

# **M**ACRO

# **U**TILITIES

# **M**ASTER

## **For Apple II Programmers**

A package of swift and sure Applesoft programming aids

### **Dear Applesoft Programmer**

You have chosen the right MUM and like every other mum this one will look after your special needs for a lifetime.

Let us have a look at the powerful assistance that MUM provides. The simplification of programming tasks now possible using the macros on our disk has to be experienced to be fully appreciated. AND, you can create seven more macros using Ctrl D = define. Editing programs becomes so easy.

Other options provided are conversion decimal/hex, checks on length of program and memory, program compaction, combination and renumbering. The search macro finds and displays for you lines containing characters, variables or strings of your choice, even all REM lines or all GOSUBs. Other macros let you find variables and cross-refer them or make easy insertions and changes.

As well as these editing functions the disk gives you some varied utilities to aid your programming still further. There is a facility to find all data statements in machine code software. A generator converts programs to text files. Another program finds specified hex sequences for you.

Look through the instructions on this sheet and you will see how much help is provided by these carefully devised utilities.

Happy programming

**HEYDEN DATASYSTEMS**

## HOW TO START THE MACROEDITOR

1. If the Applesoft program is not yet in RAM memory, simply boot the MUM Macroeditor disk by entering PR# + Slot Number (open apple, control, reset on the IIe).
2. If the Applesoft program is already in RAM memory, start the Macroeditor by entering BRUN MACROEDITOR.

The Macroeditor resides from \$7D00 through \$95FF. HIMEM is set to \$7CFF or decimal 31999. MAXFILES must have been set to 3. This is automatically done by booting the program disk.

The Macroeditor is not destroyed by NEW, FP, MAXFILES 3, HIMEM < 32000 etc., whereas it is destroyed by MAXFILES > 3, HIMEM > 32000 or pressing RESET.

## HOW TO UNHOOK AND REHOOK THE MACROEDITOR

The Macroeditor modifies the keyboard and the ampersand vectors. Applesoft programs that increase MAXFILES or HIMEM or use ampersand commands cannot be RUN while the Macroeditor is active. Any other program can be RUN if the Macroeditor is partially disabled by resetting the keyboard vectors. This is automatically done by the macro Ctrl P. Instead you can enter (on separate lines)

```
PR#0  
IN #0  
CALL 1002
```

The Macroeditor is simply unhooked by

```
&U
```

The Macroeditor is rehooked by

```
& or by CALL 32000
```

NEVER PRESS RESET while the Macroeditor is "working" (e.g. while in edit mode or while renumbering or cross-referencing) THIS IS A SURE WAY TO DESTROY YOUR PROGRAM, since pointers may not be reset correctly.

## IMMEDIATE MODE COMMANDS

Immediate mode commands must be entered after a RETURN has been issued – explicitly or implicitly, i.e. the cursor must be at the left edge of the video screen.

### 1. Non control commands

Y + line number = enter edit mode

If you type the letter Y, the word LINE will be displayed. Now enter the number of the program line you want to edit.

K + line number = enter edit mode with (k)runching

If a program line is overlong, use K instead of Y. The K command removes surplus spaces and converts PRINT to ? enabling you to edit extremely long program lines.

M + monitor command = monitor command

If you enter M300L or M300.3FF or M300:00, etc., the appropriate monitor command is executed from BASIC level e.g. M300L is equivalent to

CALL -151

300L

3DOG

### 2. Control commands

Ctrl K = CATALOG

Ctrl L = LIST

These two short macros cannot be modified.

Ctrl D = define a macro

To define a macro type Ctrl D, whereupon the word MACRO will be displayed. Now type one of the following Ctrl characters

Ctrl Q

Ctrl W

Ctrl E

Ctrl R

Ctrl T  
Ctrl Y  
Ctrl P

Then enter your macro, which may be up to 100 bytes in length: for example, Ctrl D Ctrl Q ?CHR\$ (4) "OPEN FILE" :? CHR\$ (4) "READ FILE"

To save your newly designed macros enter BSAVE MACROEDITOR, A32000, L6399

The Macroeditor uses 7 very long instead of a bunch of very short macros. Macros must be issued in immediate mode command, e.g. while entering a new program line. They are not allowed in edit mode.

### 3. Ampersand commands

&U = unhooks the Macroeditor

& = rehooks the Macroeditor (identical to CALL 32000)

&D + decimal number = converts to hexadecimal number, e.g. &D-936 results in FC58

&H + hexadecimal number = converts hexadecimal to decimal number, e.g. &HFF results in 255

&F = free sectors on disk

The disk must have been accessed prior to issuing this command, e.g. by CATALOG

&L = length of Applesoft program currently in memory

&R = renumber of Applesoft program

&R = renumber starting at 100 in 10 increments

&R1000,5 = renumber starting at 1000 in 5 increments

&K = crunches program, i.e. REM statements are removed. Always SAVE your commented version prior to issuing this command.

&S = search for string and display all the Applesoft lines containing it

&S:TOKEN searches for tokens, example

&S:REM finds all REM lines

&S:GOSUB 100 finds all Gosubs 100

&S'NONTOKEN finds all literals, variables and quoted strings, example

&S'HI PAL finds all pals

&S'X\$ finds all variables X\$

&P = protects the program currently in memory

&M = merges protected program with loaded program

These two commands work as follows: Imagine you want to append a standard subroutine to the main program you are developing.

- a) LOAD SUBROUTINE
- b) &R60000,1           renumbers subroutine at "top end"
- c) &P                   protects subroutine
- d) LOAD MAIN PROGRAM
- e) &R1,1                renumbers main program at "bottom end"
- f) &M                   merges main program with subroutine
- g) &R                   renumbers new program

Renumbering at top and bottom end avoids overlapping of line numbers.

&V = variable list = cross-reference

This command displays variables with line numbers, in alphabetical order, e.g.

```
10 A = B : X$ = "STRING"
20 PRINT X$
30 PRINT A
```

would cross-reference as

```
A     10,30
B     10
X$    10,20
```

The number of line references to be printed on one line can be poked by POKE 38395,N. For instance, POKE 38395,10 would result in 10 line references per printed line, thus producing neat printouts. Currently POKE 38395,5 is used which is convenient for screen display.

& + line number = hexadecimal -ASCII dump

This command has been implemented for Applesoft beginners. &100, for example, will display program line 100 in hexadecimal and ASCII. If you type C (for continue) the next line will be displayed, etc. This command even allows you to LIST list-protected programs. (It should be noted, that hexadecimal numbers are displayed in NORMAL and INVERSE for better reading)

## EDIT MODE COMMANDS

Edit mode commands are valid after the edit mode has been invoked by Y + line number or K + line number. The Applesoft program line to be edited is always displayed on the last 6 screen lines. The last character on the sixth line, i.e. the 240th character must remain empty.

- Ctrl B           = Jump to beginning of program line
- Ctrl E           = Jump to end of program line
- Ctrl L           = Jump 1 screen line down, i.e. 40 characters
- Ctrl T           = Tab 10 characters
- Ctrl D           = Delete character under cursor
- Ctrl I           = Invoke insert mode  
Insert mode is cancelled by right arrow, left arrow or RETURN or if 6 screen lines have been filled. Insert mode allows you to enter any control character except right arrow, left arrow, RETURN and Ctrl O
- Ctrl P           = Partial storage = quits edit mode and enters program line into memory up to the cursor position
- RETURN          = Quits edit mode and enters program line as seen on the screen. The cursor must not be at the end of the program line
- ESC             = Quits edit mode, but DOES NOT change the program line in memory (skips editing)
- Ctrl X           = ditto
- Ctrl C           = ditto