

# ATARI PROGRAMMER'S WORKSHOP



**Synergistic  
Software**

830 N. RIVERSIDE DR.  
SUITE 201  
RENTON, WA 98055  
(206) 226-3216

## **DISCLAIMER**

Synergistic Software products are sold "as is." Synergistic Software disclaims all warranties, express or implied, including but not limited to any implied warranty of merchant ability or fitness for any particular purpose or use.

# **The Programmer's Workshop**

By Dennis M. Keathley

Documentation By:  
Dennis M. Keathley  
Leslie Hornung  
Kathryn Hallgrimson Suther

©Copyright by Synergistic Software 1982

Purchaser is granted the right and license to use this program on a single CPU. By acceptance of this product the purchaser agrees that it will not sell, lease, license, or otherwise distribute copies of this software or documentation, whether or not for profit.



# TABLE OF CONTENTS

	<b>PAGE</b>
INTRODUCTION .....	4
CHAPTER 1 MICRO-DOS .....	5
CHAPTER 2 ANALYZER .....	7
CHAPTER 3 TRANSFER .....	9
Return to BASIC .....	10
File Input .....	10
Sector Input .....	10
File Output .....	10
Sector Output .....	11
Set Buffer Address .....	11
Default Buffer .....	11
Set Drive Number .....	11
CHAPTER 4 COMPARE .....	12
CHAPTER 5 VARIABLE NAME TABLE MANAGER (VARNAME) .....	13
CHAPTER 6 BAUDRATE.900 .....	14
CHAPTER 7 DISK TO CASSETTE (DISK2CAS) .....	15

# PROGRAMMER'S WORKSHOP

## INTRODUCTION

The Programmer's Workshop is a collection of utility programs written for the Atari 400/800 with 16K. Both novices and experienced programmers can use these utilities, which include disk to cassette transfer, BASIC program compare capabilities, cassette baud rate increase, analysis of program code, etc. These programs will prove to be invaluable to all dedicated Atari users who wish to facilitate their programming. The Atari becomes a more powerful machine through the utilization of the Programmer's Workshop.

Read the following instructions before using the Programmer's Workshop.

Insert your BASIC cartridge, turn on the disk drive, and insert your diskette. Turn on your computer. The Programmer's Workshop Main Menu will appear and show:

1. **DOS**
2. **ANALYZER**
3. **TRANSFER**
4. **COMPARE**
5. **VARNAMMR**
6. **BAUDRATE. 900**
7. **DISK2CAS**

When you press number 1 (DOS), then [Return], the Atari DOS 2.0S Menu will appear. To access MICRO-DOS press the [Reset] button, type DOS and the MICRO-DOS Menu will appear. If you are in the BASIC immediate mode just type DOS and the MICRO-DOS Menu will appear. Read Chapter One on MICRO-DOS for instructions on its use.

# CHAPTER ONE

## MICRO-DOS

MICRO-DOS is a handy utility allowing quick and easy access to many of the functions usually accessed thru DUP.SYS. Normally, if you would like to do a Directory of a diskette you would type DOS [Return], whereupon the computer saves your present RAM resident program to disk (if a MEM.SAV file is present) and then loads and executes DUP.SYS (all of which takes some 12 to 22 seconds). At this point you may remember you had no MEM.SAV file, and your program is now lost.

With MICRO-DOS this will never happen, as many of the options available from DUP.SYS are on-line and available at any time with no danger to your precious program. MICRO-DOS provides you with a RAM resident Menu similar to the one available in DUP.SYS. This allows access to frequently used features including:

**LOCK**  
**UNLOCK**  
**DELETE**  
**RENAME**  
**FORMAT**  
**MENU**  
**ADOS**  
**BASIC**

This is the MICRO-DOS Menu. Any command may be selected by typing the first character of the command name (inverse video on the Atari screen) followed by [Return].

After you choose the command you wish, you will see the prompt FN? (File Name) appear. You must enter your device name first and then your file name. The device name can be D1, D2, D3, or D4 for Drives 1, 2, 3, or 4. The device name must be followed by a colon. Then enter your file name. (D1:File Name). Any file name not preceded by a device name is treated as an error and will clear the screen and redisplay the MICRO-DOS Menu.

For complete instructions on the use of the MICRO-DOS commands see your Atari DOS.20S Manual page 35 for LOCK and UNLOCK, page 32 for DELETE and RENAME, page 36 for FORMAT, page 27 for MENU, and page 29 for BASIC. Briefly, this is the function of each command:

- LOCK** — “Write protects” your files, preventing you from writing over, appending, or deleting any file you choose to “lock”.  
**Note:** *Locked files have an asterisk (\*) in front of their file names in the file directory.*
- UNLOCK** — Removes locks on files you may have previously “write protected” (locked) for any reason.
- DELETE** — Removes one or more named files from the disk directory file and from the diskette.
- RENAME** — Allows you to change the name of one or more files.
- FORMAT** — Formats a diskette, allowing data to be stored and retrieved using that diskette.
- MENU** — Will display the directory of the disk drive you have chosen.
- ADOS** — Loads the Atari DOS Menu (DUP.SYS).
- BASIC** — Runs the Atari Cartridge BASIC

A few words of caution are now in order.

**First:** MICRO-DOS hooks itself into the FMS (File Management System known as DOS.SYS) so DOS.SYS must be on the diskette and can not function on its own.

**Second:** MICRO-DOS occupies all of page six (1536 to 1791), so you can not use this area without losing MICRO-DOS. MICRO-DOS also uses the bottom third of the 6502 stack area.

**Third:** If Atari DOS is called the system must be rebooted to reestablish MICRO-DOS.

**Fourth:** Most importantly, do not write DOS.SYS to any diskette if you have called Atari DOS from MICRO-DOS as many of the pointers have been changed from DOS original.



## CHAPTER TWO

### ANALYZER

The ANALYZER is a powerful utility for unlocking the mysteries of a cassette file, the computer ROM and RAM, or any portion of a diskette. The two major features are Hexdump and Disassembly, and the output of the program can be sent either to the screen or a printer.

The first question asked by the program is "Analyze: Memory, Tape, or Disk?." If you desire to explore the RAM or ROM, press [M] at this point. If analyzing a cassette file is what you need to do, press [T]. If you want to investigate your disk, either by file or by sector, press [D]. If you select "D" you will be asked "File or Sector input?." Pressing [F] will then prompt you for a filename to input. For drive #1 you may omit the "D:".

For disk or cassette files, you are then asked "# of bytes?." Enter quantity, or just press [Return] to input the entire file. Please note that in all cases numbers may be input in decimal, or hexadecimal preceeded by a "\$".

For sector input you will be prompted for "Starting sector" and "Ending sector." Enter the desired numbers (from 1 to 720, or \$1 to \$02D0 hex). Selecting sector input completely by-passes DOS and lets you examine any sector(s) on the disk.

The last question in this sequence is "Address offset?." You may simply press [Return]. The address offset allows you to input a starting address for machine language code that you may be disassembling, in order to have the proper addresses displayed on the left side of the disassembly. You may not know the starting address at first, but pressing [Return] starts the address display at zero.

Once the input (if any) into the buffer has been accomplished, you may start Analyzing. The program then asks "Relative Start Address?," or just "Start Address?" if memory is being analyzed. Enter the absolute address at which you wish to start analyzing. Valid addresses range from zero to 65535, or \$FFFF hex. For tape or disk, enter the relative address from the start of the input data.

Now, you may select the form of the output, either Hexdump or Disassembly. Any data may be hexdumped, but it may not make sense to disassemble unless it is actually machine language. If the program attempts to disassemble data that does not make sense as machine language, it will print “\*?\*” mixed in with random assembler mnemonics. You will note that all numbers in the disassembly printout are in hex. Also note that relative branching shows the actual address to which the branch command goes rather than + x or -x, except in the case of tape or disk input where the branch is backward farther than the start of the data.

Finally, you select the output device, either the screen or the printer. For disassembly, the output looks the same on either device, but for hexdump the screen displays in rows of eight columns of hex bytes, whereas the printer prints in rows of sixteen bytes. When a full screen is achieved, the display of data halts and you are asked whether you want to “Continue, Restart, or End.” Pressing [R] takes you back to “(Relative) Start Address?” You may select a new address, or return to the top choice of “Memory, Tape, or Disk?” by just pressing [Return]. Pressing [E] ends the program and returns you to BASIC. You may also end by just pressing [Return] in response to the top-level choice. Now that you have mastered the ANALYZER, no form of ATARI data should ever remain a mystery!

## CHAPTER THREE

### TRANSFER

Transfer simplifies the moving of large quantities of data between diskette and/or cassette media. The easy-to-use menu lets you select either file input (disk or cassette) or sector input from a disk. You may then select either file or sector output of the data. The menu is:

```
PRESS SELECT  
RETURN TO BASIC  
FILE INPUT  
SECTOR INPUT  
FILE OUTPUT  
SECTOR OUTPUT  
→ SET BUFFER ADR  
DEFAULT BUFFER  
SET DRIVE NUMBER
```

As SELECT is pressed, the arrow on the left moves down. When it points to the desired operation, press START. Additional prompts will appear in the lower portion (the "message window") of the screen. If an error occurs, or an improper operation is attempted, a flashing red message will appear in the message window along with an audible buzzer. Pressing OPTION allows you to change between BYTE and SECTOR output for files (see FILE OUTPUT DESCRIPTION).

Below the menu section is the current configuration section. As its name implies, it shows the present configuration of the program. It looks like:

#### **PRESS OPTION BYTES**

<b>MAX SECTORS</b>	<b>211</b>
<b>MAX BYTES</b>	<b>27008</b>
<b>BUFFER ADR</b>	<b>13184</b>
<b>DRIVE NUMBER</b>	<b>1</b>

The word under OPTION (BYTES or SECTORS) will indicate the present file output configuration of the program. MAX SECTORS is the maximum number of sectors (128 bytes from disk or cassette due to RAM limitations). MAX BYTES is this number times 128 to convert it to bytes. BUFFER ADDRESS is the RAM address of the start of the buffer that is used to hold the input data or perform output. You may set the buffer address anywhere you like for output, (see SET BUFFER ADR DESCRIPTION) but on input there are some safeguards to keep you from destroying the program or system RAM. SET DRIVE NUMBER shows you the current drive for sector input or output (I/O). For file I/O, the drive number is included in the filename as "D2:FILENAME".

Below the current configuration section is the message window section. In this area of the screen are displayed any prompts or questions the program may ask you. Also displayed here are any informational or error messages resulting from your operation of the program. Error messages are displayed in flashing red, accompanied by an audible buzzer. One other message that is displayed in red is the confirmation for output message, "PRESS Y TO OUTPUT." A successful input or output operation is indicated by a high-pitched bell sound.

## DESCRIPTION OF MENU OPTIONS

All numerical inputs (except drive number) may be either decimal, or hex preceded by "\$." Pressing only [Return] in response to a question or prompt at any time will return you to menu selection.

### RETURN TO BASIC

Gracefully ends the program.

### FILE INPUT

Allows input of a disk or cassette file into RAM buffer. You are prompted for filename by "FN?." Enter "D:--" or C:. D: means Disk Drive, then insert the disk drive number. C: means cassette. For disk files from drive #1 you may omit the D:. Press the START button. Successful input results in "—— BYTES INPUT" displayed and a bell sound.

### SECTOR INPUT

Allows input into RAM buffer of a selected disk sector or sectors. You are prompted for first "START SECTOR" and then "END SECTOR." Press the START button. Successful input results in a bell sound.

### FILE OUTPUT

Allows the output of a selected portion of the buffer in the form of either a disk file or cassette file. Enter filename in response to "FN?" prompt, then either the number of bytes or the number of 128-byte sectors to output, depending on the present setting shown under "PRESS OPTION". You will then be prompted to "PRESS Y TO OUTPUT." Any other key pressed will abort the output. Successful output results in a bell. **Note:** *Even though you select the amount of data to output as a certain number of 128 byte sectors, DOS will still store only 125 bytes per each sector in the file, reserving three for its bookkeeping. 128 byte sectors, DOS will still store only 125 bytes per each sector in the file, reserving three for its bookkeeping.*

## SECTOR OUTPUT

Allows the output to disk of a selected portion of the buffer in the form of sequential sectors of 128 bytes each. The DOS directory and allocation map sectors are unaffected, as this form of output bypasses DOS. You are prompted for "START SECTOR" and "END SECTOR," and then [Y] to output. Successful output results in a bell sound.

## SET BUFFER ADR (ADDRESS)

Allows you to set the buffer start address to be anywhere in memory (0 to 65535 decimal or 0 to \$FFFF hex). If you raise the buffer start from its initial setting, MAX SECTORS and MAX BYTES will decrease. Lowering it will not affect MAX SECTORS and bytes, but input into the buffer will be inhibited.

## DEFAULT BUFFER

Restores the buffer start address to its lowest possible setting without endangering the program or system RAM. This gives you the largest possible buffer. ***The buffer address must be set at the default value before you lower it below the default value.***

## SET DRIVE NUMBER

Allows the selection of a drive unit number for sector input or output. Some uses for TRANSFER are to move files from cassette to cassette, cassette to disk, or disk to cassette. It can also be used to transfer information from a disk that is in boot or sector format (such as a game disk) to a DOS-compatible file. This is one of its handiest uses. Also, after a machine language program is assembled and stored as a disk or cassette file it may be transferred to a disk by using sector output starting at sector 1, making it an auto-booting disk. Or, you can simply store data at a known location in memory for further manipulation, and then transfer it to disk or cassette.

## CHAPTER FOUR

### COMPARE

The BASIC Program Compare Utility will compare two BASIC files that are stored on a diskette in LIST format on a line-by-line basis. The lines which are different between the two files will be displayed on the screen. It will also create two output files, FIX1.ENT and FIX2.ENT, that can be ENTERed into the corresponding original file to make it identical to the other.

For example, suppose you have two versions of a program on your disk, and in one version you have changed two or three lines, or added or deleted some lines, to fix some bugs in the program. But now you do not remember which one is the debugged version. Simply LIST the programs to another disk as versions .001 and .002 and execute COMPARE. You will be asked for "FILENAME 1" and "FILENAME 2." Be sure you have the LISTed disk in drive #1 and enter the filenames. COMPARE will show you the lines that are different.

Another example: suppose you had a large program of which you needed several versions (for different printers, perhaps). You could keep each version on the disk and use a lot of sectors, or you could run COMPARE and create an output file for the 2nd., 3rd., 4th., etc. versions. Then just keep the first version and the FIX1.ENT output file from each of the other versions (rename FIX1.ENT appropriately for each version). The much smaller output files can just be ENTERed into version one to yield whichever version you desire. No need to go through each version to search out the differences, either.

ENTER FIX1.ENT into FILE 1 to make it identical to FILE 2. ENTER FIX2.ENT into FILE 2 to make it identical to FILE 1. **An Important Note:** COMPARE requires that four files be open at once. ATARI DOS was delivered on the MASTER DISKETTE set for only three simultaneously open files. Therefore, you must boot up using the DOS on the COMPARE diskette, or make the appropriate change to your DOS (see DOS II Reference Manual, page 79, ERROR 160 description). Please note that if you have completely renumbered the lines in a BASIC program you cannot use COMPARE to find any differences between the altered program and the original file, since COMPARE functions are based on line numbers.

## CHAPTER FIVE

### VARNAMMR

The Variable Name Table Manager allows you to recreate the Variable Name Table for a BASIC program if someone has overwritten the table with garbage to prevent LISTing of the program. Execute the Variable Name Table Manager, and it will prompt you for "FILENAME TO UNPROTECT" and "OUTPUT FILENAME". Insert the diskette containing the tokenized (SAVE format) BASIC program file, enter the filenames, and press [Return] when prompted for "SOURCE DISK." The Variable Name Table Manager will read the file, create a new Name Table, and prompt you for "DESTINATION DISK." Insert the destination diskette and press [Return]. You will then be prompted for "SOURCE DISK," "DESTINATION DISK," etc. until the entire file has been rewritten. The source and destination diskettes may be the same diskette, if desired. If you have 48K, you may only have to do this once. Please do not mix up the source with the destination diskettes as this will cause an error condition. However, the output file is properly closed after each output pass, so the destination diskette structure will be intact even if the output file is not complete.

You may use the same name for output file as you do for input file **only if** you use different disks for source and destination. In either case, the output file is quite LISTable when LOADED back into the computer.

## CHAPTER SIX

### BAUDRATE.900

BAUDRATE.900 allows you to perform cassette input and output (I/O) at a 50% faster rate by increasing the BAUD rate (bits per second input or output) from 600 to 900. This results in significantly less time spent waiting for cassette I/O. This is accomplished by modifying the ROM-resident cassette handler and installing the modified handler in RAM just below BASIC RAM.

When BAUDRATE.900 is executed, it changes the system LOMEM pointer to a higher value to make room for the new handler and then reinitializes BASIC. The program then instructs you to press [Return], at which time it will install the new cassette handler in lower RAM. You are now ready for faster cassette I/O!

Before executing BAUDRATE.900, it is recommended that you CSAVE the program at the normal 600 BAUD to all cassettes on which you plan to store 900 BAUD programs or data. After this is done, you may simply CLOAD and RUN this program from the front of the cassette upon computer power-up. Then you are ready to load in any 900 BAUD program on the cassette.

Please note that pressing SYSTEM RESET will cause your new handler to be lost, requiring you to rerun BAUDRATE.900. Also note that storing data at a higher BAUD rate increases the chances of losing data because of bad spots on the tape. You may have to use higher quality tape than is necessary at 600 BAUD. In any case, you should test the increased BAUD rate on your particular system several times before deciding to use it. Cleaning and aligning your 410 recorder will also help.



## CHAPTER SEVEN

### DISK2CAS

The Disk-to-cassette Backup Utility gives you an easy way to transfer all files from a diskette to a cassette without the effort and time spent doing it file by file. Once the options are selected and the diskette and cassette inserted, you may leave the computer and do something else, because the transfer is completely automatic.

The program options are:

- 1) **MULTIPLE CASSETTES**
- 2) **ALPHABETIC SORT, OR**
- 3) **SORT BY FILESIZE**
- 4) **END-OF-FILE ALARM**
- 5) **END-OF-CASSETTE ALARM**
- 6) **PRINT SORTED MENU**
- X) **START PROGRAM**

You will notice a high-pitched beep as you select options. This is audio feedback to let you know that an option was correctly selected. If an incorrect key is pressed, or you try to select both types of sorting, a lower-pitched buzz will be heard.

### DESCRIPTION OF MENU OPTIONS

#### **MULTIPLE CASSETTES**

This option allows you to backup from a diskette to several cassettes, one after the other, without having to re-RUN the program and reselect options.

#### **ALPHABETIC SORT**

Selecting this option causes the disk files to be transferred to cassette in alphabetical order.

#### **SORT BY FILESIZE**

Selecting this option causes the disk files to be transferred to cassette in order of increasing file size.

#### **END-OF-FILE ALARM**

When selected, this option sounds an alarm at the end of each disk file transferred. This is useful if you desire to write down the digital counter reading for the start of each cassette file.

## END-OF-CASSETTE ALARM

Selection of this option sounds an alarm after all files are transferred from diskette to cassette. This alerts you so that you may exchange cassettes or simply turn off your system when the backup operation is completed.

## PRINT SORTED MENU

If you have a printer, you may select this option. It will give you a printout of the disk filenames in the order that they will be transferred to cassette. This is very handy for keeping a record of the contents of the cassette, and the digital counter readings may be written beside the filenames. If you do not have an EPSON printer, you may have to modify or remove the printer control characters, one in line 390 and one in line 410.

## START PROGRAM

When all the desired options have been selected, press [X] to start the backup operation.

After pressing [X], you will be instructed to insert the disk to be backed up to cassette, the cassette that you want the files copied onto, and told to press RECORD and PLAY on the cassette deck. When everything is ready, press [Return] to start the disk-to-cassette transfer. You may want to turn the volume up on the TV in order to hear the alarm while you are in another room. No other noise will be heard from the TV speaker, and the standard *beep-beep* (for cassette output mode) from the console speaker will not be heard (it has been changed to a soft *click-click* to make the transfer process less annoying).

You will be constantly informed by the screen as to which files have been or are being transferred. The display is put into the ATTRACT mode most of the time to prevent damage to your TV during the long wait for the backup operation.

When the alarm sounds, you may silence it by pressing the SELECT console switch; however, the transfer operation will not continue until you press START.

The backup program itself uses 5150 bytes of RAM, so subtracting this from the total useable RAM will show the size of the largest file that may be transferred (divide this number by 125 to find the largest number of disk sectors that can be transferred). When the program finishes, you should press SYSTEM RESET before performing any other cassette I/O. *Notice! This program will cause MICRO-DOS to work incorrectly because it also uses page six of RAM. To restore MICRO-DOS, reboot the disk.*



## **PROGRAMMER'S WORKSHOP**

The Programmer's Workshop is a collection of utility programs written for the Atari 400/800 with 16K. Both novices and experienced programmers can use these utilities, which include disk to cassette transfer, BASIC program compare capabilities, cassette baud rate increase, analysis of program code, etc.

The Programmer's Workshop contains the Analyzer, a powerful utility for unlocking the mysteries of a cassette file, the computer ROM and RAM, or any portion of a diskette. Also includes on the same diskette:

MICRO-DOS

Transfer

Compare

Variable Name Table Manager

Baudrate 900

Disk to Cassette

These programs will prove to be invaluable to all dedicated Atari users who wish to facilitate their programming. The Atari becomes a more powerful machine through the utilization of the Programmer's Workshop.

**SYNERGISTIC SOFTWARE**

830 N. RIVERSIDE DR.

SUITE 201

RENTON, WA 98055

(206) 226-3216