specify the command. Check your format and spelling, and try again.

- file type mismatch

You've tried to use a program designed for sequential files on a random file (or vice versa). For example, the TYPE program works only on sequential files; if you try to use it on a random file, you see:

```
.TYPE PRTIDX.DAT RET
?Cannot open DSK0:PRTIDX.DAT - file type mismatch
```

- first logical unit is not mounted

The first logical unit accessed by the command is not mounted. Use the MOUNT command to mount the unit once you have determined which device or unit is not mounted.

- illegal block number

Your program has tried to access a disk block that doesn't exist, or that is beyond the range of the file you are using.

- illegal user code

You've tried to access an account that does not exist. For example, if there is no account [100,2] on DSK1:

```
.TYPE DSK1:MYFILE.TXT[100,2] RET
?Cannot open DSK1:MYFILE.TXT[100,2] - illegal user code
```

- insufficient free memory

There is not enough room in memory to complete the operation you are attempting. Make sure that no unnecessary modules are in your memory partition.

- insufficient queue blocks

There were not enough queue blocks to perform the requested operation. Try the command again, and if that does not work, have the System Operator increase the amount of queue block space in the system initialization command file.

- invalid filename

You've specified a filename that AMOS does not understand. For example:

```
.MAKE  .TXT RET
?Cannot open  - invalid filename
```

In this case, the filename is a space, which AMOS does not recognize as a filename.

- LOKSER queue is full

All of the available queue space in LOKSER is used up. Have your System Operator re-run LOKGEN and allocate more memory blocks.

- MFD is damaged

Your Master File Directory is damaged. Copy as many programs as you can onto a backup media, and then run the SYSACT program with the I (initialize) option.

- PPN does not exist

You've tried to access a user account that does not exist. For example, if there is no account [100,2] on DSK1:

```
.TYPE DSK1:MYFILE.TXT[100,2] RET
?Cannot open DSK1:MYFILE.TXT[100,2] - PPN does not exist
```

- protection violation

You've tried to create a file in an account other than your own that is outside of your project. For example:

```
.COPY [210,3]=ELIPSE.BAS[100,5] RET
ELIPSE.BAS[100,5] to ELIPSE.BAS[210,3]
?Cannot open ELIPSE.BAS[100,5] - protection violation
```

- record in use

You tried to access a record inside a file that is already locked by another user. Either one or both users have requested an exclusive lock on that record. This occurs only if you have the LOKSER file-locking system installed on your computer.

- record not locked for output

You are attempting to output a record that was not previously read and locked via an INPUTL statement. This only occurs if you have the LOKSER file-locking programs on your system.

- write protected

You are trying to write data to a device that is write-protected. Make sure the write-protect button is off and try again. Make sure also that you were writing to the correct disk.

A.2 OTHER ERROR MESSAGES

In addition to the error messages above, you can also see:

?Address error at nnnnnnnn

An address error occurred while your job was running. This error appears whenever a program tries to access a word or longword on an odd address boundary.

AMXXX ERROR n FOR DRIVE n BLOCK n (CYLINDER n HEAD n SECTOR n)

You will see a message something like this if you have SET DSKERR, and a hard error occurs on a system that uses an AM-500, AM-420, or AM-410 Disk Controller Hard Disk Subsystem (the XXX stands for whatever controller you have). This message indicates a hardware problem -- the Disk Controller wasn't able to successfully read data from the disk.

The message tells you what kind of error occurred (refer to the information supplied with the disk drive to find out what error conditions corresponds to that error code), and where on the device the data transfer operation failed (the drive, record, cylinder, head and sector). Check with the System Operator for help.

Bus error - PC ##

A bus error indicates that an illegal condition was recognized on the data bus. The number following the letters "PC" tell you the memory address the Program Counter was set to when the bus error occurred.

?Bus error at nnnnnnnn

A bus error occurred at location nnnnnnnn. This is a special type of error generated by the central processor hardware.

?CHK instruction at nnnnnnnn

Your job executed a CHK instruction wich caused the trap to be taken. (This instruction is not currently supported on AMOS/L systems.)

?Command terminated - insufficient memory

You tried to execute a program from inside a command file, but there was not enough room to load the program into your memory partition.

Use the MAP command to see what modules are in your memory partition; delete those you do not need. If you still cannot use the command file, talk to your System Operator about getting more memory area allocated to your job.

?Divide by zero at nnnnnnnn

Your job attempted to perform a divide (DIV or DIVS) instruction with a divisor of zero.

?EM1111 at nnnnnnnn

Your job executed on of the EM1111 instructions which are not currently used by AMOS/L; they are reserved for future use. This error is commonly caused by a program which begins executing in non-existent memory.

ERROR n

You see this message if you SET DSKERR and a soft disk-error occurs on a system using an AM-200 or AM-210 Floppy Disk Subsystem. Frequent soft errors can indicate hardware problems with the drives; check with the System Operator.

The number following ERROR indicates the type of soft error that occurred; refer to the instruction supplied with the disk drive to see what error condition corresponds to that error code.

Because this message reports a soft error (the system had to retry a data read) and not a hard error (where the data could not be read at all), you do not necessarily have to worry unless you see a DEVICE ERROR message (see above), which indicates a hard error.

ERROR n CMD n, STS n, RECORD n (TRACK n, SECTOR n)

You see this message if you have SET DSKERR and a soft disk-error occurs on a system that is using an AM-400 Hard Disk Subsystem. The number following ERROR indicates the type of soft error that occurred; refer to the instructions supplied with the disk drive to find out what error condition corresponds to that error code. The message tells you where on the disk the error occurred. Check with the System Operator.

?Illegal instruction at nnnnnnnn

Your job attempted to execute an illegal instruction. Any bit pattern not currently recognized by the CPU is considered an illegal instruction.

?Illegal user interrupt on level n

An interrupt occurred on level "n" when none was expected. This can be caused by a program that does not intitialize the interrupt vector locations properly, or by faulty hardware that generates spurious interrupts.

?Insufficient memory for program load

You do not have enough memory in your partition to load the program you want to execute. Use the MAP command to make sure that no unnecessary modules are in your memory partition. If you still receive this message, check with the System Operator to see if he or she can allocate more memory to your job.

?Login please

You've tried to enter an instruction to AMOS, but you are not logged into the system. Use the LOG command (see Section 2.4,"Logging on and off the System"). If you need help in figuring out what to do, you can use the HELP command even if you are not logged in (see Section 3.1, "The HELP Command").

?Memory allocation failed

You used the monitor call GETMEM from within an assembly language program to allocate space for a memory module within a memory partition -- there wasn't enough room in the partition to perform the allocation. Make sure that no unnecessary modules are in the memory partition, or see the System Operator about increasing your memory.

?Memory map destroyed

Each module in your memory partition maintains a pointer to the address of the next module in memory; if these connecting links become confused or broken, AMOS is not sure where your memory modules are in your partition.

You may not need to reset the computer, but you may want to delete the modules from memory and reload them just to be sure that your memory map is intact.

?Memory parity error

A memory parity error occurred while your job was running. After this error occurs on an AM-100/L system, you should examine the memory boards to determine which one the error occurred on. On both AM-100/L and AM-1000 systems, the System Operator may want to perform memory diagnostics to isolate the cause of the parity error.

?No memory available

There is no more free memory available on the system. Consult with the System Operator. The solution may be to change the amounts of memory allocated to each job on the system.

?Privilege violation at nnnnnnnn

Your job tried to execute a privileged instruction while in user mode. Certain privileged instructions, such as STOP, require that you be in supervisor mode.

?Privileged program - must be logged into OPR:

The program that you called can only be run from account DSK0:[1,2]. This is the System Operator's account, and you should have the System Operator run the command or program for you.

?Privileged program - must be logged into [1,2]

The program can only be run from account [1,2] (on any disk). Log into [1,2] and run it again.

?Trace return at nnnnnnnn

Your job returned from an instruction trace, but your job's trace trap address (JOBTRC) was not set up properly.

?TRAPV instruction at nnnnnnnn

Your job executed a TRAPV instruction which caused the trap to be taken. This instruction is not currently supported by AMOS/L.

?   ^ specification error

The format of your command line was confusing. For example:

```
.VUE RND.COMD RET
?           ^ specification error
```

In the case above, you entered too many characters for the file extension. Retype the command line, making sure that the syntax you use is the correct form for that particular command.

APPENDIX B

RESERVED FILE EXTENSIONS

You can use file extensions to categorize your files and to indicate the types of information they contain. The ones listed below have become standardized to represent specific types of files that you use frequently.

What we mean when we say that a file extension is "reserved" is that commands on the system will often work in different ways depending upon the file extension. Also, many of the commands and systems (such as AlphaVUE) have defaults built in or set in the system initialization command file.

An example is when you use the "G" command to exit from an AlphaVUE file. The "G" command is normally used on files with .TXT extensions to format them with TXTFMT and store the resulting formatted version on your disk in a file with a .LST extension. However, if you used a .CMD extension instead of .TXT on your text file, G would try to execute the file as a command, and you would get an error message.

Also, certain programs and commands create intermediate files with extensions such as .TMP or .IPF that are automatically deleted. If any of your own files has one of these extensions, it is in danger of being written over and subsequently erased from your disk.

If you use AlphaPASCAL or Alpha Micro Assembly Language, you'll find additional reserved file extensions that those lauguages recognize. These additional extensions are explained in the appropriate programmer's manual.

.TXT	A text file. A memo, a letter, or a report created using AlphaVUE.
.BAK	A backup file created automatically when you use AlphaVUE. It ensures that you have a copy of each file in case something goes wrong while you're working on the current version.

.LST A list file. This extension is created automatically when .TXT files are processed by TXTFMT, when you use DIR to create a directory listing file, and when you use MACRO or M68 to create a listing of an assembly language program.

.HLP A help file. A text file that contains information on a specific system related topic. (Type HELP and a RETURN to see a list of these topics.)

.CMD A command file. A type of text file that contains a series of system commands. You can make AMOS execute the series of commands by entering the name of the command file without the extension.

.DO A do file. A type of command file that allows you to send variable data to the command file for it to process.

.BAS An AlphaBASIC source program created from within the AlphaBASIC program or with AlphaVUE.

.RUN A compiled AlphaBASIC program.

.SBR An AlphaBASIC subroutine, usually written in assembly language, that performs some frequently performed function within an AlphaBASIC program.

.DAT A data file. This type of text file is created by an AlphaBASIC program for other programs to use.

.M68 or
.MAC An assembly language source program. (AMOS/L systems use .M68 while other AMOS systems use .MAC.)

.OBJ An object module. An assembled assembly language program.

.TMP A temporary work file created by the LIB program. LIB generates library modules from .OBJ modules, and uses a .TMP file for intermediate storage. You will only find a file with this extension if something goes wrong while LIB is processing your .OBJ module. The next time you run LIB it will disappear.

.IPF An intermediate program file. Files with .IPF extensions are used in several Alpha Micro software packages to pass data from one program to another. When processing is completed, these files are automatically erased from your disk.

.LIT or
.PRG An executable object module that has been processed by the linkage editor. (AMOS/L systems use .LIT while other AMOS systems use .PRG.)

.SYM A symbol file. Contains a symbol table for a .PRG or .LIT file. Created by using the SYMBOL or SYMLIT program on a .OBJ file.

.LSP A program written in AlphaLISP.

.PAS A program written in AlphaPASCAL.

.PSB An AlphaPASCAL assembly language subroutine.

APPENDIX C

THE CONTROL CHARACTERS

Char.	Monitor function	Function in VUE	ASCII Definition
@	Null	Null	Null
A		Last Word	Start of Heading
B		EOL* to Next Line	Start of Text
C	End run		End of Text
D		Delete Character	End of Transmission
E		End of File	Enquiry
F		Insert Character	Acknowledge
G		Next Character=Text	Bell Code
H		Cursor Left	Back Space
I		Horizontal Tab	Horizontal Tab
J	Line Feed	Cursor Down	Line Feed
K		Cursor Up	Vertical Tab
L		Cursor Right	Form Feed
M	Carriage Return	Carriage Return	Carriage Return
N		Cursor to EOL*	Shift Out
O		Concatenate	Shift In
P		Mark Block of Text	Data Link Escape
Q	Resume Display	Insert Mode	Device Control 1
R		Last Page	Device Control 2
S	Suspend Display	Center Line	Device Control 3
T		Next Page	Device Control 4
U	Erase Command Line	Cursor to Start of Line	Negative Acknowledgement
V		Delete Word	Synchronous Idle
W		Cursor to Next Word	End of Transmission blocks
X		Cursor to Next Marker	Cancel
Y		Erase to EOL*	End of Medium
Z		Delete Line	Special Sequence
[		Escape	Escape
/		Line Insert/Entry Mode	File Separator
^		Cursor Home	Group Separator

*NOTE: EOL stands for "End Of Line" -- the line the cursor is currently on.

AMOS USER'S GUIDE
DOCUMENT HISTORY

Revision A00 - AMOS Release 4.1 - (Printed 4/79) - New Document

Revision A01 - AMOS Release 4.3 - (Printed 10/79) - This edition includes an enhanced version of the PRINT command documentation, changes to the LOAD command, and changes to the SET command to support magnetic tape. Also, we have incorporated information in support of the Phoenix disk drives, new system error messages, and miscellaneous corrections.

Revision B00 - AMOS Release 5.0 and AMOS/L Release 1.1 - (Printed 1/83) - For this release, the manual was completely rewritten to incorporate the latest features of both AMOS and AMOS/L systems.

Index

TECHNICAL PUBLICATIONS FILE REFERENCE AMOS User's Guide, DWM-00100-35, Rev

TECHNICAL PUBLICATIONS READERS COMMENTS

We appreciate your help in evaluating our documentation efforts. Please feel free to attach additional comments.
If you require a written response, check here: □

NOTE: This form is for comments on documentation only. To submit reports on software problems, use Software Performance Reports (SPRs), available from Alpha Micro.

Please comment on the usefulness, organization, and clarity of this manual:

Did you find errors in this manual? If so, please specify the error and the number of the page on which it occurred.

What kinds of manuals would you like to see in the future?

Please indicate the type of reader that you represent (check all that apply):

- □ Alpha Micro Dealer or OEM
- □ Non-programmer, using Alpha Micro computer for:
  - □ Business applications
  - □ Education applications
  - □ Scientific applications
  - □ Other (please specify):
- □ Programmer:
  - □ Assembly language
  - □ Higher-level language
  - □ Experienced programmer
  - □ Little programming experience
  - □ Student
  - □ Other (please specify):

NAME: ____________________ DATE: __________

TITLE: ____________________ PHONE NUMBER: __________

ORGANIZATION: ____________________

ADDRESS: ____________________

CITY: __________ STATE: __________ ZIP OR COUNTRY: ____

STAPLE

FOLD

PLACE
STAMP
HERE

CUT ALONG LINE

alpha micro

3501 Sunflower
P.O. Box 25059
Santa Ana, CA 92799

TECHNICAL PUBLICATIONS

FOLD