The ADIOS-81™

Apple Disk Interface and Operating System

Reference Manual

November 1981

Copyright (C) - 1981
All Rights Reserved Worldwide

The SOFT WAREHOUSE
P. O. Box 11174
Honolulu, Hawaii 96828
U. S. A.

COPYRIGHT NOTICE

Copyright (C), 1981 by The Soft Warehouse. All Rights Reserved Worldwide. No part of this manual may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any human or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, without the express written permission of The Soft Warehouse, P.O. Box 11174, Honolulu, Hawaii 96828, U.S.A.

TRADEMARK NOTICE

ADIOS is a trademark of **The Soft Warehouse. Apple II**, referred to throughout this manual, is a registered trademark of Apple Computer Inc., 10260 Bandley Drive, Cupertino, California, U.S.A.

DISCLAIMER

The Soft Warehouse makes no representations or warranties with respect to the contents hereof and specifically disclaims any implied warranties of merchantibility or fitness for any particular purpose. Further, The Soft Warehouse reserves the right to revise this publication and to make changes from time to time in the content hereof without obligation of The Soft Warehouse to notify any person or organization of such revision or changes.

TABLE OF CONTENTS

Title Page i

Notices ii

Table of Contents iii

S E C T I O N S

1. Introduction 1

2. System Requirements 2

3. System Start Up 3

4. Command Prompt 4

5. Command Syntax 5

6. ADIOS Commands 7

7. Line Editing Facilities 11

8. Non-standard Apple Characters 12

9. The Diskcopy Utility 13

10. Error Messages 14

11. Application Programming Guide 16

A P P E N D I C E S

Appendix I: Memory Map 18

Appendix II: Zero Page Usage 19

Appendix III: Jump Table 20

Appendix IV: Command Summary 21

Index 22

1. Introduction

The **Apple II Computer** has become especially popular in the educational and engineering fields. This provided The Soft Warehouse with a strong incentive to make the **muMATH Symbolic Math System** and **muLISP LISP Language System** available on the Apple II. However, the large size of muMATH and muLISP made it necessary to create an operating system which used less memory space than the Apple DOS.

The **Apple Disk Interface and Operating System** is the result. ADIOS-81 is a space efficient and flexible operating system for the Apple II. It is a simple system to use and fully supports both muMATH and muLISP. Since ADIOS is a general purpose system, it can support user written 6502 machine language application programs as well.

ADIOS supports console I/O and sequential disk I/O for up to six disk drives. Application programs, such as muSIMP and muLISP, can easily be loaded and run using simple ADIOS commands. Other commands display directory information, format diskettes, and perform file maintenance, including the saving, renaming, copying, and deleting of disk files. Access to the **Apple II Monitor** (either the Autostart ROM or Monitor ROM) is gained through the use of another ADIOS Command.

2. System Requirements

The ADIOS-81 operating system requires as a minimum the following computer hardware in order to operate:

1. An Apple II or Apple II Plus Computer

2. At least 48K (48 kilobytes) of RAM memory

3. At least one standard Apple II disk drive whose controller card is installed in peripheral card slot #6

4. Apple DOS version 3.3 which records 16 sectors per track on the diskette (If you are using an earlier version of Apple DOS, a 16 sector ROM for your disk controller card is available from your Apple dealer.)

For systems having more than one disk controller card, install the second card in slot #5 and the third card in slot #4.

3. System Start Up

To start up ADIOS simply turn on the computer and insert an ADIOS diskette in drive 1. If your Apple has the Autostart ROM, ADIOS should boot automatically. Apples with the Monitor ROM must be booted by typing the number 6 followed by <CTRL-P> and <RETURN>. In a few seconds the screen should clear and the following ADIOS **header message** should appear at the top of the screen:

APPLE][ADIOS-81 VERSION mm/dd/yy

Appropriate numbers will appear for the version month (mm), day (dd), and year(yy). This **version date** should be included in all inquires concerning ADIOS. Immediately below the header message is the ADIOS prompt. The significance of the prompt is fully described in the next section.

An alternative way to start up ADIOS is from INTEGER BASIC^{tm} or APPLESOFT BASIC^{tm}. Either of the following BASIC commands will do the trick:

IN#6

or

PR#6

Once ADIOS is started, ADIOS commands can be entered as described in Section 5.

4. Command Prompt

The ADIOS **command prompt** consists of two characters displayed by the computer next to the left margin of the screen. The first prompt character is a **drive number** between 1 and 6; the second character is a right parenthesis. The prompt indicates that the system is ready to accept an ADIOS command. The commands are described in Sections 5 and 6.

ADIOS can support up to 6 disk drives. The available drives are referred to by a single number from 1 to 6. This is in contrast to Apple DOS where both a peripheral card slot number and a drive number are needed to specify a drive. ADIOS drives 1 and 2 are drives 1 and 2 of slot #6; ADIOS drives 3 and 4 are drives 1 and 2 of slot #5; and ADIOS drives 5 and 6 are drives 1 and 2 of slot #4.

The number displayed in the prompt is the drive number of the currently **logged drive.** The significance of the logged drive will be explained shortly.

5. Command Syntax

ADIOS commands take the form of a **command word**, a space, and the **command arguments**, if any. The command is not actually issued until <RETURN> is entered. For example, to look at the **catalog** or list of files on the diskette in the logged drive, simply type the command word:

CATALOG

followed by a <RETURN>. For those who don't get a thrill out of typing, the CATALOG command can be abbreviated to:

CAT

As a matter of fact, **only the first three characters of all commands are checked;** any extra characters are ignored. Thus, the following command will also produce a directory listing:

CATFISH

Unlike the CATALOG command, most ADIOS commands require one or two file names be given following the command word. Such a **file name specification** (or **fns** for short) consists of a file name followed by a period followed by a file type. A file's name and type consists of the upper-case letters and/or numerals displayed by the CATALOG command. The name can be up to ten characters long; the file type can be up to three characters long. For instance, the fns for a file named APPLE of type PIE is "APPLE.PIE".

In order to specify a file on other than the logged drive, the fns can be preceded by the other drive number and a colon. For example, if the logged drive is currently 1 and you wish to delete the file APPLE.PIE on drive 2, issue the following command:

DELETE 2:APPLE.PIE

An alternative method of deleting the file is to change the logged drive from 1 to 2 and then delete the file. This would be accomplished by the following two ADIOS commands:

2:
DEL APPLE.PIE

In general, to switch logged drives, simply enter the desired drive number followed by a colon. Thereafter, ADIOS will display the new logged drive in the prompt.

When issuing an ADIOS command there are two basic types of errors which can occur. If the command line reader is unable to recognize a command, a **syntax error** occurs. The other type of error is usually the result of insufficient disk space or trying to read a non-existent file (see Section 10 for details). For either type of

error, an audio "beep" is sounded and the ADIOS prompt is redisplayed on the next line. As described in Section 7, it is possible to re-edit the previous command and try again. Commands which are successfully accomplished do not result in a beep, and a blank line is output before the next prompt is displayed.

6. ADIOS Commands

This section lists the available ADIOS commands in alphabetical order. The first three characters of each command word are in boldface to emphasize the mandatory part of the word. The command word is immediately followed by a symbol indicating the type of argument(s) the command requires, if any. File name specifications required for command arguments are abbreviated as fns. Finally the effects of each command are detailed.

When an application program is run under ADIOS, it is first loaded into an area of memory called the **Application Program Area.** The **APA** begins at memory location $1000 (1000 hexadecimal). This relatively large block of memory is the general purpose "working storage" area for ADIOS. It is used by several of the ADIOS commands listed below.

6.1 CATALOG (Catalog Command)

Syntax: **CAT**ALOG

The CATALOG command displays all the ADIOS files on the disk in the logged drive. Specifically, each file's size, name, and type are displayed in that order. The size of the file is given as a decimal number of kilobytes. Also given is the number of kilobytes of free storage available on the disk.

6.2 DELETE (Delete Command)

Syntax: **DEL**ETE fns

The DELETE command erases the file specified by fns from the directory. The space used by that file then becomes free for re-use. Naturally this command should be used with care.

6.3 FORMAT (Format Command)

Syntax: **FOR**MAT

The FORMAT command formats a new diskette on the logged drive. It also initializes the ADIOS directory (See the INITIALIZE command, Section 6.5).

WARNING

This command erases all the files on the diskette in the logged drive and should be used with care.

6.4 GET (Get System Command)

Syntax: **GET**

The GET command reads the ADIOS system recorded on the **system tracks** (i.e. tracks 0 and 1) of the logged drive and loads them into memory beginning at the base of the APA. When the GET and PUT commands are used in succession, they effectively transfer a copy of the ADIOS operating system to a new disk.

For those few users who will have to modify ADIOS for some reason, the following memory map gives the locations where the GET command loads the various sections of ADIOS in memory.

$1000	Boot Loader (stored on track 0, sector 0)
$1100	Logical I/O routines (ultimately loaded at $800)
$1800	Physical I/O routines (ultimately loaded at $B800)
$2000	Command Processor (ultimately loaded at $B000)
$2600	End of data

While in memory and using the Apple monitor, ADIOS can be modified, at your own risk of course, before using the PUT command to resave the system on a diskette.

6.5 INITIALIZE (Initialize Directory Command)

Syntax: **INITIALIZE**

The INITIALIZE command clears the ADIOS directory of all entries. It should be used if the **SCRAMBLED DIRECTORY** error message occurs (see Section 10).

WARNING

This command erases all files in the directory on the diskette in the logged drive and should be used with care.

6.6 IN# (Input Command)

Syntax: **IN#** slot number

The IN# command determines the source of console input to ADIOS. Normally input is from the Apple keyboard; however, the IN# command can change the source to any desired slot. Its effect is similar to the BASIC IN# command. See page 83 of the Apple II Manual for details. Remember to put a space following the command word as is required in all ADIOS commands.

6.7 LOAD (Load Command)

Syntax: **LOAD** fns

The LOAD command reads the file fns from the disk and loads it into memory beginning from the base of the APA. The file

contents can then be examined and modified using the MONITOR command.

6.8 MONITOR (Monitor Command)

Syntax: **MONITOR**

The MONITOR command exits ADIOS entirely and turns control over to the Apple II Monitor in ROM memory. While in this mode the Apple's normal line edit routine is in effect instead of the ADIOS line editor. To return to ADIOS, use the Monitor's "G" command to jump to the warm start location at $800 or, if it has not been disturbed while in the monitor, you can jump directly to the ADIOS command processor at $B000.

6.9 MOVE (Move Command)

Syntax: **MOVE** fns1, fns2

The MOVE command makes a copy of fns1 and names it fns2. This command has the advantage that it does not disturb an application program residing in the APA. However, MOVE is very slow for large files. A much faster way to copy a file is to first use the LOAD command to load the file into memory and then the SAVE command to save the file under a new name or on a different diskette.

6.10 PR# (Printer Command)

Syntax: **PR#** slot number

The PR# command determines the destination of ADIOS character output. Normally output is to the 40 column Apple screen. The PR# command can change the destination to any desired slot. Its effect is similar to the BASIC PR# command. See page 83 of the Apple II Manual for details. A space is required following the command word as in all ADIOS commands.

6.11 PUT (Put System Command)

Syntax: **PUT**

The PUT command writes onto the first two tracks of the disk a memory image beginning from the base of the APA. Normally this command is issued immediately following the GET command, making possible transfers of the ADIOS operating system to newly formatted disks. Note that this command does NOT destroy an existing ADIOS directory or any files on the disk receiving the new system.

6.12 RENAME (Rename Command)

Syntax: **RENAME** fns1, fns2

The RENAME command simply renames fns1 to fns2 in the directory. Unlike the MOVE command, no copying of the file occurs.

6.13 RUN (Run Command)

Syntax: **RUN** nam

The RUN command loads the application program file whose name is nam and whose type is COM into memory beginning from the base of the APA. It then begins execution of the application program by jumping to location $1000. If the RUN command is given no argument, a jump is made to location $1000 permitting re-entry into an application program like muSIMP. This is contingent, of course, on the APA not being disturbed since the application program was loaded or run.

6.14 SAVE (Save Command)

Syntax: **SAVE** fns num

The SAVE command creates a new file named fns consisting of a memory image beginning from the base of the APA for a length of num kilobytes. num will be the number displayed by the CATALOG command for the new file.

6.15 TYPE (Type Command)

Syntax: **TYPE** fns

The TYPE command displays the text file fns on the Apple screen. The display can be "frozen" and then restarted by using <CTRL-S> (i.e. typing an S while holding down the CTRL key) as a toggle. Output will be terminated if any other character is typed.

7. Line Editing Facilities

Occasionally, typing errors will be made when entering ADIOS commands. As long as the command has not been terminated by a <RETURN>, it is possible to correct the error without having to retype the entire line. Also, since application programs which run under ADIOS inherit these same editing facilities, it is important to become familar with their use.

Line editing is done through the use of **control characters** and the left and right arrow keys on the Apple II keyboard. ADIOS line editing facilities are very similar to those normally available on the Apple, except for the lack of **escape codes.** The best way to get the "feel" of ADIOS line editing is simply to experiment with it. The following description is primarily for reference purposes.

The current input line is stored in a 256 cell **character buffer** and hence this is the maximum length of a command. Characters are displayed at the current **cursor position** as they are entered.

To enter a **control character** in a line of text, press <ESC> and then type the desired control character. The control character entered in this way is displayed in reverse video.

The left and right arrow keys (i.e. "<--" and "-->") and <CTRL-X> function in a manner similar to what you are used to on the Apple. The left arrow key is for backspacing, the right arrow for advancing the cursor, and <CTRL-X> backspaces to the beginning of the current line. However, when backspacing the characters are actually deleted from the screen. Thus in ADIOS, the cursor is always at the right end of the line with no characters to its right.

The right arrow causes previously entered characters to be resurrected and redisplayed on the screen. Similarily, typing <CTRL-R> resurrects the entire line from the current cursor position. This is useful in case <CTRL-X> was inadvertently typed, erasing the line of text. Of even more utility is the fact that <CTRL-R> can be used to resurrect the **previously typed line** of text for re-editing.

8. Non-standard Apple Characters

The standard Apple keyboard and 40 column display screen lack eight of the less common ASCII characters. ADIOS makes it possible to enter these characters by using control characters in their place. Three of the eight characters are available on the 40 column screen. The other five characters are output to the screen as similar looking characters in inverse video.

The following table lists these characters, the equivalent control characters to enter instead, and the Apple screen characters which will be displayed in their place.

Character Name	Normal Character	Control Character	Apple Screen Character
Left bracket	[	<CTRL-K>	[
Backslash	\	<CTRL-B>	\
Underscore	_	<CTRL-E>	_
Left brace	{	<CTRL-L>	[(Inverse)
Right brace	}	<CTRL-O>	] (Inverse)
Back accent	`	<CTRL-A>	' (Inverse)
Vertical bar	|	<CTRL-V>	! (Inverse)
Tilde	~	<CTRL-T>	^ (Inverse)

As is usual on the Apple, <SHIFT-M> produces the right bracket.

9. The Diskcopy Utility

The diskcopy utility provides a convenient means of copying ADIOS diskettes. It should be used to make a **working copy** of your master diskette as soon as possible after receiving it. The utility copies the **source diskette** in its entirety onto the **target diskette.**

WARNING

All information currently on the target diskette is erased when the diskcopy utility is used.

Therefore, when learning to use this utility, it is a good idea to put a write-protect tab on the source diskette to prevent inadvertently erasing valuable information.

The target diskette must be formatted before using the diskcopy utility. This can be done using the FORMAT command as described in Section 6.3.

To run the utility, be sure a copy of the file **DISKCOPY.COM** is on the logged drive. Then simply issue the following ADIOS command followed by a <RETURN>:

RUN DISKCOPY

From this point on, the system will give self-explanatory prompts on how to proceed. For two drive systems it is simply a matter of specifying the source disk drive number and the target disk drive number. The copy takes about a minute.

If the same drive number is given for the source and target disk drives, the diskcopy utility assumes that a single drive system is being used. Users of such systems will naturally have to swap diskettes several times in order to make a copy. The utility will indicate which diskette to insert by means of prompting. Since it is possible to get the source and target diskettes confused while swapping them, we strongly recommend **a write-protect tab be put on the source diskette before beginning the copy.**

10. Error Messages

The following is a list of the error messages which can occur while operating under ADIOS.

10.1 CANNOT READ DISK

If the Apple is unable to read a needed sector of information from the diskette, this message will occur. It usually results from trying to read an unformatted diskette.

10.2 WRITE-PROTECTED DISK

If an attempt is made to write information on a write-protected diskette, this message will occur. If the write-protect tab is removed, the RE-TRY option can be used to proceed with the write.

10.3 BAD OR UNFORMATTED DISK

If an attempt is made to write on a defective or unformatted diskette, this message will occur. The FORMAT command can be used to reformat the diskette; however, remember to transfer any good files to another diskette before formatting.

10.4 SCRAMBLED DIRECTORY

This error occurs when the directory information on a diskette is not in the proper ADIOS format (e.g. an Apple DOS formatted diskette). Since there is no other satisfactory alternative, control is transferred to the Apple ROM monitor after this message is displayed. The INIT command will "clean up" a scrambled ADIOS directory; unfortunately however, in the process it deletes any existing files on the diskette.

10.5 UNABLE TO FORMAT DISK

This message will only occur while using the FORMAT command. It is usually caused by a defective diskette.

10.6 INSUFFICIENT DISK SPACE

This message will only occur while using the SAVE command. It indicates that there is not enough free space on the diskette to store the file being saved. Some other files must be deleted or another diskette must be used if the file is to be saved.

10.7 FILE TOO LONG

This message will only occur while using the LOAD command. It indicates that the file being loaded is too large to entirely fit in the APA (i.e. 40 kilobytes). However, the part that was loaded remains intact and is still in memory.

The first four messages can occur either while running an application program such as muSIMP, or while processing an ADIOS command. The last three messages only occur while processing specific ADIOS commands.

The first three messages will be followed by the following available **options message:**

ESC TO ABORT
SPACE TO IGNORE, RETURN TO RE-TRY:

This message indicates the user has three alternative ways to handle the error. The **ABORT option** is selected by hitting the <ESC> key. It is the normal, and safest, response since it abandons any attempt to continue reading from or writing to the diskette. It returns control to ADIOS and results in the ADIOS prompt.

If you are able to correct the error, for example by removing a write-protect tab, the **RE-TRY option** can be selected by hitting the <RETURN> key.

Finally, if you are daring, hitting the space bar will cause the error to be ignored. ADIOS will assume the information was correctly transferred to or from the diskette and act accordingly. We do not recommend using this **IGNORE option** except in the most dire emergencies.

11. Application Programming Guide

This section is intended for those few individuals who may be interested in writing 6502 machine language application programs to run under the ADIOS operating system. Others can safely ignore this section.

ADIOS provides a convenient, logical interface between application programs and the Apple II hardware. Communication with both the console and disk is via **byte I/O.** Facilities are provided for creating, writing, and closing disk files, as well as for opening and reading disk files. Up to 6 drives can be supported.

Application programs can receive characters directly from the keyboard or from a **line input buffer.** Using the input buffer automatically provides the application with the standard ADIOS line editing features (see Section 7). The current keyboard, or **input source,** and the current screen, or **output sink,** are determined by the ADIOS IN# and PR# commands respectively.

Machine language programs use the **File Control Block,** or **FCB,** to specify a drive number and file name and type when calling ADIOS routines. The FCB begins from location $EF0 and should contain information formatted as follows:

	|d| name	type
Location from $EF0:	0 1 2 3 4 5 6 7 8 9 A B C D

"d" stands for a drive number in the range of 0 to 6. Numbers between 1 and 6 indicate the respective drive. A drive number of 0 indicates that the logged drive should be used. The name field consists of 1 to 10 standard ASCII characters padded with sufficient ASCII spaces (i.e. a binary number $20) to fill the field. The type field consists of 1 to 3 characters also padded with sufficient spaces to fill the field.

All ADIOS routines are accessed through a **jump table** located at $800. The following is a description of the 11 entry points in the table:

$800 A jump to this location effectively terminates a running application program. The **ADIOS Command Processor** is loaded and then run, resulting in the ADIOS prompt.

$803 This routine returns the 6502 zero flag set if and only if no character has been typed on the Apple keyboard. Note: Even if the ADIOS IN# command has changed the input source to an external keyboard, this routine will still check the status of the Apple keyboard.

$806 This routine inputs a character from the current input source, normally the Apple keyboard, and returns it as an ASCII character in the accumulator. True ASCII is returned with the high bit set to zero.

$809 This routine outputs the ASCII character in the accumulator to the current output sink, normally the 40 column Apple screen.

$80C This routine returns in the accumulator the next ASCII character stored in the line input buffer. The 6502 carry flag is set if and only if the ADIOS line editor had to be called to input another line of text.

$80F This routine opens an existing file for input. The drive number and file name and type given in the FCB is used in a directory search for the file. If the file is found, it is made the current input file and the 6502 carry flag is returned set; otherwise, it is reset.

$812 This routine returns the next byte read from the current input file in the accumulator. The carry flag is returned reset if and only if an attempt was made to read past the end of the file.

$815 This routine creates a new file for output. The drive number and file name and type given in the FCB is used to make a new directory entry for the file. The carry flag is returned reset if and only if the routine was unsuccessful.

$818 This routine writes the byte in the accumulator to the current output file. The carry flag is returned reset if and only if there is insufficient room on the diskette to store the file.

$81B This routine closes the current output file and modifies the directory to reflect the fact. The carry flag is returned reset if and only if the system is unable to close the file, probably because of insufficient disk space.

$81E This routine outputs to the current output sink the ASCII message immediately following the call to $81E. The end of the message is indicated by a 0 byte. The following is a typical use of this subroutine:

```
JSR     $81E
DB      'THE QUICK BROWN FOX',0
LDA     JUNK
...
```

$821 This is not an entry point but a byte which contains the number of the logged drive. It is initially one and is normally changed by the ADIOS change logged disk command (see Section 5).

The jump table information given above is summarized in Appendix III.

Appendix I: Memory Map

Address	Function
$000-$0FF	Zero page special locations
$100-$1FF	Hardware stack
$200-$2FF	Line input buffer
$400-$7FF	Text and LO-RES graphics screen memory
$800-$FFF	Logical ADIOS operating system routines
$1000-$AFFF	Application Program Area (APA)
$B000-$B5FF	ADIOS Command Processor (ACP)
$B600-$B6FF	Disk input buffer
$B700-$B7FF	Disk output buffer
$B800-$BFFF	Physical disk I/O routines
$C000-$CFFF	Apple II I/O ROM locations*
$D000-$F7FF	Integer BASIC ROM or Applesoft BASIC ROM
$F800-$FFFF	Monitor ROM or Autostart ROM

If Microsoft's RAMCARD^{tm}, or equivalent, is present, it replaces the ROM memory from location $D000 to $FFFF with RAM memory.

* See Apple II Reference Manual

Appendix II: Zero Page Usage

The following table indicates the use of a page zero memory location by the Apple monitor with an "M". ADIOS page zero usage is indicated by "X"s.

	$0	$1	$2	$3	$4	$5	$6	$7	$8	$9	$A	$B	$C	$D	$E	$F
$00																
$10																
$20	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
$30	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
$40	M	M	M	M	M	M	M	M	M	M	X	X	X	X	M	M
$50	M	M	M	M	M	M										
$60								X	X	X	X					X
$70	X															
$80	X	X	X	X	X	X	X	X	X	X	X	X	X	X		
$90																
$A0																X
$B0	X															
$C0											X	X	X	X		
$D0									X							
$E0																
$F0																

Appendix III: Jump Table

Address	Function
$800	Load and run the ADIOS ACP.
$803	Apple keyboard character ready status.
$806	Input character from input source. (unbuffered).
$809	Output character to output sink.
$80C	Input character from line input buffer.
$80F	Open a file for input.
$812	Read next character from current input file.
$815	Create a new file for output.
$818	Write a character to current output file.
$81B	Close current output file.
$81E	Print message following call to screen.
$821	Current logged drive number.

Appendix IV: Command Summary

Command	Effect
CATALOG	Display catalog of files
DELETE fns	Delete file fns
FORMAT	Format disk in default drive
GET	Get operating system into RAM
IN# slot number	Input from slot number
INIT	Initialize directory
LOAD fns	Load file fns into RAM
MONITOR	Jump to Monitor ROM
MOVE fns1, fns2	Copy file fns1 to fns2
PR# slot number	Output to slot number
PUT	Put operating system on disk
RENAME fns1, fns2	Rename file fns1 to fns2
RUN nam	Run nam.COM command file
SAVE fns num	Save file fns of num K-bytes
TYPE fns	Type file fns on console

Index

ABORT option 15
ADIOS 1
ADIOS Command Processor (ACP) 16
ADIOS commands 7-10
Apple II Microcomputer 1
Apple II Monitor 1,9
Apple DOS ii
Application Program Area (APA) 7
Application Programming Guide 16

BAD OR UNFORMATTED DISK 14
BASIC commands 3
byte I/O 16

CANNOT READ DISK 14
CATALOG 5
catalog 5
character buffer 11
command arguments 5
command prompt 4
command summary 21
command syntax 5
command word 5
control character(s) 11
copyright notice ii
CTRL-A 12
CTRL-B 12
CTRL-E 12
CTRL-K 12
CTRL-L 12
CTRL-O 12
CTRL-R 11
CTRL-T 12
CTRL-V 12
CTRL-X 11
cursor position 11

DELETE 5,7
DISKCOPY utility 13
disclaimer ii
drive number 4,5

Error messages 14-15
ESC 11,15
escape codes 11

File Control Block (FCB) 16
file name specification (fns) 5
FILE TOO LONG 15
fns 5
FORMAT 7

GET 7-8

header message 3

IGNORE option 15
INITIALIZE 8
input source 16
INSUFFICIENT DISK SPACE 14
IN# 3,8

jump table 16,20

logged drive 4
line editing 11
line editing facilities 11
line input buffer 16
LOAD 8

Memory Map 18
MONITOR 9
MOVE 9

options message 15
output sink 16

prerequisite hardware 2
PUT 9
PR# 3,9

RENAME 10
RE-TRY option 15
RETURN 5,11
RUN 10

SAVE 10
SCRAMBLED DIRECTORY 14
SHIFT-M 12
Soft Warehouse, The i,ii
syntax error 5
system requirements 2
system start up 3
system tracks 8

trademark notice ii
TYPE 10

UNABLE TO FORMAT DISK 14

version date 3

WRITE-PROTECTED DISK 14

Zero page usage	19
[left bracket	12
] right bracket	12
\ backslash	12
_ underscore	12
{ left brace	12
} right brace	12
` back accent	12
| vertical bar	12
~ tilde	12
—> right arrow key	11
<— left arrow key	11

July 8, 1982

the SOFT WAREHOUSE™
P O Box 11174 / Honolulu, Hawaii 96828-0174
Telephone (808) 734-5801

The 6502 ADIOS version of muSIMP was originally written by translating an 8086 version of the interpreter. The 8086 pseudo-registers are stored in page zero in the following locations:

Register	Location	Register	Location
AX	$0C0	SI	$0C8
BX	$0C2	DI	$0CA
CX	$0C4	BP	$0CC
DX	$0C6		

Arguments are passed to muSIMP machine language routines on the "BP" stack. The first argument is stored at location BP+2; the second is at BP+4; the third at BP+6; etc. For example, the following code retrieves the first argument and stores it in SI:

```
GETARG1:  LDY  #2
          LDA  (BP),Y
          STA  SI
          INY
          LDA  (BY),Y
          STA  SI+1
          RTS
```

The address of the muSIMP function SPACES can be found using the GETD function. A disassembly of SPACES should look like this:

```
SPACES:   JSR   GETARG1          ;SI: ARGUMENT TO SPACES
          JSR   BINARY           ;BX: BINARY EQUIVALENT OF ARG
          BCC   SPACE2           ;JUMP IF NOT 0 <= ARG < 64K
          BEQ   SPACE2           ;JUMP IF ARG = 0
SPACE1:   JSR   SPACE            ;OUTPUT A SPACE
          DEC   BX
          BNE   SPACE1           ;LOOP FOR BX SPACES
SPACE2:   LDA   CURSOR           ;A: CURRENT CURSOR POSITION
          JMP   GENNM            ;RETURN DI: CURSOR POSITION
```

This code contains two useful addresses to you: BINARY takes a muSIMP integer and returns the binary equivalent in BX. GENNM takes a binary number in the the 6502 A register and generates a muSIMP integer in DI, the register used to return values to muSIMP.

Sincerely,

Albert Rich
Applied Logician

* NOTE: Addresses in MUSIMP.COM are offset $200 when loaded using the ADIOS "LOAD" command and then examined from the Apple monitor. Therefore, if the GETD command indicates SPACES is at location $2C3C, you will be able to find it at location $2E3C.