

ATARI® PROGRAM exchange

BASIC PROGRAM COMPRESSOR (MASHER)

by
Dale Yocum
Jerry White

USER INSTRUCTIONS
(5/01/81)

USER-WRITTEN SOFTWARE FOR ATARI PERSONAL COMPUTER SYSTEMS
(APX-20008)



BASIC PROGRAM COMPRESSOR
(MASHER)

by

Dale Yocum
(Modified by Jerry White)

USER INSTRUCTIONS

COPYRIGHT 1981 Atari, Inc.

Copyright and right to make backup copies. On receipt of this computer program and associated documentation (the software), ATARI grants to you a nonexclusive license to execute the enclosed software and to make backup or archival copies of the computer program for your personal use only, and only on the condition that all copies are conspicuously marked with the same copyright notices as appear on the original. This software is copyrighted. You are prohibited from reproducing, translating, or distributing this software in any unauthorized manner.



CONTENTS

INTRODUCTION . . . 1

- Overview . . . 1
- Required accessories . . . 1
- Optional software . . . 1

GETTING STARTED . . . 2

USING MASHER . . . 3

- Setting up . . . 3
- Execution . . . 4
- Evaluating the results . . . 4

TROUBLESHOOTING . . . 5

- Error message . . . 5
- Program operation warnings . . . 5
- Suggestions for tracking down faulty program execution . . . 5

ADDITIONAL COMPRESSION TECHNIQUES . . . 6



INTRODUCTION

OVERVIEW

MASHER is a utility program to help you squeeze every last byte out of your 8K BASIC software. It compresses your BASIC programs into as little memory as possible by

- (1) Removing all unreferenced REMark statements
- (2) Converting your most frequently used constants into variables
- (3) Packing together several short lines to form longer ones

Don't use line numbers 3 - 9 or assign variable names Q0 - Q999 in your BASIC programs. MASHER uses these lines and variable names to define its variables that replace frequently used constants in your program.

REQUIRED ACCESSORIES

32K RAM
ATARI BASIC Language Cartridge
ATARI 810 Disk Drive

OPTIONAL SOFTWARE

BASIC CROSS-REFERENCE UTILITY (XREF)

This ATARI Program Exchange (APX) program tells you how many variables you've used in your program (as well as providing other information). (Order No. APX-20009)

VARIABLE CHANGER

This APX program automatically shortens your program's variable names, which often further reduces the program's memory requirements. It also tells you how many variables you've used in your program. (Order No. APX-20012)

GETTING STARTED

1. Insert the ATARI BASIC Language Cartridge in the (Left Cartridge) slot of your computer.
2. Make sure your BASIC program to be compressed is in LIST format (rather than in SAVE format). See page 10 of the BASIC Reference Manual or see the Disk Operating System Reference Manual for information about LIST format.
3. Turn on your disk drive and insert the MASHER diskette.
4. Power up your computer and turn on your video screen.
5. At the READY prompt, type RUN "D:MASHER" and press the RETURN key. If you're using more than one disk drive, remember to follow the device initial (D) with the number of the drive containing the MASHER diskette (e.g., RUN "D2:MASHER" for disk drive two). The program will load into RAM and start. The first display screen will be the COPYRIGHT 1981 ATARI notice.
6. If you're using just one disk drive, remove the MASHER diskette and insert your BASIC program to be compressed.

USING MASHER

SETTING UP

You use MASHER by responding to three prompted questions.

1 COMPLETE INPUT FILE NAME?

Type in the complete name of the file in LIST format containing your BASIC program. Use the format

`D:filename.ext`

where "D" is the device initial, "filename" stands for the name of your file (use upper case), and ".ext" is the extender, if you used one. An example is `D:ADDRESS.BAS`. If you're using more than one disk drive, remember to specify the number of the drive containing your file (e.g., `D2:ADDRESS.BAS` for disk drive two).

2 COMPLETE OUTPUT FILE NAME?

Using the format described in prompt 1, type the complete file name under which you want MASHER to store your compressed program in LIST format. Your output file must have a different name than your input file.

3 NUMBER OF VARIABLES USED IN SOURCE PROGRAM?

Type in the number of variables used in your input file. MASHER uses this information to determine how many constants it has room to define as variables (remember that BASIC is limited to 128).

You can determine this number in three ways.

(1) Count them! This method is fine if you don't mind spending the time. Increment your count by a small amount as a "fudge factor" against your having missed a few.

(2) Guess. For small programs, guessing is acceptable. If you guess too high, MASHER might not compress your program as well as it might have with the true count. If you guess too low, MASHER might unknowingly exceed BASIC's 128-variable limit and your program will generate an ERROR 4- (Too Many Variables) when you run it. It's better to guess high.

(3) Use the BASIC CROSS-REFERENCE UTILITY (XREF) or VARIALBE NAME CHANGER, available through APX. Both programs give you the exact variable count for your program.

MASHER then takes over and starts compressing your program.

EXECUTION

MASHER displays the message

PLEASE STAND BY

for a short time. It then clears the video screen to go into "attract mode", which allows MASHER to run about 40 percent faster since the screen isn't interrupting the microprocessor.

MASHER can take a long time. A small program might take only a few minutes to compress, but a large program can take hours! As long as the disk drive is read periodically, all is well. Take this opportunity to relax and browse through the latest APX software catalog for gift ideas--for yourself and others!

When MASHER finishes, it reactivates the video screen and displays the number of lines in your input and output files.

EVALUATING THE RESULTS

Occasionally, your output file might have more lines than does your input file. In these cases, check the amount of RAM used by each program before rejecting the compressed version. Because MASHER adds lines to redefine constants, the line count might increase but the compressed program will still require less RAM to run. A compressed program is rarely, if ever, larger than your original BASIC program.

Normally, MASHER saves between 5 and 60 percent of your program's original required RAM. The amount depends on your individual programming style. Programmers who use many remarks and single statement lines will save the greatest amount. In short, the more structured and easy-to-read your input file, the more MASHER can compress!

TROUBLESHOOTING

ERROR MESSAGE

The error message

```
SOURCE CODE ERROR, LAST LINE: "  
<last line read>
```

occurs when a file contains some text that is illegal in BASIC (e.g., a line with no line number). This error causes MASHER to abort. Fix the problem in your source program and rerun MASHER.

PROGRAM OPERATION WARNINGS

MASHER is quite demanding of disk drives. Because a long program can take hours to compress, even the best system might stumble. Try rerunning MASHER before throwing in the towel.

MASHER can't resolve variable GOTOs (e.g., GOTO X). Check your compressed program if you used this kind of GOTO.

SUGGESTIONS FOR TRACKING DOWN FAULTY PROGRAM EXECUTION

If you find that your compressed program doesn't run correctly, check these points:

(1) Did you leave lines 3 - 9 available for MASHER's use? MASHER will try to use these lines to define its constants

, even if you use these lines as well. MASHER's constant definitions will be overwritten, causing some strange errors!

(2) Did you leave variable names Q0 - Q999 for MASHER's use?

(3) Did you use a CLR statement anywhere in your program? This statement clears out MASHER's Q0 - Q999 definitions, causing all sorts of problems! In most applications, CLR is unnecessary and you can safely eliminate it. If you include a CLR statement, make sure your program re-executes lines 3 - 9 afterward.

(4) Did you use computed line references (e.g., GOTO X)? If so, these may reference lines that don't exist anymore if MASHER merged the statements with other lines. One way to preserve these lines when running MASHER is to place a dummy reference to these lines at the beginning of your program. For example:

```
20 ON I GOTO 1500,2000,2100  
:  
60 GOTO X
```

ADDITIONAL COMPRESSION TECHNIQUES

MASHER can save you a lot of RAM, but it's not the only method available to you as a programmer. Here are some other techniques.

- (1) Use strings instead of numeric arrays to store data.
- (2) Add two "Q" variables together to form a constant, rather than defining a new constant. For example, the constant $I=Q2+Q5$ takes less RAM than does the constant $I=7$ if you've already defined Q2 and Q5.
- (3) Use subroutines for common functions.
- (4) Recode parts of your BASIC program into machine language.
- (5) Reuse variables when they don't interfere with each other. The XREF program, available through APX, can help detect interference.
- (6) Use short variable names. The VARIABLE CHANGER program, available through APX, can help you change variable names in existing programs.
- (7) Chain in parts of your program from disk rather than maintaining your entire program in RAM.
- (8) Use the graphics mode requiring the least amount of RAM but is still suited to your application. For example, Modes 1 and 2 require less RAM than does Mode 0.
- (9) POKE data into unused RAM areas instead of keeping data in arrays. Appendix D of the BASIC Reference Manual shows where some of these areas are. You can also use the printer and cassette buffers for data if you're not using these devices.

We're interested in your experiences with APX programs and documentation, both favorable and unfavorable. Many software authors are willing and eager to improve their programs if they know what users want. And, of course, we want to know about any bugs that slipped by us, so that the software author can fix them. We also want to know whether our documentation is meeting your needs. You are our best source for suggesting improvements! Please help by taking a moment to fill in this review sheet. Fold the sheet in thirds and seal it so that the address on the bottom of the back becomes the envelope front.

1. Name and APX number of program _____

2. If you have problems using the program, please describe them here.

3. What do you especially like about this program?

4. What do you think the program's weaknesses are?

5. How can the catalog description be more accurate and/or comprehensive?

6. On a scale of 1 to 10, 1 being "poor" and 10 being "excellent", please rate the following aspects of this program:

- _____ Easy to use
- _____ User-oriented (e.g., menus, prompts, clear language)
- _____ Enjoyable
- _____ Self-instructive
- _____ Useful (for non-game software)
- _____ Imaginative graphics and sound

7. Describe any technical errors you found in the user instructions (please give page numbers)

8. What did you especially like about the user instructions?

9. What revisions or additions would improve these instructions?

10. On a scale of 1 to 10, 1 representing "poor" and 10 representing "excellent", how would you rate the documentation and why?

11. Other comments about the software or the user instructions:

[STAMP]

ATARI Program Exchange
P. O. Box 427
155 Moffett Park Drive
Sunnyvale, CA 94086

[seal here]