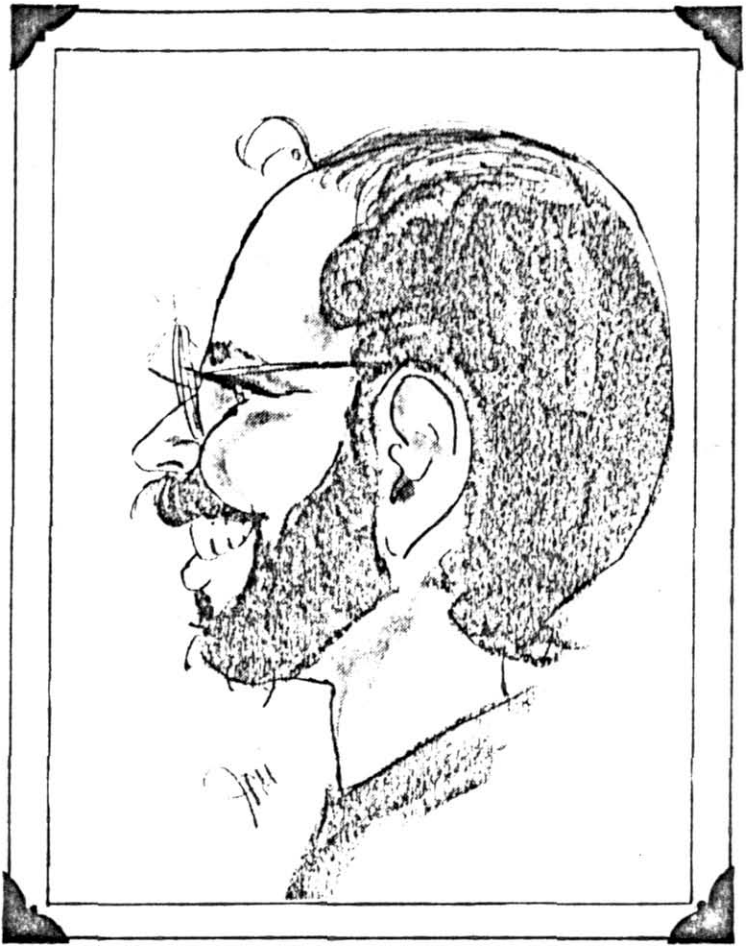
BOSS© 1980

Utility Reference Manual

Version 2.1



Written by V.B. Hester

For

Soft Sector Marketing, Inc.

BOSS 2.1 Version 2.2

Version 2.2 Boss from Soft Sector Marketing Inc. has only cosmeticly been changed from Version 2.1. The program now runs on Ultra Dos. All instructions are the same as in the manual for 2.1 (provided) execpt as per this addendum. It now comes with a universal lower case driver at no extra cost.

The enclosed lower case driver tape runs in any Level II or D.O.S. that has the hardware modification to permit lower case. This program has been tested with about 8 different types of lower case hardware mods, including the RADIO SHACK MOD and should run with no problems. Like the program BOSS 2.1 this program loads to the top of memory, under anyother program that is already there and resets the memory size to protect it's self.

TO LOAD TAPE TYPE:

```
SYSTEM    (ENTER)
*? L      (ENTER)
*? /      (ENTER)
```

Information to load lower case driver to disk.

```
starting location 5F00
ending location   5FAF
transfer location 5F00
```

If this package is on disk then to run the program just type LC from DOS READY.

Once the program is executed your in lower case. You must now hold the shift key to have a capital letter appear on the screen. To lock the program in the upper case mode hold down the SHIFT key and hit the SPACE BAR. To return to lower case, reverse the process.

Last but not least is the new low price of $18.95 on cassette and $23.95 on diskette.

B O S S OVERVIEW

INTRODUCTION

This utility is designed to aid you in creating and debugging programs written in basic. The utility will allow you to trace the program flow, to single step the basic program, to observe the conditions of variables during program execution, and to push your basic programs on the stack or pop them off the stack during program development. The utility is known to operate with the following versions of disk operating systems:

TRSDOS 2.2-2.3	NEWDOS - 80
NEWDOS 2.1	VTOS - 3.0

The utility will operate with either Level II Basic or Disk Basic. The minimum equipment configuration is a 16K Level II TRS-80 Microcomputer with cassette input. The program will automatically relocate itself for larger memory machines.

A brief description of each function is listed below.

TRACE FUNCTION

Allows you to follow the twisted (or logical) path your program takes, without messiness on the screen caused by the other trace function.

SINGLE STEPPING

Allows you to single step individual lines of a basic program of individual instructions within a line.

BREAK POINT

The trace and single step commands can be invoked by your program while it is running with this feature.

REVIEWING VARIABLES

Allows you to pause to review selected variables during program execution and return to your program with the display restored to that shown before you review the variables (great if your screen had graphics displayed).

STACKING PROGRAMS

Allows you to stack one or more programs in high memory while you work on or run another program. Of course, this ability is limited by the amount of free memory space available. You can retrieve the stacked programs at will.

BOSS

UTILITY REFERENCE MANUAL
(VERSION 2.1)

CONTENTS	PAGE
1. GENERAL INFORMATION	3
2. TRACING FUNCTION	6
3. SINGLE STEPPING	7
4. SETTING BREAKPOINTS	8
5. REVIEWING VARIABLES	9
6. STACKING PROGRAMS	11
7. USER NOTES	13

IMPORTANT NOTICE

THIS "BOSS" COMPUTER PROGRAM IS DISTRIBUTED ON AN "AS IS" BASIS WITHOUT WARRANTY. THE AUTHOR AND SELLING DEALER SHALL HAVE NO LIABILITY OR RESPONSIBILITY TO CUSTOMER OR ANY OTHER PERSON OR ENTITY WITH RESPECT TO ANY LIABILITY, LOSS OR DAMAGE CAUSED OR ALLEGED TO BE CAUSED DIRECTLY OR INDIRECTLY BY THE "BOSS" COMPUTER PROGRAM, INCLUDING BUT NOT LIMITED TO ANY INTERRUPTION OF SERVICE, LOSS OF BUSINESS OR ANTICIPATORY PROFITS OR CONSEQUENTIAL DAMAGES RESULTING FROM THE USE OF THE "BOSS" PROGRAM. THE AUTHOR AND SELLING DEALER MAKE NO CLAIM AS TO THE "BOSS" COMPUTER PROGRAMS FITNESS, SUITABILITY FOR A PARTICULAR USE, OR PERFORMANCE.

BOSS - UTILITY REFERENCE MANUAL (VERSION 2.1)

1 - GENERAL INFORMATION

1.1 INTRODUCTION

THIS UTILITY IS DESIGNED TO AID YOU IN CREATING AND DEBUGGING PROGRAMS WRITTEN IN BASIC. THE UTILITY WILL ALLOW YOU TO TRACE THE PROGRAM FLOW, TO SINGLE STEP THE BASIC PROGRAM, TO OBSERVE THE CONDITIONS OF VARIABLES DURING PROGRAM EXECUTION, AND TO PUSH YOUR BASIC PROGRAMS ON THE STACK OR POP THEM OFF THE STACK DURING PROGRAM DEVELOPMENT. THE UTILITY IS KNOWN TO OPERATE WITH ALL CURRENT DISK OPERATING SYSTEMS.

THE UTILITY WILL OPERATE WITH EITHER LEVEL II BASIC OR DISK BASIC. THE MINIMUM EQUIPMENT CONFIGURATION IS A 16K, LEVEL II, S-80 UNIT WITH CASSETTE INPUT. THE UTILITY IS UPWARDS COMPATIBLE WITH 32K AND 48K TAPE AND DISC BASED S-80 SYSTEMS.

1.2 BACKUP

BEFORE YOU BEGIN, IT IS RECOMMENDED THAT YOU MAKE A BACKUP COPY OF THE UTILITY FOR YOUR PROTECTION.

1.3 TRANSFERRING YOUR "BOSS" CASSETTE TO DISKETTE

A. TAPEDISK/CMD

```
    PREPARE THE "BOSS" CASSETTE FOR LOADING.
    UNDER "DOS READY" TYPE TAPEDISK <ENTER>
    ANSWER THE "?" PROMPT WITH "C" <ENTER> (LOADS "BOSS" INTO RAM)
    AFTER "BOSS" HAS BEEN LOADED, ANOTHER "?" WILL APPEAR
    TO SAVE "BOSS" ONTO THE DISKETTE, ENTER THE FOLLOWING:
    ?F BOSS/CMD:0 6000 6B39 6070 <ENTER>  (NOTE: DRIVESPEC REQUIRED)
    ?E <ENTER>
    NOW "BOSS" IS ON YOUR DISKETTE FOR "DOS READY" EXECUTION.
```

B. LMOFFSET/CMD

```
    PREPARE THE "BOSS" CASSETTE FOR LOADING.
    UNDER "DOS READY" TYPE LMOFFSET <ENTER>
    ANSWER THE QUERIES AS FOLLOWS:
    "SOURCE FROM DISK OR TAPE? REPLY "D" OR "T"?"  T <ENTER>
    AFTER A SUCCESSFUL LOAD, LMOFFSET WILL RESPOND WITH THE LOAD
    ADDRESSES, THE ENTRY POINT, AND TELL YOU IT WILL OVERLAP THE
    "CMD" PROGRAM AREA. (NOTE: RELOCATION IS NOT NECESSARY)
    RESPOND TO "NEW LOAD BASE ADDRESS (HEX)?" WITH <ENTER>
    "DESTINATION FILESPEC?"  BOSS/CMD <ENTER>
    NOW "BOSS" IS ON YOUR DISKETTE FOR "DOS READY" EXECUTION.
```

1.4 INITIALIZING THE UTILITY (DISC SYSTEMS)

THIS UTILITY IS SELF-RELOCATING AND DOES NOT HAVE TO RESIDE IN THE HIGHEST BYTE OF USER RAM AVAILABLE. LOAD ANY OTHER MACHINE LANGUAGE ROUTINES OR UTILITIES NEEDED BEFORE YOU LOAD AND INITIALIZE "BOSS" TO INITIALIZE THE UTILITY FROM THE "DOS READY" MODE, TYPE "BOSS". DO TRY TO "LOAD" THIS UTILITY.

THE UTILITY WILL LOAD, EXECUTE AND RESPOND WITH:
"ENTER MEMORY SIZE (DECIMAL) YOU WANT TO PROTECT?"

IF YOU DO NOT WANT TO PROTECT ANY MEMORY OTHER THAN THAT REQUIRED FOR "BOSS", PRESS "ENTER". IF YOU HAVE SOME OTHER MACHINE LANGUAGE CODE YOU WANT TO PROTECT, ENTER THE LOWEST DECIMAL MEMORY LOCATION WHICH THAT CODE (NOT "BOSS") WILL OCCUPY. "BOSS" WILL RESPOND BY TELLING YOU WHAT ACTUAL MEMORY SIZE TO USE TO PROTECT "BOSS" AND ANY OTHER MACHINE LANGUAGE CODE YOU ASKED TO PROTECT, IF THE OTHER CODE HAS BEEN LOADED ABOVE "BOSS".

NOTE: USERS OF TRSDOS 2.2 OR LATER WILL HAVE THEIR PROGRAM PROTECTED FROM THE "64 BYTE BOMB", UPON "BASIC" OR "BASICR" INITIALIZATION.

EXAMPLE #1 BELOW SHOWS INITIALIZATION STEPS FOR A USER WHO ALSO HAS LOADED A 1024 BYTE ROUTINE TO RESIDE IN HIGH MEMORY OF A 48K MACHINE.

```
**************************************************************
*  DOS READY                                                 *
*  BOSS                                                      *
*                                                            *
*  (THE SCREEN WILL CLEAR AND THE FOLLOWING TEXT APPEARS)    *
*                                                            *
*  ENTER MEMORY SIZE (DECIMAL) YOU WANT TO PROTECT? 64512    *
*                                                            *
*  (THE SCREEN WILL CLEAR AND THE FOLLOWING TEXT APPEARS)    *
*                                                            *
*  ANSWER THE MEMORY SIZE QUESTION WITH XXXXX                *
**************************************************************
```

EXAMPLE 1

THE UTILITY MOVES ITSELF TO MEMORY JUST BELOW THE MEMORY SIZE YOU ASKED TO PROTECT.

AFTER "BOSS" HAS RELOCATED ITSELF, TYPE:

(FOR TRSDOS) BASIC
ANSWER THE FILES QUESTION AS NEEDED.
ANSWER THE MEMORY SIZE QUESTION WITH XXXXX, WHERE XXXXX IS THE MEMORY SIZE WHICH "BOSS" HAS ASKED YOU TO USE.

(FOR NEWDOS) BASIC F,XXXXX
WHERE F= THE NUMBER OF FILE BUFFERS TO BE USED AND XXXXX = THE MEMORY SIZE WHICH BOSS HAS ASKED YOU TO USE. AFTER THE SYSTEM GIVES YOU THE READY PROMPT, THE FIRST LETTER KEY YOU PRESS WILL INITIALIZE THE UTILITY. YOU ARE NOW READY TO USE THE FUNCTIONS DESCRIBED IN SECTIONS 2, 3, 4, OR 5.

1.5 INITIALIZING THE UTILITY (CASSETTE SYSTEMS)

AS WITH DISC BASED SYSTEMS, THE UTILITY IS SELF-RELOCATING AND DOES NOT HAVE TO RESIDE IN THE HIGHEST BYTE OF USER RAM AVAILABLE. IF YOU HAVE ANY OTHER MACHINE LANGUAGE UTILITY OR DRIVER ROUTINES TO GO INTO HIGH MEMORY, ANSWER THE MEMORY SIZE QUESTION AS NORMALLY REQUIRED AND LOAD THE MACHINE LANGUAGE CODE BEFORE YOU LOAD AND INITIALIZE "BOSS". IF YOU DO NOT LOAD ANY OTHER MACHINE LANGUAGE CODE EXCEPT "BOSS", A SPECIFIC MEMORY SIZE REPLY IS NOT NEEDED.

PREPARE THE "BOSS" CASSETTE FOR LOADING AND TYPE "SYSTEM". ANSWER THE "*?" PROMPT WITH "BOSS". AFTER A SUCCESSFUL LOAD (APPROXIMATELY 50 SECONDS) ANOTHER "*?" WILL APPEAR. PRESS THE "/" AND "ENTER" KEYS. THE SCREEN WILL CLEAR AND THE FOLLOWING PROMPT WILL APPEAR:

```
ENTER MEMORY SIZE (DECIMAL) YOU WANT TO PROTECT?
```

ENTER THE MEMORY SIZE USED TO PROTECT THE PREVIOUSLY LOADED MACHINE LANGUAGE CODE OR, IF "BOSS" WAS THE ONLY PROGRAM LOADED, PRESS "ENTER". "BOSS" WILL RESPOND WITH THE FOLLOWING MESSAGE:

```
YOUR MEMORY SIZE IS XXXXX
```

1.6 CONTROL KEY

TO ENABLE YOU TO USE THE UTILITY A CONTROL KEY HAS BEEN DEVELOPED. THIS KEY IS THE "@" KEY AND WILL BE REFERRED TO AS <CON> IN THE REMAINING SECTIONS OF THIS DOCUMENTATION. THE "@" SYMBOL IS NOW A SHIFTED ZERO. REMEMBER, THE <CON> KEY IS JUST ABOVE THE "ENTER" KEY AND IS LABELED "@".

NOTE: THE SHIFT "@" STILL CANNOT BE USED IN "PRINT@", YOU MUST USE THE SHIFT "0" KEY TO OBTAIN THE PROPER "@" SYMBOL. SHIFT "@" CONTINUES TO FUNCTION AS "PAUSE EXECUTION AND FREEZE DISPLAY".

2 - TRACING FUNCTION

2.1 INTRODUCTION

THIS UTILITY WILL ALLOW YOU TO FOLLOW THE TWISTED (OR LOGICAL) PATH YOUR PROGRAM TAKES, WITHOUT THE MESSINESS ON THE SCREEN CAUSED BY SOME OTHER TRACE FUNCTIONS. THERE ARE THREE MAJOR TRACE COMMANDS.

<CON> 1	=	TRACE OFF
<CON> 2	=	TRACE ON TO VIDEO DISPLAY
<CON> 3	=	TRACE ON TO PRINTER

2.2 TRACE OFF

PRESSING "<CON> 1" WILL TURN OFF THE TRACE. WHEN THE UTILITY IS LINKED, IT WILL SLOW DOWN. PROGRAM EXECUTION. IF YOU WANT TO CHECK CRITICAL TIMING LOOPS YOU SHOULD TURN OFF THE UTILITY AND UNLINK IT BY PRESSING "<CON> 1". THIS WILL ALLOW YOUR PROGRAM TO RUN AT NORMAL SPEED.

2.3 TRACE TO DISPLAY

PRESSING "<CON> 2" WILL TURN ON THE TRACE FUNCTION AND WILL DISPLAY EACH BASIC PROGRAM LINE NUMBER, AS IT IS EXECUTED. THE LINE NUMBERS WILL BE DISPLAYED ON THE TOP RIGHT AREA OF THE SCREEN IN THE FOLLOWING FORMAT:

```
50
60
30
40
```

THE LAST FOUR LINE NUMBERS EXECUTED WILL BE DISPLAYED. THE LINE NUMBER CURRENTLY BEING EXECUTED WILL BE PREFIXED BY THE "→" SIGN. THIS AREA WILL NOT SCROLL AND WILL OVERPRINT ANYTHING DISPLAYED BY YOUR PROGRAM AS LONG AS THE TRACE IS DIRECTED TO THE SCREEN.

2.4 TRACE TO THE PRINTER

PRESSING "<CON> 3" WILL TURN THE TRACE FUNCTION ON AND DIRECT THE OUTPUT TO THE PRINTER. NO TRACE INFORMATION WILL APPEAR ON THE SCREEN. THE OUTPUT WILL LOOK SIMILAR TO THIS:

```
10    20    30    50    60  1000  1010  1020  1030    70
```

TRACING WILL CONTINUE TO THE PRINTER UNTIL THE TRACE FUNCTION IS REDIRECTED TO THE DISPLAY VIA "<CON> 2" OR TURNED OFF VIA "<CON> 1".

THE TRACE FUNCTION WILL SHOW EACH LINE NUMBER AS THAT LINE IS ENTERED FOR EXECUTION. IF MULTIPLE STATEMENT LINES ARE USED SUCH AS "100 FOR X=1 TO 10: A(X)=3*Y: NEXT X ", THE TRACE FUNCTION WILL DISPLAY LINE 100 ONE TIME, NOT TEN TIMES. CALLS TO SUBROUTINES, GOTOS AND REMARK LINE NUMBERS WILL BE SHOWN AS ENCOUNTERD.

3 - SINGLE STEPPING

3.1 INTRODUCTION

THIS UTILITY WILL ALLOW YOU TO SINGLE STEP INDIVIDUAL LINES OF A BASIC PROGRAM OR INDIVIDUAL INSTRUCTIONS WITHIN A LINE. IN ADDITION, YOU CAN VARY THE DELAY IN WHICH YOUR PROGRAM STEPS BETWEEN LINES OR INDIVIDUAL INSTRUCTIONS. THERE ARE FOUR SINGLE STEP COMMANDS.

<CON> 4	=	SINGLE STEP OFF
<CON> 5	=	SINGLE STEP TO END OF LINE
<CON> 6	=	SINGLE STEP INSTRUCTION
<CON> 7	=	VARIABLE DELAY STEP

3.2 SINGLE STEP OFF

PRESSING "<CON> 4" WILL TURN OFF THE SINGLE STEP FUNCTION AND ALLOW YOUR PROGRAM TO RUN AS NORMAL. IF THE TRACE FUNCTION WAS IN USE, IT WILL CONTINUE TO FUNCTION UNTIL TURNED OFF. (SEE SECTION 2.2)

3.3 SINGLE STEP TO END OF LINE

PRESSING "<CON> 5" WILL CAUSE YOUR PROGRAM TO PAUSE AT THE END OF EACH LINE UNTIL THE SPACE BAR (OR ANY OTHER KEY) IS PRESSED. THE TRACE TO VIDEO DISPLAY MODE WILL ALSO BE INITIATED TO SHOW YOU WHICH LINE NUMBER IS BEING EXECUTED. THIS TRACE MODE CAN BE DISABLED BY A "<CON> 1", WHILE THE SINGLE STEP MODE WILL CONTINUE.

3.4 SINGLE STEP INSTRUCTION

PRESSING "<CON> 6" WILL CAUSE YOUR PROGRAM TO PAUSE WHEN AN INSTRUCTION SEPARATOR, :, IS FOUND AND AT THE END OF EACH LINE. PRESS THE SPACE-BAR (OR ANY OTHER KEY) TO CONTINUE TO THE NEXT INSTRUCTION. AGAIN, THE TRACE TO VIDEO DISPLAY MODE WILL BE INVOKED TO SHOW YOU WHICH LINE NUMBER IS BEING EXECUTED. THIS FUNCTION CAN BE USEFUL, BUT BE WARY OF USING IT IF A PROGRAM CONTAINS LINES SUCH AS:

```
90 FOR X=1 TO 100:A(X)=6+3*X*Z:NEXT X
```

TO SINGLE STEP THROUGH THIS LOOP WOULD REQUIRE 300 PRESSES OF A KEY. INSTEAD USE "SINGLE STEP TO END OF LINE."

3.5 VARIABLE DELAY STEP (AUTO STEP)

PRESSING "<CON> 7" WILL CAUSE YOUR PROGRAM TO DELAY APPROXIMATELY 0.25 SECONDS AT THE END OF EACH LINE. AGAIN THE TRACE TO VIDEO WILL BE INVOKED TO SHOW YOU WHICH LINE NUMBER IS BEING EXECUTED. "<CON> 5" AND "<CON> 6" BECOME SUB-COMMANDS AFTER "<CON> 7" IS INITIATED.

PRESSING "<CON> 6" AFTER "<CON> 7" IS INITIATED WILL CAUSE THE DELAY TO OCCUR AT AN INSTRUCTION SEPARATOR, IN ADDITION TO THE END OF A LINE. PRESSING "<CON> 5" WILL CAUSE THE DELAY TO OCCUR AT THE END OF A LINE ONLY. THIS DELAY HAS NINE SETTINGS FROM APPROXIMATELY 4 MILLISECONDS TO APPROXIMATELY 0.9 SECONDS. TO SPEED UP EXECUTION (DECREASE DELAY) PRESS "<CON> ↑". TO SLOW DOWN EXECUTION (INCREASE DELAY) PRESS "<CON> ↓". THE AMOUNT OF DELAY CAN BE ADJUSTED ANY TIME AFTER BOSS IS INITIALIZED IN BASIC. THE INITIAL SETTING PROVIDES 0.25 SECONDS DELAY. THE AMOUNT OF DELAY IS HALVED EACH TIME "<CON> ↑" IS PRESSED, OR DOUBLED EACH TIME "<CON> ↓" IS PRESSED. DURING THIS DELAY, KEY PRESSES ARE NOT RECOGNIZED.

4 - SETTING BREAK POINTS

4.1 BREAK POINTS

THE TRACE AND SINGLE STEP COMMANDS DESCRIBED IN SECTIONS 2 AND 3 CAN BE INVOKED BY YOUR PROGRAM WHILE IT IS RUNNING BY INSERTING A POKE INSTRUCTION IN YOUR PROGRAM AT THE LOCATION WHERE YOU WANT TO INVOKE THE COMMAND. THE FOLLOWING CODES ARE USED:

FUNCTION	POKE 16667,
TRACE OFF	1
TRACE TO DISPLAY	2
TRACE TO PRINTER	3
SINGLE STEP OFF	4
SINGLE STEP TO END OF LINE	5
SINGLE STEP INSTRUCTION	6
VARIABLE DELAY STEP	7

4.2 EXAMPLES OF BREAKPOINT USE

IF YOU WANT NORMAL PROGRAM EXECUTION TO LINE 1540, THEN SINGLE STEPPING WITH TRACE TO THE SCREEN, INSERT JUST PRIOR TO LINE 1540, THE INSTRUCTION "POKE 16667,5".

```
      EXAMPLE 1                 EXAMPLE 2
1530 (USERS TEXT)          1530 (USERS TEXT)
1535 POKE 16667,5          1540 POKE 16667,5: (USERS TEXT)
1540 (USERS TEXT)
```

NOTE: IF YOUR PROGRAM LOGIC HAS GOTO'S, GOSUB'S, ETC., BE SURE TO POSITION THE BREAKPOINT WHERE THE CODE WILL BE EXECUTED.

MULTIPLE POKES ARE PERMITTED; POKE16667,7:POKE16667,1:POKE16667,6
THIS WILL INVOKE "VARIABLE DELAY STEP" BETWEEN INSTRUCTIONS WITH THE TRACE DISABLED

NOTE: YOU CAN INSERT AS MANY BREAK POINTS IN YOUR PROGRAM AS YOU DESIRE.

5 - REVIEWING VARIABLES

5.1 GENERAL INFORMATION

THIS UTILITY WILL ALLOW YOU TO PAUSE TO REVIEW SELECTED VARIABLES DURING PROGRAM EXECUTION AND THEN RETURN TO YOUR PROGRAM WITH THE DISPLAY RESTORED TO THAT SHOWN BEFORE YOU REVIEWED THE VARIABLES. THERE ARE TWO COMMANDS USED FOR THIS FUNCTION.

<CON> N	=	SELECT VARIABLES FOR REVIEW
<CON> O	=	REVIEW THE SELECTED VARIABLES

5.2 SELECTING VARIABLES

PRESSING "<CON> N" WILL ALLOW YOU TO SELECT THE VARIABLES YOU WANT TO REVIEW DURING PROGRAM EXECUTION. THIS COMMAND CAN BE ENTERED AT ANY TIME BEFORE YOU RUN THE PROGRAM OR DURING PROGRAM EXECUTION.

AFTER INVOKING "<CON> N", THE QUERY "ENTER MAXIMUM VARIABLE LENGTH?" WILL BE DISPLAYED. RESPOND FROM THE FOLLOWING CHOICES:

RESPONSE	RESULT
BREAK	EXIT FUNCTION & RETURN TO BASIC PROGRAM
1	1 CHARACTER VARIABLE NAMES
2 OR 3	MAXIMUM OF 3 CHARACTER VARIABLE NAMES
4 - 7	MAXIMUM OF 7 CHARACTER VARIABLE NAMES
8 - 15	MAXIMUM OF 15 CHARACTER VARIABLE NAMES
16 - 31	MAXIMUM OF 31 CHARACTER VARIABLE NAMES
ENTER	DEFAULT TO MAXIMUM OF 7 CHARACTER NAMES

THE MAXIMUM NUMBER OF VARIABLES FOR REVIEW IS LIMITED BY THE MAXIMUM VARIABLE NAME LENGTH SELECTED, AS SHOWN BELOW.

NAME LENGTH	NUMBER OF VARIABLES TO REVIEW
1	MAXIMUM OF 128
2-3	MAXIMUM OF 64
4-7	MAXIMUM OF 32
8-15	MAXIMUM OF 16
16-31	MAXIMUM OF 8

NOTE: THE NAME LENGTH INCLUDES ALL CHARACTERS. THE VARIABLE NAME A$(21,5) IS CONSIDERED TO HAVE A LENGTH OF EIGHT (8), F(R(3,8)) IS NINE CHARACTERS IN LENGTH.

AFTER SUCCESSFULLY ENTERING A VARIABLE LENGTH, THE MESSAGE "INPUT VARIABLES FOR REVIEW, "BREAK" TERMINATES FUNCTION" WILL BE DISPLAYED AND ALL PREVIOUSLY ENTERED VARIABLE CHOICES WILL BE ERASED. THIS FUNCTION WILL ALLOW YOU TO ENTER VARIABLES USING THE FOLLOWING SYNTAX:

```
A           X!            B#          Q!(F(G,Q))
K$          A!(F(G,Q))    F(2,3)      T#(F,(B(A,N),(G(E)))
A(B)        WEEKDAY       S%          A(B,C)
                            (ETC.)
```

ANY NUMBER OF PARENTHESES ARE ALLOWED, PROVIDED YOU CLOSE THEM WITHIN THE VARIABLE LENGTH ENTERED. ALTHOUGH ILLEGAL VARIABLE NAMES SUCH AS A$3 OR A(3H) ARE NOT REJECTED, THEY WILL CAUSE ERRORS LATER WHEN REVIEW OF THE VARIABLES IS ATTEMPTED.

WHEN YOU HAVE FINISHED ENTERING THE VARIABLES, PRESS "BREAK" TO CONTINUE WITH THE REVIEW? IF YOU ENTER THE MAXIMUM NUMBER OF VARIABLES ALLOWED, "BOSS" WILL AUTOMATICALLY PROCEED WITH THE REVIEW. AT THIS POINT "BOSS" INVOKES A "<CON> O" AS DESCRIBED IN SECTIÓN 5.3 BELOW.

5.3 DISPLAY VARIABLES

PRESSING "<CON> O" AT ANY TIME DURING PROGRAM EXECUTION WILL IMMEDIATELY SAVE THE CONTENTS OF THE VIDEO DISPLAY AND REPLACE IT WITH THE MESSAGE "VARIABLES - PRESS "C" (CHANGE), "BREAK" (END), OTHERS ADVANCE" WILL APPEAR ALONG WITH THE FIRST VARIABLE AND ITS VALUE. VARIABLES ARE DISPLAYED IN THE ORDER ENTERED BY THE "<CON> N" FUNCTION.

PRESSING "BREAK" WILL CAUSE THE MESSAGE "END - PRESS "BREAK" TO RETURN, OTHERS TO REVIEW AGAIN" TO APPEAR. IF YOU ARE REALLY FINISHED WITH THE VARIABLE REVIEW, PRESS "BREAK" AGAIN AND YOUR ORIGINAL VIDEO DISPLAY WILL BE RETURNED. YOUR PROGRAM WILL RESUME EXECUTION AT THAT POINT. IF YOU INSTEAD WANT TO REVIEW MORE VARIABLES, PRESS ANY KEY OTHER THAN "BREAK". PRESSING "C" WILL ALLOW YOU TO SELECT ANOTHER VARIABLE IN PLACE OF THE LAST VARIABLE DISPLAYED. THE NEW VARIABLE SELECTED AND ITS VALUE WILL BE DISPLAYED. REMEMBER, THE VARIABLE NAME IS LIMITED IN LENGTH PER YOUR ORIGINAL CHOICE WHEN "<CON> N" WAS SELECTED.

PRESSING ANY KEY OTHER THAN "BREAK" OR "C" WILL ADVANCE THE DISPLAY TO THE NEXT VARIABLE SELECTED. IF YOU ATTEMPT TO REVIEW A VARIABLE WHOSE SUBSCRIPT IS OUT OF RANGE OR WITH AN ILLEGAL NAME, THE MESSAGE "ERROR, RE-ENTER" WILL BE DISPLAYED AND THE "C" COMMAND WILL AUTOMATICALLY BE INVOKED. YOU MUST SELECT A VALID VARIABLE TO EXIT FROM THIS SUB-COMMAND. IF YOU EVALUATE AN ELEMENT OF AN ARRAY (SUBSCRIPT < 11) AND THE ARRAY HAS NOT YET BEEN DIMENSIONED BY YOUR PROGRAM, THIS ARRAY WILL BE DIMENSIONED FOR ELEVEN ELEMENTS (0-10). IF YOUR PROGRAM SUBSEQUENTLY ATTEMPTS TO DIMENSION THIS ARRAY VIA THE "DIM" INSTRUCTION, AN ERROR WILL OCCUR. DIMENSION ALL USED ARRAYS BEFORE YOU REVIEW THEM WITH THE REVIEW VARIABLES FUNCTION.

6 - STACKING PROGRAMS

6.1 GENERAL INFORMATION

THIS UTILITY WILL ALLOW YOU TO STACK ONE OR MORE PROGRAMS IN HIGH MEMORY WHILE YOU WORK ON OR RUN ANOTHER PROGRAM. OF COURSE, THIS ABILITY IS LIMITED BY THE AMOUNT OF FREE MEMORY SPACE AVAILABLE. YOU CAN RETRIEVE THE STACKED PROGRAM(S) AT WILL. THE INTERRUPTS ARE DISABLED IN ORDER FOR THIS FUNCTION TO OPERATE PROPERLY. IN ADDITION THIS FUNCTION INVOKES A "CLEAR 50" UPON COMPLETION. THERE ARE FIVE MAJOR COMMANDS FOR THIS FUNCTION.

<CON> -	=	SAVE THE BASIC PROGRAM IN HIGH MEMORY (PUSH)
<CON> :	=	RECALL THE LAST SAVED PROGRAM FROM MEMORY (POP)
<CON> 8	=	APPEND THE LAST SAVED PROGRAM TO THE CURRENT PROGRAM
<CON> 9	=	APPEND THE NEXT TO LAST SAVED PROGRAM TO THE CURRENT PROGRAM
<CON> 0	=	RECALL THE NEXT TO LAST SAVED PROGRAM (SWITCH)

6.2 PUSHING PROGRAMS

PRESSING "<CON> -" WILL "SAVE" THE RESIDENT PROGRAM IN HIGH MEMORY USING THE "MEMORY SIZE" PREVIOUSLY ENTERED AS THE END AND WILL AUTOMATICALLY ADJUST "MEMORY SIZE" TO THE BEGINNING OF THE OF THE PUSHED PROGRAM, THUS PROTECTING IT FROM BASIC. YOUR CURRENT PROGRAM WILL ALSO BE LEFT AVAILABLE IN BASIC RAM, IF MEMORY PERMITS. SINCE MEMORY SIZE IS ADJUSTED, SUBSEQUENT PUSHES CAN BE MADE AS DESIRED.

WHEN A PROGRAM IS PUSHED, A GRAPHIC VERTICAL BAR WILL APPEAR IN THE UPPER RIGHT CORNER OF THE VIDEO DISPLAY. THIS BAR INDICATES THAT A PROGRAM HAS BEEN PUSHED INTO HIGH MEMORY AND A PROGRAM IS IN BASIC MEMORY FOR USER EXECUTION OR MODIFICATION. IF INSUFFICIENT MEMORY IS AVAILABLE TO PUSH YOUR PROGRAM AND ALSO MAINTAIN IT IN BASIC RAM (I.E. TRYING TO PUSH A 30K PROGRAM IN A 48K MACHINE), YOUR PROGRAM WILL BE PUSHED AND A "NEW" INVOKED. THIS CONDITION WILL BE INDICATED BY A CLEAR SCREEN WITH A SMALL GRAPHIC BLOCK IN THE UPPER RIGHT OF THE VIDEO DISPLAY.

6.3 POPPING PROGRAMS

PRESSING "<CON> :" WILL POP OR RETRIEVE THE LAST SAVED PROGRAM FROM HIGH MEMORY. THE RESIDENT PROGRAM WILL BE LOST. MEMORY SIZE WILL AUTOMATICALLY BE ADJUSTED. IF NO SAVED PROGRAM REMAINS, THE ERROR MESSAGE "NOTHING TO POP" WILL BE DISPLAYED.

6.4 SWITCHING PROGRAMS

PRESSING "<CON> 0" WILL RETRIEVE THE NEXT TO LAST SAVED PROGRAM FROM HIGH MEMORY. THE RESIDENT PROGRAM WILL BE LOST. IF YOU WANT TO SWITCH THE RESIDENT PROGRAM WITH THE LAST SAVED PROGRAM FIRST PUSH THE CURRENT PROGRAM VIA THE "<CON> -" COMMAND AND THEN RETRIEVE THE NEXT TO LAST PROGRAM VIA THE "<CON> 0" COMMAND.

6.5 APPENDING PROGRAMS

PRESSING "<CON> 8" WILL RETRIEVE THE LAST SAVED PROGRAM AND APPEND IT TO THE RESIDENT PROGRAM.

PRESSING "<CON> 9" WILL RETRIEVE THE NEXT TO LAST SAVED PROGRAM AND APPEND IT TO THE RESIDENT PROGRAM.

EXAMPLES OF MEMORY CONTENTS

BEFORE <CON> 8	AFTER <CON> 8	BEFORE <CON> 9	AFTER <CON> 9
UTILITIES	UTILITIES	UTILITIES	UTILITIES
PROG 1	FREE RAM	PROG 1	PROG 2
FREE RAM	FREE RAM	PROG 2	FREE RAM
FREE RAM	PROG 1	FREE RAM	PROG 1
PROG 2	PROG 2	PROG 3	PROG 3

LINE NUMBER SEQUENCE IS MANDATORY FOR PROPER EXECUTION OF THE APPENDING COMMANDS. THE STACKED PROGRAM SHOULD HAVE IT'S LOWEST LINE NUMBER GREATER THAN THE HIGHEST LINE NUMBER IN THE CURRENT PROGRAM IN BASIC RAM. THESE COMMANDS APPEND, THEY DO NOT MERGE.

NOTE: THE "PUSHED" ITEM CAN BE AS LITTLE AS ONE PROGRAM LINE. THIS MAKES THIS FUNCTION USEFUL TO APPEND/INSERT A FAVORITE ROUTINE INTO SEVERAL PROGRAMS.

7.0 USER NOTES

7.1 NOTES

IF A "LINE" PRINTER IS BEING USED, THE TRACE INFORMATION WILL SHOW THE COMPLETE PROGRAM FLOW, BUT WILL ONLY BE PRINTED WHEN AN ENTIRE PRINT LINE IS AVAILABLE. THIS IS BECAUSE THE PRINTER WILL NOT OUTPUT ANY DATA UNTIL A COMPLETE PRINT LINE IS SENT. IF A CHARACTER AT A TIME, PRINTER IS BEING USED, THE TRACE INFORMATION WILL BE PRINTED AS EACH PROGRAM LINE NUMBER IS EXECUTED.

THE PRINT FORMAT FOR A TRACE TO PRINTER IS SPACE, 10K DIGIT, 1K DIGIT, 100'S DIGIT, 10'S DIGIT, AND UNIT DIGIT. ZERO SUPRESSION IS USED AND EACH LINE REQUIRES SIX CHARACTER POSITIONS. THIS FITS IN WELL WITH 72 CHARACTER PER LINE (12 LINE TRACES) AND 132 CHARACTER PER LINE (22 LINE TRACES) PRINTERS. THIS OUTPUT CAN BE CONTROLLED BY ANY TYPE OF USER LINKUP TO THE PRINTER DEVICE CONTROL BLOCK, SINCE IT IS RAW SPACE/NUMERIC OUTPUT. NO LINEFEEDS, CARRIAGE RETURNS, OR CONTROL CHARACTERS ARE SENT.

7.2 THINGS YOU SHOULD KNOW

A. THIS UTILITY DEFEATS THE BUILT-IN TRACE FUNCTION.

B. THIS UTILITY LINKS TO THE KEYBOARD CONTROL BLOCK AND INCORPORATES ANY KEYBOARD ROUTINE PRESENT WHEN "BOSS" IS INITIALIZED (I.E. KEYBOARD DEBOUNCE ROUTINE).

C. THIS UTILITY LINKS TO THE VIDEO CONTROL BLOCK AND INCORPORATES ANY ROUTINE PRESENT WHEN "BOSS" IS INITIALIZED.

D. REBOOTS WILL NOT RECOVER THE UTILITY. DOS USERS USE OF "BASIC *" WILL NOT RECOVER THE UTILITY.

E. BREAK POINTS ARE INVOKED AFTER THE BREAKPOINT INSTRUCTION IS EXECUTED.

F. "BOSS" DOES NOT INTERFERE WITH ANY UTILITIES WHICH LOAD BELOW BASIC.

G. YOU CAN RESERVE ADDITIONAL MEMORY BY USING AN APPROPRIATE NUMBER LESS THAN THAT WHICH "BOSS" ASKED YOU TO USE, OR YOU CAN START BOSS AT A LOWER LOCATION AS YOU SEE FIT.

H. USERS OF TRSDOS VERSIONS 2.2 OR LATER MAY USE THE "ENTER" COMMAND IN PLACE OF A SPECIFIC MEMORY SIZE IF "BOSS" IS LOADED LAST. USERS OF NEWDOS EDITION 05/29/79 (ZAP 013) OR LATER CAN ALSO USE THE "ENTER" COMMAND.

ORIGINAL DOCUMENTATION WRITTEN BY C. A. P. THOMAS
VERSION 2.1 ADDITIONS BY V. B. HESTER

7260