

SOFT-STEP™

Debugger for Applesoft programs

DEBUGGER

STEP

TRACE

BREAK

LIST

ACCENT

software

UPDATE NOTICE: SOFT-STEP VERSION 1.1

Dear Soft-Step User:

SOFT-STEP 1.1 overcomes a limitation imposed by Apple's DOS mentioned on page 44 of the Apple DOS manual, as well as by the earlier SOFT-STEP 1.0. Specifically, you could not Trace, Single Step through, or stop at a Breakpoint in a section of code which had a DOS READ or WRITE in effect.

In overcoming this limitation, the added logic to make this improvement increases the length of SOFT-STEP slightly, and the address parameters listed on page 2 of the SOFT-STEP user's manual should now read as follows:

SYSTEM SIZE	48K	32K
HIMEM	35572	19188
SOFT-STEP START ADDRESS	35573	19189
SOFT-STEP LENGTH	2827	2827
SOFT-STEP END ADDRESS	38399	22015

SOFT-STEP™

Debugger for Applesoft programs

REPLACEMENT: If this disk becomes worn or damaged, Accent Software, Inc., will gladly replace it. Send the damaged disk with proof of purchase and \$10.00 to:

**Accent Software, Inc.
3750 Wright Place
Palo Alto, CA 94306**

ACCENT SOFTWARE, INC., and APPLE COMPUTER, INC., make no warranties, either expressed or implied, regarding the enclosed computer software package, its merchantability or its fitness for any particular purpose.

APPLESOFT and APPLE are registered trademarks of Apple Computer, Inc.

TABLE OF CONTENTS

1.0 INTRODUCTION	1
2.0 GETTING STARTED	2
2.1 BOOTING UP	2
2.2 OPTION MENU	2
2.3 BLOADING SOFT-STEP	2
2.4 ENTERING SOFT-STEP AND THE AMPERSAND	2
2.5 LOADING YOUR BASIC PROGRAM	3
2.6 QUITTING	3
3.0 EXECUTION COMMANDS	3
3.1 STEP	3
3.2 GO TO BREAKPOINT	3
3.3 RUN	3
4.0 DIRECT COMMANDS	3
4.1 PRINT	3
4.2 PEEK	4
4.3 POKE	4
4.4 LET	4
5.0 INTERRUPTIONS	4
5.1 BREAK	4
5.2 SCROLL-STOP	4
5.3 SCROLL-START	4
5.4 FORCED BREAKPOINT	4
6.0 TABLE COMMANDS	5
6.1 DISPLAYING TABLE ENTRIES	5
6.2 INSERTING TABLE ENTRIES	5
6.3 DELETING SINGLE TABLE ENTRIES	6
6.4 DELETING ALL TABLE ENTRIES	6
7.0 THE AUTOMATIC COMMANDS	6
7.1 DISPLAYING AUTO-MODES	6
7.2 AUTO-TRACE	6
7.3 AUTO-LIST	6
7.4 AUTO-DO	7
7.5 CONTINUOUS AUTOMATIC COMMANDS	7
7.6 CANCELLING AUTOMATIC COMMANDS	7
8.0 MISCELLANEOUS COMMANDS	7
8.1 CLEAR	7
8.2 FROM	7
8.3 LIST	7
8.4 WARM START	8
8.5 EXIT	8

8.6 EXECUTE THE DO COMMANDS	8
8.7 HOME	8
9.0 APPLICATION NOTES	8
9.1 VIDEO MODE	8
9.2 SCREEN POSITIONING	8
9.3 ARRAYS	9
9.4 IMPLICIT DIMENSIONING	9
9.5 PAGE ZERO AND PAGE THREE	9
9.6 HIMEM	9
9.7 MAXFILES	9
SOFT-STEP COMMAND SUMMARY	10

1.0 INTRODUCTION

SOFT-STEP is an interactive debugger for Applesoft Basic programs. **SOFT-STEP** allows you to:

- STEP** — Single step through your basic program, statement by statement, pausing to allow you to examine or define variables or memory.
- BREAK** — Set breakpoints at the line of your choice and the program will pause there and await your command.
- DO** — Predefine a command to **DO** whenever the debugger pauses. Automatically print or define memory or variables!
- LIST** — See the line to be executed next; automatically, if you wish. No guessing or printout required!
- TRACE** — **TRACE** all lines, or only the ones you are interested in. Much clearer and more versatile than Apple's Trace.
- EXAMINE** — **PRINT** the value of any variable or **PEEK** at any memory address whenever the debugger has paused for your command.
- DEFINE** — **LET** a variable change its value, or **POKE** values into memory.
- SMALL** — Only 2.7 kbytes long! (32K or 48K required)

All of this, and more without ever altering or inserting logic in your source program!

2.0 GETTING STARTED

SOFT-STEP was assembled to be run on a 32K or 48K machine, and resides just below DOS in memory.

2.1 BOOTING UP

With Autostart Rom, just insert the SOFT-STEP disk, close the drive door, and turn on the Apple's power switch.

Without the Autostart Rom, or with your Apple turned on, insert the SOFT-STEP disk, close the drive door, and type:

]PR#n

where n is the slot number of your disk controller (usually 6).

2.2 OPTION MENU

After a successful boot, you should see a title page, and a prompt to press any key to continue. Pressing any key will cause the OPTION MENU as shown below to appear.

OPTIONS:

- (1) BLOAD SOFT-STEP BELOW DOS
- (2) LOAD BASIC PROGRAM
- (3) QUIT

ENTER INDEX (1-3): _____

2.3 BLOADING SOFT-STEP

To BLOAD SOFT-STEP enter the digit "1" and press [return].

After Bloading SOFT-STEP the loader will display a MEMORY USAGE REPORT containing the starting address, length, and end address of the program displayed in decimal. Depending on the memory of your system you will see:

SYSTEM SIZE	48K		SYSTEM SIZE	32K
HIMEM	35691		HIMEM	19307
SOFT-STEP START	35692	or	SOFT-STEP START	19308
SOFT-STEP LENGTH	2708		SOFT-STEP LENGTH	2708
SOFT-STEP END	38399		SOFT-STEP END	22015

2.4 ENTERING SOFT-STEP AND THE AMPERSAND

To enter SOFT-STEP use Applesoft's ampersand (&) command. SOFT-STEP will then display a title banner, a message indicating the starting line of your basic program, and a prompt for you to enter a command. If you have no Applesoft program loaded, the title banner will be repeatedly displayed. Should this happen, press [RESET] and LOAD your program.

2.5 LOADING YOUR BASIC PROGRAM

After SOFT-STEP is BLOADED, all that is left to do is load your basic program. Choosing option (2) from the OPTION MENU will prompt you to enter your source disk, and then present you with the following prompt:

ENTER PROGRAM NAME:

After you type the basic program name and press (return) your program will be loaded. This causes the loader program to end. Control then passes to Applesoft. If this is undesirable, use the QUIT option. When you are ready, load your program with the normal] LOAD command.

2.6 QUITTING

To exit the loader, choose option (3) from the OPTION MENU. At this point if SOFT-STEP was *not* successfully loaded, the message "SOFT-STEP NOT BLOADED" will appear. If SOFT-STEP was successfully loaded, the MEMORY USAGE REPORT will once again be displayed for your information. Additionally, the ampersand vector status will be displayed as ON.

3.0 EXECUTION COMMANDS

3.1 STEP

Command Format: **S**

The STEP command instructs SOFT-STEP to execute the next Applesoft statement and then pause and wait for a debugger command. If the line contains multiple statements separated by colons, only the next segment is executed, not the entire line.

3.2 GO TO BREAKPOINT

Command Format: **G**

The GO command instructs SOFT-STEP to begin execution of the Applesoft program and continue execution until one of the breakpoint lines is reached. SOFT-STEP then pauses and waits for a command. Program Execution stops just before the breakpoint line; it is *not* executed. See Section 6 for instructions describing how to set breakpoints.

3.3 RUN

Command Format: **R**

The RUN command instructs SOFT-STEP to begin execution of the basic program with no debugger pauses. Only the debugger pauses are disabled, not any of the operations to be performed according to the AUTOMATIC commands described in Section 7. The key press scroll control is also still enabled.

4.0 DIRECT COMMANDS

4.1 PRINT

Command Format: **PRINTexp**

The PRINT command allows you to print the value of an expression. The

expression may be any combination that would be valid in BASIC, of variable names, numeric constants, quoted text, and the operators: plus (+), minus(-), times(*), divide(/), equals(=), comma(,), and semicolon(;).

4.2 PEEK Command Format: **PEEK(exp)**

The PEEK command actually peeks at the address defined by the expression argument, and then prints the value. See PRINT command description for expression rules.

4.3 POKE Command Format: **POKEexpA,expV**

POKE the value of expression V into the address defined by expression A. The value of expression V must be in the range 0 to 255.

4.4 LET Command Format: **LETv=exp**

The LET command allows you to define or re-define the value of any variable. The variable may be a real, integer or string variable. This useful feature allows you to set up initial conditions before executing a segment of code, hence you can check portions of your program whether the preceding portions of your program work correctly or not!

V is any legal variable name (although only 2 characters are significant) and expression is described further under the PRINT command.

5.0 INTERRUPTIONS

5.1 BREAK Command Format: **CTRL-C**

Pressing the control key and the C key simultaneously has the same effect as in Applesoft basic, that is, to stop execution and enter Applesoft basic. Use the ampersand (&) and FROM commands to restart SOFT-STEP.

5.2 SCROLL-STOP Command Format: **ANY KEY**

Pressing any key while the basic program is executing will cause it to temporarily suspend processing to allow you to examine the screen.

5.3 SCROLL-START Command Format: **ANY KEY**

Press any key except ESCAPE to resume execution when suspended by a key press.

5.4 FORCED BREAKPOINT Command Format: **ESC**

Pressing the [ESCAPE] key when the program is suspended by a key press will cause a forced breakpoint. NOTE that any actions to be executed at pauses, such as AUTO-LIST or AUTO-DO will be executed since a breakpoint is a pause.

7.4 AUTO-DO

Command Format: AD

Setting **AUTO-DO** instructs **SOFT-STEP** to automatically execute the **DO** commands at pauses. This command works only in the **STEP** and **GO** execution modes.

7.5 CONTINUOUS AUTOMATIC COMMANDS

Command Format: AT*
AL*
AD*

This mode, valid with all three **AUTO**-commands, instructs **SOFT-STEP** to perform the desired option (**TRACE**, **LIST**, or **DO**) before each statement is executed. This mode is valid in all three execution modes (**RUN**, **STEP**, **GO**).

7.6 CANCELLING AUTOMATIC COMMANDS

Command Format: A/T
A/L
A/D

To cancel any of the **AUTO**-commands, type an "A" followed by a slash (/), followed by either a "T" for **TRACE**, an "L" for **LIST**, or a "D" for **DO**.

8.0 MISCELLANEOUS COMMANDS

The following commands may be entered at any time the Debugger is in the pause state.

8.1 CLEAR

Command Format: C

This command clears the variables and resets the memory pointers described in your Applesoft manual. This command, unlike Applesoft's **clear**, does not restore the **DATA** pointer to its beginning.

8.2 FROM

Command Format: FNNNNN

The **FROM** command allows you to change where the Debugger starts executing the basic program. The command may be entered at any time **SOFT-STEP** is awaiting a command. As with all **SOFT-STEP** commands, do not leave any blanks between the command **F**, and its argument, the line number represented here by **NNNNN**.

ERRORS: If the line cannot be found, an error results and control returns to Applesoft. To recover use the ampersand (&) to re-enter **SOFT-STEP**, and the **FROM** command to reset the line to be executed to the same one as before the error occurred.

8.3 LIST

Command Format: L

The **LIST** command lets you see the next line to be executed. As with all **SOFT-STEP** output, the line will be in inverse video. There are always

exceptions to any rule, though, and the LIST command demonstrates one. If the line to be executed contains multiple statements as Applesoft allows, only the next segment to be executed is in inverse, not the entire line, so you know exactly where execution begins.

8.4 WARM START

Command Format: W

The preferred way to exit SOFT-STEP is by performing a D.O.S. warm start. This re-connects D.O.S. and leaves you in Applesoft Basic. Variables are cleared. An unconditional JMP is performed to \$3D0.

8.5 EXIT

Command Format: X

An alternate, but not always appropriate way to exit SOFT-STEP is with the EXIT command. An RTS command (return from subroutine) is executed. This command will not work, however, when the Debugger is inside of a FOR/NEXT loop or in a Basic subroutine. This is because the RTS command looks on top of the stack, an area of the computer's memory, for the address it must go to. But both the FOR/NEXT and the GOSUB commands temporarily put information on the stack.

8.6 EXECUTE THE DO COMMANDS Command Format: E

Entering an "E" forces execution of the DO commands, described in Section 6. This feature effectively gives you a keyboard macro capability.

8.7 HOME

Command Format: H

Entering an "H" performs the Applesoft HOME command, clearing the screen and placing the cursor in the upper left hand corner of the screen.

9.0 APPLICATION NOTES

9.1 VIDEO MODE

All SOFT-STEP output is done in inverse video so that you may easily distinguish it from your program's normal output. Whenever SOFT-STEP tries to print something, it sets the print mode to INVERSE, prints its message, then sets the print mode to NORMAL. NOTE that this may cause your program's output to be changed to NORMAL even though you may have previously set it to INVERSE.

9.2 SCREEN POSITIONING

SOFT-STEP's output is printed on the next available line on the screen, with the screen scrolling up when full. This method of output was chosen over using a reserved area of the screen to allow you to access to the entire screen. This method does have its peculiarities though, since your program's output and SOFT-STEP's output may occasionally conflict with each other. Remember, too, that disk commands must be preceded by a carriage return, and SOFT-STEP doesn't always provide one.

9.3 ARRAYS

Occasionally some odd things will happen with arrays, but by reading this section carefully, you should be able to understand and avoid problems.

When SOFT-STEP begins execution, it does not automatically clear the variables and arrays, as Applesoft's RUN command does; therefore, if a DIM statement is executed by SOFT-STEP after the program has already executed a DIM statement by either SOFT-STEP or Applesoft, a REDIMENSION ERROR will occur. To remedy this, just use the SOFT-STEP C (clear) or Applesoft CLEAR command before re-executing a program with dimension statements

9.4 IMPLICIT DIMENSIONING

Applesoft allows array variables to be dimensioned implicitly by ten (10) simply by using the variable as an array element. For example, the line

```
10 A(1) = 100.0
```

as the first line of your program would implicitly dimension an array of length 10 and give element one the value of 100.0. So what does this have to do with SOFT-STEP? Plenty! SOFT-STEP allows you to enter the command

```
PRINT A(1)
```

either as a direct or DO command. This will cause a redimension error if used before the DIM statement in your program. The solution is not to reference array variables with SOFT-STEP before they have been dimensioned. Placing all dimension statements as the first lines in your program, in addition to being good programming practice, will help overcome the problem.

9.5 PAGE ZERO AND PAGE THREE

Assembly language users will be pleased to learn SOFT-STEP uses no page zero or page three addresses, other than those used for normal Applesoft and Monitor operation.

9.6 HIMEM

To avoid any memory conflict between your program and SOFT-STEP, HIMEM is set just below the start of SOFT-STEP.

9.7 MAXFILES

WARNING! Setting MAXFILES greater than the default will overwrite SOFT-STEP.

SOFT-STEP COMMAND SUMMARY

• EXECUTION COMMANDS

R	RUN normally. No debugger pauses.
G	GO . Run until a breakpoint line is reached, then pause.
S	STEP , pausing for debugger command after each statement.

• MISCELLANEOUS CONTROL COMMANDS

C	CLEAR the variables.
E	EXECUTE the DO commands.
FNNNN	Set execution to begin FROM line NNNN.
H	Applesoft HOME command. Clears screen.
L	LIST the next line to be executed.
W	EXIT to basic using jump to dos WARM start.
X	EXIT to caller, using "RETURN".

• TABLE COMMANDS

T	DISPLAY line numbers to TRACE .
B	DISPLAY line numbers to BREAK at.
D	DISPLAY commands to DO at pauses.
TNNNN	INSERT line number NNNN into TRACE table.
BNNNN	INSERT line number NNNN into BREAK table.
Ddirect	INSERT direct command into DO table.
T/NNNN	DELETE line number NNNN from TRACE table.
B/NNNN	DELETE line number NNNN from BREAK table.
D/N	DELETE Nth command from DO table. (N starts at 0)
T/*	DELETE ALL line numbers from TRACE table.
B/*	DELETE ALL line numbers from BREAK table.
D/*	DELETE ALL commands from DO table.

• AUTOMATIC COMMANDS

A	DISPLAY the Automatic modes.
AT	AUTOMATICALLY TRACE lines in TRACE table.
AL	AUTOMATICALLY LIST lines in BREAK table.
AD	AUTOMATICALLY execute the DO commands at pauses.
AT*	TRACE ALL lines immediately before executing.
AL*	LIST ALL lines immediately before executing.
AD*	DO commands at each statement.
A/T	CANCEL AUTO-TRACE .
A/L	CANCEL AUTO-LIST .
A/D	CANCEL AUTO-DO .

- **DIRECT COMMANDS**

- PRINTexp** PRINT the value of the expression.
- PEEK(exp)** PEEK and PRINT the value at address defined by the expression.
- POKEexpA,expV** POKE the value of expression V into the address defined by expression A.
- LETv=exp** LET variable V = value of expression.

Expression is any valid combination of variable names, numeric constants, quoted text, and the operators plus (+), minus(-), times(*), divide(/), equals (=), semicolon (;), and comma(,).

- **DO COMMANDS**

- Ddirect** Any of the direct commands, preceded by a "D" places that command in the DO table, a list of commands to be executed according to the AUTO options chosen.

- **INTERRUPTIONS**

- CTRL-C** Pressing CONTROL-C while the program is executing will cause a break and debugger exit.
- SCROLL-STOP** Pressing any key while the program is executing will cause it to temporarily suspend processing, allowing the user to examine the screen.
- SCROLL-START** Press any key, except ESCAPE, to resume program execution when suspended by a keypress.
- ESC** Pressing the ESCape key, when the program is suspended by a keypress, will cause a forced breakpoint.

**Apple II or II Plus
ROM Applesoft
DOS 3.3
32k or 48k Memory**

A new, easy-to-use debugger for the expert and beginner alike. No modification to your source program needed. Fast assembly language. Only 2.7 kbytes long.

- Single-step thru your program
- Set breakpoint at any line
- List next line to be executed
- Trace all or only chosen lines
- Examine values of variables, memory
- Define values of variables, memory